Encryption, authentication and copyright protection techniques for digital images

Fawad, Ahmed

2010

Fawad, A. (2010). Encryption, authentication and copyright protection techniques for digital images. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46246>

<https://doi.org/10.32657/10356/46246>

ENCRYPTION, AUTHENTICATION AND COPYRIGHT PROTECTION TECHNIQUES FOR DIGITAL IMAGES

FAWAD AHMED

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2010

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other university or institution.

Date

Fawad Ahmed

Acknowledgement

I am extremely grateful to my supervisor Dr. Mohammad Yakoob Siyal for his guidance, support and encouragement at every step of my Ph.D. study. He not only provided me useful insight, but also allowed me considerable freedom to explore a variety of topics to develop my abilities in research and problem solving. He allowed me to work with different researchers to help broaden my vision. He would calm me down whenever things would not work out for me. His knowledge, kindness and vision have provided me lifetime benefits.

I am lucky to have worked with Dr. Farook Sattar. He used to hold in-depth discussions with me and provided me valuable suggestions concerning my work on copyright protection techniques. I would like to thank Dr. Vali Uddin Abbas for his thoughtful comments and enlightening discussions towards exploring new ideas in my work. The role of Dr. Yap Kim Hui towards my research has also been very instrumental. He used to give deep insight in his lectures which would trigger my thought process. I would also like to express my gratitude to Dr. Nassar Ikram, Dr. Athar Mahboob and Dr. Arshad Aziz.

I am indebted to Dr. Faeyz Karim and Mohammad Yousuf for their invaluable friendship. They justified that a friend in need is a friend indeed. Special thanks to Ghazanfar Monir for looking after administration issues on my behalf. Thanks also to Dr. Abdul Waheed Umrani, Dr. Liu Mingchun, Dr. Bikal Kumar, Dr. Niladri Puhan, Gurdeep Singh, Mayank Gupta, Jitandar Kumar Kella and Asad Sidiqqi.

I would also like to thank the panel examiners of my oral exam, first-year qualifying exam and the external examiners for their comments and useful suggestions

that has definitely helped to improve the quality of this thesis. The prompt help and response of the staff of ISRL laboratory is really appreciated. I take this opportunity to thank Nanyang Technological University for providing me with a scholarship for Ph.D. studies. I am also thankful to National University of Sciences and Technology for facilitating me when I resumed my job along with my PhD studies.

I must acknowledge the patience that my wife Madiha and my two lovely children Aamna and Ebrahim bore whenever I was busy with my research. I would like to express my appreciation to my uncles Capt. Jaleel Ahmed, Dr. Khalil Mukaddam and Faiz Mukaddam for their timely help. I am also thankful to my brother Jawad Ahmed and my sister Ayesha and my brother-in-law Tariq. Special thanks to my parents-in-law, Hashim Noorani and Khadija Noorani for their constant support.

Lastly, I must mention two people to whom I dedicate this thesis, my parents Shakeel Ahmed and Fauzia Shakeel. Your constant love, support, nurturing, care and prayers have always been a blessing for me.

Abstract

For the past several years, a lot of research work has been underway to devise security techniques to protect multimedia data like audio, image and video. A question that arises is why new techniques for multimedia security are required when the field of cryptography is already well matured? There are several reasons that actually impede the direct use of cryptographic techniques for solving multimedia security problems. For example, an image encrypted using the AES algorithm cannot be correctly decrypted if it undergoes JPEG lossy compression. In multimedia scenarios, content preserving operations like lossy compression, enhancement, transcoding, etc are very common. Applying standard cryptographic techniques directly to the entire multimedia data stream will have problems because a single bit error in the transmitted data stream will cause errors in the data recovery stage. Hence there is a need to devise security algorithms that are focused specifically towards multimedia environment keeping in view its very intrinsic nature.

This thesis is geared towards addressing three closely related problems pertaining to multimedia security, i.e., encryption, authentication and copyright protection. A JPEG compression tolerant image encryption scheme is proposed using some very interesting properties of orthogonal matrices. The proposed scheme provides high level of security against various cryptographic attacks, and in addition, can also tolerate JPEG compression. This means if the encrypted image data is JPEG compressed up to a certain quality factor, decryption is possible with some distortion. Detailed experimental results and security analysis is presented to show the viability of the proposed scheme.

To address the issue of image authentication, a wavelet-based hashing scheme is proposed that simultaneously addresses the core issues pertaining to the security of an image authentication algorithm, tamper detection and robustness. Instead of using the secret key to randomly select the image features, a randomized pixel modulation method is proposed that makes the entire feature space random with high sensitivity towards change in the secret key and effective tamper detection capability. The proposed image hash is robust to JPEG compression, low-pass and high-pass filtering; which are content preserving operations. A 4-bit quantization scheme is also proposed that significantly reduces the size of the image hash. Detailed experimental results and security analysis depict that the proposed scheme generates a hash of reasonable size with good security features. To address the issue of copyright protection in digital images, a wavelet-based watermarking scheme is proposed that is integrated with standard cryptographic techniques to form a set of copyright protection algorithms and protocols. The preamble behind this proposal is that a watermarking algorithm alone cannot completely address the complex issues involved in copyright protection. For a reliable and secure watermarking scheme, it is necessary that watermarking algorithms are well integrated with secure protocols. Specifically, three issues related to copyright protection have been focused on, that is, buyer-seller identification, copyright infringement and verification of ownership.

Table of Contents

Statement of Originality	i
Acknowledgement	ii
Abstract	iv
List of Figures	xvi
List of Tables	xvii
1 Introduction	1
1.1 Background	1
1.2 Basic Requirements of Data Security	2
1.3 Multimedia Security	2
1.3.1 Image Encryption	5
1.3.2 Hash-Based Image Authentication	7
1.3.3 Watermarking for Copyright Protection	8
1.4 Motivation	9
1.5 Thesis Contribution	10

TABLE OF CONTENTS

1.6	Thesis Organization	12
1.7	Chapter Summary	13
2	Image Security and Review of Related Work	14
2.1	Introduction	14
2.2	Review of Image Encryption Schemes	14
2.3	Review of Image Authentication Schemes	27
2.4	Review of Copyright Protection Schemes	35
2.5	Chapter Conclusion	42
3	A JPEG Compression Tolerant Image Encryption Scheme	44
3.1	Introduction	44
3.2	Image Encryption using Orthogonal Matrices	47
3.3	The Proposed Image Encryption Scheme	52
3.3.1	Image Encryption Algorithm	53
3.3.2	Image Decryption Algorithm	55
3.4	Experimental Results and Discussion	57
3.4.1	Image Encryption and Decryption	57
3.4.2	Robustness to JPEG Compression	59
3.4.3	PSNR Comparison	63
3.4.4	Effect of JPEG Compression on Images Encrypted using AES	66
3.4.5	Histogram of the Encrypted Image	68
3.4.6	Effect of Repeated Multiplication on the Cipher-image . .	70

TABLE OF CONTENTS

3.4.7	Effect of Tampering on the Decrypted Output	70
3.4.8	System's Throughput	74
3.5	Security Analysis	77
3.5.1	Key Space and Key Sensitivity	77
3.5.2	Brute Force Attack	80
3.5.3	On the Use of an Image-Independent Encryption Key . . .	83
3.5.4	Other Attacks	84
3.6	Conclusion	89
4	A Robust and Secure Hash-Based Technique for Image Authentication	91
4.1	Introduction	91
4.2	Proposed Algorithm for Image Authentication	94
4.2.1	Hash Generation Module	95
4.2.2	Image Verification Module	97
4.3	Quantization Scheme for Hash Size Reduction	99
4.3.1	Quantization Procedure	101
4.3.2	Hash Size Reduction Due to Quantization	104
4.3.3	Non-Malicious and Malicious Operation	105
4.3.4	Condition for Detection of Tampered Blocks	106
4.4	Experimental Results and Discussion	106
4.4.1	System's Robustness and Discussion on Threshold Selection	107
4.4.2	Detection of Tampering	111

TABLE OF CONTENTS

4.4.3	The Receiver Operating Curve	113
4.4.4	Hash Size	114
4.5	Security Analysis	115
4.5.1	Randomized Pixel Modulation and its Impact on System's Security	116
4.5.2	Effect of Secret Key on the Hash	118
4.5.3	Probability of Hash Collision	118
4.5.4	Quantization and System's Security	120
4.6	Discussion and Conclusion	120
5	A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection	123
5.1	Introduction	123
5.2	Preliminaries	125
5.2.1	Hash Function	126
5.2.2	Digital Signature	126
5.2.3	Blind Source Separation using Independent Component Anal- ysis	127
5.3	Watermark Embedding	129
5.3.1	Private Watermark Embedding	130
5.3.2	Public Watermark Embedding	131
5.4	Watermark Extraction	135
5.4.1	Public Watermark Extraction	135
5.4.2	Private Watermark Extraction	136

TABLE OF CONTENTS

5.5	Copyright Protection Protocols	137
5.5.1	Watermarked Image Generation and Distribution Protocol	138
5.5.2	Buyer-Seller Identification Protocol	141
5.5.3	Copyright Infringement Protocol	143
5.6	Resolving Multiple Ownership Claims	145
5.6.1	Multiple Watermarked Image Attack	146
5.6.2	Invertible Watermark Attack	146
5.6.3	Watermark Removal Attack	151
5.7	Security Analysis	153
5.8	Improvement in the Key Size and System's Robustness to JPEG Compression	156
5.9	Conclusion	159
6	Conclusion and Future Directions	161
6.1	Conclusion	161
6.2	Future Directions	164
	Publications	167
	Bibliography	170

List of Figures

1.1	Illustration of malicious and non-malicious manipulation in an image. (a) Original Cameraman image. (b) Tampered version of the Cameraman image. (c) Compressed version of the image shown in Fig. 1.1a, JPEG QF=50.	4
3.1	Encryption and decryption results for the Cameraman image. (a) Cameraman image (plaintext-image). (b) Encrypted Cameraman image. (c) Decrypted Cameraman image: PSNR=43.2dB.	58
3.2	Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=39.8dB) when the encrypted Cameraman image data was JPEG compressed with QF=100. (b) Decrypted output (PSNR=42.7dB) when the encrypted Baboon image data was JPEG compressed with QF=100.	60
3.3	Effect of JPEG compression on the decrypted output (a) Decrypted output (PSNR=24.2dB) when the encrypted Cameraman image data was JPEG compressed with QF=90. (b) Decrypted output (PSNR=24.4dB) when the encrypted Baboon image data was JPEG compressed with QF=90.	61

LIST OF FIGURES

3.4	Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=21.5dB) when the encrypted Cameraman image was JPEG compressed with QF=80. (b) Decrypted output (PSNR = 21.2dB) when the encrypted Baboon image data was JPEG compressed with QF=80.	61
3.5	Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=20.6dB) when the encrypted Cameraman image data was JPEG compressed with QF=70. (b) Decrypted output (PSNR=21.2dB) when the encrypted Baboon image data was JPEG compressed with QF=70.	62
3.6	Effect of encryption on PSNR. (a) PSNR comparison of JPEG decompressed Cameraman image with and without encryption. (b) PSNR comparison of JPEG decompressed Baboon image with and without encryption (c) PSNR comparison of JPEG decompressed Fruit image with and without encryption (d) PSNR comparison of JPEG decompressed Tank image with and without encryption. . .	65
3.7	PSNR of JPEG decompressed decrypted images.	66
3.8	Response of AES Encryption to JPEG compression. (a) AES encrypted Cameraman image. (b) Histogram of AES encrypted Cameraman image. (c) Decrypted output when AES encrypted Cameraman image data was JPEG compressed with QF=100.	67
3.9	Histogram illustration of cipher-images. (a) Histogram of encrypted Cameraman image. (b) Histogram of encrypted Baboon image. (c) Histogram of encrypted binary image. (d) Histogram of encrypted noise (uniform randomly distributed) image.	69

LIST OF FIGURES

3.10	Effect of repeated multiplication on the cipher-image. (a) Minimum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.4. (b) Maximum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.4. (c) Minimum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.5. (d) Maximum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.5. . . .	71
3.11	Effect of tampering on the decrypted output. (a) Decrypted output when a 32×32 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (b) Decrypted output when a 64×64 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (c) Decrypted output when a 128×128 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (d) Decrypted output when the encrypted Cameraman image was replaced by encrypted Baboon image.	73
3.12	Throughput and speed analysis of the proposed scheme. (a) Percentage of time occupied by each task. (b) Throughput in Mbps when SVD is applied at odd iterations. (c) Throughput when SVD is calculated once before the start of iterations.	76
3.13	Checking effective key space by generating 1000 random R_x matrices. (a) Comparison result between $(R_a, R_b) \in R_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (b) Comparison result between $(U_a, U_b) \in U_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (c) Comparison result between $(V_a, V_b) \in V_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (d) Comparison result between $U_a \in U_x$ and $V_a \in V_x$, where $1 \leq x \leq 1000$	79

LIST OF FIGURES

3.14	Illustration of sensitivity of the seed δ_x towards the cipher-image and the decrypted-image. (a) Comparison of a cipher-image with cipher-images generated using different seed values. (b) Comparison of the decrypted-image with the actual plaintext-image for different values of δ_x . The value δ_{500} is the correct seed value to recover the plaintext-image.	81
3.15	Illustration of Chosen-plaintext attack. (a) Cipher-image after one iteration. (b) Cipher-image after two iterations. (c) Cipher-image after three iterations.	88
4.1	Hash Generation Module.	96
4.2	Image Verification Module.	98
4.3	Hash Generation Module with Quantization.	100
4.4	Image Verification Module with Quantization.	100
4.5	Illustration of quantization procedure.	102
4.6	System robustness to JPEG compression, low pass and high pass filtering. (a) Effect of JPEG compression. (b) Effect of Low pass filtering. (c) Effect of High pass filtering.	108
4.7	Minute tampering that may go undetected at small threshold. (a) Original Baboon image. (b) Tampered Baboon Image. Tampering is shown inside the circle.	110
4.8	Change in $\max(E(p, q))$ as the receiver receives compressed version of the Baboon image with varying JPEG QF.	110
4.9	Tamper detection capability of the proposed scheme. (a) Original Cameraman image. (b) Tampered Cameraman image. Tampering is shown inside the circle. (c) Result of tamper detection.	112
4.10	The Receiver Operating Curve.	114

LIST OF FIGURES

4.11	The maximum value of wavelet coefficient for each image in the database.	115
4.12	For smooth texture blocks, most of the DCT AC coefficients are zero. (a) An 8×8 block marked on the Cameraman image. (b) DCT coefficient of the 8×8 shown in (a).	117
4.13	The proposed RPM technique increases the entropy of the feature space. (a) RPM-transformed Cameraman image. (b) DCT coefficient after the 8×8 block shown in Fig. 4.12a is RPM-transformed.	117
4.14	Effect of the change of secret key K_1 on the intermediate hash. (a) Number of blocks identified as malicious from a total of 256 blocks. (b) Magnitude of error between the hashes.	119
4.15	Probability of collision of hash values.	120
5.1	Block diagram of watermark embedding.	130
5.2	Illustration of the public watermark. (a) Public Watermark. (b) Segment 1 of the public watermark. (c) Segment 2 of the public watermark.	132
5.3	Watermark embedding. (a) Original image. (b) Intermediate watermarked image. (c) Watermarked image. (d) Public-watermark key.	134
5.4	Block diagram of public watermark extraction.	136
5.5	Public watermark extraction. (a) Public Watermark. (b) Segment 1 of the public watermark. (c) Segment 2 of the public watermark.	136
5.6	Flow diagram of watermarked image generation and distribution protocol.	139
5.7	Block diagram of buyer-seller identification protocol.	142
5.8	Block diagram of copyright infringement protocol.	145

LIST OF FIGURES

5.9	Public watermarks. (a) Segment 1 of the public watermark. (b) Segment 2 of the Public Watermark.	158
5.10	Watermark extraction in the spatial domain. (a) Extracted public watermark at JPEG QF=100. (b) Extracted public watermark at JPEG QF=75.	158
5.11	Wavelet domain public watermarks. (a) Segment 1 of the public watermark. (b) Segment 2 of the public watermark.	158
5.12	Watermark extraction in the wavelet domain. (a) Extracted public watermark at JPEG QF=100. (b) Extracted public watermark at JPEG QF=75.	159

List of Tables

3.1	Robustness to JPEG compression: Cameraman image.	62
3.2	Robustness to JPEG compression: Baboon image.	63
3.3	PSNR of decrypted images without JPEG compression.	65

Chapter 1

Introduction

1.1 Background

For the past several years, a lot of research work has been underway to devise security techniques to protect multimedia data like audio, image and video. A question that instantly comes to mind is why new techniques for multimedia security are required when the field of cryptography is already well matured? For example, if an image needs to be transmitted over the Internet in a secure manner, symmetric key encryption techniques like the AES [1], [2] or public key algorithm like the RSA [3], [4] can be used. Similarly, if authentication of an image is required, it can be done using cryptographic hash functions like MD5 or SHA1 along with the RSA algorithm [3]. In other words, if multimedia data is considered as a bit stream, then the field of cryptography seems to provide the required solution. There are, however, several reasons that actually impede the direct use of cryptographic techniques for solving multimedia security problems.

In this thesis, three closely related problems pertaining to multimedia security; encryption, authentication and copyright protection are addressed. Specifically, encryption, authentication and copyright protection algorithms are developed for digital images. This chapter provides the basic ground work towards defining the fundamental goals of this thesis. General aspects of data security are introduced in Section 1.2. In Section 1.3, core issues related to image encryption, authenti-

cation and copyright protection are presented along with the rationale that why traditional cryptographic techniques are not adequate to solve multimedia security problems. As a result of this discussion, the motivation behind this thesis is presented in Section 1.4. In Section 1.5, the original contribution of this thesis is outlined. This is followed by thesis structure and concluding remarks in Sections 1.6 and 1.7, respectively.

1.2 Basic Requirements of Data Security

Generally speaking, data security encompasses four important aspects; confidentiality, integrity, authentication and non-repudiation [5], [6]. These aspects are briefly described as follows:

1. **Confidentiality:** This feature enables only the authorized users to see the data. Encryption algorithms are employed to achieve data confidentiality.
2. **Integrity:** Integrity means that the received message has not been altered during transmission. Even if a single bit of the received message is changed, the integrity of the message will be lost.
3. **Authentication:** Authentication enables two or more communicating parties to identify each other. If information is transmitted over a channel, the authentication function enables to check its origin, date, integrity of the information, information source and intended recipient(s).
4. **Non-repudiation:** Non-repudiation ensures that in future, a sender cannot deny his/her previous commitments or actions.

1.3 Multimedia Security

Before multimedia security issues are discussed, it is important to understand the difference between multimedia data and multimedia content. Multimedia data

represents the exact data values in a multimedia bitstream whereas multimedia content refers to the meaning or semantic of multimedia data. For example, if an image is JPEG compressed, its pixel values might change but the literal meaning i.e. its contents remain intact as long as compression does not cause severe distortion in the image. Multimedia data like image, video or audio is exposed to two types of distortions; *malicious* and *non-malicious*. Malicious manipulation means that the data values are changed in such a way that the content of the multimedia signal is changed. For example, in Fig. 1.1a an original image (Cameraman) is shown while in Fig. 1.1b the tampered Cameraman image is shown. Tampering is done by increasing the length of the lens of the camera in the image shown in Fig. 1.1a. This is an example of malicious tampering. On the other hand, Fig. 1.1c shows the JPEG compressed version of the Cameraman image; the JPEG quality factor was kept at 50. The peak-signal-to-noise ratio (PSNR) [8] between the two images is 31dB. The value PSNR indicates that pixels value of the images shown in Fig. 1.1a and Fig. 1.1c are not the same. However, from the perceptual point of view, the two images are same. This is an example of non-malicious manipulation that changes the pixel values, but keeps the semantic of the multimedia data intact.

This brings us to a very important point; a multimedia security algorithm has to face a different type of environment as compared to conventional text-based security systems. For example, in text-based security, the transmission channel is lossless, hence there is no issue of non-malicious operation taking place to the data. For example, in a conventional text-based authentication system, if any single bit of data is changed, the authentication system will report a failure in verification. This is not the case with multimedia systems; the authentication algorithm should by-pass any non-malicious distortion that does not alter its basic semantic. On the other hand, any minute tampering that alters the semantic of the multimedia signal should be positively detected. This makes the design of multimedia security system a very challenging task as sometimes it becomes extremely difficult to draw a boundary that separates the malicious and non-malicious distortion. One of the fundamental aims of this thesis is to come up

(a)

(b)

(c)

Figure 1.1: Illustration of malicious and non-malicious manipulation in an image. (a) Original Cameraman image. (b) Tampered version of the Cameraman image. (c) Compressed version of the image shown in Fig. 1.1a, JPEG QF=50.

with security algorithms that besides providing high level of security can work in a lossy compression environment.

In recent years, a number of interesting and novel security technologies have been developed for multimedia applications that involve images, video and or audio [5] - [7]. It should be noted that the secrecy of a security algorithm does not rest on the obstruction of the algorithm, but rather on the secret key as outlined by Kerckhoff [3]. The secret key, ideally, should allow only the genuine participants to use the algorithm effectively and prohibit all others. This thesis is specifically focused towards developing encryption, authentication and copyright protection algorithms for digital images. Before the actual motivation and contribution of this thesis is presented, it is imperative to outline the fundamental requirements of image encryption, authentication and copyright protection schemes.

1.3.1 Image Encryption

The process of disguising a message is called encryption [4]. The unencrypted message is called the *plaintext* and the encrypted message is called the *ciphertext*. The process of obtaining the plaintext back from the ciphertext is called decryption [4]. There are two types of algorithms used for encryption; *Symmetric algorithms* and *Public-key algorithms*. In symmetric algorithms, the encryption key can be calculated from the decryption key and vice versa. In most of the symmetric algorithms, the two keys are same [4]. In public key algorithms (also referred to as asymmetric algorithms), encryption and decryption keys are different. The encryption key is made public so that anyone can encrypt a message, however, only the person who has the correct private key can decrypt the message. It is believed that in a reasonable amount of time, it is infeasible to calculate the decryption key from the encryption key [4].

Nearly all the still image encryption algorithms proposed in the literature are symmetric. The following properties are always desired for any still image encryption algorithm [9]:

1. The encryption procedure should not decrease the compression ratio or increase the size of the image.
2. The encryption algorithm should be robust against the general digital image processing procedures and should have error resilience. Networks may cause transmission errors and packet loss.
3. The decryption procedure should not degrade the quality of the decrypted image.
4. The encrypted image should have no perceptual correlation with the actual unencrypted image.
5. The encryption/decryption algorithm has to be very fast in order to meet the performance requirements of real-time applications.

There are two levels of security for digital image encryption; low-level security encryption and high-level security encryption [9]. In low-level security encryption, the encrypted image has degraded visual quality compared to that of the original one, but the content of the image is still visible and understandable to the viewers. In the high-level security case, the content is completely scrambled and the image looks like random noise. An encryption algorithm besides holding these properties should be strong enough to withstand cryptanalysis. Cryptanalysis is the science of breaking the encryption machinery to recover the plaintext and or the secret key used by the encryption machinery [4]. Generally there are four types of cryptanalytic attacks [3], [4]:

1. **Ciphertext-only attack.** The attacker has the ciphertext of one or more plaintext messages. The objective of the attack is to recover the plaintext and or the secret key.
2. **Known-plaintext attack.** The attacker has access to one or more ciphertext and the corresponding plaintext messages. The objective is to find the secret key.

3. **Chosen-plaintext attack.** The attacker besides having ciphertext and corresponding plaintext messages also has the liberty to choose a plaintext of his/her choice and get the corresponding ciphertext. Again the objective is to find the secret key. This attack is more powerful since the attacker can choose plaintext of his/her choice. If the underlying encryption mechanism is weak, the attacker can choose a plaintext whose ciphertext output by the encryption machinery leaks information that may be used to guess the secret key.
4. **Chosen-ciphertext attack.** The attacker can choose ciphertext and get the corresponding plaintext. This attack is more applicable to public key cryptosystems [4].

A good encryption scheme should be capable enough to withstand all these attacks.

1.3.2 Hash-Based Image Authentication

A hash provides a compact representation of any data. Primarily hash is used to form digital signature for the purpose of authentication [3], [4]. Consider the example of a digital image. A simple way to authenticate digital images is to use cryptographic hash functions like MD5 or SHA1 [3], [4] along with public key encryption algorithms such as the RSA [3], [4]. One of the biggest problems of using cryptographic hash functions for image authentication is that even a single bit change in the image will produce a completely different hash value. In practice, it is common that a digital image may undergo some content preserving manipulations such as compression, enhancement, etc. These operations may not change the visual appearance of the image, however, the cryptographic hash value will be completely different. From this discussion, we note that multimedia image authentication requires techniques which should be somewhat resilient to content preserving manipulations like JPEG compression, while at the same time be fragile enough to detect malicious manipulations. For digital images, the hash

function is referred to as perceptual hash function (PHF).

Following are some important requirements that are expected from a PHF used to authenticate an image [10], [11]:

1. A PHF should be robust against non-malicious manipulations, for example, lossy compression, low pass/high pass filtering, minor contrast enhancement, etc.
2. A PHF should be highly sensitive to malicious tampering and should be able to detect the location of the tampering.
3. A PHF should be key dependent. Without the knowledge of the correct key, it should be extremely difficult to generate the correct hash.
4. A key-based PHF should satisfy the property of modified weak collision [10]. According to this property, given an image, I and its corresponding robust hash, $H(I)$, it should not be feasible to find another image, P such that $H(I) = H(P)$ when I is significantly different from P .
5. A PHF should have a high level of security against counterfeit attacks or attacks that are meant to fool the authentication system.
6. By looking at the hash values generated by a PHF, it should be extremely difficult for an attacker to guess the contents of the image or reverse engineer the secret key that was used to calculate the hash.

1.3.3 Watermarking for Copyright Protection

Digital watermarking is a technique that provides a way to protect digital images from illicit copying and manipulations. A digital watermark is an imperceptible signal added to digital data, called cover work, which can be detected later for buyer/seller identification, ownership proof, and so forth [8]. Depending on the type of application, a watermark can either be visible or invisible. For example, in most of the currency bills, a visible watermark is embedded that helps to

distinguish between genuine and fake currency. Invisible watermarks find their usage in a number of multimedia applications like copyright protection, fingerprinting, broadcast monitoring, etc [5]. In recent years, a number of techniques have been proposed in the area of digital watermarking for audio, imaging and video applications [5], [6]. Typical uses of watermarks include identifying the origin of content, tracing illegally distributed copies, and disabling unauthorized access to content.

Watermarks designed for copyright protection, fingerprinting, or access control must be embedded in a secure manner. This means that an attacker who knows all the details of the embedding-extracting algorithms except the secret key should not be able to detect or extract the watermark. Secondly, watermarks should be robust i.e. they should remain intact within the image that undergoes common image processing operations, like lossy compression, filtering, noise addition, histogram manipulation, and various geometrical transformations. The underlying watermarking algorithm either alone or with a set of protocols should be resistant to a number of attacks, as discussed in Chapter 2 and Chapter 5 of this thesis.

1.4 Motivation

From the discussion in Section 1.3, we conclude that devising encryption, authentication and copyright protection schemes for multimedia, like digital images is not an easy task. The proposed scheme should be designed in such a way that it encompasses a high level of security and at the same time offers resilience to content preserving operations like compression, filtering, etc. This is very different from conventional cryptographic algorithms in which there is no robustness issue. The addition of robustness starts effecting the security of the underlying algorithm. This is the main motivation behind the nuts and bolts that make up this thesis. The theme of this thesis is to come up with encryption, authentication and copyright protection schemes for digital images that possess high-level of security and are also robust to content preserving manipulations. There are

quite a number of robust and secure schemes proposed in the literature in the area of media hashing and watermarking. In the area of image encryption, the proposed image cryptosystem has a unique feature that encryption and decryption algorithms are independent entities and not an integral part of any compression algorithm. Further, it is shown that the proposed scheme has the capability to withstand traditional cryptographic attacks and can tolerate JPEG compression. In fact devising such an encryption scheme was the most challenging part of this thesis.

1.5 Thesis Contribution

This thesis addresses the problem of designing robust and secure techniques in the area of image encryption, authentication and copyright protection. The main contributions of this thesis in each of these areas are as follows:

Image Encryption

1. In the area of image encryption we are interested in devising an image cryptosystem that provides a high level of security and at the same time provides flexibility to work in a JPEG lossy compression environment. In this regard, a JPEG compression tolerant image encryption is proposed in Chapter 3 of this thesis. This scheme relies on some very interesting properties of orthogonal matrices.
2. Detailed mathematical proofs are presented to show how the encryption works with orthogonal matrices. Further, it is statistically shown that the proposed encryption scheme has the capability to hide the original histogram of the plaintext image.
3. Several experimental results are presented to show how the proposed scheme works in a JPEG lossy environment.

4. Detailed cryptanalysis is carried out to show how the proposed image cryptosystem can resist a number of well known attacks, like the brute force attack, ciphertext-only attack, known-plaintext attack and chosen plaintext/ciphertext attacks.

Image Authentication

1. A robust and secure hashed-based authentication scheme using wavelet transform is proposed for image authentication. The primary motivating factor behind this proposal is to encompass robustness, tamper detection with localization capability and security against various attacks designed to break a hash-based system.
2. Instead of finding some image features first and then randomizing these features, the proposed scheme first makes the initial space from which the features would be extracted as random.
3. Detailed experimental results are presented to show how the proposed scheme can tolerate JPEG compression, low-pass and high-pass filtering.
4. A detailed security analysis is carried out to show the practical viability of the proposed scheme in case of attacks that are meant to find the secret hash function, or to create counterfeit images that could be correctly authenticated.
5. A 4-bit quantization scheme is proposed to reduce the size of the hash. Thorough analysis of the proposed quantization scheme is carried out to ascertain its robustness, tamper detection and security capabilities.

Copyright Protection

1. A robust and secure wavelet-based watermarking scheme is proposed using Independent Component Analysis.

2. The proposed scheme is further integrated with public key cryptography to devise a set of protocols to identify genuine buyer and seller of a water-marked image.
3. A copyright infringement protocol is proposed that enables the seller to identify the specific buyer from whom an illegal copy of a watermarked image has originated and further prove this fact to a third party.
4. Protocols to resolve multiple ownership claims are proposed to thwart a number of attacks common in copyright protection systems.

1.6 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 3, a JPEG compression tolerant image encryption scheme is proposed that is devised using multiplicative properties of orthogonal matrices. The proposed scheme besides providing high level of security against various cryptographic attacks can also tolerate JPEG compression. This means if the encrypted image is JPEG compressed up to a certain quality factor, decryption is still possible with some distortion. Detailed security analysis is presented that shows the viability of the proposed scheme. In Chapter 4, a secure and robust hash-based image authentication scheme is presented. This scheme uses wavelet transform to generate the hash and randomized-pixel modulation to achieve security. In the same chapter, a 4-bit quantization scheme is also proposed that can be used to reduce the size of the hash. In Chapter 5, a secure and robust watermarking scheme is presented that effectively integrates watermarking with cryptography to address issues related to buyer-seller identification and copyright protection. Concluding remarks and future research directions are presented in Chapter 6.

1.7 Chapter Summary

In this chapter, the need for multimedia security and the fundamental requirements that an image security system should meet were presented. The discussion presented in this chapter was instrumental in designing image security algorithms presented in Chapters 3, 4 and 5. The next chapter presents a comprehensive literature review about state-of-the-art image security technologies and relevant issues.

Chapter 2

Image Security and Review of Related Work

2.1 Introduction

In this chapter, several important issues related to image security are presented by reviewing a number of techniques proposed in the literature related to encryption, authentication and copyright protection of digital images. Each section to follow, sets the background and presents a comprehensive review in the respective area. In Section 2.2 image encryption schemes are reviewed, while Section 2.3 and Section 2.4 present review of work related to image hashing and copyright protection.

2.2 Review of Image Encryption Schemes

The tremendous growth in multimedia technology has led to widespread use of digital multimedia services and applications. However, most of the multimedia transmission medium and storage devices are generally not secure. Hence there is a need to develop technologies that can protect multimedia data from unauthorized access and usage. Multimedia encryption is a technology that can be effectively used to provide confidentiality and access control mechanism to digital contents such as images, video and audio. Using traditional cryptographic tech-

niques to encrypt multimedia data is possible and this technique is called naïve encryption [5]. In a multimedia environment, it is the delivery of the content that is usually important and not the exact data values. It is because of this reason that traditional ciphers are not effective to directly encrypt multimedia data because they treat multimedia data as a bit stream and generally do not utilize the intrinsic structure of the data [7]. The transmission of multimedia data usually has issues such as bandwidth adaptation, error protection, etc. Further, multimedia data often has high redundancy, large size and requires real-time processing. Another important requirement of multimedia data is that after encryption it should be perceptually distorted. Generally speaking, multimedia encryption has three important applications [6] mentioned below.

1. **Confidentiality:** Multimedia encryption prevents unauthorized leakage of the contents during storage or transmission.
2. **Access Control:** Authorized access to subscription-based services like satellite and cable TV channels, pay-per-view, etc is possible using multimedia technologies.
3. **Digital Rights Management (DRM):** A DRM system provides a mechanism for persistent rights management of multimedia content right from its inception to usage. Multimedia encryption is a core component of a DRM system.

Multimedia encryption can be broadly classified into three types; image encryption, video encryption and audio encryption. Though multimedia encryption technology can be traced back to 1980 [12], however, serious work in this area started in 1990. Most of the initial work was based on the idea of randomly permuting the pixels of an image to get the encrypted image [12], [13], [14]. Pixel permutation however makes the adjacent pixels de-correlated thus effecting the compression performance. Further, it has been shown in [15] that permutation-based encryption techniques are highly vulnerable to chosen plaintext and ciphertext attacks. These factors along with advances in multimedia technology in the

Image Security and Review of Related Work

90s motivated researchers to develop new encryption techniques especially tailored to meet the requirements of multimedia applications such as compression, error recovery, transcoding, etc .

- *Raw image encryption*: One possibility is to perform encryption before it enters the compression stage. This is the simplest technique that is referred to as naïve encryption. As pointed out by Shannon [16], encryption and compression both tend to remove the redundancies in data. Hence encrypting an image before it enters the compression stage shall effect the performance of the compression stage. Further, naïve or raw image encryption does not take into account the image semantic and treats the entire image as textual data. In case of video, audio or images with large sizes, this can be a potential bottleneck in meeting real-time requirements. The advantage of raw encryption is that it can be done independently and does not need any modification in the image encoder.
- *Encryption after compression*: Encryption after compression has the advantage that compression reduces the redundancy thus making cryptanalysis difficult. However, compression introduces known patterns in the compressed bitstreams like headers and markers that can facilitate plaintext attacks [7].
- *Encryption during compression*: The compression algorithm can be modified so that encryption and compression can be done simultaneously.
- *Encryption of the final compressed bitstream*: Another simple way is to encrypt the entire multimedia sequence using some well known encryption cipher, like the DES, AES, RSA, etc [1]- [4]. However, in such a case, it would not be possible to perform any intermediate processing like bit rate adjustment and or DC image extraction for content searching without having access to the decryption key.

A comprehensive review of image encryption scheme is now presented. Image and video encryption technologies are closely knitted. Due to this reason,

advancement in either technology has benefited the other. Although the review presented is for image encryption schemes, video encryption techniques may also be referred to present a more in-depth knowledge to the reader. The outcome of this review has provided many insights to come up with a new image encryption scheme that is presented in Chapter 3 of this thesis.

Permutation Based Encryption

Early research to protect multimedia signals was mainly based on scrambling techniques using permutation and or affine transformation either in time or frequency domain. One such example is the raw data encryption in which the encryption process is directly applied to the raw image or video data. One of the simplest and the fastest way for raw data encryption is to apply permutation. For example, pay-per-view TV programming [13] uses a permutation algorithm that permutes the TV field line by line. Permutations can also be done using chaotic maps. Scharinger [14] was the first to apply a class of such maps called Kolmogorov. The initial value of the chaotic map is taken as the coordinates of the image pixels. The chaotic map is then iterated to get new coordinates which are random. The control parameter of the chaotic map acts as the encryption or decryption key. Permutation has also been used in the first MPEG encryption algorithm that was reported by Tang [17]. In this proposal, the quantized DCT coefficients obtained from the zig-zag scanning are permuted using a random permutation list. Instead of mapping the 8×8 DCT blocks to 1×64 vectors in the zig-zag order, Tang proposed to reorder the 8×8 blocks to 1×64 vectors according to a random permutation list. This method increases the overall bit rate reducing the efficiency of the Huffman encoding stage. Realizing the fact that most videos start with standard header clips, Tang proposed two permutation lists. Qiao and Nahrstedt [18] have pointed out how a known-plaintext attack can be launched to break the permutation lists used in Tang's algorithm [17].

Generally speaking, raw data encryption is a simple encryption technique that can change the entire semantic of an image/video thus making it unintelligible.

However, a problem with this technique is that it changes the entire statistics of the image, video or audio data. This can have a negative effect on the compression process. Therefore raw data encryption is better suited for applications that do not need compression like TV signals, uncompressed image/video etc. There are a number of cryptanalysis indicating that permutation-based image and video encryption schemes are not sufficiently secure from a cryptographical point of view. Firstly it is well established that secret permutations are insecure against chosen plaintext and known plaintext attacks. Secondly, since natural images and videos have large redundancy, a ciphertext-only attack may lead to partial information leakage [18], [15].

Selective Encryption

In recent years, research in multimedia encryption has been focused towards integrating encryption and compression. The basic idea is to encrypt multimedia data in such a way that the encryption overhead is small, and secondly, the encrypted data can be represented in a standard compliant format. The term that is often used in the literature to describe such encryption schemes is *selection encryption* or *partial encryption*. Spanos and Maple [19] were the first to introduce the idea of selective encryption in 1995. Encryption and compression have a strong link as pointed out by Claude Shannon in his classical paper “Communication Theory of Secrecy Systems” [16]. Lookabaugh and Sicker [20] have pointed out Shannon’s perspective that redundancy in a source can make encryption weak. Shannon suggested that removing redundancy in the source can strengthen the encryption process. The basic idea behind selective encryption is to encrypt selected coefficients from either the intermediate or the final result of a compression system. The coefficients that are not selected are sent unencrypted. If no meaningful information can be deduced from the non-selected coefficients, then the security level of the system is same as if all the coefficients have been encrypted.

Cheng and Li gave the notion of *partial encryption*, which is also called *selective encryption*. Specifically, they have proposed image encryption for quadtree com-

pression algorithms and wavelet compression algorithms based on zerotrees [21]. The authors then extend the idea of image encryption to video encryption. One of the main reasons for the authors to select these two algorithms is because both of them are suitable for low bit rate applications. The quadtree compression algorithm produces two logical parts; the quadtree structure and the parameters that describe each block in the tree. Considering that the parameter used to describe each block is the average intensity, Cheng and Li proposed to encrypt the quadtree structure only. They have shown that certain leaf orderings are prone to cryptanalysis, hence choosing the correct leaf ordering is important for the security of the encryption. The second algorithm that is proposed works with the zerotree wavelet compression. These algorithms generally transmit the structure of the zerotrees along with the significant coefficients. One such example is the SPIHT compression algorithm [22] which the authors have used to demonstrate their encryption scheme. In the encoding stage, this algorithm generates different types of significant bits, for example, sign bits, refinement bits, significance of pixels and significance of sets. They exploit the fact that even if a small amount of significant information is missing, the SPHIT decoder cannot perform correct decoding. Based on this argument, Cheng and Li proposed to encrypt only the significant information in the two highest pyramid levels of the wavelet transform. Cheng and Li argue that it is difficult to cryptanalyze this scheme because the unencrypted data contains different types of bits which cannot be identified unless the significant information is obtained by correct decryption. Further, there is many-to-many mapping between the significant part and the non-significant part produced by the compression algorithm thus making it difficult to extract useful information for cryptanalysis. One main limitation of this scheme as pointed out by the authors is that for each compression scheme, a different encryption scheme needs to be devised to get optimum results in terms of performance and security.

Wu and Kuo [23] describes several important issues that must be taken into account while devising a selective encryption scheme for multimedia. Through a number of experimental results, they argue that many selective encryption schemes fail to properly encrypt image/video contents. Using the Lena image as

one of the examples, they demonstrate the security weakness in Tang's scheme [17] that encrypts the DCT DC coefficients of each block and performs scrambling on the AC coefficients. After making all the DCT DC coefficients equal to 128, Wu and Kuo were still able to recover the edges of the Lena image. They further demonstrate that even if nine or half of the AC coefficients are encrypted in the zig-zag pattern, the image semantic can still be recovered. The recovery was done simply by setting the encrypted coefficient to their mean value and performing some simple image enhancement techniques. In a similar way, they also did cryptanalysis on the scheme proposed by Shi and Bhargava [24] that encrypt the sign bit of every DCT coefficient in JPEG and every motion vector in MPEG. Using the Lena image as an example, they were able to recover the edges by simply assigning all DCT DC coefficients to 128 and making all DCT AC coefficients sign as positive. Surprisingly, edges information was recovered even though DCT AC coefficients up to four significant bits were encrypted. Wu and Kuo concluded that selective encryption is not feasible for a compression system that has an orthogonal transform followed by a scalar quantization, for example, JPEG, MPEG and MP3 audio compression, because the image/video/audio semantic is scattered among all the frequency components. Further, performing selective encryption before the entropy coding stage affects the compression performance [23], [25]. Similarly, if selective encryption is performed after the entropy stage, then it becomes difficult to identify the bits that are closely related to the semantic of the multimedia signal. After defining these constraints, Wu and Kuo proposed a multimedia encryption scheme to fit within the entropy coding stage [23]. They used multiple Huffman coding tables to implement a secure Huffman coder. The data stream was encoded with the multiple Huffman coding table. The secret key for encryption and decryption is the contents of these tables and the order in which they are used. A challenge in this scheme is to come up with Huffman tables that do not affect the compression performance. The authors tackle this issue by permuting four different Huffman trees to derive a large number of Huffman trees using a technique which they define as *Huffman tree mutation*. For a Huffman table with t entries, 2^{t-1} Huffman trees can be de-

rived. Security of this scheme lies in the fact that if an attacker does not know the order in which the trees are used in the encryption process, correction decryption will not take place because of lack of synchronization between the symbol stream and the encoded bitstream. This scheme has the capability to resist ciphertext only and known-plaintext attack, however, it is vulnerable to chosen-plaintext attacks as reported by the authors. Some suggestions are also presented to make the scheme resistant to the chosen-plaintext attacks.

Wen, Muttrell and Severa [26] proposed a format compliant encryption scheme by assigning a fixed-length index to each variable length codeword (VLC). The indices are then concatenated and encrypted. The encrypted indices are then mapped back to the codeword domain to form the encrypted bitstream. Mao and Wu [27] have pointed out that this scheme works well with Huffman codes and Golomb-Rice code that associate each symbol coming from a finite set with a unique codeword of integer length. However, applying this scheme directly to VLC schemes that allow fractional codeword length per symbol is not feasible. To address this issue, Mao and Wu have proposed a *Generalized Index Mapping* approach [27] which is the extension of [26]. Instead of applying fixed-length index to VLC as in [26], the authors in [27] proposed to apply encryption directly to symbols. After encryption, any appropriate VLC coding scheme can be applied to the encrypted bitstream. For a large sequence, this method produces uniformly distributed encrypted symbols. This in turn increases the entropy of the encrypted symbols thus increasing the bit-rate overhead. To overcome the bit-rate overhead issue, a partition methodology is proposed that partitions the symbol range into more than one subset and restricts the output of the encryption to be in the same subset to which the input symbols belongs. The choice of partition should be done carefully as it increases the chance of attacks. The other method proposed for encryption is called the *Intra Bitplane Shuffling*. This technique can be applied to MPEG4 video that uses fine granularity scalability (FGS). FGS uses a base layer to provide basic quality level at a lower bit rate and the enhancement layer that provides further refinement. To achieve FGS, the enhancement layer is formed by encoding bitplanes starting from the most

significant bitplane. Within an image block, each bitplane is represented by the runs of zeros and the end-of-plane (EOP) flag. The run-EOP symbols are encoded using VLC codes and interleaved with sign bits. Mao and Wu proposed to apply random shuffling to each bit plane [27]. The randomly shuffled bitplanes are then encoded using run-EOP approach. Further, the sign bit of each coefficient is also randomly flipped using one-time pad. The number of bitplanes should be carefully selected keeping in view the required perceptual distortion. One problem with this approach is that different shuffle tables should be used for different bitplanes and the shuffling tables should be changed frequently to avoid known plaintext and chosen plaintext attacks.

Van Droogenbroeck and Benedett [28] have proposed selective encryption schemes for uncompressed images and for compressed JPEG images. They observe that a gray level image is made up of eight bitplanes. The most significant bitplanes are correlated to the gray level image while the least significant bitplanes look random. The proposed idea is to encrypt the least significant bitplanes so that an image becomes noisy. Droogenbroeck and Benedett however observe that at least four to five least significant bitplanes should be encrypted to get visual distortion in the image. They argue that this method is relatively robust to plaintext attacks because encryption is performed on the bitplanes that contain uncorrelated bit values. The second method proposed by Droogenbroeck and Benedett can be used for JPEG image encryption. In JPEG compression, the codewords formed by the Huffman coder are followed by appended bits that specify the sign and magnitude of the non-zero coefficients. This scheme is good for progressive image transmission where the encrypted image can be considered a low quality version and a client can pay to get the decrypted high quality version. However, both the methods do not make the image random and cannot be used if the image information is to be completely hidden. As reported in their paper, for Lena image, the PSNR for 7 bit plane encryption is 13.8 dB; the image semantic is clearly visible. Similarly for the second method, the PSNR is 17.7 dB when only the DCT DC coefficient is left unencrypted.

Podesser, Schmidt and Uhl [29] look at bitplane encryption from a different point of view. They argue that visual distortion can only be achieved if most significant bitplanes (MSB) and a few other bitplanes after the MSB are encrypted. For selective encryption, they propose to encrypt at least 12.5% of the MSB, increasing in steps of 12.5% for each additional bitplane. They used AES [1] with a key size of 128 bits. The authors presented some important results related to the security of bitplane selective encryption in the spatial domain. Through *replacement attack*, they show how the structural information of an image can be revealed if only the MSB and the next bitplane is encrypted. This attack is carried out by replacing the encrypted data by similar looking data taken from other images. As mentioned in [29], the encrypted bitplane is replaced by zero, and the resulting decrease in the average intensity is adjusted by adding 64 to each pixel if only MSB is encrypted, 94 if MSB and the next bitplane is encrypted, and so on. The next attack illustrated in [29] is called the *reconstruction attack* which is designed to get back the original unencrypted image. Assuming that only the MSB is encrypted, the idea is to get back the unencrypted image using the other bitplanes that are left in the clear. The authors exploit the fact that in natural images, there are many regions where the change in texture is smooth due to which the MSBs of most of the pixels are close in their respective values. Once such areas are detected, a binary operation is performed to separate the smooth texture areas thus making the edges visible. Using two different versions of reconstructed images, the authors demonstrate how the original unencrypted image can be recovered. It is not clear whether the recovered image has same pixel values as that of the original. However, the perceptual quality is as good as the original image.

Zeng and Lei [30] have proposed frequency domain selective scrambling schemes for digital video. Though the scheme is meant for video, however it is also applicable to images. Encryption is done directly in the wavelet and the DCT domain. The basic notion is that some bits of individual transform coefficients have high entropy. These bits can be selectively encrypted without having significant effect on the compression performance. In addition, shuffling in the transform domain

can help to achieve security as long as shuffling does not destroy the low entropy aspect of the transform domain map that is relied upon by the bitstream coder. The authors point out that a transform coefficient is made up of three types of bits; the significant bit (the MSB), the refinement bits and the sign bits. The significant bit has low entropy and is therefore highly compressible. However, the sign and the refinement bits have high entropy and can be used for encryption. The first strategy is to flip these high entropy bits using some key-based cryptographically secure pseudo-random number generator. In addition, block shuffling and block rotation is also proposed to further increase the security level. Each subband is divided into equal number of blocks. The block size can be different for different blocks. For each subband, blocks of coefficients can be shuffled using a different shuffling table. For layered encryption, the authors suggest to scramble only the high-resolution subbands. The proposed idea of sign encryption, block shuffling and rotation is then extended to 8×8 DCT blocks so that it can be applied to MPEG video compression standards. This also includes scrambling the motion vectors. The results in [30] depict that combining all the three methods, i.e. sign encryption, block shuffling and block rotation render an image perceptually distorted with no clue about the edges. The paper, however, does not present cryptanalysis for the two proposed methods against known plaintext and chosen plaintext attacks.

Amir Said, one of the inventors of the SPHIT algorithm [22] looks at security of selective encryption scheme from a similar perspective. His notion is that in a multimedia encryption environment, security is not only to keep the encryption/decryption key secure [31]. He argues that an otherwise secure scheme would be consider weak if a high quality image can be obtained from a low quality encrypted image. He argues that due to low quality demo images being freely available on the Internet, this attack is practically feasible, requires low complexity and does not need any knowledge of the secret key. Amir demonstrates these attacks on two selective image encryption schemes [21] and [30]. The pixel values of the recovered images are not exactly equal to the original images, however, the quality is as good as the original image. The designer of partial selective image

encryption should also cryptanalyze their scheme keeping in view this attack. Definitely, recovering a high quality image from a low quality encrypted image defeats the basic purpose of encryption.

Grangetto, Magli and Olmo have proposed a multimedia encryption framework [32] that is based on the modification of the arithmetic coder used in image and video coding standards. The notion of *randomized arithmetic coding* (RAC) is introduced to provide randomization in the arithmetic coding (AC) procedure. The basic idea behind the RAC approach is that arithmetic decoding is very sensitive to errors in the compressed data. Any error that occurs in the compressed bitstream is propagated throughout the decoded block. As compared to Huffman coding's ability to recover after a certain number of decoding errors, the AC exhibits a very poor resynchronization capability [33]. The authors in [32] have used this inherent flaw in the AC to design an encryption scheme by providing the decoder with a key to do the proper resynchronization. Specifically, the RAC is based on random organization of the encoding interval thus enabling only the synchronized decoder to undo the randomization. The seed that is used for interval randomization is actually the encryption key that is shared with the encoder and the decoder. The authors in [32] cryptanalyze their scheme by extending the results presented in [33]. In [33], it is shown that if the first m bits of the AC stream are unknown to the decoder, then the number of decoding steps required by an attacker to achieve complete synchronization is lower-bounded by $O(2^{m/2})$. In case of the RAC approach, since all the bits are encrypted, considering p as the block size, the lower bound on the number of attempts for full resynchronization is $O(2^{p/2})$. Taking $p = 1000$, the authors in [32] claim that it is extremely difficult for a decoder to successfully decrypt the stream without knowledge of the secret key.

Van De Ville *et al* [34]. have recently proposed an image encryption scheme based on two-dimensional discrete prolate spherical sequences (DPSS). The work in [34] is basically an extension of Wyner's 1D speech scrambling scheme [35] and has the advantage that it introduces very small expansion in the bandwidth.

The DPSSs are the normalized eigenvectors of a specific matrix which is real and symmetric. These eigenvectors form an orthonormal basis that spans the subspace of sequences with an energy concentration within a certain band that is bounded. Any arbitrary sequence can be encrypted by performing a simple linear transformation of the sequence with the DPSS. The new sequence is then scrambled using a secret matrix. Inverse transformation is then applied to get the scrambled sequence in the original domain. Wyner [35] has shown that bandwidth expansion will be negligible if the smallest eigenvalue of the scrambling coefficient is sufficiently large. This is achieved by only scrambling those coefficients of the DPSS transformed sequence which have large eigenvalues. This makes the scheme proposed in [34] a selective encryption scheme. Besides scrambling, it is proposed in [34] to swamp the coefficients in the DPSS. Further, the secret matrix used in the encryption and decryption is derived from the Hadamard matrix. Li *et al.* [36] have carried out a detailed cryptanalysis of the scheme proposed in [34] showing that the scheme is not at all secure against ciphertext only, known-plaintext, chosen plaintext attacks and chosen ciphertext attacks. Further, they have also shown that if the secret matrix is generated from a square Hadamard matrix, the decryption is not sensitive to the key mismatch thus significantly reducing the key space. Li *et al.* finally conclude that the scheme proposed in [34] can only be used for perceptual encryption and is not capable of providing content encryption to digital images. For further security analysis on selective encryption schemes, readers are referred to [37]- [41]

Chaos-Based Image Encryption

Over the past decade, a substantial amount of work has been done to apply chaos in the field of encryption. Chaos is a phenomenon that exists in nonlinear deterministic systems which are very sensitive to initial conditions and parameters. In addition, the mixing characteristics of chaos are quite beneficial towards designing cryptosystems [42], [43], [44]. The main difference between conventional cryptosystems and chaotic maps is that the former works on a finite set of integers, while the latter is defined on an infinite set of real numbers [42], [43].

There has been a surge towards the use of chaos in cryptography after Pecora and Carroll [45] demonstrated the possibility for self-synchronization of chaotic oscillators. Self-synchronization makes it possible for two chaotic systems to synchronize with each other using a driving signal that may be sent from one system to another through an external source. Chaos-based cryptosystems can be analog or digital [44]. Digital chaos-based systems work with digital computers that use one or more chaotic maps in finite computing precision. A number of chaos-based image encryption schemes have been recently proposed. One of the early work in this area was done by Scharinger [14] who used Kolmogorov flow to make a key-controlled chaotic system for permuting pixels in an entire image. In [46], a chaotic key-based algorithm was proposed that first generates a time series, based on a chaotic map. This time series was then used to make a binary sequence that forms the secret key. The image pixels were then XORed or XNORed with the image pixels to get the cipher image. This method has been found to be weak against chosen/known plaintext attack [47]. A chaos-based symmetric block encryption was proposed in [48] that uses an invertible two dimensional chaotic map torus on a square. Chaos-based block ciphers have the advantage that a large image can be divided into a number of non-overlapping blocks and then the blocks can be serially encrypted [43]. A number of methods have been recently developed for chaos-based image and video encryption [49]. It is pertinent to mention that chaos-based cryptography techniques are still passing through an evolutionary stage as many of them have not been proved to be cryptographically secure [50]- [52]

2.3 Review of Image Authentication Schemes

Compared to cryptographic hash functions [3], [4], the field of image hashing is passing through an evolution stage. Since cryptographic hash functions are matured and well studied for more than a decade, it is very natural to design image hash functions that besides meeting the requirements of multimedia applications follow the security features of cryptographic hash functions. It should

be noted that the objectives of a cryptographic hash function and an image hash function are not exactly the same. For example, there is no robustness or tamper localization requirement in case of a cryptographic hash function. However, a good hash function for an image authentication system should address three important issues: robustness to non-malicious manipulations, ability to detect malicious tampering with localization capability and security. Interestingly, all these issues are related to each other. For example, increasing robustness to non-malicious manipulations generally reduces the tamper detection capability and security of the over all scheme. Similarly, if the system is made very sensitive to detect malicious tampering, the robustness parameter will suffer. Depending on the type of application, a trade-off is usually adopted to balance these three parameters. The security aspect is of paramount importance in any authentication scheme. A compromise in security means complete failure of the authentication system.

In recent years, a number of researchers have proposed many interesting and novel ideas to formulate image hash functions. In 1993, Friedman [53] introduced the idea of a “*trustworthy digital camera*” to authenticate digital images using cryptographic hash functions and public key cryptography. The camera itself is considered a secure environment which contains an embedded circuitry to generate the digital signature of the captured image. The digital signature is generated by first taking the cryptographic hash (MD5 or SHA1 [4]) of the image and then encrypting the hash with the camera’s private key. The receiver can then authenticate the image using the camera’s public key. As long as the camera is kept secure, this scheme is extremely secure as it uses standard cryptographic primitives which are now well established. For this scheme to work, it is necessary that the signature verifier should receive exactly the same image that was used while calculating the signature inside the camera. Even an LSB change in any of the pixel will generate an incorrect digital signature. This is because of the use of cryptographic hash function in calculating the image hash. In a real world scenario, it is difficult to realize this scheme because digital images often undergo content-preserving operations like compression, format change, etc. These oper-

ations may change the pixel values; however they do not change the content of the image.

This limitation motivated researchers to devise content-based hash functions that are robust to non-malicious manipulations and fragile to malicious manipulations. One of the early works in designing content-based hash function for image authentication was carried out by Schneider and Chang [54]. They divided the image into blocks and calculated the intensity histogram of each block. To incorporate details of smaller and larger regions of an image, blocks of different sizes were used. The Euclidean distance between intensity histograms was used to measure the distortion in the content of the image in question. There are two serious limitations with this method. First, an image or a specific block can be manipulated in such a way that its intensity histogram remains same or within the distortion bound. The second problem is the storage requirement as the histograms should be stored in the encrypted form otherwise given the knowledge of the block size, an attacker can easily create forged image without even requiring the genuine image.

Xie, Arce and Graveman [55] used mean of an image block to construct short binary representation of an image which they call Approximate Image Message Authentication Codes (AIMAC). They use the Approximate Message Authentication Code (AMAC) [56] to get the AIMAC. The AMAC is a probabilistic checksum calculated using pseudorandom permutations, masking and majority voting. The image is first divided into non-overlapping blocks of size 8×8 pixels. The mean of each block is computed and the most significant bit of the mean is taken. In this way, a binary map of the image is formed. The AIMAC bits are then calculated by applying the AMAC algorithm to each row and column of the binary map. The algorithm generates short binary hash of around 100-450 bits and survives JPEG compression to some extent. However, since the AIMAC only uses the MSB of the block's mean value to generate the hash, therefore, it is quite easy to manipulate an image block in such a way that the visual appearance of the block is changed but the block is still authenticated.

A somewhat similar signature scheme having a compact hash is proposed by Lou and Liu [57]. They divide the image into non-overlapping blocks of 8×8 pixels and calculate the mean of each block. The mean value is then quantized to get the image hash. A 2-bit code is used to get 4-level intensity quantization of the mean value of each block. For an image of size 256×256 pixels, the hash size reported is 2048 bits, tolerance to compression up to 0.5bpp and tamper localization capability. Another novel feature of their scheme is that it can recover the tampered blocks to some extent thus reducing the re-transmission data. However, since the scheme uses the mean of each block, which is then further quantized to 2 bits, an attacker can easily manipulate the image block without changing its mean quantization value.

Queluz [58] proposed a signature scheme that uses moments and edges of an image. Moments may be a good feature to use from the global perspective; however, local tampering does not have much effect on the moment. Secondly, it is possible to modify an image such that its moment does not change. Edge based features capture more information of the image semantic as compared to moments. However, using edges as feature vector has several issues like the length of the hash, robustness of the edges against non-malicious manipulations, for example, JPEG compression, etc.

Bhattacharjee and Kutter [59] proposed the use of scale interaction model in the wavelet domain to extract visually salient image feature points which are used for authentication. The location of these points is used to form the hash of the image. The results demonstrate that their scheme is tolerant to JPEG compression up to a quality factor of 80. Although their scheme can detect the position of malicious manipulations, however, there are some errors in detecting the exact location of these positions due to wrap-around effect of the wavelet transform.

Fridrich and Goljan [60] proposed a robust hashing algorithm that is based on the observation that visual changes to an image cannot be made without significantly changing the low frequency DCT coefficients. The input image is

divided into blocks of 64×64 pixels and each block is projected on 50 random smooth patterns generated from a secret key. Therefore each block hashes to a 50-bit code. This is one of the pioneering works done in the area of image authentication that uses key-dependent feature extraction. Due to large size of image blocks, the tamper detection capability verses lossy compression is not clear. Radhakrishnan, Xiong and Memon [10] have used the Boosting technique to estimate the hash function, thus making malicious tampering go undetected with very high probability.

Lin and Chang [11] have proposed an image authentication technique that relies on the invariant relationship between any two selected DCT coefficients which are at the same position of two different 8×8 image blocks. To get the hash, the image is first divided into 8×8 pixels non-overlapping blocks and DCT of each block is taken. Based on a secret key, randomly generated DCT block pairs are obtained. For each DCT block pair, a few frequency bins are randomly selected using the secret key. This also includes the mean value of DCT coefficients in each selected position for all the blocks. The purpose of including the mean value is to prevent an attacker from making uniform changes to all blocks. The hash of each block is generated by comparing the selected frequency bins of that block with the other secretly selected block to get a binary value. Different levels of thresholds are used to record the sign of comparison and absolute difference between the coefficients in increasing precision. The relationship between the selected DCT coefficients remains unchanged for compression/re-compression cycle the image has passed through. Due to this reason, this scheme is resilient to JPEG compression while being sensitive towards catching malicious manipulations with localization capability. Radhakrishnan and Memon [61], [62] have shown a method to attack this scheme. Through this attack, the secret mapping key can be found.

Sun and Chang [63] also use a DCT-based approach to generate what they call as the media signature. The signature or the hash is generated by first dividing the input image into 8×8 pixels non-overlapping blocks and then taking the

DCT of each block. For each block, the DC and 3 AC coefficients are selected. The selection of the AC coefficients is based on a secret key. The DC and AC coefficients are then adjusted within the quantization boundary (the authors refer to this process as the rounding quantization) such that they remain within the same quantization boundary for a bounded distortion. This helps to make the system robust against non-malicious operations, like JPEG compression. All the rounded quantized coefficients are then concatenated and hashed using a cryptographic hash function to get the final image hash. During the rounding quantization process, 1-bit information for each block is generated, which is called the remainder. This information is embedded as the watermark and is used at the verification stage for authentication.

Lu and Liao [64] have proposed an image authentication scheme that uses the structure of an image as the digital signature. The image structure is obtained by identifying the parent-child pairs located at the multiple scales in the wavelet domain. Their scheme is based on the observation that the magnitude difference between a parent node and its four child nodes at consecutive scales in the wavelet domain mostly remains preserved for content preserving manipulations like JPEG compression and blurring. However, for malicious manipulations, this relationship changes, which enables the detection of tampering. The main drawback of this scheme is that the signature obtained does not depend on any secret key and can be extracted/verified by anyone who has knowledge of the algorithm. Secondly, since no secret key is used, therefore for a single image, only one unique signature can be generated.

Venkatesan *et al.* [65] have proposed a robust image hashing scheme for the purpose of image indexing. Wavelet decomposition of an image is first computed and each sub-band is then randomly tiled into small rectangles. For each rectangle, some statistical measure such as mean or variance is calculated and quantized using a random quantizer. The quantized outputs are decoded using an error-correction code to get the final hash value. The results reported show that the extracted hash is tolerant towards acceptable manipulations like JPEG compres-

sion, low pass filtering, etc and is different for different images. However, this scheme is meant to compare the image as a whole and cannot localize tampered area(s) in an image. Furthermore, it is not very clear how effectively tampering within an image can be successfully detected.

Lefebvre *et al.* [66] used the invariance properties of the radon transform to construct the image hash. A set of random vectors is first obtained by projecting the luminance plane of the input image in different angular directions. To achieve robustness against average luminance changes, the angular increment between two successive projections is used instead of directly using the projections. To reduce the size of the signature, the covariance matrix of the radon vectors is estimated. The digital signature is formed by selecting the eigenvector that corresponds to the two largest eigenvalues. To compare two images, the cross-correlation between the corresponding eigenvalues is computed and compared against a certain threshold. The proposed scheme is resilient against Gaussian and average filtering, JPEG compression, minor scaling and rotation. The results, however, show some false alarm cases. In addition, it is not clear whether the scheme can detect localized tampering of a single image in question. The scheme is not very robust for an image that has a high frequency texture. The issue of forgery and counterfeit attacks is also not addressed as the hash extraction does not depend on any secret key.

Monga and Evans [67] used visually significant feature points to generate the image hash. A wavelet-based iterative feature detection algorithm is used to extract these feature points. A feature vector is generated from the extracted feature points. Deterministic and randomized algorithms are developed to extract the hash of the image from the feature vector. The deterministic algorithm does not use any secret key and is weak from the security point of view. This problem is solved using the randomized algorithm that uses a secret key to achieve randomness in various stages of the algorithm. At first the input image is divided into overlapping circular/elliptical regions with randomly selected radii. These regions are then approximated as rectangles using water-filling like approach.

Image Security and Review of Related Work

The deterministic algorithm is then applied to the random rectangles to get the binary hash value. The proposed scheme is robust to several non-malicious manipulations like JPEG compression, contrast enhancement Gaussian smoothing, scaling, rotation, etc. The scheme detects malicious manipulations like object addition/removal, excessive noise addition, face morphing etc. However, all the malicious attacks reported in [67] are based on changing a significant portion of the image. The feature points are generated mainly in image areas where significant image geometry lies. Due to this reason, feature points do not cover the entire image area, especially the background. It is not very clear how the scheme will perform under small tampering, especially cloning which may not affect the feature points.

Swaminathan, Mao and Wu [68] have proposed an image hashing scheme that is resilient to geometric and filtering operations. The input image is first preprocessed by applying low pass filtering, resizing and histogram equalization. Fourier transform is then applied to the preprocessed image and the resulting transformed image is then converted into polar co-ordinates to obtain the feature vector. To incorporate security, the feature vector is modulated using pseudo-random numbers generated from a secret key. The result is then quantized and losslessly compressed to obtain the final hash. The normalized Hamming distance between the hashes is used as the performance metric. The results reported in [68] show that the scheme is quite robust to several non-malicious operations like JPEG compression, rotation of around 10 degrees, cropping, shearing. However, it is not clear whether the scheme can detect tampering in a small area of an image.

Monga and Mihcak [69] have used a dimensionality reduction technique called the non-negative matrix factorization (NMF) to generate the image hash. The NMF technique uses non-negativity constraints that enable parts-based representation. The results reported in [69] show that the proposed scheme is robust to JPEG compression, rotation, cropping and resizing. However, it is not clear whether the scheme can detect tampering in small areas of the image. Furthermore, the security analysis of this scheme is not presented.

Motivated by the work of Monga and Mihcak [69], Lv and Wang have recently proposed a Fast Johnson-Lindenstrauss transform for image hashing [70]. Swaminathan *et al.* [71] proposed an image hashing scheme resilient to geometric and filtering operations by using the properties of discrete polar Fourier transform. Lu and Hsu [72] have proposed geometric distortion-invariant image hashing scheme by extracting robust meshes from an image. The normalized meshes are then used for generating the hash in the DCT domain. The proposed scheme has good robustness to several types of geometric distortions. This idea is further extended in the watermarking domain [73] in which hash-dependent watermarks are generated using meshes that are extracted from the image to be watermarked. Interestingly, Deng *et al.* [74] uses affine covariant regions for watermark embedding. An affine covariant point detector is used to extract feature points for the constructions of affine covariant elliptical regions. These regions are then normalized into a circle with specific rotation to align with the dominant gradient orientation of the corresponding feature points. The proposed scheme has good robustness to many common image processing operations, especially geometric distortions, like cropping, scale invariance and rotation. The ideas proposed in [73] and [74] can be further investigated to come up with hashing schemes that besides being robust to lossy compression and image filtering can offer good resilience properties for geometric distortions.

2.4 Review of Copyright Protection Schemes

With rapid growth of the Internet, security of digital images is becoming a great concern. It has now become very easy to illegally copy, modify, and retransmit a digital image. Digital watermarking is a technique that provides a way to protect digital images from illicit copying and manipulation. A digital watermark is an imperceptible signal added to digital data, called cover work, which can be detected later for buyer/seller identification, ownership proof, and so forth [8]. Depending on the type of application, a watermark can be either visible or invisible. For example, in most of the currency bills, a visible watermark is

Image Security and Review of Related Work

embedded that helps to distinguish between genuine and fake currency. Visible watermarks find their usage in a number of multimedia applications like copy right protection, fingerprinting, broadcast monitoring, etc [5]. In recent years, a number of techniques have been proposed in the area of digital watermarking for audio, imaging and video applications, [5], [6].

Our focus in this thesis is on invisible watermarking for multimedia security applications. A digital watermarking scheme can either be symmetric or asymmetric. A symmetric watermarking scheme uses identical keys for watermark embedding and detection [75]. This possesses a security weakness as the information used to detect a watermark can be used to remove it. This restricts the use of symmetric watermarking, as the number of authorized detectors have to be strictly controlled. To solve this problem, asymmetric watermarking schemes have been proposed that use different keys for watermark embedding and detection [76]- [79]. This makes the use of watermarking possible for public domain applications where any one with the detection key can check the embedded watermark. However, the practical use of asymmetric watermarking requires careful considerations [80]

It is difficult to come up with a single watermarking technique that covers all the multimedia applications. In fact, each application has its own special requirement for which a watermarking scheme is tailored. For example, *robust watermarking* schemes are designed to survive severe attacks [75]. There can be different forms in which the attacks can be done. For example, a simple attack can be to compress a watermarked image so that the presence of the watermark in the watermarked image is lost. This attack may be done either intentionally or unintentionally. For example, Alice sends a watermarked image to Bob with Alice's and Bob's identity embedded into the image using one or more than one watermarks. The embedded watermarks can serve different purposes. Alice's watermark, for example, can help to assert Alice's genuine ownership of the watermarked image. This application of watermarking is called *copy protection* [5], [6]. Bob's watermark can help Alice to trace Bob in case Bob starts illegally

distributing the watermarked image. This application of digital watermarking is called fingerprinting [6]. If Bob can somehow remove the watermark from the watermarked image, then Bob can freely distribute this image illegally. In this case, Alice's ownership claim on the image is lost. If the underlying watermarking scheme is weak, Bob can perform some simple signal processing operations like compression, filtering, geometric transformation to remove the watermark. This is the case of an intentional attack. On the other hand it might be the case that when Alice sent the watermarked image to Bob, the channel performed lossy compression and the watermark was removed. It may also be the case that Bob wanted to save his hard disk memory and compressed the watermarked image before storing it. This is an example of an unintentional attack, that may actually remove or severely reduce the watermark's energy in the image.

There is another type of watermarking called *fragile watermarking* that is basically used for multimedia authentication [81]. These schemes are designed to be sensitive to identify a slight modification in the watermarked image done intentionally or unintentionally. In between robust and fragile watermarking, *semi-fragile watermarking* [82] is another approach used for protection of multimedia content. Semi-fragile watermarking schemes are designed to survive some specific attacks. Watermarking schemes can also be classified depending upon the watermark detection algorithm. A watermarking scheme is called *non-blind* if the watermark detection algorithm requires the presence of the original unmarked image. On the other hand, watermarking schemes in which the watermark detection algorithm does not require the presence of the unmarked host falls into the category of *blind watermarking* [5], [6], [8].

The above mentioned watermarking schemes can be realized by embedding the watermark either in the spatial domain or the transform domain. The earlier work in the field of watermarking was mostly done in the spatial domain. A simple spatial domain technique is the LSB replacement scheme [83] in which the least significant bit (LSB) of every pixel in an image is replaced by the binary watermark. LSB techniques are easy to implement and are computationally less

complex. However, in general, they are not robust to common signal processing operations and are weak from the security perspective.

In 1996, the field of watermarking reached its new horizon when Ingemar J. Cox and his coworkers presented the spread spectrum watermarking techniques [75]. This is a non-blind frequency domain watermarking technique that uses an independent and identically distributed Gaussian random vector as the watermark. The basic idea is to insert the watermark in such a way that it is spread over a large number of frequency bins so that energy in any single bin is very small. However, destroying such a watermark would require high amplitude noise added to all the selected frequency bins. Hence an attacker trying to destroy the watermark will have to make significant perceptual changes in the image quality that shall be noticeable. It is pertinent to mention here that before this proposal was floated, people working in the watermarking community had the opinion that a watermark should be embedded in the perceptually insignificant regions of multimedia data so that minimal distortion is caused. Cox and his coworkers argued that for the watermark to be robust against a number of signal processing operations, the watermark should be placed in the perceptually most significant components of the image spectrum. The technique proposed in [75] has been found to be extremely robust against a number of signal processing operations, like, scaling, JPEG compression, dithering distortion, print and scan, cropping, multiple watermarking and collusion attacks. A number of proposals can be found in the literature that are based on spread spectrum techniques. Besides spread spectrum, researchers have proposed to use different embedding domains like DWT [84], DFT [85], [86], etc.

Hartung, Su Bernd Girod [87] have listed several conceivable attacks on spread spectrum techniques. These attacks include simple attacks, detection-disabling attacks, ambiguity attacks, and removal attacks. Simple attacks attempt to damage the embedded watermark by manipulating the entire watermark data instead of targeting the hidden watermark. Disabling attack attempts to make the recovery of the watermark impossible or infeasible mostly by geometric distortion

like rotation, sub-sampling, etc. Ambiguity attacks are designed to produce fake original data or fake watermarked data. This puts the credibility of the original data or original watermarked image in question. Removal attacks are designed to estimate the watermark from the watermarked data and then remove the watermark. Collusion attack belongs to this class of attack. Cox and Linnartz also describe tampering issues with watermarks [88].

Max Costa in his classical paper, “Writing on Dirty Paper” [89], has given some extremely amazing results that have been instrumental in paving the way to devise watermarking schemes that use side information, or informed coding. In conventional watermarking systems, the watermark is generated independently of the cover image. Hence there is a chance that the cover image will interfere with the watermark such that the watermark detector may have problems. Informed embedding can be used to solve this problem by adjusting the embedded strength for each individual cover data so that the watermarked data will have a fixed-magnitude linear correlation with the watermark. Costa showed that the capacity of a channel with additive Gaussian noise and power constrained is not affected if some extra noise sequence that is independently identically distributed (i.i.d.) is added to the channel’s output if the encoder has complete information about the extra noise. Costa’s work inspired a lot of research in the watermarking community because it enabled to establish an upper bound on the watermark capacity. Direct application of Costa’s work in watermarking is not feasible for large number of messages and watermarks since a lot of computation and storage will be required while searching for optimum watermarks.

A practical realization of this idea paved way for a new class of watermarking scheme proposed by Chen and Wornell which is called Quantization Index Modulation (QIM) [90]. The basic idea is to structure the watermarks in such a way that efficient search is possible while selecting the optimum watermark for a given cover data. The QIM technique is based on using a set of N -dimensional quantizers; one quantizer for each message. The message to be watermarked selects the quantizer. The selected quantizer then embeds the information by quantizing

Image Security and Review of Related Work

the cover data. In the decoding process, the embedded information is recovered by using a distance metric for all the quantizers and then selecting the quantizer which gives minimum distance from the received cover data.

The above mentioned watermarking schemes are algorithms used to insert messages into a cover work. A good algorithm should do this in a robust and secure manner. However, someone else can also insert another watermark and then claim the ownership of the multimedia content. How can Alice claim her genuine ownership? Similarly in case of fingerprinting, assume Alice sells an image to Bob by embedding a watermark so that she can trace Bob if the watermarked image is distributed illegally. Alice then illegally distributes the watermarked image that has Bob's fingerprint and falsely accuses Bob of distributing the watermarked image illegally. How can Bob prove that he is innocent? Therefore, merely using a watermarking algorithm does not completely address the issues of copyright protection. To devise a secure watermarking scheme that can address the issues related to the buyer and seller of a watermarked image, it is necessary that a watermarking algorithm is well integrated with a secure protocol [91], [92]. In this thesis, our main focus is to integrate cryptographic protocols with a watermarking scheme to address some important issues related to copyright protection.

Craver *et al.* [93] argue that a watermarking scheme alone in many cases is not sufficient to resolve copyright ownership claims unless they follow certain standardization and specific requirements. For example, if a watermarked image contains watermarks of both Alice and Bob, who is the rightful owner? This problem relates to multiple copyright ownership claims. A simple solution seems that both Alice and Bob present their original (unwatermarked image). If Alice is the genuine owner, and later Bob watermarked Alice's image with a fake watermark, then Bob's original and watermarked image will contain Alice's watermark. On the other hand, Alice's original will not contain Bob's watermark. This way Alice can prove that she is the genuine owner. Craver *et al.* however presented an amazing counterfeit watermarking scheme that can allow Bob to show his watermark in Alice's unwatermarked original image even though Bob has no access to

Image Security and Review of Related Work

Alice's original image. This is something very amazing and interesting. Craver *et al.* call such watermarking schemes as *invertible* in which the fake original is created by subtracting the watermark. They further suggest a method that may enable a watermarking scheme to avoid the condition of invertibility, which they refer as noninvertible watermarking schemes. Noninvertibility is achieved by selecting a watermark that is generated by a cryptographically secure one-way hash function. Since Bob does not have the fake original before hand, he cannot obtain the watermark since it will require the hash of the fake original. Craver *et al.* further show multiple watermark attacks on noninvertible watermarking schemes that enable to introduce counterfeit watermarks and can create an ownership deadlock. They call such attacked schemes as *quasi-invertible*, for which they describe the condition of nonquasi-invertibility.

Memon and Wong [91] present another interesting scenario in watermarking between the buyer and seller. Suppose Alice (seller) watermarks an image and sells it to Bob (buyer). Now suppose Bob illegally distributes the image. If Alice finds an illegal copy originated from Bob, she can prove that Bob actually distributed this copy by showing the unique watermark that associates to Bob. Bob, however, can claim that the unauthorized copy was actually originated by Alice or it may have originated due to a security breach in Alice's system. If Alice is innocent, how can Alice prove to a third party that Bob is lying? On the other hand, if Bob is innocent, how can he prove his innocence to a third party? To solve this problem, Memon and Wong [91] proposed a buyer-seller watermarking protocol for invisible watermarking schemes. Under this protocol, the seller does not know the actual watermarked copy that the buyer receives. This means that the seller cannot further create watermarked copies that contain the watermark of the buyer to which the seller has previously sold a watermarked copy. In case if the seller finds an illegal copy, he/she can identify the buyer from this copy and further prove this fact to a third party using a dispute resolution protocol.

Ramkumar and Akansu [92] proposed a restricted protocol that helps a genuine owner to claim the ownership unambiguously, while making it extremely difficult

for a pirate. Their work is basically an advancement to the work proposed by Craver *et al.*. Towards defining their protocol, they suggest a number of restrictions such as distortion in the cover data, unsuitability of using strictly oblivious watermarking schemes and the choice of signature used to create the watermark. They show a simple attack on the noninvertible watermarking methodology proposed by Craver *et al.* [93]. The attack relies on the fact that if a single LSB of an image is flipped, virtually there will be no perceptual distortion between the modified image and the actual image. However, the cryptographic hash of the modified image and the actual image will be completely different. This is due to the sensitivity property of cryptographic hash function [4]. Ramkumar and Akansu show that this sensitivity property of a cryptographic hash function actually goes in favor of an attacker who can change 1 or 2 LSBs of an image to obtain a large number of signatures out of which one or more than one may be used to show its presence in the actual original that the attacker does not have in the first place. Despite these effective and interesting schemes, efforts in finding more robust protocols for watermarking are still underway. Some more interesting watermarking related protocols that aim to resolve disputes in a buyer-seller environment can be found [94] - [97].

2.5 Chapter Conclusion

The fundamental goal of this thesis is to address three closely related problems pertaining to multimedia security, i.e., encryption, authentication and copyright protection. In this chapter the basic requirements of multimedia security, specifically related to image encryption, image authentication and copyright protection of digital images were discussed. A comprehensive review was presented highlighting important issues in these areas. In light of these issues, the next three chapters will present novel encryption, authentication and copyright protection schemes for digital images. The basic theme behind the proposed encryption and authentication algorithms is to make them robust against content preserving operations and at the same time provide sufficient security. This is a drift from

Image Security and Review of Related Work

standard cryptographic techniques in which there is no robustness requirement. Copyright protection protocols are proposed to address important issues related to the identification of buyer and seller of a watermarked image. Specifically, three issues related to copyright protection shall be focused on, that is, buyer-seller identification, copyright infringement and verification of ownership.

Chapter 3

A JPEG Compression Tolerant Image Encryption Scheme

3.1 Introduction

For the past few years, a lot of research work has been conducted to devise new encryption technologies to protect multimedia data like audio, image and video. A question that arises is why new techniques for multimedia encryption are required when the field of cryptography is already well matured? There are several reasons that actually impede the direct use of cryptographic techniques for solving multimedia encryption problems. For example, if an image data is encrypted using a traditional cipher, like the AES [1] and the encrypted image data undergoes JPEG lossy compression, the decryption mechanism will not be able to recover the un-encrypted image data. In multimedia scenarios, content preserving operations like lossy compression, enhancement, transcoding, etc are very common. Conventional encryption algorithms are designed to recover the exact values of the plaintext data. As for multimedia data that involves audio, image or video, an approximation of the original multimedia content is enough for most applications, as long as the error between the original and the recovered data is small. In view of this discussion, we have a strong feeling that there is a need to devise a new class of encryption algorithms that can tolerate multimedia content preserving operations to some extent. This means if the encrypted

A JPEG Compression Tolerant Image Encryption Scheme

multimedia data passes through a channel that changes the values of the encrypted data within a certain limit, the decryption algorithm should recover a close approximation of the multimedia content.

In Chapter 2, a number of image encryption schemes were reviewed to show the current and past trends in this area. Nearly all of these schemes are designed for a lossless channel, i.e. a single bit error in the encrypted data would not yield successful decryption. In this chapter, we present a new class of encryption and decryption algorithms for digital images that can work in a lossy compression environment. This idea is a drift from the standard cryptographic paradigm in which the encrypted data stream always passes through a lossless compression channel and for successful decryption, it is necessary that all the bits of the encrypted bitstream remain the same. This is one of the main motivating factors that has driven us towards proposing this scheme; decryption should be possible even if encrypted data passes through a lossy compression channel. Adding this feature can have a negative impact on the security of the encryption scheme. This brings us towards the second goal of this scheme, i.e. to incorporate security features that make the scheme compatible with the standard cryptographic norms. The third aim is to present detailed analysis illustrating the security of the scheme against well known cryptographic attacks [3], [4], [98]. To achieve the security objectives, the confusion and diffusion mechanism proposed by Shannon [16] should be carved into the very roots of the encryption scheme. As mentioned in [44], any proposed multimedia encryption scheme should properly define the key space, various cryptographic attacks, like the brute force attack to break the key, ciphertext only attacks, known plaintext attacks and chosen plaintext attacks.

A brief review is now presented to show the work related to image encryption in the JPEG domain. In [99], a $\{2, 2\}$ secret sharing scheme for images is presented that is designed to work directly in the JPEG domain. The quantized DCT coefficients obtained from the zig-zag scan are used in conjunction with a PRNG to generate two noise-like shares. The decryption process is lossless that

A JPEG Compression Tolerant Image Encryption Scheme

uses an XOR operation to recover the plaintext-image data. The value of the shares should not change once they are encrypted otherwise the XOR operation will not be able to recover the plaintext image correctly. In [100], a selective encryption scheme for JPEG images is presented that uses the AES algorithm to encrypt the Huffman's bitstream such that the ciphertext reflects the syntax of the Huffman's bitstream to ensure JPEG compliance. Due to AES encryption, the encryption and decryption process is lossless. In [101], encryption is achieved by a technique called reversible histogram spreading approach in which iteratively, the JPEG encoded blocks are shuffled followed by varied-length codes (VLC) fix-permutation to obtain an encrypted image in the JPEG domain that is perceptually different from the plaintext image. The scheme works in the JPEG domain with JPEG block shuffling and VLC permutation iteratively, thus making it sensitive to changes in the encrypted bitstream. To preserve file size, [102] develops a scheme by decoding the JPEG data stream to obtain the Huffman code which in turn gives the DC differential residues which are encrypted through XOR with the key that is of the same length as the data stream. After some further steps like DCT block confusion and key embedding, Huffman coding is performed to get the final encrypted JPEG bitstream. For correct decryption, the JPEG encrypted bitstream should not change, otherwise the XOR operation would not correctly yield the DC differential residues. Another selective encryption scheme is presented in [103] in which specific high frequency DCT coefficients are selected which are then encrypted using a proposed Non-linear shift back register used as a stream cipher by XORing the DCT coefficients with the output of the key generated stream cipher. To enhance security, an additional shuffling mechanism is also proposed. The proposed algorithm is lossless and requires the actual encrypted data at the decryption end to perform the XOR operation correctly.

Specifically, we propose an image encryption scheme that has the capability to tolerate JPEG compression. This means if the encrypted image data is JPEG compressed, the decryption algorithm should still be able to recover the plaintext image data with certain amount of distortion that depends upon the compression ratio. It is to be noted that a unique feature of the encryption scheme proposed

A JPEG Compression Tolerant Image Encryption Scheme

in this chapter is that the cryptosystem is an independent entity and is not an integral part of any compression algorithm like JPEG as is the case with the schemes reviewed in the preceding paragraph. In fact, the user is free to decrypt the encrypted image directly or by compressing the encrypted image using JPEG compression and then decrypting the JPEG decompressed encrypted image. We directly work on the image pixel values without disturbing the headers and other information, thus making our scheme format independent. A detailed security analysis is presented to show that the proposed scheme can withstand a number of attacks. We believe that the proposed scheme shall open new research directions in the area of image and video encryption technologies by providing the transmission channel or storage medium with a flexibility to allow lossy compression on the encrypted multimedia data stream. This idea can be extremely useful in applications in which it is difficult to maintain a 100% lossless channel between the transmitter and the receiver.

The rest of the chapter is organized as follows. The utilization of orthogonal matrices towards designing the proposed image encryption scheme shall be presented in the section to follow. Section 3.3 describes the proposed encryption and decryption algorithms. Section 3.4 presents a number of experimental results to show the robustness of the proposed scheme against JPEG compression, comparison with the AES algorithm to show robustness improvement, histogram of the encrypted image, tamper detection capability and system's throughput. In Section 3.5, detailed security analysis of the proposed scheme is presented. Section 3.6 finally concludes this chapter.

3.2 Image Encryption using Orthogonal Matrices

The image encryption scheme proposed in this chapter utilizes a number of interesting properties of orthogonal matrices. If Φ is an $n \times n$ orthogonal matrix, then

A JPEG Compression Tolerant Image Encryption Scheme

$\Phi = \{\vec{\phi}_1, \vec{\phi}_2, \dots, \vec{\phi}_n\}$ contains n orthogonal set of nonzero vectors in $\mathbb{R}^n$ which are linearly independent and forms a basis for the subspace spanned by Φ . If the vectors in the set Φ are both orthogonal and unit vectors, then Φ contains a set of vectors that forms an orthonormal basis [104]. In this paper, we have used Singular Value Decomposition (SVD) [104], [105] to generate orthogonal matrices. Singular value decomposition is a tool used in linear algebra that has found many interesting applications in image processing and multimedia security [106]–[109]. The SVD of an $m \times n$ matrix, A , gives the factored form: $A = USV^T$. The matrices U , S and V have dimensions, $m \times m$, $m \times n$ and $n \times n$, respectively. The U and V matrices have columns and rows that forms orthonormal vectors in R^m and R^n , respectively. The matrix S contains the singular values of A on the diagonal [104], [105]. If the matrix A is $n \times n$, then both U and V will have dimensions $n \times n$. In the remaining part of this chapter, we use the notations U_x , V_x and Φ_i to represent an $n \times n$ orthogonal matrix that has orthonormal columns and rows. Let $\vec{y}$ be a column vector in $\mathbb{R}^n$, T be the transpose operator and $\|\cdot\|$ represents the function that evaluates the Euclidean length of a vector. Following are some well known properties related to Φ_i [104], [105]:

1. $\Phi_i^{-1} = \Phi_i^T$
2. $\|\Phi_i \vec{y}\| = \|\vec{y}\|$
3. $\|(\cdots \Phi_{i-1} \Phi_i \Phi_{i+1} \cdots) \vec{y}\| = \|\vec{y}\|$
4. $\Phi_i^T \Phi_i \vec{y} = \vec{y}$
5. $\Phi_i \vec{y} \neq \Phi_j \vec{y} \neq \Phi_j^T \Phi_i \vec{y} \neq \vec{y}$

The fact that $\Phi_i^T = \Phi_i^{-1}$ has two important features. The inverse of Φ_i matrix will always exist, and, secondly, it can simply be obtained by taking its transpose. Obtaining an inverse of a matrix by simply taking its transpose is computationally less expensive as compared to the adjoint method of taking the inverse of a matrix. According to properties 2 and 3, the length of the vector $\vec{y}$ remains unchanged whether it is premultiplied by a single orthogonal matrix or more than

A JPEG Compression Tolerant Image Encryption Scheme

one orthogonal matrices. Property 3 holds because multiplication of $n \times n$ orthogonal matrices that have orthonormal columns and rows generates another $n \times n$ orthogonal matrix that has orthonormal columns and rows [104], [105]. Properties 4 and 5 helps to develop the basic decryption mechanism and incorporates a number of security features that shall be discussed in the subsequent sections.

Using the above properties, the basic mathematical formulation for proposed image encryption scheme is now developed. Let P be an $n \times n$ gray scale plaintext image matrix and C be an $n \times n$ ciphertext image matrix. Let $\vec{p}_{row_j^T}$ and $\vec{p}_{col_k}$ be column vectors, where $\vec{p}_{row_j^T}$ represents transpose of the j^{th} row of P , and $\vec{p}_{col_k}$ represents the k^{th} column of P . Similarly for the C matrix, we define the column vectors, $\vec{c}_{row_j^T}$ and $\vec{c}_{col_k}$. To represent an element within a matrix, we use the subscripts j and k for row and columns, respectively. For example, P_{jk} represents an element of P at row j and column k .

Proposition 1

- 1a: If $C = \Phi_i P$, then $\|\vec{c}_{col_k}\| = \|\vec{p}_{col_k}\|$.
 1b: For the column k , $\max(|C_{jk}|) \leq \|\vec{p}_{col_k}\|$, $1 \leq j \leq n$.

Proof 1a

The k^{th} column of C can be written as:

$$\begin{aligned} \vec{c}_{col_k} &= \Phi_i \vec{p}_{col_k} \\ \Rightarrow \begin{bmatrix} C_{1k} \\ C_{2k} \\ \vdots \\ C_{nk} \end{bmatrix} &= \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \begin{bmatrix} P_{1k} \\ P_{2k} \\ \vdots \\ P_{nk} \end{bmatrix} \\ \Rightarrow \begin{bmatrix} C_{1k} \\ C_{2k} \\ \vdots \\ C_{nk} \end{bmatrix} &= P_{1k} \begin{bmatrix} \phi_{11} \\ \phi_{21} \\ \vdots \\ \phi_{n1} \end{bmatrix} + P_{2k} \begin{bmatrix} \phi_{12} \\ \phi_{22} \\ \vdots \\ \phi_{n2} \end{bmatrix} + \cdots + P_{nk} \begin{bmatrix} \phi_{1n} \\ \phi_{2n} \\ \vdots \\ \phi_{nn} \end{bmatrix}. \end{aligned} \quad (3.2.1)$$

A JPEG Compression Tolerant Image Encryption Scheme

According to property 2, $\|\Phi_i \vec{y}\| = \|\vec{y}\| \Rightarrow \|\vec{c}_{col_k}\| = \|\vec{p}_{col_k}\|$ since the matrix Φ_i contains column vectors that forms orthonormal basis. The elements of the vector $\vec{p}_{c_k}$ basically forms coordinates with respect to the orthonormal basis vectors contained in Φ_i . Multiplication of Φ_i with $\vec{p}_{col_k}$ generates the coordinates with respect to the standard basis given by $\vec{c}_{col_k}$.

Proof 1b

Since $\|\vec{c}_{col_k}\| = \|\vec{p}_{col_k}\| \Rightarrow \sqrt{(C_{1k})^2 + (C_{2k})^2 + \dots + (C_{nk})^2} = \|\vec{p}_{col_k}\|$,
 $\Rightarrow \max(|C_{jk}|) \leq \|\vec{p}_{c_k}\|, 1 \leq j \leq n$.

Proposition 2

2a: If $C = P\Phi_i$, then $\|\vec{c}_{row_j^T}\| = \|\vec{p}_{row_j^T}\|$.

2b: For the row j , $\max(|C_{jk}|) \leq \|\vec{p}_{row_j^T}\|, 1 \leq k \leq n$.

Proof 2a

The operation $C = P\Phi_i$ when expanded gives:

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} P_{11}\phi_{11} + \cdots + P_{1n}\phi_{n1} & P_{11}\phi_{12} + \cdots + P_{1n}\phi_{n2} & \cdots & P_{11}\phi_{1n} + \cdots + P_{1n}\phi_{nn} \\ P_{21}\phi_{11} + \cdots + P_{2n}\phi_{n1} & P_{21}\phi_{12} + \cdots + P_{2n}\phi_{n2} & \cdots & P_{21}\phi_{1n} + \cdots + P_{2n}\phi_{nn} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1}\phi_{11} + \cdots + P_{nn}\phi_{n1} & P_{n1}\phi_{12} + \cdots + P_{nn}\phi_{n2} & \cdots & P_{n1}\phi_{1n} + \cdots + P_{nn}\phi_{nn} \end{bmatrix}.$$

Writing the j^{th} row of the matrix C as a column vector:

$$\begin{bmatrix} C_{j1} \\ C_{j2} \\ \vdots \\ C_{jn} \end{bmatrix}^T = \begin{bmatrix} P_{j1}\phi_{11} + P_{j2}\phi_{21} + \cdots + P_{jn}\phi_{n1} \\ P_{j1}\phi_{12} + P_{j2}\phi_{22} + \cdots + P_{jn}\phi_{n2} \\ \vdots \\ P_{j1}\phi_{1n} + P_{j2}\phi_{2n} + \cdots + P_{jn}\phi_{nn} \end{bmatrix}^T$$

A JPEG Compression Tolerant Image Encryption Scheme

$$\Rightarrow \begin{bmatrix} C_{j1} \\ C_{j2} \\ \vdots \\ C_{jn} \end{bmatrix}^T = \begin{bmatrix} P_{j1} \begin{bmatrix} \phi_{11} \\ \phi_{12} \\ \vdots \\ \phi_{1n} \end{bmatrix} + P_{j2} \begin{bmatrix} \phi_{21} \\ \phi_{22} \\ \vdots \\ \phi_{2n} \end{bmatrix} + \cdots + P_{jn} \begin{bmatrix} \phi_{n1} \\ \phi_{n2} \\ \vdots \\ \phi_{nn} \end{bmatrix} \end{bmatrix}^T. \quad (3.2.2)$$

Since the Φ_i matrix contains both columns and rows that forms orthonormal basis, therefore,

$$\begin{aligned} \sqrt{(C_{j1})^2 + (C_{j2})^2 + \cdots + (C_{jn})^2} &= \sqrt{(P_{j1})^2 + (P_{j2})^2 + \cdots + (P_{jn})^2} \\ \Rightarrow \|\vec{c}_{row_j^T}\| &= \|\vec{p}_{row_j^T}\|. \end{aligned}$$

Proof 2b

$$\begin{aligned} \text{Since } \|\vec{c}_{row_j^T}\| &= \|\vec{p}_{row_j^T}\| \Rightarrow \sqrt{(C_{j1})^2 + (C_{j2})^2 + \cdots + (C_{jn})^2} = \|\vec{p}_{row_j^T}\|, \\ \Rightarrow \max(|C_{jk}|) &\leq \|\vec{p}_{row_j^T}\|, \quad 1 \leq k \leq n. \end{aligned}$$

Remarks

Proposition 1 and 2 basically states that if the image matrix P is premultiplied/postmultiplied by a single orthogonal matrix or more than one orthogonal matrices that have orthonormal columns/rows, then each element of the resultant matrix C will always be bounded. In case of premultiplication, the absolute value of any element of the k^{th} column of C will always be less than or equal to the Euclidean length of the k^{th} column of P . In case of postmultiplication, the absolute value of any element of the j^{th} row of C will always be less than or equal to the Euclidean length of the j^{th} row of P . Single or repeated premultiplications or postmultiplications of the image matrix P with orthogonal matrices creates a cipher image matrix C whose entries have different values as compared to P . On the other hand, each entry of C has a fixed bound related to premultiplication and postmultiplication no matter how many multiplication rounds are carried out. This fact is the basic building block of the proposed image encryption scheme. The coordinates of $\vec{p}_{col_k}$ and $\vec{c}_{col_k}$ in Eq. 3.2.1 basically represents the

A JPEG Compression Tolerant Image Encryption Scheme

same vector having different coordinate values with respect to its basis. Similar is the case with $\vec{p}_{row_j^T}$ and $\vec{c}_{row_j^T}$ in Eq. 3.2.2. As described in Section 3.3, this helps to create a cipher image matrix C that has no visual similarity with the image P . In fact, the absolute value of each element of the k^{th} column or j^{th} row of C is quite smaller than their respective bounds. This issue is further discussed in Section 3.4.6. In view of the above discussion, we define the following two basic encryption transformations:

1. $C = \Phi_i P$. The corresponding decryption transformation will be: $P = \Phi_i^T C = \Phi_i^T \Phi_i P$.
2. $C = P \Phi_i$. The corresponding decryption transformation will be: $P = C \Phi_i^T = P \Phi_i \Phi_i^T$.

The transformation selected for the first round of encryption is used for all the other rounds. For example, if $C = \Phi_i P$ is selected as the encryption transformation, then it is used in all subsequent rounds. This ensures $\|\vec{c}_{col_k}\| = \|\vec{p}_{col_k}\|$ for any number of encryption rounds. The proposed encryption mechanism although very simple, is very effective in terms of security and robustness to JPEG compression as explained in the following sections. In the next section, the proposed image encryption and decryption algorithms are described. Detailed experimental results and security analysis shall be presented in the subsequent sections.

3.3 The Proposed Image Encryption Scheme

With the background knowledge presented in the previous section, the proposed image encryption and decryption algorithms are now described. The image to be encrypted is called the *plaintext-image* represented by P_0 . The encrypted image is called the *cipher-image* represented by C and the decrypted image is called the *decrypted-image* denoted by P_D . The images P_0 , C and P_D are gray-scale images of size 256×256 pixels. We use the term *iteration* to represent one round of permutation and multiplication. Let K be a 128-bit secret parameter only known

A JPEG Compression Tolerant Image Encryption Scheme

to the sender. To introduce diffusion in the encryption process with respect to the plaintext-image, the cryptographic hash of P_0 is taken using the MD5 hashing algorithm [4] that gives a 128-bit random hash value for any arbitrary input. Let this hash be represented by $H(P_0)$. The hash $H(P_0)$ is bitwise XORed with K to generate a seed, δ_1 . In our proposed scheme, the seed δ_1 is an image-dependent secret key used for encryption and decryption and is different for each image. The key δ_1 is generated by the sender of an encrypted image and is securely transmitted to the receiver to perform decryption. As shown in Section 3.5, this additional overhead of using a different key for a different image greatly helps to enhance the security of the proposed scheme which is actually designed to show flexibility for small changes in the pixel values of the cipher-image due to JPEG compression.

3.3.1 Image Encryption Algorithm

With the above preamble, various steps of the proposed image encryption algorithm are now described.

1. The seed δ_1 initializes a pseudo-random number generator (PRNG) with uniform distribution to generate a random matrix, R_1 of size 256×256 and a block permutation vector, $\vec{\omega}_1$ of size 1×64 . As described in Step 3 of this algorithm, the random matrix R_1 and the permutation vector $\vec{\omega}_1$ shall be used in the first and second iterations. For the third and fourth iterations, the seed δ_2 is used to initialize the PRNG for generating R_2 and $\vec{\omega}_2$, where $\delta_2 = \delta_1 + 1$. In general, the seed δ_x is used to initialize the PRNG for generating a sequence of random matrices R_x and block permutation vectors $\vec{\omega}_x$, where $\delta_x = \delta_{x-1} + 1$ for $x > 1$ and $\delta_x = \delta_1$ for $x = 1$. For each iteration, i , the parameter x is calculated as follows:

$$x = \begin{cases} \frac{i+1}{2} & : i \bmod 2 = 1 \\ \frac{i}{2} & : i \bmod 2 = 0. \end{cases} \quad (3.3.3)$$

A JPEG Compression Tolerant Image Encryption Scheme

2. For each iteration, permutation is carried out by dividing the input image into non-overlapping blocks of size $Z \times Z$ pixels. Each block is then permuted using the values in $\vec{\omega}_x$. We choose $Z = 32$, hence for a 256×256 image, there shall a total of 64 blocks. This is the reason that the vector $\vec{\omega}_x$ has 64 entries. Let $\Gamma_{\vec{\omega}_x}(\cdot)$ represents the permutation function that uses the permutation vector $\vec{\omega}_x$ to perform the permutation. For example, $\Gamma_{\vec{\omega}_x}(P_0)\Phi_x$ means that the image P_0 is first permuted and the permutation result is then multiplied by the matrix Φ_x .
3. Let the iteration count starts from 1 and goes up to N . For each odd iteration, i.e. when $i \bmod 2 = 1$, SVD is applied to R_x to get U_x and V_x of size 256×256 . Let $\Phi_i = U_x$ for the i^{th} odd iteration ($i \bmod 2 = 1$) and $\Phi_i = V_x$ for the next even $i^{th} + 1$ iteration ($i \bmod 2 = 0$). This strategy is adopted to save computational time as SVD is required to be taken once for every two iterations.
4. For each iteration, $P_i \in \mathbb{R}^{256 \times 256}$ is calculated using either of the rules given by the following equations:

$$P_i = \Phi_i \Gamma_{\vec{\omega}_x}(P_{i-1}). \quad (3.3.4)$$

$$P_i = \Gamma_{\vec{\omega}_x}(P_{i-1}) \Phi_i. \quad (3.3.5)$$

As mentioned in Section 3.2, the encryption transformation selected for the first iteration is then used in all iterations to follow.

5. At the end of the last iteration N , let the resultant image be represented by P_N . The image P_N is uniformly quantized with a fixed offset such that $\min(P_N) = 0$ and $\max(P_N) = 255$. This transformed image is the cipher-image, C . To utilize both U_x and V_x matrices, the parameter N is kept even.

Equation 3.3.4 and Eq. 3.3.5 show the permutation and mixing mechanisms. For each iteration, the current image, P_{i-1} is first permuted using the permutation

A JPEG Compression Tolerant Image Encryption Scheme

function $\Gamma_{\tilde{\omega}_x}(\cdot)$ to get the permuted image $\Gamma_{\tilde{\omega}_x}(P_{i-1})$. This permuted image is then multiplied with the matrix U_x or V_x . The matrices U_x and V_x are obtained by taking SVD of R_x . The matrix R_x is generated using a 128-bit seed δ_x whose initial value, δ_1 is derived by XORing K with $H(P)$. Thus, a single-bit change in K or P_0 will change δ_1 thus causing the PRNG to change its initial state. As a result, the PRNG will produce a different initial random matrix R_1 which is used to generate subsequent orthogonal matrices and permutation vectors. The use of cryptographic hash function ensures that a single-bit change in the plaintext-image would yield an entirely different δ_1 , thus completely changing the cipher-image. This shows how a cryptographic hash function is used to cause diffusion with respect to the plaintext-image and the secret parameters on the cipher-image. It ensures that two plaintext-image with even a difference of 1-bit would map to entirely different cipher-images.

Since the cipher-image C is obtained by quantizing and scaling P_N so that $\min(C) = 0$ and $\max(C) = 255$, therefore, some information is lost at the encryption end due to which the decryption algorithm will not recover the exact pixel values of P_0 . However, the proposed encryption scheme is designed for digital images with the emphasis to work in a JPEG lossy compression environment. Our objective at the decryption end is to recover a high quality reconstruction of the original plaintext-image. The exact recovery of the value of the plaintext-image pixels is not the focus of this work.

3.3.2 Image Decryption Algorithm

To decrypt the encrypted image, the transformations used in the encryption algorithm are applied in the reverse order by generating the seed δ_x from $\delta_{N/2}$ to δ_1 . Let the received cipher-image at the decryption side be represented by $\hat{C}$. If there is no distortion in the channel, then $\hat{C} = C$. The channel can be considered as either the transmission or storage medium. The distortion in the channel can be because of malicious tampering or non-malicious operation like JPEG compression, in our particular case. Besides the received cipher-image, the decryption

A JPEG Compression Tolerant Image Encryption Scheme

algorithm will require the 128-bit seed δ_1 , the value of N , the scaling/quantization parameters and the minimum and maximum pixel values of the plaintext-image.

Following are the steps involved in image decryption:

1. Inverse quantization and offset is applied to $\hat{C}$ to get $\hat{P}_N$. Due to quantization in the encryption algorithm, $\hat{P}_N \neq P_N$ even if $\hat{C} = C$. Let $P_{D_0} = \hat{P}_N$.
2. Let the iteration count starts from 1 and goes up to N . For each iteration, the parameter x is calculated using the rule given by Eq. 3.3.3, i.e. $x = (i + 1)/2$ for odd iteration and $x = i/2$ for even iteration.
3. For each odd iteration, the seed δ_{f+1} is calculated using the following equation:

$$\delta_{f+1} = \delta_1 + f, \quad (3.3.6)$$

where $f = (N/2) - x$. The seed δ_{f+1} is then used to initialize the PRNG to generate the random matrix R_{f+1} and the permutation vector $\vec{\omega}_{f+1}$. SVD is applied to R_{f+1} to obtain the matrices U_{f+1} and V_{f+1} . For example, if $N = 8$, then for the first iteration, $i = 1$, $\delta_4 = \delta_1 + 3$. This shall generate R_4 and $\vec{\omega}_4$. Through the random matrix R_4 , the matrices U_4 and V_4 will be obtained.

4. For correct decryption, it is necessary that for all the iterations, $\Phi_i = V_{f+1}$ for the i^{th} odd iteration ($i \bmod 2 = 1$) and $\Phi_i = U_{f+1}$ for the $i^{th} + 1$ even iteration ($i \bmod 2 = 0$).
5. For each iteration, one of the following rules will be applied:

$$P_{D_i} = \Gamma_{\vec{\omega}_{f+1}}^{-1} (\Phi_i^{-1}(P_{D_{i-1}})). \quad (3.3.7)$$

$$P_{D_i} = \Gamma_{\vec{\omega}_{f+1}}^{-1} ((P_{D_{i-1}})\Phi_i^{-1}). \quad (3.3.8)$$

A JPEG Compression Tolerant Image Encryption Scheme

Equation 3.3.7 will be used if Eq. 3.3.4 is used by the encryption algorithm. Similarly, Eq. 3.3.8 will be used if Eq. 3.3.5 is used by the encryption algorithm.

6. After N iterations, the resultant image P_{D_N} is adjusted such that its minimum and maximum pixel values is equal to that of the plaintext-image. We call this image as the decrypted image that is represented by P_D .

3.4 Experimental Results and Discussion

In this Section, several experimental results are presented to demonstrate the effectiveness of the proposed scheme. All the experiments are based on images of size 256×256 pixels. The outcome of these results have been instrumental towards analyzing important characteristics of the proposed scheme. In the following sections, experimental results related to quality of encryption and decryption, robustness of the encryption algorithm to JPEG compression, effect of tampering, statistical analysis and the throughput of encryption and decryption algorithms are presented.

3.4.1 Image Encryption and Decryption

Due to quantization in the encryption algorithm, the pixel values of the decrypted-image are not 100% same as that of the plaintext-image. The error in the pixel values between the plaintext-image and the decrypted-image is, however, very small. Figure 3.1a shows the original Cameraman image (plaintext-image). Figure 3.1b shows the encrypted Cameraman image in which the pixel values are scaled and then quantized between 0 and 255. The number of encryption rounds were 10. The decryption result is shown in Fig. 3.1c. The decrypted image has a PSNR of 43.2dB and is indistinguishable from the plaintext-image shown in Fig. 3.1a. The decrypted image has a high PSNR which indicates a good quality reconstruction of the respective plaintext-image. Similar results were obtained upon

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.1: Encryption and decryption results for the Cameraman image. (a) Cameraman image (plaintext-image). (b) Encrypted Cameraman image. (c) Decrypted Cameraman image: PSNR=43.2dB.

testing this algorithm with other images in our database. These results indicate the following:

- The cipher-image has no perceptual similarity with the plaintext-image.
- Despite of quantization at the encryption side, the perceptual quality of the decrypted-image is as good as the original plaintext-image with a high value of PSNR.
- The texture of the encrypted-image is different as compared to conventional encryption in which a random noise-like pattern is obtained. This point is elaborated in the next section.

3.4.2 Robustness to JPEG Compression

In this section, experimental results are presented to show the robustness of the proposed encryption scheme against JPEG compression. By robustness it is meant that if the encrypted image data is JPEG compressed, the decryption algorithm should still be able to recover the plaintext-image. This feature is one of the main contributions of this work. The distortion in the decrypted image, however, will increase if the encrypted image data is subjected to JPEG compression with decreasing JPEG quality factor (QF). As a test case, the Cameraman image (low texture) and the Baboon image (high texture) images were taken; both having a size of 256×256 pixels. These two images were encrypted using 10 rounds of iterations. The scaled and quantized version of the encrypted image matrices were stored as BMP files. It is imperative to mention that due to quantization, the size of the plaintext-images and the encrypted BMP image files of both these images is same, i.e. 65kB. Hence the proposed encryption algorithm does not increase the file size.

We then observed the effect of JPEG compression on the quality of the decrypted-image and the file size. This was done by obtaining the encrypted Cameraman and Baboon image matrices from the Cameraman and Baboon encrypted BMP image files, each having a file size of 65kB. The uncompressed encrypted image matrices were then converted into JPEG format with a QF of 100. The size of JPEG encrypted Cameraman and Baboon image files was 48.2kB and 52.5 kB, respectively. The JPEG converted files were then again uncompressed to obtain lossy encrypted Cameraman and Baboon image matrices. Decryption was then performed on these matrices. The result of decryption is shown in Fig. 3.2a and Fig. 3.2b, respectively. The PSNR of the decrypted Cameraman and Baboon images was 39.8db and 42.7 dB, respectively. The effect on the quality of the decrypted output was further observed by increasing the JPEG compression ratio with QF of 90, 80 and 70. The corresponding decryption results are shown in Fig. 3.3 to Fig. 3.5. Tables 3.1 and 3.2 show the size of JPEG files obtained by compressing the encrypted Cameraman and Baboon BMP files at different

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.2: Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=39.8dB) when the encrypted Cameraman image data was JPEG compressed with QF=100. (b) Decrypted output (PSNR=42.7dB) when the encrypted Baboon image data was JPEG compressed with QF=100.

JPEG QF. In addition, the quality of decrypted Cameraman and Baboon image in terms of PSNR is also shown. In all these cases, the size of the Cameraman and Baboon image BMP files was 65kB.

It is noteworthy to mention that with conventional cryptographic ciphers or with most of the image encryption schemes so far proposed in the literature, it is not possible to recover correct plaintext-image if JPEG lossy compression is performed on the encrypted image data. However, with the proposed scheme, successful decryption is possible even if the encrypted image data is JPEG lossy compressed as demonstrated by the results depicted in Fig. 3.2 to Fig. 3.5. At JPEG QF of 100, the quality of decrypted output is quite good with a high PSNR.

Increasing the JPEG QF would start degrading the quality of the decrypted output. Nevertheless, the proposed scheme is a step forward to demonstrate a cryptosystem that can work in a lossy compression environment. The proposed cryptosystem itself is an independent entity and is not part of any compression algorithm, for example, JPEG. Another important feature of the proposed scheme is that the size of the encrypted file is never greater than the size of the plaintext-image file. For example, the size of the encrypted BMP image files and

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.3: Effect of JPEG compression on the decrypted output (a) Decrypted output (PSNR=24.2dB) when the encrypted Cameraman image data was JPEG compressed with QF=90. (b) Decrypted output (PSNR=24.4dB) when the encrypted Baboon image data was JPEG compressed with QF=90.

Figure 3.4: Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=21.5dB) when the encrypted Cameraman image was JPEG compressed with QF=80. (b) Decrypted output (PSNR = 21.2dB) when the encrypted Baboon image data was JPEG compressed with QF=80.

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.5: Effect of JPEG compression on the decrypted output. (a) Decrypted output (PSNR=20.6dB) when the encrypted Cameraman image data was JPEG compressed with QF=70. (b) Decrypted output (PSNR=21.2dB) when the encrypted Baboon image data was JPEG compressed with QF=70.

Table 3.1: Robustness to JPEG compression: Cameraman image.

JPEG QF	JPEG Compressed File	PNSR of Decrypted Image
100	48.2kB	39.8dB
90	20.3kB	24.2dB
80	14.6kB	21.5dB
70	12.6kB	20.6dB
60	12.1kB	19.4dB
50	11.2kB	18.6dB
40	10.3kB	17.4dB
30	9.4kB	16.6dB

A JPEG Compression Tolerant Image Encryption Scheme

Table 3.2: Robustness to JPEG compression: Baboon image.

JPEG QF	JPEG Compressed File	PNSR of Decrypted Image
100	52.5kB	42.7dB
90	23.3kB	24.4dB
80	15.4kB	21.2dB
70	12.9kB	21.2dB
60	12.2kB	19.4dB
50	11.3kB	19.7dB
40	10.4kB	19.2dB
30	9.4kB	18.4dB

their respective plaintext-image files is same. The size of the JPEG compressed encrypted image file decreases considerably as the JPEG QF is decreased.

3.4.3 PSNR Comparison

In this section, we study the effect of encryption algorithm on the quality of JPEG decompressed decrypted images with respect to decompressed plaintext-images. In particular, we compare the PSNR of JPEG decompressed plaintext-images with the PSNR of JPEG decompressed images after passing through the encryption-decryption system. We further compare the PSNR of decrypted images without JPEG compression with JPEG decompressed decrypted images. For PSNR comparison, we have used four images; Cameraman, Baboon, Fruit and Tank.

We first see the effect of the encryption-decryption process on the PSNR of JPEG decompressed decrypted images. This is done by observing the change between the PSNR of JPEG decompressed decrypted output and the PSNR of the corresponding JPEG decompressed plaintext-image without encryption-decryption. Encryption and decryption were done using 10 rounds of iterations. Figure 3.6 shows the results of this comparison. The solid line in Fig. 3.6a shows the PSNR of the Cameraman image when it was subjected to JPEG compression with JPEG QF varying from 100 to 10. This is the effect of JPEG compression

A JPEG Compression Tolerant Image Encryption Scheme

on the plaintext-image without encryption. To see the effect of JPEG compression on the encryption-decryption process, the line marked with stars in Fig. 3.6a shows the PSNR when the encrypted Cameraman image was JPEG compressed and the corresponding decompressed JPEG result was decrypted to reconstruct the plaintext Cameraman image. Similarly, Fig. 3.6b, Fig. 3.6c and Fig. 3.6d show the results for the Baboon, Fruit and the Tank images. From these results, we observe that the PSNR for JPEG decompressed decrypted images is low as compared to the corresponding PSNR of the plaintext JPEG decompressed images.

To show the effect of JPEG compression on the quality of the decrypted output, the results are summarized in Table 3.3 and Fig. 3.7. For example, in case of the Cameraman image, the PSNR of the decrypted output without applying JPEG compression is 43.2dB. From Fig. 3.7, we observe that as the encrypted Cameraman image is subjected to JPEG compression, the PSNR of the JPEG decompressed decrypted output starts decreasing. Among the results shown in Table 3.3 and Fig. 3.7, the Tank image which has the highest texture gives the worse case performance in terms of PSNR. As the JPEG quality factor decreases, the difference in the pixel values of the uncompressed encrypted image and the JPEG decompressed encrypted image increases. However, the Φ_i matrices at the encryption and decryption ends are same as they are generated using the secret key. As a result, distortion is introduced in the decrypted output. The other reason for distortion in the decrypted output, we believe, is the quantization at the encryption side which is done so that each pixel value of cipher-image is 8 bits. The inherent diffusion in the encryption/decryption algorithm due to matrix multiplication, which although is useful in case of security as shown in Section 3.5 (Security Analysis), effects the quality of the JPEG decompressed decrypted output.

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.6: Effect of encryption on PSNR. (a) PSNR comparison of JPEG decompressed Cameraman image with and without encryption. (b) PSNR comparison of JPEG decompressed Baboon image with and without encryption (c) PSNR comparison of JPEG decompressed Fruit image with and without encryption (d) PSNR comparison of JPEG decompressed Tank image with and without encryption.

Table 3.3: PSNR of decrypted images without JPEG compression.

Image	PSNR
Cameraman	43.2dB
Baboon	43.0dB
Fruit	45.4dB
Tank	38.6dB

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.7: PSNR of JPEG decompressed decrypted images.

3.4.4 Effect of JPEG Compression on Images Encrypted using AES

In this section, we show the effect of JPEG compression on an image that is encrypted using the AES encryption algorithm [1]. To test this, a 256×256 Cameraman image was encrypted using the AES algorithm. The size of the BMP file in this case was 65kB. Figure 3.8a shows the AES encrypted Cameraman image, while Fig. 3.8b shows its respective histogram. The AES encrypted image data was then JPEG compressed with a QF of 100. The size of the corresponding JPEG compressed file was 100kB which is much larger than the uncompressed encrypted BMP file. Figure 3.8c shows the result of AES decryption when the JPEG compressed encrypted file was used to reconstruct the lossy uncompressed AES encrypted image matrix. Since 128-bit AES algorithm works with 16 bytes of data at a time, i.e. 4×4 blocks, hence within a 4×4 block of an encrypted AES image, if a single pixel value is changed by 1-bit, the entire 4×4 block will not be successfully decrypted. In contrast, the proposed encryption scheme enables to perform correct decryption with some distortion even if the encrypted image data passes through a lossy compression channel. The flat histogram shown in Fig. 3.8b depicts the fact that the AES algorithm transforms an arbitrary image

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.8: Response of AES Encryption to JPEG compression. (a) AES encrypted Cameraman image. (b) Histogram of AES encrypted Cameraman image. (c) Decrypted output when AES encrypted Cameraman image data was JPEG compressed with QF=100.

into a random noisy image that follows the uniform distribution having a texture in which the neighboring pixels are highly de-correlated. As a result, the JPEG compression efficiency is hampered. In contrast, the texture of the encrypted image obtained using the proposed scheme has smooth sections of patterns in which the neighboring pixels have some correlation. This enables the cipher-image to have a good compression efficiency.

3.4.5 Histogram of the Encrypted Image

An image encryption scheme is required to resist statistical attacks [43]. Generally speaking, this is accomplished by ensuring that the cipher-image has a flat histogram for any plaintext-image, thus concealing the pixel frequency distribution of the plaintext-image [7], [43]. In contrast, the proposed image encryption scheme generates a cipher-image whose histogram closely approximates a Gaussian distribution function. After doing experiments with a number of different images, we have observed that this fact remains consistent for cipher-images whose corresponding plaintext-images are either natural, binary or random noise images. For the purpose of demonstration, Fig. 3.9a to Fig. 3.9d show the histograms of cipher-images whose corresponding plaintext-image are natural, binary and random noise. All these histograms follow approximate Gaussian distribution with a mean value that is within a certain proximity from the mid-range gray-scale value of 128.

The Gaussian shape of the cipher-image histogram can be explained using the Central Limit Theorem (CLT) which basically says that whenever a random experiment is repeated, the average or sum of the random variables obtained over this repetition tends to have a Gaussian distribution as the sample size tends to infinity [110]. In most of the cases, if the sample size, $n \geq 30$, the Gaussian approximation will be true regardless of the shape of the population [110], [111]. To show how the CLT fits well with respect to the proposed encryption scheme, we give the following proposition.

Proposition 3

The distribution of the matrix P_i given by Eq. 3.3.4 and Eq. 3.3.5 in Section 3.3.1 would be approximate Gaussian for any number of encryption rounds.

Proof 3

For simplicity, let us represent $\Gamma_{\vec{\omega}_x}(P_{i-1})$ in Eq. 3.3.4 and Eq. 3.3.5 by a matrix A . Therefore, $P_i = \Phi_i A$ or $P_i = A \Phi_i$. If $P_i = \Phi_i A$, then:

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.9: Histogram illustration of cipher-images. (a) Histogram of encrypted Cameraman image. (b) Histogram of encrypted Baboon image. (c) Histogram of encrypted binary image. (d) Histogram of encrypted noise (uniform randomly distributed) image.

$$P_i(j, k) = \phi_{j1}A_{1k} + \phi_{j2}A_{2k} + \cdots + \phi_{jn}A_{nk}. \quad (3.4.9)$$

If $P_i = A\Phi_i$, then:

$$P_i(j, k) = A_{j1}\phi_{1k} + A_{j2}\phi_{2k} + \cdots + A_{jn}\phi_{nk}. \quad (3.4.10)$$

Each entry ϕ_{jk} of the matrix Φ_i can be considered as an independent random variable because the matrix Φ_i is generated from a random matrix whose each entry is independently identically distributed as outlined in Section 3.3.1. Therefore, each term of Eq. 3.4.9 or Eq. 3.4.10 can also be considered as an independent random variable. Since the parameter $n \geq 30$ in Eq. 3.4.9 and Eq. 3.4.10, there-

A JPEG Compression Tolerant Image Encryption Scheme

fore according to the central limit theorem, the elements of P_i in Eq. 3.4.9 and Eq. 3.4.10 would follow an approximate Gaussian distribution.

The results presented in this section show that the proposed encryption scheme maps any plaintext-image into a cipher-image that has a Gaussian looking histogram, thus concealing the gray-level frequency distribution of the plaintext-image pixels.

3.4.6 Effect of Repeated Multiplication on the Cipher-image

By applying Proposition 1, it can be shown that each entry, $P_i(j, k)$ of P_i in Eq. 3.3.4 cannot be greater than the length of the k^{th} column of $\Gamma_{\vec{\omega}_x}(P_{i-1})$. Similarly, proposition 2 can be used to show that each entry $P_i(j, k)$ of P_i in Eq. 3.3.5 cannot be greater than the length of the j^{th} row of $\Gamma_{\vec{\omega}_x}(P_{i-1})$. In reality, however, each entry $P_i(j, k)$ is quite small as compared to these bounds. Figure 3.10a and Fig. 3.10d show the minimum and maximum values of P_i obtained at each iteration for 500 consecutive multiplications ($i = 500$) using the encryption transformations given by Eq. 3.3.4 and Eq. 3.3.5. This shows how orthogonal matrices helps to keep each entry of P_i bounded irrespective of the fact that how many multiplications rounds are carried out.

3.4.7 Effect of Tampering on the Decrypted Output

The proposed scheme offers resilience to JPEG compression which implies that change in the pixel values of the encrypted image up to a certain level still ensures correct decryption with some distortion as demonstrated by the results shown in Section 3.4.2. This robustness feature bring us towards an important security issue. Is it possible for an attacker to cause tampering in the encrypted image that goes undetected at the decryption end? The attacker may cause tampering in the encrypted image such that either the entire decrypted image is changed, or a part of the decrypted image is modified. For example, if Alice sends an

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.10: Effect of repeated multiplication on the cipher-image. (a) Minimum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.4. (b) Maximum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.4. (c) Minimum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.5. (d) Maximum value of P_i at the i^{th} iteration using encryption transformation given by Eq. 3.3.5.

A JPEG Compression Tolerant Image Encryption Scheme

encrypted image of Mr. A to Bob, can Eve intercept the cipher-image, modify it, and then retransmit the tampered encrypted image to Bob such that decryption of the tampered image by Bob yields picture of Mr. B instead of Mr. A. In this section, we show that the proposed scheme has the capability to resist such an attack. As mentioned in the encryption algorithm, the seed δ_1 is generated by XORing the hash of the plaintext-image and the secret key. This means that for each plaintext-image, the seed value used to initialize the PRNG will be different. The proposed scheme resists the possibility of allowing an attacker to intercept a cipher-image encrypted using a specific seed and then tamper it with the objective that the tampered encrypted image decrypts to an image of attacker's choice. For this attack to work, the attacker should know the correct seed which is only known to the sender and the receiver. Without knowledge of the correct seed, an attacker cannot map an arbitrary plaintext-image into a cipher-image that can be correctly decrypted.

Some experimental results are now presented to show how the proposed scheme can resist such a tampering. For the purpose of illustration, we take encrypted Cameraman and encrypted Baboon images. Figure 3.11a to Fig. 3.11c show the decryption result when randomly selected 32×32 , 64×64 and 128×128 blocks of encrypted Baboon image were pasted on the top left corner of the encrypted Cameraman image. Similar results were obtained if the encrypted Baboon image blocks were pasted on any part of the encrypted Cameraman image. Figure 3.11d shows the decryption result when the entire encrypted Cameraman image is replaced by the encrypted Baboon image. These results show that if a genuine encrypted image is tampered, visual distortion will be caused in the decrypted image. The amount of visual distortion will increase with the increase in the tampering. Tampering, however, cannot make the decrypted image to look like any other arbitrary image either locally or globally other than the genuine plaintext-image.

With slight modification, the proposed scheme can be made fragile so that even if 1-bit of tampering is done in the cipher-image data, decryption would render

A JPEG Compression Tolerant Image Encryption Scheme

(a)

(b)

(c)

(d)

Figure 3.11: Effect of tampering on the decrypted output. (a) Decrypted output when a 32×32 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (b) Decrypted output when a 64×64 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (c) Decrypted output when a 128×128 randomly selected encrypted block of the Baboon image was pasted on the top left corner of the encrypted Cameraman image. (d) Decrypted output when the encrypted Cameraman image was replaced by encrypted Baboon image.

A JPEG Compression Tolerant Image Encryption Scheme

a gibberish result. However, with this modification, the scheme will not be able to tolerate JPEG compression. The modification to make the proposed scheme fragile to 1-bit change is as follows:

Modification in the encryption algorithm

After Step 5 of the image encryption algorithm in Section 3.3.1, cryptographic hash of the cipher-image C is calculated using MD5. This hash is then XORed with the seed δ_1 to generate another seed, $\hat{\delta}_1$. The seed $\hat{\delta}_1$ is now dependent on C and δ_1 . Instead of δ_1 , the seed $\hat{\delta}_1$ is securely passed to the decryption algorithm.

Modification in the decryption algorithm

The decryption algorithm will use C and $\hat{\delta}_1$ to get the original seed δ_1 . This is done by calculating the MD5 hash of C and then XORing it with $\hat{\delta}_1$. Let $\bar{C}$ be the encrypted image in which one or more bits have been flipped. Even if a single bit in C is changed, then $H(\bar{C}) \neq H(C)$. XORing $H(\bar{C})$ with $\hat{\delta}_1$ will not produce δ_1 . This in turn will not generate the correct random matrices R_x and the permutation vectors Γ_{ω_x} that were used at the encryption side. As a result, the decryption output would be completely gibberish.

The proposed scheme can therefore be used at both semi-fragile and fragile levels. At the semi-fragile level, if the encrypted image data is passed through JPEG lossy compression channel, correct decryption is possible with some distortion. In this case, 1-bit tamper detection in the cipher-image is not possible. If the proposed scheme is used at a fragile level, even a single bit change in the cipher-image shall produce a decrypted image that would be completely gibberish.

3.4.8 System's Throughput

In this section, analysis is presented to show the relative speeds of various sub-components of the proposed encryption/decryption algorithms. This analysis

A JPEG Compression Tolerant Image Encryption Scheme

does not depict optimized throughput as the algorithms were tested in the MATLAB programming environment. The size of the image was taken as 256×256 pixels. Testing was done on a Pentium M, 1.6 GHz machine having a RAM of 512MB running under WINDOWS XP operating system. Following is a breakdown of the major tasks that need to be carried out in the encryption and decryption algorithms. With each task, a symbol is used to represent the time it takes to complete that task.

1. Generating $n \times n$ random matrix: t_{rand}
2. Generating the permutation vector: t_P .
3. Calculating the SVD of the $n \times n$ random matrix: t_{SVD} .
4. 32×32 block permutation of the image: t_{perm} .
5. Multiplication of P_i or P_{D_i} with Φ_i : t_{mul}
6. Scaling/inverse scaling, quantization/inverse quantization of the encrypted/decrypted image matrix: t_s .

For N iterations, the total time, t_T for encryption/decryption can be approximated estimated as:

$$t_T = 0.5 N t_{rand} + 0.5 N t_P + N t_{perm} + 0.5 N t_{SVD} + N t_{mul} + t_{sq}. \quad (3.4.11)$$

Figure 3.12a shows the percentage of time taken by each task. The most expensive operation is the SVD that takes approximately 89% of the total time. Figure 3.12b shows the throughput (Mbps) when SVD is applied at odd iterations. One SVD operation gives U and V matrices. The U matrix is used in the i^{th} odd iteration and the V matrix is used in the next $i^{th} + 1$ even iteration. Figure 3.12c shows the throughput when a single U or V matrix is used for all the iterations. In this case, if the number of iterations are more than

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.12: Throughput and speed analysis of the proposed scheme. (a) Percentage of time occupied by each task. (b) Throughput in Mbps when SVD is applied at odd iterations. (c) Throughput when SVD is calculated once before the start of iterations.

one, a substantial improvement in the throughput is achieved as compared to the throughput shown in Fig. 3.12b. This technique to boost the throughput can still give good security as long as the number of iterations is greater than one. It is pertinent to mention that the proposed encryption technique simply uses SVD as a means to generate the orthogonal matrices. Obtaining optimized throughput is not the focus of this work. For practical implementation of the proposed scheme, orthogonal matrices can be generated using much simpler techniques, such as Gram-Schmidt Orthogonalization [104]. An advantage of SVD is that it does not require a full-rank matrix to generate orthogonal matrices. On the other hand, Gram-Schmidt Orthogonalization requires linearly independent columns vectors

to generate orthonormal basis.

3.5 Security Analysis

An encryption scheme is required to fulfill a number of security requirements such as large key space, resistance against ciphertext-only attack, known-plaintext and chosen-plaintext attacks, chosen-ciphertext attack, statistical attacks, etc. In the following sections, a number of important security characteristics of the proposed scheme are analyzed.

3.5.1 Key Space and Key Sensitivity

In this section, we analyze the security of the secret key used in the encryption and decryption algorithms. As mentioned in Section 3.3, the initial seed value δ_1 is the actual key used by the encryption and decryption algorithms. The seed δ_1 is generated at the encryption side by XORing K and $H(P_0)$. The security of key is analyzed from two different perspectives. First is the key space, i.e., how many different combinations of the seed value is possible to generate R_x and $\Gamma_{\tilde{\omega}_x}$ such that each R_x and $\Gamma_{\tilde{\omega}_x}$ is random and distinct. Secondly, do all these combinations provide strong cryptographic security? For example, in chaotic-based encryption, the effective key space is limited and there are sets of keys that provide more security than the others [43], [44]. One may ask why $H(P_0)$ is not directly used as the seed? The use of K gives the sender of an encrypted image the flexibility to use different seed values for same images. For example, if Alice wants to send the same image encrypted to 10 different recipients, she can use a different secret parameter K for each recipient. This enables Alice to generate different seed values for the same plaintext-image. This prevents a recipient from decrypting cipher-images of other recipients. Further, at some stage, even if the actual plaintext-image is exposed, the seed value used to encrypt the plaintext-images intended for other recipients remain secret.

Key Space

As for the proposed encryption scheme, the key used is 128-bit, hence there are 2^{128} different combinations of δ_x from which any one can be used. The effective key space, however, is limited due to the period of the PRNG; the number of individual numbers generated by the PRNG before it begins to repeat. For a 256×256 image, the total entries in R_x and $\Gamma_{\vec{\omega}_x}$ are $65536 + 64 = 65600$. For all the possible combinations of δ_x , the total random sequences generated by the PRNG will be 2^{128} , with each sequence having 65600 numbers. Therefore, the total random numbers required to be generated will be $2^{128} \times 65600 = 2.2323^{43}$. If the period of the PRNG is greater than this number, then the effective key space will be 2^{128} . Generally, the period of the PRNG should be greater than $2^{128} \times (n^2 + (n^2/1024))$, where $n \times n$ is the size of the image that is divided into 32×32 blocks for permutation.

Another important aspect of the effective key space is to check how the change in key effects the matrices U_x and V_x . It is extremely important that two un-correlated matrices, R_a and R_b should generate the corresponding U_a , U_b , V_a and V_b matrices that are also un-correlated, otherwise the matrices R_a and R_b will be considered as weak. To check this, a simulation was carried out by generating 1000 random R_x matrices, where $1 \leq x \leq 1000$. At first, all the R_x matrices were compared with each other using the normalized correlation coefficient, η [112]. If there is a total of Y number of R_x matrices, i.e. $1 \leq x \leq Y$, then the total number of comparisons, C_T will be

$$C_T = \frac{Y!}{2!(Y-2)!}. \quad (3.5.12)$$

For the 1000 random R_x matrices, the total number of comparisons will be 499500. The result of this comparison for the R_x matrices is shown in Fig. 3.13a. The maximum value of $|\eta|$ is 0.0797. We are now interested to see whether the correlation pattern obtained for R_x matrices shown in Fig. 3.13a also exists for the U_x and V_x matrices. The results are quite encouraging as depicted in Fig.

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.13: Checking effective key space by generating 1000 random R_x matrices. (a) Comparison result between $(R_a, R_b) \in R_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (b) Comparison result between $(U_a, U_b) \in U_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (c) Comparison result between $(V_a, V_b) \in V_x$, where $1 \leq x \leq 1000$ and $a \neq b$. (d) Comparison result between $U_a \in U_x$ and $V_a \in V_x$, where $1 \leq x \leq 1000$.

3.13b, Fig. 3.13c and Fig. 3.13d. Figure 3.13b shows the result when all the U_x matrices were compared with each other. Similar to Figure 3.13a, each U_a and U_b was taken from the set U_x such that $a \neq b$ and η was calculated. In this case, $\max(|\eta|) = 0.0816$ which shows that a random R_x will also generate a random U_x . Figure 3.13c shows the result when all the V_x matrices were compared with each other. In this case, $\max(|\eta|) = 0.0971$. Figure 3.13d shows the result when all the U_x matrices were compared with all the V_x matrices. In this case, the number of comparisons were 1000000 and $\max(|\eta|) = 0.0920$. If $\eta(Z_a, Z_b)$ represents the normalized correlation coefficient between two matrices, Z_a and Z_b , then for a certain scaler, $\epsilon \ll 1$, and $a \neq b$, if $|\eta(R_a, R_b)| < \epsilon$, then $|\eta(U_a, U_b)| <$

A JPEG Compression Tolerant Image Encryption Scheme

$$\epsilon, |\eta(V_a, V_b)| < \epsilon \text{ and } |\eta(U_a, V_b)| < \epsilon.$$

The above results show that if R_x is a random matrix, then the corresponding U_x and V_x matrices will also be random. Since R_x will be generated using a PRNG, so its randomness is guaranteed provided that the period of PRNG is greater than $2^{128} \times (n^2 + (n^2/1024))$.

Key Sensitivity

The next security issue w.r.t. the secret key is that how much the cipher-image C is sensitive towards the change in the key. A 1-bit change in δ_x should cause significant change in the cipher-image. This is achieved due to the fact that the PRNG would output a new random matrix R_x if δ_x is flipped by a single bit. To check the key sensitivity, 1001 random matrices R_x were generated using 1001 different seed values δ_x starting from 1 to 1001. For each R_x , the matrix U_x was generated using the SVD. Let $C_x = U_x P_0$ represent the cipher-image, where P_0 was taken as the Cameraman image shown in Fig. 3.1a. The first cipher-image, C_1 was then compared with all the cipher-images, C_x where $2 \leq x \leq 1001$. The result of this comparison obtained using the normalized correlation coefficient is shown in Fig. 3.14a. For all the comparisons, the value of $|\eta| \ll 1$. Figure 3.14b shows the result of decryption when the seed value of 500 was correct and the remaining seed values were incorrect. Again, the value of $|\eta| \ll 1$. These results demonstrate the sensitivity of the encryption and decryption algorithms towards a 1-bit change in the seed value. Since the seed value is obtained by XORing K and $H(P_0)$, therefore, a 1-bit change in either K or the plaintext-image would yield similar results that is obtained by flipping a single bit of δ_x .

3.5.2 Brute Force Attack

The basic objective of a brute force attack is to try all possible combinations of the secret key to recover the plaintext-image and or the secret key. It is, however, pertinent to mention that a security loophole in the encryption algorithm can

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.14: Illustration of sensitivity of the seed δ_x towards the cipher-image and the decrypted-image. (a) Comparison of a cipher-image with cipher-images generated using different seed values. (b) Comparison of the decrypted-image with the actual plaintext-image for different values of δ_x . The value δ_{500} is the correct seed value to recover the plaintext-image.

provide a passage to find the secret key without trying all possible key combinations. The analysis of brute force attack should therefore consider the effect of the secret key on various security sub-components of an encryption algorithm. The 128-bit secret key δ_1 used in the proposed scheme helps to conceal the Φ_i matrices and the permutation vectors $\Gamma_{\vec{\omega}_x}$ that are used by the encryption and decryption algorithms. A brute force attack will require trying all the 2^{128} combinations of the secret key. Instead of trying all the possible combinations of the seed value, an attacker may exploit potential weaknesses in the encryption algorithm to find the plaintext-image or some parameter that can enable the attacker to encrypt or decrypt images.

Let us analyze the difficulty of finding the Φ_i matrices and the permutation vector $\Gamma_{\vec{\omega}_x}$ through the brute force attack. For a 256×256 image, the total number of entries in the Φ_i matrix will be 65536. Let each entry in Φ_i be represented by ω_L , where $-1 < \omega_L < 1$ due to SVD computation. Since ω_L is not an integer, it will require 32-bit or 64-bit floating point precision, or generally v bits to store. For correct decryption, it is necessary that the entries in the Φ_i matrices are correct and are placed at the correct position. For a 256×256 image, the total possible combinations required to find the exact Φ_i is $2^{(v)(n^2)} = 2^{(32)(65536)}$, in case

A JPEG Compression Tolerant Image Encryption Scheme

$v = 32$. This involves even much larger calculations than applying brute force on the 128-bit secret key.

Let us now create an easy scenario that goes in favor of an attacker. Assume that each entry in the Φ_i matrix is represented by one decimal digit of accuracy that may be a positive or negative number in the range between -0.8 to 0.8 excluding the number zero. Hence there will be a total of 16 different numbers that requires 4 bits to represent. Further, assume that all the entries of a specific column of the Φ_i matrix are same. Even with such an easy assumption, the number of possibilities required to find the correct Φ_i are $2^{(4)(256)}$. With this assumption, even for small images of 64×64 pixels, the number of possibilities are $2^{(4)(64)}$. This analysis shows that it is extremely difficult to find the Φ_i matrix by brute force. This hypothetical scenario was created to demonstrate the difficulty of finding the columns of Φ_i even if each column has similar entries. In reality, brute force on Φ_i is much more difficult because SVD generates U and V matrices whose columns are different and each column usually has different entries. Hence, even if a single U or V matrix is used by the encryption and decryption algorithm and if the number of iterations are say 2, 3 or more, then the proposed scheme shall still have a high security to resist the brute force attack. The impact of the number of iterations on system's security is discussed in Section 3.5.4. Using a single U or V matrix has the advantage that SVD is only required to be calculated once. This increases the system's throughput as explained in Section 3.4.8.

It is pertinent to mention that merely finding the correct Φ_i is not sufficient to perform decryption because each iteration also involves permutation. The permutation is done by dividing the image into 32×32 blocks. For an $n \times n$ image, the total number of permutations will be $(n^2/1024)!$. For a 256×256 image, there shall be $64!$ permutations. Permutation makes the brute force attack even more difficult. For each iteration, the attacker needs to find the correct Φ_i matrix and the permutation sequence.

3.5.3 On the Use of an Image-Independent Encryption Key

The proposed encryption algorithm in Section 3.3.1 uses the seed δ_1 as the encryption key, which is obtained by XORing K and $H(P_0)$. Hence for each encrypted image, the sender needs to transmit a different δ_1 to the receiver. The size of δ_1 is 128 bits irrespective of the size of the input plaintext-image and the cipher-image. This additional overhead is not very significant if we compare it with the size of the encrypted image. For example, the overhead is only 0.083% for small gray-scale images of size 160×120 pixels. The overhead will become less significant as the size of the image increases. To securely transmit δ_1 from the transmitter to the receiver, a symmetric encryption algorithm, like the 128-bit AES [1] can be used. However, one may still argue that this 128-bit overhead per image frame might cause bandwidth/storage issues in case if a very large number of image frames are required to be transmitted or stored.

One of the main reasons for using an image-dependent encryption/decryption key is to create a very high diffusion mechanism with respect to the plaintext-image. Since the seed δ_1 is obtained by XORing K and $H(P_0)$, hence even a single bit change in the plaintext-image P_0 would create a different random $H(P_0)$ which in turn would generate a different cipher-image. The diffusion in the cipher-image with respect to the plaintext-image is obtained due to the cryptographic property of the MD5 hash function and is very useful towards preventing known-plaintext and chosen-plaintext attacks [4] as shown in the next section. Interestingly, the proposed encryption algorithm has an additional inbuilt diffusion mechanism due to block permutation and matrix multiplication. Due to this inbuilt diffusion mechanism, we suggest that an image-independent encryption/decryption key can also be used. This simply means that the seed δ_1 used to initialize the PRNG can be any large number selected at random, which would then serve as a single session key for image encryption and decryption. In this case, diffusion in the cipher-image with respect to the plaintext-image is obtained due to the inherent permutation and matrix multiplication mechanism in the encryption algorithm.

A JPEG Compression Tolerant Image Encryption Scheme

In the next section, detailed cryptanalysis is presented to show that even if an attacker has a collection of plaintext-images and cipher-images encrypted with a single encryption key, the proposed scheme can still resist known-plaintext and chosen-plaintext attacks.

3.5.4 Other Attacks

In this section, we analyze the possibility of launching ciphertext-only, known-plaintext and chosen-plaintext attacks on the proposed scheme. The analysis presented in this section covers both the cases of the encryption/decryption key. In the first case, which we call as the image-dependent encryption/decryption key, the seed δ_1 is derived by XORing K and $H(P_0)$ and is different for every encrypted image. In the second case, which we call as the image-independent encryption/decryption key, the seed δ_1 is simply a large random number, preferably 128 bits that is securely shared by the sender (encryption side) and the receiver (decryption side). In this case, the same seed value can be used in a single session that may encompass encryption/decryption of a single image or multiple images.

Ciphertext-Only Attack

This is one of the most difficult among the three attacks in which the attacker has access only to a ciphertext or a collection of ciphertexts [4]. The objective of this attack is to find the plaintext-image and or the secret key. As an example, suppose the cipher-image C is obtained in three rounds of iterations; $C = \Phi_3 \Gamma_{\vec{\omega}_2} (\Phi_2 \Gamma_{\vec{\omega}_1} (\Phi_1 \Gamma_{\vec{\omega}_1} (P_0)))$. From the cipher-image C , the plaintext-image P_0 is obtained as: $P_0 = \Gamma_{\vec{\omega}_1}^{-1} (\Phi_1^{-1} \Gamma_{\vec{\omega}_1}^{-1} (\Phi_2^{-1} \Gamma_{\vec{\omega}_2}^{-1} (\Phi_3^{-1} C)))$.

Without knowledge of the secret key δ_1 and the plaintext-image, the only option an attacker has is the use of brute force to find P_0 . The difficulty of finding a single Φ_i matrix using a brute force attack has been described in Section 3.5.2. Using a brute force attack, if an attacker wishes to find $P_0 = \Gamma_{\vec{\omega}_1}^{-1} (\Phi_1^{-1} \Gamma_{\vec{\omega}_1}^{-1} (\Phi_2^{-1} \Gamma_{\vec{\omega}_2}^{-1} (\Phi_3^{-1} C)))$, then he/she needs to find the correct Φ_i matrix

A JPEG Compression Tolerant Image Encryption Scheme

and the $\vec{\omega}_x$ vector for each iteration. Suppose that the attacker has a collection of cipher-images, can he/she deduce some meaningful information from this data to find Φ_i and or $\vec{\omega}_x$? In case of image-dependent secret key, the Φ_i matrices and the $\vec{\omega}_x$ vectors will be different for different images. Further, a brute force attack on a specific cipher-image would only enable an attacker to find Φ_i and $\vec{\omega}_x$ for that particular image. In case of an image-independent key, if an attacker is successful in finding Φ_i and $\vec{\omega}_x$, then it can be used to decrypt all the images that were encrypted using the Φ_i and $\vec{\omega}_x$. This, however, would still not enable the attacker to find δ_1 because Φ_i and $\vec{\omega}_x$ are the outcome of the PRNG whose initial state is δ_1 . Since the attacker does not have the knowledge of the plaintext-images, the cipher-images of different plaintext-images would not give any clue to an attacker that could pave the way for finding Φ_i and $\vec{\omega}_x$. The only means the attacker could resort is the use of brute force which has been shown to be extremely difficult to launch.

Known-Plaintext Attack

In this attack, the attacker has a ciphertext or a set of ciphertexts for which he/she knows the corresponding plaintext [4]. The objective is to find the secret key or any side information that can be used to encrypt or decrypt arbitrary images. To thwart this attack, the permutation step in the encryption algorithm plays a very important role. Assume that the cipher-image is obtained without permutation. Then for three rounds of iteration, the cipher-image will be; $C = \Phi_3\Phi_2\Phi_1P_0$, if Φ_i is premultiplied with P_0 . If the attacker has C , P_0 and if the inverse of P_0 exists, then the attacker can obtain $\hat{\Phi} = CP_0^{-1} = \Phi_3\Phi_2\Phi_1P_0P_0^{-1} = \Phi_3\Phi_2\Phi_1$.

Using the product matrix $\Phi_1\Phi_2\Phi_3$, the individual matrices Φ_1 , Φ_2 and Φ_3 cannot be calculated. As a result, the seed value δ_1 and the secret parameter K cannot be obtained. In case of image-dependent key, knowledge of the product matrix $\hat{\Phi} = \Phi_3\Phi_2\Phi_1$ would not help the attacker to decrypt any arbitrary cipher-image because for each plaintext-image, the seed value is different thus generating different set of Φ_i matrices. However, in case of image-independent key, once the

A JPEG Compression Tolerant Image Encryption Scheme

attacker has $\Phi_3\Phi_2\Phi_1$, he/she can use it to decrypt all the cipher-images that have been encrypted using the set of Φ_i matrices. For example, if the attacker has a cipher-image, $C_g = \Phi_3\Phi_2\Phi_1G_0$, then $(\Phi_3\Phi_2\Phi_1)^{-1}\Phi_3\Phi_2\Phi_1G_0 = G_0$. The permutation step thwart these scenarios; for example after three rounds of encryption, the cipher image C will be $C = \Phi_3\Gamma_{\vec{\omega}_2}(\Phi_2\Gamma_{\vec{\omega}_1}(\Phi_1\Gamma_{\vec{\omega}_1}(P_0)))$. The cipher-image is a complicated matrix obtained by block permutation and multiplication with Φ_i at each iteration. Hence CP_0^{-1} will not give any meaningful information to the attacker. This shows that if an attacker is able to find the cipher-image and the corresponding plaintext-image, he/she will not be able to find the secret key, the secret parameter K , Φ_i matrices and or the $\Gamma_{\vec{\omega}_x}$ vectors to perform decryption of arbitrary cipher-images.

Chosen-Plaintext Attack

This attack gives more flexibility to an attacker by allowing to select a plaintext of his/her choice and observe the corresponding ciphertext [4]. The attacker can choose any plaintext-image, give it to the encryption machinery and get the corresponding cipher-image. Using this attack, an attacker can create pairs of plaintext-image and cipher-image of his/her choice to deduce the secret key or any side information that can be used to encrypt or decrypt arbitrary images.

We first analyze this attack by assuming that an image-dependent encryption key is used. Suppose the attacker selects a set of plaintext-images such that each plaintext-image has a difference of only 1 bit from the other. Can this helps the attacker to get some useful information, paving the way to find the secret key, the Φ_i matrices and or the permutation vectors $\Gamma_{\vec{\omega}_x}$? To foil this attack, the MD5 hash used in the encryption algorithm plays a very important role. Assume P_1 and P_2 are same plaintext-images with only 1 bit difference. Due to the sensitivity property of MD5 hashing algorithm [4], $H(P_1) \neq H(P_2) \Rightarrow \delta_1(P_1) \neq \delta_1(P_2)$. The Φ_i matrices and the permutation vectors generated in both these cases will be different. Hence by trying different combinations of plaintext-images, the attacker will not get any useful information since each cipher-image would look like a new

A JPEG Compression Tolerant Image Encryption Scheme

random image. In addition, there shall be no single encryption and decryption key for all these images as each key will only be valid for a single pair of plaintext-image/cipher-image.

Let us now analyze this attack by assuming that an image-independent key is used. Assume that the attacker has access to the encryption machinery that can perform encryption rounds using the same secret key. Let us assume for a moment that the permutation function is disabled and the encryption takes place in three rounds with premultiplication of Φ_i matrices with the plaintext-image P_0 ; $C = \Phi_3\Phi_2\Phi_1P_0$. If the attacker chooses P_0 such that all the entries of P_0 are zero and only the first entry of P_0 is 1, i.e., $P_0(j, k) = 1$ for row $j = 1$ and column $k = 1$, then the first column of C will reveal the first column of $\Phi_3\Phi_2\Phi_1$. Similarly, if the attacker chooses P_0 such that $P_0(j, k) = 1$ for only $j = 2, k = 1$, then then the first column of C will reveal the second column of $\Phi_3\Phi_2\Phi_1$. In general, if $P_0(j, k) = 1$ for only $j = n, k = 1$, then then the first column of C will reveal the n^{th} column of $\Phi_3\Phi_2\Phi_1$. This means an attacker will be able to find the entire matrix $\Phi_3\Phi_2\Phi_1$. Once the attacker has $\Phi_3\Phi_2\Phi_1$, he/she can decrypt all the images that were encrypted using these matrices. Hence, without permutation function, a chosen-plaintext attack can be easily launched to find the product of Φ_i matrices.

Let us see how the permutation function helps to resist this attack. Assume the same rounds of iteration with premultiplication of Φ_i matrices with the plaintext-image. Hence after three rounds of encryption, the cipher image will be; $C = \Phi_3\Gamma_{\vec{\omega}_2}(\Phi_2\Gamma_{\vec{\omega}_1}(\Phi_1\Gamma_{\vec{\omega}_1}(P_0)))$. Suppose that the attacker chooses P_0 such that $P_0(j, k) = 1$ for only $j = 1, k = 1$. After the first iteration, the intermediate cipher-image, c_1 will be; $c_1 = \Phi_1\Gamma_{\vec{\omega}_1}(P_0)$. Suppose the permutation function maps $P_0(1, 1)$ to some location at row j and column k . Then after the first iteration, the k^{th} column of c_1 will contain the j^{th} column of Φ_1 . The intermediate cipher-image, c_2 after the second iteration will be; $c_2 = \Phi_2\Gamma_{\vec{\omega}_1}(c_1)$. For an image of size 256×256 pixels with a block permutation size of 32×32 pixels, the permutation function will extract 8 sub-columns from the k^{th} column of c_1 , each

A JPEG Compression Tolerant Image Encryption Scheme

Figure 3.15: Illustration of Chosen-plaintext attack. (a) Cipher-image after one iteration. (b) Cipher-image after two iterations. (c) Cipher-image after three iterations.

sub-column having a size of 32×1 pixels. Due to permutation, each of these 8 sub-columns will be placed in randomly selected columns and the result will then be multiplied by Φ_2 to generate c_2 . Due to random position of the sub-columns, the intermediate matrix c_2 will have columns with random entries. The same process will repeat for the third iteration. Permutation is followed by matrix multiplication which basically creates diffusion in the entire column. Therefore, an attacker cannot actually pinpoint the exact permutation block. This impedes the attacker from calculating the inverse permutation function.

Figure 3.15 shows the images c_1 , c_2 and C generated under a simulation that was done using the above parameters, the plaintext-image contained all entries as

A JPEG Compression Tolerant Image Encryption Scheme

zeros except for $P_0(1, 1) = 1$. As shown in Fig. 3.15a, the matrix multiplication operation diffuses a single number into the entire column. It is pertinent to mention that this intermediate result would not be available to the attacker if the encryption machinery uses three rounds of iterations. Fig. 3.15b and Fig. 3.15c show the intermediate cipher-image c_2 and the final cipher-image C . Looking at the image C , the attacker would not be able to find the inverse permutation function. In addition, the values in the columns of C are random that have no correlation with corresponding columns of any of the Φ_i matrices. From this analysis, we conclude that the proposed scheme can resist chosen-plaintext attack for both image-dependent and image-independent keys.

3.6 Conclusion

In this chapter, a JPEG compression tolerant image encryption has been proposed that uses some very interesting properties of orthogonal matrices. A unique feature of the proposed scheme is that the cryptosystem is an independent entity and is not an integral part of any compression algorithm like JPEG. The user is free to decrypt the encrypted image directly or by compressing the encrypted image using JPEG compression and then decrypting the decompressed encrypted image. The proposed image encryption scheme uses the actual image data in a file to perform encryption and decryption. Due to this reason, even if the encrypted image is stored as a BMP file, for example, decryption is possible regardless of whether the encrypted file is converted into some other format, for example JPEG, GIF, TIFF, etc. Further, the proposed scheme has the capability to recover the plaintext-image from the cipher-image even if the cipher-image data is compressed using JPEG lossy compression. The distortion in the recovered plaintext-image will increase if the JPEG QF is decreased. The security analysis presented in Section 3.5 shows that the proposed scheme can withstand typical cryptographic attacks that are used to test an encryption scheme.

In this chapter, we have considered images of size $N \times N$ pixels. However, practically, a plaintext-image can be of any size, say, $M \times N$ pixels. In such a

A JPEG Compression Tolerant Image Encryption Scheme

case, we need to select the size of the orthogonal matrices and the corresponding option of using either premultiplication or postmultiplication keeping in view the fact that the size of the cipher-image should be equal to the size of the plaintext-image. For this, it is necessary to keep the size of the orthogonal matrix as $M \times M$ in case if the orthogonal matrix is required to be premultiplied with the plaintext-image. On the other hand, if the orthogonal matrix is required to be postmultiplied with the plaintext-image, the size of the orthogonal matrix should be $N \times N$.

It may be noted that the proposed encryption approach is not designed for text-based lossless encryption that demands each alphabet to be recovered correctly. The proposed decryption mechanism, however, recovers an image that is perceptually similar to the plaintext-image with a high PSNR. The recovery of exact pixel value is not the focus of this work. There are several multimedia applications in which the exact recovery of each pixel value is not required at the receiver's end. Consider, for example, the case of a police station securely sending an encrypted photo of a criminal to a group of detectives on their respective hand-held devices. In such a scenario, it may be the case that each hand-held device has a different image storing or compression standard. The fundamental requirement in this case is that each detective should receive the correct photo with an image quality that can enable the detective to recognize the criminal. Small changes in pixel values is not important in such a case.

Chapter 4

A Robust and Secure Hash-Based Technique for Image Authentication

4.1 Introduction

In the previous chapter, the confidentiality issue of multimedia data was addressed by devising a compression friendly encryption technique for digital images. This chapter focuses on the integrity protection of the multimedia data by devising a hash-based authentication technique for digital images. With tremendous growth in the field of multimedia technology and the availability of powerful image processing software, it has now become easy to tamper digital images [113]. This possesses a serious issue, especially, if a digital image is to be used as evidence in a court of law. Hence, there should be some mechanism to prove the authenticity of the image in question. Cryptographic digital signatures [4] can be used to authenticate a digital image as proposed in [53]. The main drawback of this technique is that even a single bit change in the image due to some content preserving operation like compression or filtering will completely change the hash value. Multimedia image authentication, therefore, requires techniques which should be some what resilient to content preserving manipulations and at the same time be fragile enough to detect malicious manipulations.

A Robust and Secure Hash-Based Technique for Image Authentication

Several image authentication schemes have been proposed in recent years. A comprehensive review of a number of hashing schemes was presented in Chapter 2. Generally speaking, image authentication schemes can be classified into two types: watermark-based and hash-based. Watermarking techniques embed an imperceptible signal into a cover work to form a watermarked image. At the receiver's end, the extracted watermark from the watermarked image is used for authenticating purpose [81]. In contrast to watermark-based techniques, hash-based (or digital signature-based) techniques use a Perceptual Hash Function (PHF) to extract a set of features from the image to form a compact representation that can be used for authentication [67]. One main disadvantage of a PHF-based scheme for image authentication is that the hash is an extra overhead that needs to be transmitted or stored besides the image. However, hash-based schemes have many useful advantages as compared to watermark-based authentication schemes, for example:

- There is no distortion introduced in the image.
- The size of the authentication watermark is limited by the image embedding capacity. This is not the case with image hash as it is a separate entity.
- It is not necessary that the image hash is transmitted along with the image. It can even be transmitted before or after the image in question is transmitted. Similarly the image and its respective hash can be stored at two different physical locations.

Following are some important requirements that are expected from a PHF used to authenticate an image [61], [10], [63].

- Robustness against non-malicious manipulations, for example, lossy compression, low pass/high pass filtering, minor contrast enhancement, etc.
- The PHF should be highly sensitive to malicious tampering and should be able to detect the location of tampering.

A Robust and Secure Hash-Based Technique for Image Authentication

- The PHF should be key dependent. Without knowledge of the correct key, it should be extremely difficult to generate the correct hash.
- A key-based PHF should satisfy the property of modified weak collision [10]. According to this property, given an image, I and its corresponding robust hash, $H(I)$, it should not be feasible to find another image, P such that $H(I) = H(P)$ when I is significantly different from P .
- The PHF should possess a high level of security against counterfeit attacks or attacks that are meant to fool the authentication system.
- By looking at the hash values, it should be extremely difficult for an attacker to guess the contents of the image or reverse engineer the secret key that was used to calculate the hash.

To address the above mentioned issue related to image authentication, a wavelet-based hashing scheme is proposed in this chapter. The proposed scheme is designed to simultaneously addresses the core issues pertaining to security of an image authentication algorithm, tamper detection and robustness. Instead of using a secret key to randomly select the image features, a randomized pixel modulation method is proposed that makes the entire feature space random. The randomized pixel modulation has been found to be very sensitive to changes in the key and can effectively detect tampering. The image hash is robust to JPEG compression, low pass and high pass filtering; which are content preserving operations. A 4-bit quantization scheme is also proposed that significantly reduces the size of the image hash. Detailed experimental results and security analysis depict that the proposed scheme generates a hash of reasonable size with good security features that can be effectively used to authenticate a digital image.

The remaining chapter is organized as follows. In Section 4.2, the proposed hash generation and image authentication algorithms are presented. Section 4.3 proposes a 4-bit quantization scheme that significantly decreases the size of the

hash. In Section 4.4, several experimental results are presented that show the Receiver Operating Curve, tamper detection and robustness capabilities of the proposed scheme. Section 4.5 presents the security analysis of the proposed scheme. Section 4.6 concludes this chapter.

4.2 Proposed Algorithm for Image Authentication

Our aim in this chapter is to develop an image authentication system using PHF that encompasses the three core components i.e. security, tamper detection and robustness. To meet these requirements, a wavelet-based hashing scheme is proposed in this section. Some initial work on this technique had been presented in [114] and [115]. The wavelet transform is used due to its good time-frequency localization property. It has been experimentally observed that the use of LL, LH and HL wavelet coefficients gives good robustness for non-malicious distortions like JPEG compression with high compression ratios, low pass and high pass filtering. In addition, for some malicious tampering in an image area, there is a significant change in these coefficients. To address the security issue, pixels in the spatial domain are randomly modulated using a secret key before taking the wavelet transform. In this chapter, this technique is called as Randomized Pixel Modulation (RPM). The RPM randomly modulates each pixel of the input image using a secret key to obtain the RPM-transformed image. The RPM-transformed image is a random pattern which is a function of the input image pixel values and the secret key.

Most of the PHF-based image authentication techniques, like the ones proposed in [11], [63] use the secret key in the feature extraction stage to enforce security. In such a strategy, the attacker knows the feature space that is used to generate the hash, but without knowledge of the secret key cannot actually pinpoint the exact subset of the features that are used to generate the hash. Depending upon the type of underlying algorithm used to generate the hash, this strategy has a security loop hole that is discussed in Section 4.5 of this chapter. To prevent

A Robust and Secure Hash-Based Technique for Image Authentication

an attacker from knowing the feature space, the secret key is applied before the feature extraction stage using the RPM technique. No matter what type of features are used, they all depend upon the values of the image pixels. Generally speaking, changing the pixel values in a random fashion can possibly make the feature space random. Therefore, if an attacker replaces an image block with a visually different image block, without knowledge of the secret key, he/she cannot predict how the replaced block would contribute towards forming the feature space. There are many interesting properties of RPM related to system's security and tamper detection capabilities that are discussed in later sections of this chapter. The proposed image authentication system consists of a hash generation module and an image verification module. There are two secret keys, K_1 and K_2 used by the system. These keys are shared by both the modules. To distinguish between the parameters of sender and receiver, a bar over the parameters will be used that relates to the receiver. Details of the proposed technique are presented in the following sections.

4.2.1 Hash Generation Module

The block diagram of hash generation module is shown in Figure 4.1. Following are the various steps involved in the hash generation process:

1. Let the input image be represented by I of dimension $N \times N$ pixels. The image I is partitioned into non-overlapping blocks, each of dimension $J \times J$ pixels. This gives a total of N^2/J^2 blocks. Each block is represented by B_i , where $i = 0, \dots, N^2/J^2 - 1$. In the proposed scheme, each block consists of 16×16 pixels, i.e. $J = 16$ [116]. Choosing block size of 8×8 pixels may increase the accuracy, however it would also increase the hash size. After doing experiments on a number of different images, it has been found that block size of 16×16 pixels gives a good-trade off between the tamper detection capability and size of the block.
2. Let $B_i(x, y)$ represent the gray value of a pixel at spatial location (x, y)

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.1: Hash Generation Module.

in the block B_i . Let K_1 be the secret key that is shared between the sender and the receiver (verifier). Since $J = 16$, each block consists of 256 pixels. Pixels of each block are modulated using a permutation sequence that has 256 elements. The permutation sequences are obtained using the RC4 algorithm [117] that is initialized using the secret key K_1 . The RC4 is a well studied algorithm in cryptography that produces sequences having a repetition cycle of 10^{100} . Let the permutation sequence of each block be represented by $S_i(m)$, where i is the block index and m is the index of a specific element in S_i .

3. To make the image hash key-dependent, pixels of each block B_i are RPM-transformed using the permutation sequence S_i to get a new image called the RPM-transformed image, I_{RPM} . Each pixel, $B_i(x, y)^*$ of I_{RPM} is calculated as follows:

$$B_i(x, y)^* = [B_i(x, y) + \alpha \times S_i(m)] \quad (4.2.1)$$

for $0 \leq x, y \leq J - 1$ and $0 \leq m \leq J^2 - 1$.

The parameter $S_i(m)$ can have any value between 1 and 256. Due to the random property of RC4, even similar looking blocks are mapped to different hash values. This point is illustrated in Section 4.5.

4. The I_{RPM} image is wavelet transformed such that each sub-band coefficient approximately represents a 16×16 area in I_{RPM} . For example, if I has a dimension of 256×256 pixels, then I_{RPM} will also have the same dimension.

A Robust and Secure Hash-Based Technique for Image Authentication

The 4th level wavelet decomposition of I_{RPM} would yield LL , LH , HL and HH sub-bands of dimension 16×16 such that each coefficient in the sub-band exactly represents a 16×16 area in I_{RPM} . We have chosen 16×16 area as it gives good precision for tamper detection with reasonable hash size. Due to time-frequency localization property of the wavelet transform, each sub-band coefficient relates to its corresponding spatial area in I_{RPM} . Thus tampering in a specific image area of I would effect the same area in I_{RPM} according to Eq. 4.2.1 thus only perturbing the corresponding wavelet coefficient in the sub-bands obtained from I_{RPM} . This helps to detect tampering with localization.

5. Let the position of a wavelet coefficient in the sub-bands be indexed by p and q , where $0 \leq p, q \leq Z$. Let each coefficient in the respective sub-band be represented by $LL(p, q)$, $LH(p, q)$, $HL(p, q)$ and $HH(p, q)$. The intermediate hash of I is calculated by the following two equations.

$$H_{\bar{I}_1}(p, q) = LL(p, q) + LH(p, q), \quad (4.2.2)$$

$$H_{\bar{I}_2}(p, q) = LL(p, q) + HL(p, q), \quad (4.2.3)$$

for all $0 \leq p, q \leq Z$.

6. The final hash of I is obtained by permuting the entries of $H_{\bar{I}_1}$ and $H_{\bar{I}_2}$ with the secret key, K_2 . The permuted matrices are represented by H_{I_1} and H_{I_2} , respectively.
7. The sender also determines the threshold, τ . The proper selection of τ is very important as it defines the boundary between non-malicious distortion and malicious tampering. The strategy to select the parameter τ is described in Section 4.4.

4.2.2 Image Verification Module

The image verification module shown in Figure 4.2 is used to authenticate the received image, $\bar{I}$ w.r.t its hash. The actual image I may be transmitted through

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.2: Image Verification Module.

an insecure channel. Therefore $\bar{I}$ may undergo non-malicious operations like JPEG compression, etc or malicious tampering. The hash of I that shall be used to authenticate its received version $\bar{I}$ and other system parameters N, K_1, K_2, α and J are transmitted securely. Following are the steps involved in the image verification process:

1. Using the secret key K_1 , the received image $\bar{I}$ of $N \times N$ pixels is passed through the same steps as mentioned in Section 4.2.1 to get the intensity transformed version, $\bar{I}_{RPM}$.
2. The intermediate hash of $\bar{I}$ is calculated using the formula outlined by Eq. 4.2.2 and Eq. 4.2.3. Let the intermediate hash of $\bar{I}$ be represented by $\bar{H}_{\bar{I}_1}$ and $\bar{H}_{\bar{I}_2}$.
3. Using the secret key K_2 , inverse permutation is applied to the received hash H_{I_1} and H_{I_2} to get the matrices of intermediate hash $H_{\bar{I}_1}$ and $H_{\bar{I}_2}$, respectively.
4. Difference matrices D_1 and D_2 are calculated as follows:

$$D_1(p, q) = |H_{\bar{I}_1}(p, q) - \bar{H}_{\bar{I}_1}(p, q)|, \quad (4.2.4)$$

$$D_2(p, q) = |H_{\bar{I}_2}(p, q) - \overline{H}_{\bar{I}_2}(p, q)|, \quad (4.2.5)$$

for all $0 \leq p, q \leq Z$.

5. An error matrix $E(p, q)$ is calculated that contains the maximum value of D_1 or D_2 .

$$E(p, q) = \max(D_1(p, q), D_2(p, q)), \quad (4.2.6)$$

for all $0 \leq p, q \leq Z$.

6. Each entry of the matrix E is compared with the threshold, τ . If any entry of E is greater than τ , then the corresponding spatial area in $\bar{I}$ shall be considered as tampered and the image $\bar{I}$ will not be positively authenticated.

4.3 Quantization Scheme for Hash Size Reduction

In this Section, a quantization method is proposed to reduce the size of the hash. Interestingly, the proposed method not only reduces the size of the hash, but it also enhances the security of the system. When quantization is used, the hash permutation module shown in Figure 4.1 and its corresponding inverse permutation module shown in Figure 4.2 are not required. Therefore, the secret key K_2 is not required. The block diagrams of hash generation and image verification system with quantization module added are shown in Figure 4.3 and Figure 4.4, respectively.

In the hash generation phase, the inputs to the quantization module are the intermediate hash matrices $H_{\bar{I}_1}$ and $H_{\bar{I}_2}$ of dimension $Z \times Z$. These two matrices are scanned row-wise to form vectors, $V_{H\bar{I}_1}$ and $V_{H\bar{I}_2}$ respectively. The two vectors $V_{H\bar{I}_1}$ and $V_{H\bar{I}_2}$ are then concatenated to form a vector, $V_H = [V_{H\bar{I}_1} \ V_{H\bar{I}_2}]$. Let the intermediate hash coefficients in V_H be represented by ψ_κ , where $0 \leq \kappa \leq 2Z^2 - 1$.

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.3: Hash Generation Module with Quantization.

Figure 4.4: Image Verification Module with Quantization.

A Robust and Secure Hash-Based Technique for Image Authentication

The output of the quantization module consists of two parts. The first part is the 160-bit hash of the input image I obtained using the cryptographic hash function, SHA1 [4]. The second part consists of a perturbation vector, Ω_κ that is generated as part of the quantization process. The dimensions of V_H and Ω_κ are same and each entry in Ω_κ requires 4 bits and corresponds to the entry at the same position in V_H . The information contained in Ω_κ is used to adjust the intermediate hash coefficients, $\overline{\psi_\kappa}$ in the vector $\overline{V_H}$ during the image verification stage before performing quantization. This adjustment ensures that if the drift between ψ_κ and $\overline{\psi_\kappa}$ due to non-malicious operation is less than or equal to the defined threshold τ , then $\overline{\psi_\kappa}$ coefficients would quantize to the same value that was calculated in the hash generation phase. To positively authenticate an image, it is necessary that all the coefficients ψ_κ and $\overline{\psi_\kappa}$ fall in the same quantization interval in the hash generation and image verification phases.

4.3.1 Quantization Procedure

The details of the quantization process are now presented. The sender calculates the quantization intervals for each ψ_κ in the vector V_H . All the quantization intervals are then concatenated and hashed using the cryptographic hash function SHA1 [4] that gives a 160-bit hash. For each ψ_κ , a 4-bit value called perturbation information is calculated and recorded in Ω_κ . As described above, the perturbation information is used at the receiver's end to adjust the value of $\overline{\psi_\kappa}$ coefficients in $\overline{V_H}$ before performing quantization. The quantization procedure is described with the help of Figure 4.5. Let Q_κ represent a quantization interval, where κ represents the index to a specific intermediate hash coefficient in V_H . Let τ be the maximum allowable difference between ψ_κ and $\overline{\psi_\kappa}$ values due to non-malicious operation. Each quantization interval will then be equal to τ . The quantization interval for ψ_κ is calculated by dividing ψ_κ by τ and rounding the result using the floor operator.

$$Q_\kappa = \left\lfloor \frac{\psi_\kappa}{\tau} \right\rfloor. \quad (4.3.7)$$

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.5: Illustration of quantization procedure.

Let $H_{SHA1}(\cdot)$ represent the cryptographic hash function using the SHA1 algorithm [4] and $\parallel$ represents the concatenation operator. The sender calculates the hash, H_I of the image I using the following equation.

$$H_I = H_{SHA1}(Q_0 \parallel Q_1 \parallel Q_2 \parallel \dots \parallel Q_\kappa). \quad (4.3.8)$$

Let the 4-bit perturbation information for a quantization interval Q_κ be represented by two 2-bit variables $\rho_{1\kappa}$ and $\rho_{2\kappa}$. Due to non-malicious operation, it is possible that a single or multiple hash coefficients may drift from their original values such that one or more quantization intervals are different from their original counterparts. According to Eq. 4.3.8, to positively authenticate an image, it is necessary that $Q_\kappa = \overline{Q_\kappa} \forall \kappa$. Therefore, before the receiver calculates $\overline{Q_\kappa}$, $\forall \kappa$ $\overline{\psi_\kappa}$ is adjusted such that $\overline{Q_\kappa} = Q_\kappa$ if $|\psi_\kappa - \overline{\psi_\kappa}| \leq \tau$.

For a specific $\overline{\psi_\kappa}$, its corresponding recorded values $\rho_{1\kappa}$ and $\rho_{2\kappa}$ are used by the receiver to adjust $\overline{\psi_\kappa}$ before calculating $\overline{Q_\kappa}$. The parameters $\rho_{1\kappa}$ and $\rho_{2\kappa}$ enable the receiver to decide whether shifting in $\overline{\psi_\kappa}$ is required or not. Further, if shifting is required, whether the shifting should be in the positive or the negative direction.

For a quantization interval Q_κ , the sender calculates $\rho_{1\kappa}$ as follows:

$$\rho_{1\kappa} = Q_\kappa \bmod 4 \quad (4.3.9)$$

As shown in Figure 4.5, each quantization interval is divided into four equal parts. To show how the two bits of $\rho_{2\kappa}$ are recorded, consider the interval between

A Robust and Secure Hash-Based Technique for Image Authentication

τ and 2τ , i.e. $Q_\kappa = 1$. The two bits of the parameter $\rho_{2\kappa}$ are recorded as follows:

$$\rho_{2\kappa} = \begin{cases} 0 & : \quad \tau \leq \psi_\kappa < 1.25\tau \\ 1 & : \quad 1.25\tau \leq \psi_\kappa < 1.50\tau \\ 2 & : \quad 1.50\tau \leq \psi_\kappa < 1.75\tau \\ 3 & : \quad 1.75\tau \leq \psi_\kappa < 2.0\tau \end{cases} \quad (4.3.10)$$

The receiver will adjust $\overline{\psi_\kappa}$ coefficients to cater for non-malicious operation to get the adjusted coefficient, $\overline{\psi_\kappa^*}$. Let $\&$ and $|$ represent logical AND and OR operators, respectively. Let σ_κ be a parameter that enables the receiver to know if $\overline{\psi_\kappa}$ has drifted one quantization interval in either direction. In case if any $\overline{\psi_\kappa}$ coefficient is drifted more than one quantization interval, then it will be considered as a drift due to malicious tampering, hence no adjustment in $\overline{\psi_\kappa}$ will be done. This implies $Q_\kappa \neq \overline{Q_\kappa}$. This is the reason a single quantization interval is equal to τ . The receiver calculates σ_κ as follows:

$$\sigma_\kappa = \begin{cases} 0 & : \quad \overline{\rho_{1\kappa}} = (\rho_{1\kappa} - 1) \bmod 4 \\ 1 & : \quad \overline{\rho_{1\kappa}} = (\rho_{1\kappa} + 1) \bmod 4 \\ 2 & : \quad otherwise \end{cases} \quad (4.3.11)$$

Before calculating $\overline{Q_\kappa}$, the receiver will adjust $\overline{\psi_\kappa}$ according to the following rule:

$$\overline{\psi_\kappa^*} = \begin{cases} \overline{\psi_\kappa} + \tau & : \quad (\sigma_\kappa = 0) \ \& \ (\rho_{2\kappa} = 0) \\ \overline{\psi_\kappa} + \tau & : \quad (\sigma_\kappa = 0) \ \& \ (\rho_{2\kappa} = 1) \ \& \ (\overline{\rho_{2\kappa}} = 1 \mid \overline{\rho_{1\kappa}} = 2 \mid \overline{\rho_{1\kappa}} = 3) \\ \overline{\psi_\kappa} + \tau & : \quad (\sigma_\kappa = 0) \ \& \ (\rho_{2\kappa} = 2) \ \& \ (\overline{\rho_{2\kappa}} = 2 \mid \overline{\rho_{1\kappa}} = 3) \\ \overline{\psi_\kappa} + \tau & : \quad (\sigma_\kappa = 0) \ \& \ (\rho_{2\kappa} = 3) \ \& \ (\overline{\rho_{2\kappa}} = 3) \\ \overline{\psi_\kappa} - \tau & : \quad (\sigma_\kappa = 1) \ \& \ (\rho_{2\kappa} = 0) \ \& \ (\overline{\rho_{2\kappa}} = 0) \\ \overline{\psi_\kappa} - \tau & : \quad (\sigma_\kappa = 1) \ \& \ (\rho_{2\kappa} = 1) \ \& \ (\overline{\rho_{2\kappa}} = 0 \mid \overline{\rho_{2\kappa}} = 1) \\ \overline{\psi_\kappa} - \tau & : \quad (\sigma_\kappa = 1) \ \& \ (\rho_{2\kappa} = 2) \ \& \ (\overline{\rho_{2\kappa}} = 0 \mid \overline{\rho_{2\kappa}} = 1 \mid \overline{\rho_{2\kappa}} = 2) \\ \overline{\psi_\kappa} - \tau & : \quad (\sigma_\kappa = 1) \ \& \ (\rho_{2\kappa} = 3) \\ \overline{\psi_\kappa} & : \quad otherwise. \end{cases} \quad (4.3.12)$$

After obtaining $\overline{\psi_\kappa^*}$ coefficients, the receiver calculates the quantization intervals and the image hash.

$$\overline{Q}_\kappa = \left\lfloor \frac{\overline{\psi}_\kappa^*}{\tau} \right\rfloor. \quad (4.3.13)$$

$$\overline{H}_I = H_{SHA1} (\overline{Q}_0 || \overline{Q}_1 || \overline{Q}_2 || \dots || \overline{Q}_\kappa) \quad (4.3.14)$$

The received image $\overline{I}$ will be positively authenticated if:

$$H_I = \overline{H}_I. \quad (4.3.15)$$

Since the final hash is calculated using the cryptographic hash function, therefore, if tampering in any 16×16 image area is such that its corresponding $|\psi_\kappa - \overline{\psi}_\kappa| > \tau$ then $Q_\kappa \neq \overline{Q}_\kappa$ which implies that H_I and $\overline{H}_I$ will be completely different.

4.3.2 Hash Size Reduction Due to Quantization

It is to be noted that without quantization, the hash of the image consists of coefficients in the matrices H_{I_1} and H_{I_2} . As discussed later in Section 4.4.4, after performing experiments on a number of different images, it has been observed that each hash coefficient takes 14 bits of storage. For an image of size 256×256 pixels, the total entries in H_{I_1} and H_{I_2} will be $2 \times 16 \times 16$. With each entry taking 14 bits, the size of the hash will be 896 bytes. With quantization, the information contained in Ω_κ requires $2 \times 16^2 \times 4$ bits of storage, plus additional 160 bits for the hash. Therefore the receiver requires 276 bytes of information to perform the authentication. By using the quantization process, the overhead is 3.24 times less thus saving the bandwidth or storage space. As stated above, an additional benefit of quantization is that it also enhances the security of the system. This point is discussed in Section 4.5.

4.3.3 Non-Malicious and Malicious Operation

In this Section, the quantization procedure is further elaborated. Let the system threshold $\tau=10$.

Non-Malicious Operation

Consider the case of a non-malicious operation. Assume the value of a hash coefficient $\psi_\kappa = 46$, falling in $Q_\kappa = 4$. Then $\rho_{1\kappa} = 4 \bmod 4 = 0$ and $\rho_{2\kappa} = 2$. Now suppose at the receiver, due to a non-malicious operation, the same coefficient, is drifted from 46 to 38, thus falling in $\overline{Q}_\kappa = 3$, instead of $\overline{Q}_\kappa = 4$. If no adjustment is made, then $H_I \neq \overline{H_I}$, hence $\overline{I}$ will not be positively authenticated. To make adjustment in $\overline{\psi}_\kappa$, the receiver calculates $\overline{\rho}_{1\kappa}$, $\overline{\rho}_{2\kappa}$ and σ_κ using Eq. 4.3.9, Eq. 4.3.10 and Eq. 4.3.11, respectively yielding $\overline{\rho}_{1\kappa} = 3$, $\overline{\rho}_{2\kappa} = 3$ and $\sigma_\kappa = 0$. Using Eq. 4.3.12, the receiver will adjust $\overline{\psi}_\kappa$ such that $\overline{\psi}_\kappa^* = \overline{\psi}_\kappa + 10 = 48$. Using Eq. 4.3.13, $\overline{Q}_\kappa = 4 = Q_\kappa$. It is easy to check that for any κ , as long as $|\psi_\kappa - \overline{\psi}_\kappa| \leq \tau$, $\overline{Q}_\kappa = Q_\kappa$ thus $H_I = \overline{H_I}$.

Malicious Tampering and its Detection Condition

Let us consider the case of malicious tampering, i.e. $|\psi_\kappa - \overline{\psi}_\kappa| > \tau$, where $\tau = 10$. Assume the same condition at the sender's side, i.e. $\psi_\kappa = 46$, which implies $\rho_{1\kappa} = 0$ and $\rho_{2\kappa} = 2$. At the receiver's side, let $\overline{\psi}_\kappa = 35$, i.e. $|\psi_\kappa - \overline{\psi}_\kappa| > 10$. In this case, $\overline{\rho}_{1\kappa} = 3$, $\overline{\rho}_{2\kappa} = 2$ and $\sigma_\kappa = 0$, therefore, $\overline{\psi}_\kappa^* = \overline{\psi}_\kappa + 10 = 45$. This implies that $\overline{Q}_\kappa = Q_\kappa$, despite the fact that $|\psi_\kappa - \overline{\psi}_\kappa| > \tau$. This is a contradiction from the fact that $\overline{Q}_\kappa \neq Q_\kappa$ if $|\psi_\kappa - \overline{\psi}_\kappa| > \tau$. The reason for this discrepancy is the resolution of the quantization interval. Since each interval is divided into 4 equal parts, therefore $\overline{\rho}_{2\kappa}$ will not change if the drift in $\overline{\psi}_\kappa < 0.25\tau$. However, it is guaranteed that $\overline{Q}_\kappa \neq Q_\kappa$ if $|\psi_\kappa - \overline{\psi}_\kappa| \geq 1.25\tau$.

Let us take another example in which for the same drift of 11 in $\overline{\psi}_\kappa$, tampering will be detected. At the sender's side, let $\psi_\kappa = 40$, this implies $Q_\kappa = 4$, $\rho_{1\kappa} = 0$ and $\rho_{2\kappa} = 0$. At the receiver's side, let $\overline{\psi}_\kappa = 29$, which implies that $\overline{\rho}_{1\kappa} = 2 \neq$

A Robust and Secure Hash-Based Technique for Image Authentication

$(\rho_{1\kappa} - 1) \bmod 4$, $\overline{\rho_{2\kappa}} = 3$ and $\sigma_\kappa = 2$. There will be no change in $\overline{\psi_\kappa}$, hence, $\overline{\psi_\kappa^*} = \overline{\psi_\kappa}$. Therefore, $\overline{Q_\kappa} = 2 \neq Q_\kappa$. Therefore, $H_I \neq \overline{H_I}$.

To generalize this discussion; using quantization, tampering will be detected if $|\psi_\kappa - \overline{\psi_\kappa}| > \tau_\kappa$, where $\tau \leq \tau_\kappa \leq 1.25\tau$. Let x be the distance between any two divisions of a quantization interval, where $0 \leq x \leq 0.25\tau$.

$$x_\kappa = \left\lceil \left(\frac{\psi_\kappa}{\tau} \right) \bmod 0.25 \right\rceil \tau. \quad (4.3.16)$$

Depending upon the position of ψ_κ in a quantization interval, the value of τ_κ at which tampering will be detected is given by the following equation:

$$\tau_\kappa = \tau + x_\kappa. \quad (4.3.17)$$

Hence, in case of quantization, for any κ if $|\psi_\kappa - \overline{\psi_\kappa}| \geq 1.25\tau$, $H_I \neq \overline{H_I}$.

4.3.4 Condition for Detection of Tampered Blocks

If tampering is detected in $\overline{I}$, then $H_I \neq \overline{H_I}$ which indicates that $\overline{I}$ is not authentic. The quantization algorithm, however, cannot detect the location of tampering in $\overline{I}$. Interestingly, the perturbation information stored in Ω_κ can be used by the receiver to detect the location of tampering provided $\overline{\rho_{1\kappa}^*} \neq \rho_{1\kappa}$, where $\overline{\rho_{1\kappa}^*} = \overline{Q_\kappa} \bmod 4$. Once the index κ is determined, the corresponding location in $\overline{I}$ can be identified.

4.4 Experimental Results and Discussion

An image hashing scheme should be robust against non-malicious manipulations like JPEG compression, low pass/high pass filtering, contrast enhancement, etc and simultaneously be sensitive enough to detect malicious manipulations. In addition, the entire hashing scheme should fulfill a number of security requirements. The degree of robustness is a trade-off with respect to the amount of fragility required to detect malicious manipulations and the level of security that

is desired. In this section, several experimental results are given to demonstrate the robustness and tamper detection capabilities of the proposed scheme. The parameter α in Eq. 4.2.1 is used for enforcing security by generating a hash that depends not only on the image pixels but also on the secret key. Increasing the value of α though increases the randomness in the hash, also increases the size of the wavelet coefficients that form the hash. For example, in case of the Cameraman image, if α is kept at 0.5, then $\max(H_{\bar{I}_1}(p, q), H_{\bar{I}_2}(p, q)) = 4039$ and around 74 percent blocks of the Cameraman image are detected as tampered for incorrect keys. However, if α is kept at 1.5, then $\max(H_{\bar{I}_1}(p, q), H_{\bar{I}_2}(p, q)) = 6278$ and around 97 percent blocks of the Cameraman image are detected as tampered for incorrect keys. After doing experiments on a number of images, the value of $\alpha = 1.5$ has been found to give good randomness in the hash along with a reasonable hash size.

4.4.1 System's Robustness and Discussion on Threshold Selection

The threshold τ draws the boundary between robustness, tamper detection capability and security of the authentication system. In the proposed scheme, increasing τ will increase the system's robustness while decreasing tamper detection capability and security. The proposed scheme offers good robustness against JPEG compression, low pass and high pass filtering. It is important to note that robustness characteristics vary from image to image. To explore this fact, a database of 1000 different images was formed with each image having a size of 256×256 pixels. Figure 4.6a shows the response of the authentication system due to JPEG compression. The JPEG quality factor, QF was kept at 20. The parameter $\max(E(p, q))$ was calculated using Eq. 4.2.6. The value of $\max(E(p, q))$ varies between 33 and 78. Similarly, Figure 4.6b and Figure 4.6c show the response of the system when the receiver gets low pass and high pass filtered versions of the transmitted image, respectively. The average values of $\max(E(p, q))$ for JPEG compression, low pass and high pass filtering are 42, 28 and 23, respectively. In view of these results, it is proposed that defining a global

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.6: System robustness to JPEG compression, low pass and high pass filtering. (a) Effect of JPEG compression. (b) Effect of Low pass filtering. (c) Effect of High pass filtering.

A Robust and Secure Hash-Based Technique for Image Authentication

threshold for all images will not prove efficient. For example, in case of Image No. 200, the $\max(E(p, q))$ value due to JPEG compression is 39. Using this set of data, if a global threshold strategy is adopted that caters for JPEG compression, then for any image, tampering will only be detected if $\max(E(p, q)) > 78$ (the response of Image No. 645 for JPEG compression). In such a case, for Image No. 200, tampering that results in $\max(E(p, q)) < 79$ will not be detected despite the fact that $\max(E(p, q))$ for this image is 39. Keeping in view this fact, it is proposed that when the sender is sending an image I to a receiver, the sender should also estimate the desired threshold that the receiver shall be bound to use. This implies that the sender is also putting a constraint on the amount of robustness that can be tolerated. The threshold parameter can be included in the digital signature so that a third party can also correctly authenticate $\bar{I}$.

Bounding the receiver by the sender to use a specific threshold makes sense because the sender has the actual image that needs to be authenticated at the receiver's end. For a specific image, the sender can have some estimate about the authentication system response to different thresholds. The sender can then select a specific threshold that gives a reasonable trade-off between robustness, tamper detection and security.

Consider the Baboon image shown in Figure 4.7a. In Figure 4.7b, a slightly tampered version of the Baboon image is shown in which the right eye ball has been replaced with the left eye ball. The value of $\max(E(p, q))$ for this tampering is 60. To detect such a minute tampering, $\tau < 60$. Figure 4.8 shows the $\max(E(p, q))$ for the Baboon image by varying JPEG QF from 100 to 1. If such minute tampering is to be detected, then the Baboon image should not be compressed below JPEG $QF = 12$ (at JPEG $QF = 12$, $\max(E(p, q)) = 56$, whereas at JPEG $QF = 11$, $\max(E(p, q)) = 73$) while being transmitted from the sender to the receiver. This discussion emphasizes the point that in robust image hashing, the selection of threshold is quite a complex issue, as the sender does not know in advance what may be the magnitude of malicious and non-malicious distortion in the image while it is in transit. However, with the empirical analysis

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.7: Minute tampering that may go undetected at small threshold. (a) Original Baboon image. (b) Tampered Baboon Image. Tampering is shown inside the circle.

Figure 4.8: Change in $\max(E(p, q))$ as the receiver receives compressed version of the Baboon image with varying JPEG QF.

shown above, one has some idea about robustness versus tamper detection capability of the system. As a future research direction, algorithms based on artificial intelligence or other techniques can be devised that can ask the user to specify important image areas that need more protection and the type of non-malicious operation that the image may undergo during transmission. Based on this data, the algorithm can give some optimum threshold that the sender can bound the receiver to use while authenticating the received image.

4.4.2 Detection of Tampering

In this section, some further experimental results are presented to show the proposed system sensitivity towards detecting tampering. For the purpose of illustration the Cameraman image was used for which the value of $\max(E(p, q))$ for JPEG QF between 100 and 20, low pass and high pass filtering is 48. Suppose robustness requirement is to bypass any distortion in the received image if it has undergone JPEG compression up to QF of 20. Let us choose the system's threshold as 50 and analyze the sensitivity of the system to detect malicious tampering. Tampering can be either low contrast or high contrast. High contrast tampering is easy to detect as compared to low contrast tampering. Figure 4.9a shows the original Cameraman image while Figure 4.9b shows its tampered version with tampered areas marked inside a circle. The first tampering is with the lens which is a high contrast tampering. In the second tampering, the roof of the small building is tampered. This tampering is low contrast as compared to the first one. Figure 4.9c shows the detection result in which both the tampered areas were successfully detected. The interesting thing to note is that for the lens tampering, $E(p, q) = 330$, while for the building tampering, $E(p, q) = 98$. The former is a high contrast tampering as compared to the latter. Both these value are quite high as compared to the system's threshold. This result reveals that the proposed system can detect minute tampering at a high compression ratio.

A Robust and Secure Hash-Based Technique for Image Authentication

(a)

(b)

(c)

Figure 4.9: Tamper detection capability of the proposed scheme. (a) Original Cameraman image. (b) Tampered Cameraman image. Tampering is shown inside the circle. (c) Result of tamper detection.

4.4.3 The Receiver Operating Curve

The receiver operating curve (ROC) is a plot of the probability of false positive, P_{FP} verses the probability of false negative P_{FN} as the system threshold is varied. In context to the authentication system, the two probabilities are defined as follows:

$$P_{FP} = \frac{\text{Number of Tampered Blocks Detected as Genuine}}{\text{Total Number of Tampered Blocks}}. \quad (4.4.18)$$

$$P_{FN} = \frac{\text{Number of Genuine Blocks Detected as Tampered}}{\text{Total Number of Genuine Blocks}}. \quad (4.4.19)$$

To estimate P_{FP} , an experiment was conducted using the Cameraman image shown in Figure 4.9a and the Baboon image shown in Figure 4.7a. The Cameraman image is considered as the genuine image whose hash is compared with the hash of the Baboon image. The hash comparison is done by comparing each block of the Cameraman image with all blocks of the Baboon image and identifying those blocks for which $E(p, q) < \tau$. For each τ , the total number of comparisons are 65536. A tampered block shall be considered genuine if the respective $E(p, q)$ for the block is less than τ . The P_{FP} is then calculated using Eq. 4.4.18. By looking at the Baboon image, it is easy to judge that nearly all blocks of the Baboon image are visually different from the blocks of the Cameraman image. Due to this reason, the Baboon image was considered as a tamper test case w.r.t. the Cameraman image. The false negative probability P_{FN} is estimated by choosing JPEG compression as the non-malicious operation. The Circusman image is used that gives the worst robustness response to JPEG compression. The genuine hash is first calculated using the Circusman's image, which is then compared with the hash of the same image JPEG compressed at $QF = 20$. Each block of the Circusman's image is compared with the corresponding block of its compressed version. The comparison though genuine shall be considered as tampered if for any block $E(p, q) > \tau$. The probability P_{FN} is calculated using Eq. 4.4.19 using the same values of thresholds that were used to calculate P_{FP} .

Figure 4.10: The Receiver Operating Curve.

While estimating the ROC curve, the parameter τ was varied from 0 to 150. The result of this experiment is presented in Figure 4.10. Even at low values of P_{FN} , the proposed system gives a reasonably low P_{FP} . This is basically a trade-off between robustness, tamper detection capability and security of the authentication system. For example, if P_{FP} is kept at 0.01, then the corresponding P_{FN} is 0.005. These results are quite encouraging when compared to the results in the literature, for example, [67].

4.4.4 Hash Size

The size of the hash depends upon the size of the image. Figure 4.11 shows the maximum value of the wavelet coefficient for each image in our database of 1000 images. The block size used was 16×16 pixels. The maximum value of the wavelet coefficient is 7549 that requires 13 bits of storage. Let us allocate 14 bits of storage for each wavelet coefficient so that its maximum value can span up to 16384. In this case, the size of hash for a 256×256 image will be 896 bytes.

Figure 4.11: The maximum value of wavelet coefficient for each image in the database.

4.5 Security Analysis

In this section the security of the proposed scheme is analyzed. To enforce security, two secret keys K_1 and K_2 are used in the hash generation and verification stages. The purpose of K_1 is to change the gray levels of all the pixels within a specific image block by the random pixel modulation given by Eq. 4.2.1. For each image block, this transformation is random and unique. The basic idea behind the proposed RPM technique is to make the intermediate hash dependent on the input image and the secret key K_1 . Due to the small size of an image block, a brute force attack can be launched to estimate K_1 . To thwart this possibility, the second secret key K_2 is used to randomly permute the intermediate hash coefficients. By looking at the permuted hash coefficients, an attacker will not be able to relate a hash coefficient to its respective image block. This shall prevent the attacker from launching a brute force attack to estimate K_1 . In case if quantization is used, the secret key K_2 is not required. This point is discussed in Section 4.5.4. In the following sub-sections, the impact of the proposed randomized pixel modulation on the security of the proposed scheme is discussed. Further, the effect of the secret key K_1 on the security of the system and the probability of hash collision is also discussed.

4.5.1 Randomized Pixel Modulation and its Impact on System's Security

Most of the image hashing techniques proposed in the literature, for example [11], [63] use the secret key to select the subset of the feature space to generate the image hash. This strategy enables the attacker to know the feature space, however, the attacker is not aware which features are selected from the feature space. Depending on the type of underlying hashing algorithm, this strategy can have security loop holes. For example, the DCT scheme proposed by Sun and Chang [63] forms the hash features by dividing an image into 8×8 non-overlapping blocks. From each block, the DCT DC coefficient and 3 DCT AC coefficients are selected. The selection of AC coefficients is done using a secret key. To illustrate the security weakness in this scheme, consider the 8×8 block marked in the background of the Cameraman image shown in Figure 4.12a. Figure 4.12b shows the DCT coefficients of this block. Since the block belongs to the image background with smooth texture, hence most of the DCT AC coefficients are zero. In such a scenario, the impact of the secret key that is used to randomly select the DCT AC coefficient is significantly reduced. The same is true for the scheme proposed by Lin and Chang [11].

Generally speaking, any hashing scheme that has weak secrecy in the feature selection stage can have such security issues. In this chapter, a different approach is adopted. Instead of using the secret key to randomly select features, the secret key is used to transform the feature space using the RPM transformation given by Eq. 4.2.1. Through this transformation, each image pixel is changed into a new value which depends upon the original pixel value and the secret key K_1 . Figure 4.13a shows how an RPM-transformed Cameraman image looks like. It is to be noted that the RPM-transformed image is used at run-time by the algorithm to calculate the hash and is not transmitted to the receiver. Figure 4.13b shows the RPM-transformed DCT coefficients of the same block that was shown in Figure 4.12a. In Figure 4.12b most of the DCT coefficients were zero because of the smooth texture of the block. However, due to randomized pixel modulation,

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.12: For smooth texture blocks, most of the DCT AC coefficients are zero. (a) An 8×8 block marked on the Cameraman image. (b) DCT coefficient of the 8×8 shown in (a).

Figure 4.13: The proposed RPM technique increases the entropy of the feature space. (a) RPM-transformed Cameraman image. (b) DCT coefficient after the 8×8 block shown in Fig. 4.12a is RPM-transformed.

random key-dependent DCT coefficients are obtained. Hence, even for image areas that have very low entropy, the proposed RPM technique increases the entropy of the feature space as in this case by generating DCT coefficients that are random. Increasing the entropy of the feature space increases the security of the hash that is generated using the feature space. The same applies for the LL, LH and HL wavelet coefficients that are used in this chapter to generate hash of an image. In the next section, the sensitivity of the secret key on the randomness of wavelet coefficients will be shown.

4.5.2 Effect of Secret Key on the Hash

In this section, the effect of secret key K_1 on the intermediate hash values is shown. The change in the secret key should significantly change the hash. To verify this fact, hash of the Cameraman image was generated and then compared with 1000 hashes of the same image generated with 1000 randomly generated keys. The system threshold was kept at 48, which means that the receiver will tolerate the incoming image even if it is JPEG compressed with a QF of 20. Figure 4.14a shows the result of this experiment. On an average, 96 percent of image blocks for a different key are detected as malicious. Figure 4.14b shows the $\max(E(p, q))$ for each random key. The average value of $\max(E(p, q))$ taken over all the 1000 random keys is 240 which is five times higher than the threshold value of 48. Hence these results show that the proposed RPM-technique is very sensitive to the change in the secret key.

4.5.3 Probability of Hash Collision

Probability of hash collision means that if an attacker replaces any block of a genuine image with any arbitrary block, what is the probability that the arbitrary block will get positively authenticated. To carry out this test, the Cameraman image was used and its hash was compared with hashes of 1000 different images in our database. For each comparison, hash coefficient of every single block of the Cameraman image was compared with hash coefficients of all blocks of the target image. A collision occurs if for any block of the Cameraman and the target image, $E(p, q) < \tau$. Since all the images used are of size 256×256 pixels with block size of 16×16 pixels, therefore, for each target image, total block comparisons are 65536. The probability of collision for a single block is calculated by dividing the total number of collisions with the total comparisons. The threshold used was 48, i.e. the system can withstand high compression ratios. Figure 4.15 shows the probability of collision if any block of the Cameraman image is replaced by any block of the target image. The average probability of collision for a single block is 0.0052. The probability of collision per block is not low as we find in

A Robust and Secure Hash-Based Technique for Image Authentication

Figure 4.14: Effect of the change of secret key K_1 on the intermediate hash. (a) Number of blocks identified as malicious from a total of 256 blocks. (b) Magnitude of error between the hashes.

cryptographic hash functions.

It should be noted that an image hash function is required to bypass non-malicious operations like compression etc, which is not the case with cryptographic hash functions. Interestingly, the chance of success of an attacker to randomly replace all blocks of an image with visually different blocks such that all the replaced blocks get positively authenticated is extremely low. If average probability of collision for a single block is taken as 0.0052, then the probability that all the replaced blocks get positively authenticated is 0.0052^N , where N in this case is 256. This result explains the importance of the secret key in an image authentication system. If a secret key is not used, the attacker knows the hash coefficient. Therefore, it is easy for an attacker to find collision in the hash values. The attacker will be successful even if the hash is sent encrypted because the hashing algorithm is public. However, with the proposed RPM-transformation, by simply looking at the image, an attacker cannot estimate the wavelet coefficients without knowledge of the secret key. Furthermore, in this scheme, even if the final permuted hash is exposed, it is not feasible for an attacker to estimate the secret keys K_1 and K_2 . This is because a permuted hash value does not give any clue to its respective image block. Hence, K_2 cannot be estimated. This implies that the secret key K_1 cannot be estimated. If more robustness is

Figure 4.15: Probability of collision of hash values.

required, the system threshold has to be increased. This, however, shall increase the probability of collision. Depending upon the type of application, a trade-off is required between robustness and security of the authentication system.

4.5.4 Quantization and System's Security

When quantization is used, the intermediate hash coefficients are quantized and the quantization intervals are hashed using the cryptographic hash function. Without knowledge of K_1 , an attacker cannot guess the quantization intervals. To launch a brute force attack, the attacker has to search the key space such that all the quantization intervals are correctly found. This is the advantage of using the cryptographic hash function. For example if the attacker is close to finding the key and the quantization interval differs say by even a unit value, still the hashed output will be entirely different. Hence the attacker has no idea whether he/she is even close to the actual key or not. Further, the perturbation vector does not leak any information regarding the quantization interval as observed from Eq. 4.3.9 and Eq. 4.3.10.

4.6 Discussion and Conclusion

In this chapter, a wavelet-based image hashing scheme is presented that simultaneously tackles robustness, security and tamper detection with localization.

A Robust and Secure Hash-Based Technique for Image Authentication

Unlike other schemes that apply randomness to select the features, a randomized pixel modulation (RPM) method has been proposed that makes the entire feature space random. As mentioned in Section 4.5.1, applying randomness to select feature does not reveal which features have been selected, however, the feature space used to generate the hash remains open to an attacker. Depending upon the structure of the underlying hashing scheme, this can create a security loophole, especially if entropy in the feature space is not high. In Section 4.5.1, this issue is highlighted for the hashing schemes proposed by Lin and Chang [11] and Sun and Chang [63]. Similar is the case with other image hashing schemes, for example, [65], [67] that rely on random partitioning to select the features. In contrast, the proposed scheme uses RPM to first make the entire feature space random and then extract some useful features to form the image hash. The proposed RPM method is very sensitive to change in the key and can effectively detect tampering. Hence looking at an image, an attacker cannot guess in advance what feature a specific area of an image would generate. This helps to make it extremely difficult for an attacker to generate counterfeit images that would get positively authenticated.

When image hash is used for authentication, secret key should be used to enforce security. Without a secret key, anyone with the knowledge of the hashing algorithm can generate hash of an image. Once the hash and its corresponding image is known, counterfeit attacks or attacks to fool an authentication system can easily be launched. For example, the wavelet-based hashing scheme proposed in [64] does not use a secret key and creates image signature using the parent - child pairs located at the multiple scales in the wavelet domain. If the image to be authenticated is stored or transmitted without encryption, then anyone with the knowledge of the algorithm can extract the hash. The proposed scheme in this chapter uses two secret keys K_1 and K_2 to randomize the feature space and the sequence of hash values. This gives a number of security benefits as mentioned in Section 4.5. In addition, a quantization scheme is also proposed that significantly reduces the size of the hash. The information used by the quantization algorithm at the receiver's end does not leak any useful information

A Robust and Secure Hash-Based Technique for Image Authentication

that may help to estimate the hash or the secret key. The system is very sensitive to the change in the secret key. If the final permuted hash is exposed, it will not leak any useful information that can be used to estimate the secret keys K_1 and K_2 . In addition, the permuted hash values do not reveal their respective image blocks.

The proposed scheme is robust to JPEG compression, low pass and high pass filtering that are content preserving operations. Since the image hash is formed by taking into account the LL sub-band wavelet coefficients, the proposed system is not robust to other parameters like rotation, scaling, change in brightness, contrast enhancement, etc. This is a performance trade-off as practically it is very difficult to come up with a PHF that is robust to all non-malicious manipulations and still has a high level of security and tamper detection capability. Discarding the LL sub-band coefficients in the hash, for example, will help to improve the system's robustness to contrast enhancement, however, the system would then fail to detect tampering that involves a change in gray scales. This is the reason that LL coefficients were used. Combinations of LL, LH and HL subbands provides useful information about a specific image block thus enabling a better tamper detection capability. The proposed scheme takes the entire image into account for calculating the hash, unlike the scheme in [67] that works with image areas that have the feature points used to generate the hash. Besides taking features from the entire image, it is also important that the features should have good performance-security trade off. For example, Xie *et al* [55] uses the block mean values as the feature. Although this feature has good robustness property, however, it is very easy to find image blocks that have approximately the same mean value but are perceptually different.

Chapter 5

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

5.1 Introduction

With rapid growth of the Internet, security of digital images is becoming a great concern. It has now become very easy to illegally copy, modify and retransmit a digital image. Digital watermarking is a technique that provides a way to protect digital images from illicit copying and manipulation. A digital watermark is an imperceptible signal added to digital data, called cover work, which can be detected later for buyer/seller identification, ownership proof, etc, [8]. A digital watermarking scheme can either be *symmetric* or *asymmetric*. A symmetric watermarking scheme uses identical keys for watermark embedding and detection [75]. This possesses a security weakness as the information used to detect a watermark can be used to remove it. This restricts the use of symmetric watermarking, as the number of authorized detectors have to be strictly controlled. To solve this problem, asymmetric watermarking schemes have been proposed that use different keys for watermark embedding and detection [76] - [79]. This makes the use of watermarking possible for public domain applications where anyone with the detection key can check the embedded watermark. However, the prac-

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

tical use of asymmetric watermarking requires careful considerations [80]. It is worth noting that merely using a watermarking algorithm does not completely address the issues of copyright protection. To devise a secure watermarking scheme, it is necessary that a watermarking algorithm is well integrated with a secure protocol [91], [92], [94], [118]. For example, in [91], an interactive buyer-seller protocol is proposed that prevents a seller from knowing the exact watermarked copy he/she creates for a buyer. Therefore, the seller cannot create copies of the original content that contains the buyer's watermark. The protocol further allows the seller to identify a buyer from whom an unauthorized copy has originated and prove this fact to a third party.

The primary aim of this chapter is to devise cryptographic protocols and integrate them with some of the existing watermarking techniques in order to address the issues related to buyer/seller identification and copyright protection. To further elaborate our motivation, few scenarios are presented. Suppose Alice sells a watermarked image to Bob. Later in time, Bob starts selling Alice's watermarked image using his fake watermarks. How will Alice prevent Bob from doing this? If the watermarked image consists of both Alice's and Bob's watermarks, how will the actual owner (Alice) be identified? If Bob somehow removes Alice's watermark from the image in dispute, is there any way for Alice to claim her genuine ownership? Consider another scenario. Alice wants to sell a watermarked image I_w to Bob such that the extraction of the watermark from I_w is a legal proof that Bob has indeed purchased I_w from Alice. How will such a watermark be designed whose extraction reveals identities of the buyer/seller? In this chapter, it will be shown that combination of cryptography with watermarking not only increases the security of the overall scheme but it also enables to associate identities of the buyer/seller with their respective watermarked images. Specifically, three issues of copyright protection will be focused, that is, buyer-seller identification, copyright infringement and verification of ownership. By buyer-seller identification, it is meant that a successful watermark extraction at the buyer's end will reveal identities of the buyer and seller of a watermarked image. In case of copyright infringement from a buyer, the proposed scheme enables the seller

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

to identify the specific buyer from whom an illegal copy of a watermarked image has been originated, and further prove this fact to a third party. By ownership verification, it is meant that the seller of a watermarked image should be able to prove his/her legal ownership in case of multiple ownership claims. The rest of the chapter is organized as follows. Section 5.2 presents an overview of some of the terminologies used in this chapter and describes certain assumptions. In Section 5.3 and Section 5.4, the watermark embedding and extraction processes are described. In Section 5.5 and Section 5.6, details of the copyright protection protocols are presented. Section 5.9 concludes the chapter.

5.2 Preliminaries

Before describing watermarking scheme and related protocols, an overview of some of the terminologies and certain assumptions made in the chapter are presented. It is assumed that there exists a Certification Trusted Authority (CTA) whose purpose is to generate watermarks and issue them to any user upon request. The CTA is memory-less and does not keep a track record of the watermarks issued to different users. At any instant in time, the CTA can issue watermarks to a single seller. It is further assumed that each time a seller requests for watermarks, the CTA issues unique watermarks. The seller of a watermarked image is represented as Alice. For encryption/decryption and digital signatures, the RSA public key cryptosystem is used [119]. Encryption and decryption operations are denoted by the functions $E_K(\cdot)$ and $D_K(\cdot)$, respectively. The subscript K is used to represent the cryptographic key used for encryption/decryption. For the purpose of illustration, assume (K_C^{pub}, K_C^{pri}) to be the respective public and private key pair of the CTA. Let (K_A^{pub}, K_A^{pri}) be the respective public and private key pair of Alice. Digital signature is represented by the function $S_S(\cdot)$. The subscript S represents the signer's identity. For example, for a message X , the digital signature of the CTA will be represented by $S_C(X)$. A brief overview of hash function, digital signature and blind source separation is given below:

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

5.2.1 Hash Function

Suppose a message is to be sent that contains ' p ' symbols and it is required to reduce the length of the message to say ' k ' symbols. A cryptographic hash function [4], [128] $H(x)$ maps the set of ' p ' symbols to a set of ' k ' symbols if $H(x)$ is easy to compute from x , however:

1. It is computationally difficult to find two different values of x that give the same $H(x)$, i.e., a hash function is *collision free*.
2. Given y in the image of $H(\cdot)$ no one can feasibly find an x such that $H(x) = y$, i.e., a hash function is *pre-image resistant*.

There are a number of hash functions proposed in the literature. The two famous ones are SHA1 and MD5 that give 160-bit and 128-bit hash values, respectively, for any length of a message [4]. Hash functions are also called message digest algorithms.

5.2.2 Digital Signature

A digital signature of a message is a number dependent on some secret known only to the signer, and additionally, on the content of the message being signed. It provides a way to protect the integrity of a digital document and to verify who signed it. One way to implement a digital signature scheme is to use a one-way hash function and the RSA public key cryptosystem [3], [4].

Signature Generation

Suppose Alice wants to send a digitally signed message, m to Bob. Alice will calculate the digital signature as follows:

1. Transform the message m to a message digest $H(m)$.
2. Encrypt $H(m)$ with her private key to get the digital signature $S_A(m)$:

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

$$S_A(m) = E_{K_A^{pri}}(H(m)).$$

3. Send the pair $[m, S_A(m)]$ to Bob.

Signature Verification

At the receiving end, Bob will verify Alice's signature as follows:

1. Decrypt $S_A(m)$ with Alice's public key to obtain $H(m)$:

$$H(m) = D_{K_A^{pub}}(S_A(m)).$$

2. Compute the hash $\overline{H(m)}$ of the message m . (For the purpose of clarity, the notation $\overline{H(X)}$ is used in the signature verification stage to represent the computed hash of a message X).
3. If $\overline{H(m)} = H(m)$, the signature will be considered valid.

5.2.3 Blind Source Separation using Independent Component Analysis

Independent Component Analysis (ICA) is probably the most widely used method for performing Blind Source Separation (BSS). It is a very general-purpose statistical technique to recover the independent sources, given only sensor observations that are linear mixtures of independent source signals [120], [121]. ICA model consists of two parts; the mixing process and the un-mixing process. In the mixing process, the observed linear mixtures $x_1, \dots, x_m$ of n number of independent components are defined as:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \dots + a_{jn}s_n \quad 1 \leq j \leq m \quad (5.2.1)$$

where $\{s_k, k = 1, \dots, n\}$ denote the source variables, i.e. the independent components, and $\{a_{jk}, j = 1, \dots, m; k = 1, \dots, n\}$ are the mixing coefficients. In vector-matrix form, the above mixing model can be expressed as:

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

$$x = As, \quad (5.2.2)$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (5.2.3)$$

is the mixing matrix, $x = [x_1 \ x_2 \ \dots \ x_m]^T$, $s = [s_1 \ s_2 \ \dots \ s_n]^T$ and T is the transpose operator. The un-mixing process [120], [121] can be formulated by computing the separation/un-mixing matrix Q so that the independent components can be obtained as:

$$s = Qx. \quad (5.2.4)$$

The simplest BSS model assumes that the number of linear mixtures and the independent components or sources is same. The objective of BSS is to find a linear representation in which the components are statistically independent. For performing BSS, techniques such as Principal Component Analysis (PCA) [122] are not feasible as they give components that are uncorrelated. However, there are many uncorrelated representations of signals that are actually not independent. As a matter of fact, independence is a much stronger property than uncorrelatedness. Independence implies uncorrelatedness, however, the opposite is not true. The goal of ICA is much broader than PCA as it gives components that are not only uncorrelated but statistically independent as well. This makes ICA suitable for performing BSS. It is to be noted that for the ICA model in Eq. 5.2.2, two major ambiguities exist. The first ambiguity is that the variances or energies of the extracted independent components cannot be determined as both the mixing matrix and the original independent components are unknown. This may also create ambiguity in the sign of the extracted components. The second ambiguity is that the original order of the independent components cannot be determined as both the mixing matrix and the original independent components are

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

unknown. In Section 5.4, it will be shown, how these ambiguities are addressed in watermarking schemes.

The use of ICA in watermarking application is not new. Noel and Szu [123] were among the first to introduce ICA in watermarking applications. Likewise, Dan Yu *et al.* [124] have proposed a blind watermarking technique using ICA. In this chapter, ICA is used for extracting the public watermark from the watermarked image. The basic idea is to use some specific image pattern as the public watermark; an example is shown in Figure 5.2. In Section 5.4 and Section 5.5, it is demonstrated how such patterns can be used within a cryptographic framework to represent the identities of the buyer and seller of a watermarked image. The image to be watermarked is linearly mixed with the public watermark to get the watermarked image. Hence, watermark extraction can be viewed as a blind source separation problem. To perform BSS, ICA is used to extract the public watermark from the watermarked image. ICA, however, poses a restriction that all the independent components with the possible exception of one component, must be non-Gaussian [120].

5.3 Watermark Embedding

In this section, the procedure for watermark embedding is described. Let the seller's original image be denoted by I and the watermarked image by I_w . To address buyer-seller identification and copyright protection, the proposed watermarking scheme uses two different watermarks. The first watermark is used to reveal the identity of the buyer and seller of the watermarked image. This watermark is named as the *public watermark*, W^{pub} . The second watermark serves two purposes. Firstly, it enables a legal seller to prove his/her ownership in case of multiple ownership claims. Secondly, in case of copyright infringement by a buyer, the extraction of this watermark will enable the seller to identify the malicious buyer from whom an illegal copy of the watermarked image has originated and further prove this fact to a third party. This watermark is called the *private watermark*, W^{pri} . Figure 5.1 shows the block diagram for watermark embedding.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.1: Block diagram of watermark embedding.

The private watermark W^{pri} is first embedded into the original image I to get an *intermediate-watermarked image*, I^* . The image I^* is then further watermarked with the public watermark W^{pub} to get the final watermarked image I_w and the public-watermark key. It will be shown in the watermark extraction procedure that embedding the public watermark after embedding the private watermark does not have any significant degradation on the private watermark extraction. In the following sections, procedure for embedding private and public watermarks is given.

5.3.1 Private Watermark Embedding

The private watermark is required to be very robust because of three main reasons. Firstly, it is used to resolve copyright infringement and multiple ownership claims. Secondly, since the public watermark is embedded after embedding the private watermark, the private watermark should withstand the distortions introduced due to public watermark embedding. The third reason for the private watermark to be robust is because it is the only means through which a genuine owner can prove his/her ownership in case the public watermark is destroyed. As Mintzer and Braudaway [125] have pointed out, in case of multiple watermark embedding, different watermarks might have different robustness requirements. Secondly, the order of embedding the watermarks is also very important. Mintzer and Braudaway suggest that the ownership watermark should be the

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

most robust and should be embedded first; the most fragile watermark should be embedded last, while moderately robust watermark(s) should be inserted in between. For successful multiple watermark embedding, the robust watermark that is embedded first should be able to withstand all the subsequent watermark insertions [125]. Since the private watermark is the most important, it is embedded first which is then followed by public watermark embedding. The spread spectrum watermarking technique proposed by Cox *et al.* [75] is used to embed the private watermark. This technique is very robust against a number of attacks as discussed in [75]. In addition, the watermark pattern can be detected even if an image is watermarked multiple number of times.

The private watermark W^{pri} is a sequence of real numbers,

$$W^{pri} = \{w_1, w_2, \dots, w_n\}, \quad (5.3.5)$$

where each w_i is chosen independently according to a normal distribution with zero mean and unit variance. The DCT of the original image I is taken and the watermark sequence w_i is embedded in the 1000 ($N=1000$) highest-valued AC coefficients using the following relation [75]:

$$\hat{z}_i = z_i(1 + \alpha_1 w_i), \quad (5.3.6)$$

where z_i is the i^{th} highest-valued AC DCT coefficient of I and α_1 controls the strength of the watermark. The modified DCT coefficients $\hat{z}_i$ are then inserted back in place of z_i and an inverse DCT is taken to get the intermediate-watermarked image I^* .

5.3.2 Public Watermark Embedding

The purpose of embedding the public watermark is to enable anyone having knowledge of the *public-watermark key* to extract the public watermark. The

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.2: Illustration of the public watermark. (a) Public Watermark. (b) Segment 1 of the public watermark. (c) Segment 2 of the public watermark.

public watermark is used to identify the buyer and seller of a watermarked image. An important question that arises is how to design a public watermark that can be associated with some information. For example, if the public watermark is required to reveal the identities of the buyer and seller of a watermarked image, how can this be achieved? This issue is addressed by using a watermark that portrays the hash of the information that is required to be associated with the watermark. Some commonly used hash functions are MD5 and SHA1 that give 128-bit and 160-bit hash values respectively [4]. Suppose M is a piece of information that uniquely identifies the buyer and seller of a watermarked image. In the proposed scheme, public keys of the buyer and seller are used for identification purpose. It should be noted that in a real world scenario, public keys are certified by some trusted certification authority and therefore can be used for identification purpose. The message M is calculated by concatenating the public keys of the buyer and seller of the watermarked image. The seller calculates the hash of M to get $H(M)$ and generates a watermark that portrays $H(M)$. The reason for using a cryptographic hash function is that no matter how long M is, the hash output will be compressed to 128-bit in case of MD5, or 160-bit in case of SHA1. For the purpose of illustration, suppose we have a hash sequence $H(M) = 0101010 \dots 10$. The public watermark for such a sequence is shown in Fig. 5.2a. The watermark pattern can accommodate 256-bit of information. The box in black represents a '0' while the box in white represents a '1'. In case the hash function used is MD5, the hash output will be 128-bit. Since a watermark pattern can accommodate 256-bit of information, the remaining blocks in the watermark

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

can be zero-padded or the hash pattern can be tiled to cover the entire image area of the watermark. For the sake of convenience, the public watermark pattern shown in Fig. 5.2a will be used in the discussion to follow. In our experiments, a black pixel in Fig. 5.2a is represented by a gray value of zero, while a white pixel is represented by a gray value of 255. The public watermark is segmented into W^{pub1} and W^{pub2} as shown in Fig. 5.2b and Fig. 5.2c, respectively.

$$W^{pub} = W^{pub1} + W^{pub2}. \quad (5.3.7)$$

A third level wavelet decomposition of the intermediate-watermarked image I^* is performed and the public watermark is embedded in the LL sub-band. If the image I^* is of dimension $N \times N$, then the size of the public watermark will be $N/8 \times N/8$. As pointed out in [75], for a watermark to be robust, it should be embedded in the perceptually most significant components of the image spectrum. The public watermark is embedded in the LL sub-band for increased robustness as the LL sub-band contains the most important information of an image. The embedding coefficients should, however, be carefully chosen keeping in view that for a particular value of the embedding coefficient, the LL sub-band is more susceptible to perceptual distortion as compared to the other sub-bands. Let us denote the third level LL sub-band wavelet coefficient of I^* by Y_{LL3} . Following are the steps for public watermark embedding:

1. Perform the third level discrete wavelet decomposition of I^* . Embed W^{pub1} and W^{pub2} separately in Y_{LL3} according to the following rules:

$$\hat{Y}_{1_{LL3}} = Y_{LL3} + \alpha_2 W^{pub1}, \quad (5.3.8)$$

$$\hat{Y}_{2_{LL3}} = Y_{LL3} + \alpha_2 W^{pub2}, \quad (5.3.9)$$

where α_2 controls the watermark embedding strength and $\hat{Y}_{1_{LL3}}$, $\hat{Y}_{2_{LL3}}$ are the modified LL sub-band wavelet coefficients after embedding the watermark .

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.3: Watermark embedding. (a) Original image. (b) Intermediate watermarked image. (c) Watermarked image. (d) Public-watermark key.

2. The watermarked image, I_w is obtained by replacing the Y_{LL3} coefficients of I^* by the modified $\hat{Y}1_{LL3}$ coefficients and then taking the inverse discrete wavelet transform. The inverse discrete wavelet transform takes into account all the frequency sub-bands.
3. The modified wavelet coefficients $\hat{Y}2_{LL3}$ are then scaled and rounded-off into an n -bit integer to obtain the public-watermark key, K_w . The purpose of scaling and rounding-off is to compress the size of K_w . The coefficient of $\hat{Y}2_{LL3}$ that has the minimum value is always mapped to zero while the coefficient of $\hat{Y}2_{LL3}$ that has the maximum value is mapped to 2^n . The remaining coefficients are linearly mapped between the values zero and 2^n using the equation of a straight line. The mapped wavelet coefficients are then rounded-off to the nearest integer. For an 8-bit gray level image having 256×256 pixels, there will be a total of 1024 $\hat{Y}2_{LL3}$ coefficients. By scaling and rounding-off these coefficients to 8 bits, the size of K_w will be compressed to 1024 bytes. By choosing different values of n , the size of K_w can be controlled. It has been experimentally observed that scaling and rounding-off to a 10-bit integer gives good extraction results. In this case, the size of K_w will be 1280 bytes.

Figure 5.3a shows the original cameraman image I , while Fig. 5.3b shows the intermediate-watermarked image I^* obtained by following the steps outlined in Section 5.3.1. Figure 5.3c and Fig. 5.3d show the watermarked image I_w and the corresponding public-watermark key K_w obtained by the steps outlined above.

5.4 Watermark Extraction

5.4.1 Public Watermark Extraction

The public watermark is extracted from the watermarked image using the public-watermark key K_w . Figure 5.4 shows the block diagram of public watermark extraction. A third level discrete wavelet decomposition of the watermarked image is first performed to get the LL -sub-band coefficients, Y_w . Note that the dimension of Y_w and K_w is same. The matrix Y_w is a linear mixture of Y_{LL3} and W^{pub1} (Eq. 5.3.8), while the matrix K_w is a linear mixture of Y_{LL3} and W^{pub2} (Eq. 5.3.9). There are a total of three sources Y_{LL3} , W^{pub1} and W^{pub2} in the two mixtures Y_w and K_w . To extract the public watermark, blind source separation is used as discussed in Section 5.2.3. Cardoso's JADE ICA algorithm [126] is used for watermark extraction. The mixtures Y_w and K_w are treated as inputs to the blind source separation process. Since two mixtures are being used, the BSS process will give two outputs, as shown in Fig. 5.5a and Fig. 5.5b. The first output consists of a distorted version of Y_{LL3} . We call this as the residue output. The second output consists of two parts. The left half is similar to the left half of W^{pub1} (Fig. 5.2b). The right half is, however, exactly the opposite of the right half of W^{pub2} (Fig. 5.4). This change in sign is because of the sign ambiguity present in the ICA algorithm as discussed in Section 5.2.3. By scaling the pixels values in Fig. 5.5b to a gray level range between 0 and 255 and flipping the right half, the extracted watermark $\hat{W}^{pub}$ is obtained, as shown in Fig. 5.5c.ⁱ The binary-pattern of the extracted watermark shown in Fig. 5.5c is similar to the embedded watermark (Fig. 5.2). In some cases, it might be required to flip the left half. Since the seller of the watermarked image knows the exact pattern of

ⁱThe JADE algorithm used in this chapter is based on linear mixing model, fourth-order statistics and non-iterative approach. Although this algorithm works well in the wavelet domain; like other BSS algorithms, there are ambiguities present with this algorithm, like scaling and sign change. Because of this reason, the hash bits of the right half of the extracted public watermark (Fig. 5.5b) are toggled, i.e. a binary 'one' becomes 'zero' and a binary 'zero' becomes 'one'. This will therefore give an incorrect value of the hash. To compensate this problem, the portion of the extracted output that has been inverted is flipped. Furthermore, the left half of Fig. 5.5b appears different from the left half of Fig. 5.5c. This is due to the high contrast between the left and the right half of Fig. 5.5b.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.4: Block diagram of public watermark extraction.

Figure 5.5: Public watermark extraction. (a) Public Watermark. (b) Segment 1 of the public watermark. (c) Segment 2 of the public watermark.

the public watermark, he/she can carry out BSS to see which half of the extracted output shown in Fig. 5.5b is required to be flipped. This information can then be conveyed to the recipient. Because the $\hat{Y}_{2_{LL3}}$ coefficients obtained from Eq. 5.3.9 are scaled and rounded-off into an n -bit integer, a simple subtraction of Y_w and K_w will not work.

5.4.2 Private Watermark Extraction

The private watermark extraction is non-blind and requires the original image I . To extract the private watermark, the DCT of the watermarked image I_w and the original image I is taken and the watermark sequence is extracted from the embedding locations using Eq. 5.3.6.

$$\bar{w}_i = \frac{1}{\alpha_1} \left(\frac{\bar{z}_i}{z_i} - 1 \right) \quad (5.4.10)$$

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

where $\bar{w}_i$ is the extracted watermark sequence and $\bar{z}_i$ are the DCT coefficients of I_w . The extracted watermark is then compared with the original watermark using some similarity measure. The normalized correlation coefficient [112] is used as the similarity measure. In our experiments, the values of α_1 and α_2 were chosen as 0.08 and 0.07 respectively. Since the normalized correlation coefficient is used as the similarity measure, the value of α_1 is not required to be known for comparing the extracted watermark with the reference watermark. It is important to note that embedding of the public watermark should not cause any significant degradation in the private watermark. Interestingly, due to the robustness property of the spread spectrum watermarking technique, the public watermark introduces a slight decrease in the correlation value of the private watermark from 1.00 to 0.96. This interference can be further minimized by carefully choosing the domain where both the watermarks are embedded. For example, if the public watermark is embedded in the *LH* or *HL* sub-band instead of the *LL*-sub-band, it will cause less interference with the private watermark. However, there will be a loss in robustness of the public watermark.

5.5 Copyright Protection Protocols

The proposed watermarking scheme consists of the following protocols to deal with copyright protection issues.

1. Watermarked Image Generation and Distribution Protocol.
2. Buyer-Seller Identification Protocol.
3. Copyright Infringement Protocol.

In Section 5.6, a few more protocols shall be discussed that can be used for resolving ownership claims in case of multiple ownership disputes.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

5.5.1 Watermarked Image Generation and Distribution Protocol

Suppose Alice wants to sell a watermarked image to Bob. This protocol will enable Alice to acquire a watermark certificate, Cer from the CTA that contains a valid private watermark and digital signature. Let (K_B^{pub}, K_B^{pri}) be the respective public and private key pair of Bob. Figure 5.6 shows the flow diagram of the watermarked image generation and distribution protocol. The protocol proceeds as follows:

1. Alice hashes her original image I to get $H(I)$. She then sends $H(I)$, her public key K_A^{pub} and certificate of her identity to the CTA along with a request for issuing a watermark.
2. CTA verifies Alice's identity. It then generates the private watermark W_A^{pri} for Alice. The private watermark is a pseudo-random noise sequence as described by Eq. 5.3.5.
3. CTA calculates the hash $H(W_A^{pri}, H(I), T_1)$. The parameter T_1 indicates the time stamp that is used to resolve ownership disputes. CTA encrypts $H(W_A^{pri}, H(I), T_1)$ with its private key K_C^{pri} to get digital signatures for $H(I)$ and W_A^{pri} .

$$S_C(W_A^{pri}, H(I), T_1) = E_{K_C^{pri}}(H(W_A^{pri}, H(I), T_1)). \quad (5.5.11)$$

4. A tuple X_A is formed as shown by Eq. 5.5.12. A digital signature $S_{CA}(X_A, T_1)$ is obtained by encrypting $H(X_A, T_1)$ with CTA's private key and then with Alice's public key.

$$X_A = \{W_A^{pri}, S_C(W_A^{pri}, H(I), T_1), T_1\} \quad (5.5.12)$$

$$S_{CA}(X_A, T_1) = E_{K_A^{pub}}(E_{K_C^{pri}}(H(X_A, T_1))). \quad (5.5.13)$$

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.6: Flow diagram of watermarked image generation and distribution protocol.

5. CTA sends the watermark certificate Cer_A to Aliceⁱⁱ

$$Cer_A = \{X_A, S_{CA}(X_A, T_1)\}. \quad (5.5.14)$$

6. Alice verifies Cer_A by first decrypting $S_{CA}(X_A, T_1)$ with her private key and then further decrypting the result with CTA's public key to get $H(X_A, T_1)$. She then hashes X_A and T_1 to get $\overline{H(X_A, T_1)}$. If $\overline{H(X_A, T_1)} = H(X_A, T_1)$, it will be verified that Cer_A has been generated by the CTA and that it has not been tampered. Alice then uses the watermark W_A^{pri} obtained from Cer_A to generate the intermediate-watermarked image I_A^* using the steps outlined in Section 5.3.1.
7. Alice hashes K_A^{pub}, K_B^{pub} and uses the hash bits to generate the public watermark, W_A^{pub} . She then segments W_A^{pub} into W_A^{pub1} and W_A^{pub2} using Eq. 5.3.7. Using the steps outlined in Section 5.3.2, she generates the watermarked image, I_{Aw} and the public-watermark key, K_{Aw} . She then encrypts

ⁱⁱInstead of using W_A^{pri} and its corresponding digital signature in Cer_A , the CTA can also use a seed (that can be used with a secure publicly known pseudorandom number generator) and its corresponding digital signature. This will save bandwidth. For further security, the CTA can also encrypt Cer_A with Alice's public key and then transmit the encrypted version of Cer_A to Alice.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

K_{Aw} with her private key to get CK_{Aw}

$$CK_{Aw} = (E_{K_A^{pri}}(K_{Aw})). \quad (5.5.15)$$

8. Alice calculates $H(W_A^{pri})$ and sends it along with I_{Aw} and CK_{Aw} to Bobⁱⁱⁱ
9. In this step, Bob will verify the genuine buyer-seller transaction between him and Alice. Bob performs the following steps:
 - (a) Decrypt CK_{Aw} with Alice's public key to get K_{Aw} . Using I_{Aw} and K_{Aw} extract the public watermark, $\hat{W}_A^{pub}$ according to the procedure outlined in Section 5.4.1.
 - (b) Hash K_A^{pub} , K_B^{pub} and compare the output of the hash function with the binary pattern obtained from $\hat{W}_A^{pub}$.

After performing Step 9a, Bob shall only be successful in extracting a genuine watermark pattern (like the binary pattern shown in Fig. 5.2a, if the public-watermark key has been encrypted with Alice's private key. This shall also prove that the extracted watermark has been embedded by Alice as no one else is supposed to know Alice's private key other than herself. Furthermore, if Step 9b is successful, Bob will be convinced that $\hat{W}_A^{pub}$ reflects his and Alice's identities.

10. After positive verification of Step 9, Bob sends the following to Alice:

$$S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub}) = E_{K_B^{pri}}(H(I_{Aw}, H(W_A^{pri}), K_A^{pub})) \quad (5.5.16)$$

11. Alice verifies $S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ and stores Cer_A , I_{Aw} , CK_{Aw} and $S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ as a record of this transaction with Bob^{iv}.

ⁱⁱⁱAlthough not mentioned, the flipping information for post-processing of the public watermark will also be transmitted.

^{iv}It is not necessary that Alice stores I_{Aw} as this will add an extra storage overhead. Instead she can re-generate the watermarked image I_{Aw} when required using Step 7 of the Watermarked Image Generation and Distribution Protocol. This will require some extra storage requirements

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

5.5.2 Buyer-Seller Identification Protocol

Suppose Alice makes a selling transaction with Bob as discussed in Section 5.5.1. The protocol discussed in this section can be used by Bob or any other party to show that Bob is a genuine buyer of the watermarked image I_{Aw} sold to him by Alice. The protocol requires I_{Aw} and CK_{Aw} that Bob obtained from Alice in the watermarked image generation and distribution protocol along with Alice's and Bob's public keys. Figure 5.7 shows the block diagram of the proposed buyer-seller identification protocol. The protocol proceeds as follows:

1. Decrypt CK_{Aw} with Alice's public key to get K_{Aw} .

$$K_{Aw} = (D_{K_A^{pub}}(CK_{Aw})). \quad (5.5.17)$$

2. Using I_{Aw} and K_{Aw} , extract the public watermark $\hat{W}_A^{pub}$ according to the procedure outlined in Section 5.4.1.
3. Hash the public keys of Alice and Bob to get $HPub$ of length L bits. For example, if the hash function used is SHA1, then L will be 160 bits.

$$HPub = H(K_A^{pub}, K_B^{pub}). \quad (5.5.18)$$

4. Compare the binary bit sequence of $HPub$ with the bit pattern obtained from $\hat{W}_A^{pub}$. If all the bits are compared successfully, then it shall be proved that Bob is the legal buyer of I_{Aw} , sold to him by Alice.

Remark

Is it possible for Bob to embed the binary sequence $HPub$ in any arbitrary image J and then claim that he is the legal buyer of J sold to him by Alice? It is easy

like the watermark embedding strength parameters and any secret key used in watermark embedding, etc. This storage, however, will be quite less as compared to storing the entire watermarked image. In order to make sure that the watermarked image re-generated in the future is 100% similar to the one that was generated in the past, Alice can also store the cryptographic hash of I_{Aw} .

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.7: Block diagram of buyer-seller identification protocol.

for Bob to generate the pattern $HPub$ shown by Eq. 5.5.18 since it only requires the knowledge of Alice's and Bob's public keys that are available in the public domain. However, Bob cannot obtain Eq. 5.5.15 (Step 7 of the watermarked image generation and distribution protocol), since it requires the knowledge of Alice's private key. If the correct private key of Alice is not used in this step, then the result of decryption in Step 1 of the buyer-seller identification protocol will not be correct. As a result, the extracted watermark will be gibberish. This shows that it is not possible for Bob to insert the pattern $HPub$ in any arbitrary image J and then claim that he is the legal buyer of J sold to him by Alice.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

5.5.3 Copyright Infringement Protocol

Suppose Alice finds an illegal copy, $\hat{I}_{Aw}$ of the watermarked image I_{Aw} that she had previously sold to Bob. Using this protocol, a judge can check whether $\hat{I}_{Aw}$ has originated from the watermarked image I_{Aw} .^v Figure 5.8 shows the block diagram of this protocol. This protocol requires either one or two stages to complete.

Stage 1

In this stage, the judge will follow the steps outlined in the buyer-seller identification protocol (Section 5.5.2) to extract the public watermark, $\hat{W}_A^{pub}$ from $\hat{I}_{Aw}$ using CK_{Aw} (supplied by Alice from Bob's transaction record). If the extracted watermark depicts Alice's and Bob's identities, then Bob will be liable for copyright infringement. Bob can however be smarter. Since he knows the public watermark, he can subtract its scaled version from $\hat{I}_{Aw}$ such that $\hat{W}_A^{pub}$ is not detected in $\hat{I}_{Aw}$. In such a case, Stage 2 of the protocol will be used.

Stage 2

In this stage, the judge will extract an estimation of the private watermark, $\hat{W}_A^{pri}$ from $\hat{I}_{Aw}$ to check whether Bob is guilty or not. In this stage, Alice will supply the judge with I_{Aw} , W_A^{pri} and $S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ from Bob's transaction record along with her original image I . The protocol proceeds as follows:

1. Use Alice's original image I to extract the private watermark $\hat{W}_A^{pri}$ from $\hat{I}_{Aw}$ using the procedure outlined in Section 5.4.2.
2. Decrypt $S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ with Bob's public key to get $H(I_{Aw}, H(W_A^{pri}), K_A^{pub})$:

$$H(I_{Aw}, H(W_A^{pri}), K_A^{pub}) = D_{K_B^{pub}}(S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})). \quad (5.5.19)$$

^vIn this chapter, the case in which an unauthorized copy of a watermarked image is distributed by a malicious seller or due to a security breach in the buyer/seller system is not considered. This problem is addressed in [91].

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

3. Use I_{Aw} , W_A^{pri} (supplied by Alice from her transaction record for Bob) and K_A^{pub} to get $\overline{H(I_{Aw}, H(W_A^{pri}), K_A^{pub})}$.
4. If $\overline{H(I_{Aw}, H(W_A^{pri}), K_A^{pub})} = H(I_{Aw}, H(W_A^{pri}), K_A^{pub})$, it shall be proved that Bob had purchased the watermarked image I_{Aw} from Alice that contains the private watermark W_A^{pri} . The reason for this is because in Step 2, $H(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ is obtained by decrypting $S_B(I_{Aw}, H(W_A^{pri}), K_A^{pub})$ using Bob's public key and that it contains Alice's public key as an argument.
5. Bob shall be considered guilty of copyright infringement if the following are true:
6. $\hat{W}_A^{pri}$ and W_A^{pri} match with high correlation.
7. $\hat{I}_{Aw}$ and I_{Aw} match with high correlation.

Remark

If Alice has sold different watermarked versions of the same cover image to different customers, how will she identify the particular customer from whom an illegal copy has originated? This task may become complicated, especially if the number of clients grows huge. Before reporting the case to the judge, Alice will first have to find out the identity of the buyer from whom the illegal copy has originated. For example, for each cover work, I_1 , I_2 , I_3 and so on that she watermarks and sells, she can maintain a separate data base of all the private watermarks that she had embedded into that particular cover work. Now if she finds, for example, an illegal image, I' , first she will sort out that for which cover work I' belongs. This can be done by using a number of image processing techniques that are available for efficient and effective comparison of images. Once I' is matched with a particular cover work, say I_2 , Alice will then narrow her search by extracting the watermarks she had embedded in I_2 and then comparing them with all the watermarks that she had stored w.r.t. to the cover work I_2 . The match with the highest correlation will enable her to decide about the buyer. After this, she may report that particular buyer to the judge.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.8: Block diagram of copyright infringement protocol.

5.6 Resolving Multiple Ownership Claims

In this section, a number of problems that may arise in case of multiple ownership claims over a watermarked image are discussed. In particular, the following three attacks show how the proposed scheme can resist such attacks.

1. Multiple Watermarked Image Attack.
2. Invertible Watermark Attack.
3. Watermark Removal Attack.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Throughout the discussion to follow, assume that Alice watermarks her original image I using the private watermark W_A^{pri} obtained from the CTA's watermark certificate given by Eq. 5.5.14 to get a watermarked image I_{Aw} .

5.6.1 Multiple Watermarked Image Attack

Suppose Bob obtains a copy of I_{Aw} and further watermarks it using his private watermark W to get the watermarked image I_{ABw} , for which he claims to be the legal owner. Resolving an ownership dispute between Alice and Bob over I_{ABw} is quite straightforward if the watermarking technique is robust [75], [93]. For example, in case of the spread spectrum technique used in this chapter, both Alice's and Bob's watermarks can be detected in the disputed image I_{ABw} . Bob with his fake original, I_{Aw} can show the presence of his watermark W in I_{ABw} . However, he cannot show the presence of W in Alice's original image I . Alice on the other hand can show the presence of her watermark W_A^{pri} both in Bob's fake original I_{Aw} and as well as in the disputed image I_{ABw} . In this way, Alice can prove her legal ownership of I_{ABw} . To show a numerical example, the cameraman image I shown in Fig. 5.3a was watermarked with a PN sequence, W_A^{pri} to represent I_{Aw} . Then I_{Aw} was watermarked with another PN sequence, W to get another watermarked image, say I_{ABw} . In both the cases, the embedding strength of the watermark α_1 was kept as 0.08. Using I as the original image, the watermark W_A^{pri} was detected in I_{Aw} and I_{ABw} with normalized correlation coefficient of 0.998 and 0.691 respectively. Similarly, using I_{Aw} as the original image, the watermark W was detected in I_{ABw} with a normalized correlation coefficient of 0.995. However, the normalized correlation coefficient for W in I was only -0.0316. These results confirm the above discussion.

5.6.2 Invertible Watermark Attack

The scenario depicted in Section 5.6.1 enables Alice to claim her legal ownership because Bob cannot show the presence of his watermark in Alice's original image

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

I. What if Bob is able to show the presence of his fake watermark in Alice's original image? This might lead to an ownership deadlock. In fact, Craver *et al.* [93] were the first to show such a scenario. The attack proposed in [93] works for the decoding strategy shown by Eq. 5.4.10 in which the extraction of the private watermark is non-blind. The idea is quite simple. In contrast to what was shown in Section 5.6.1, Bob does something smarter. Instead of embedding a watermark W in I_{Aw} , he subtracts W from I_{Aw} to get an image, $\bar{I}_{Aw}$ which he calls his original. Let us denote the watermark embedding and subtraction operators by $\oplus$ and $\ominus$ respectively. With this notation, Alice's watermarked image and Bob's fake original are represented by Eq. 5.6.20 and Eq. 5.6.21 respectively.

$$I_{Aw} = I \oplus W_A^{pri}. \quad (5.6.20)$$

$$\begin{aligned} \bar{I}_{Aw} &= I_{Aw} \ominus W, \\ &= (I \oplus W_A^{pri}) \ominus W. \end{aligned} \quad (5.6.21)$$

In terms of I_{Aw} and I , Eq. 5.6.21 can be written as follows:

$$\begin{aligned} I_{Aw} &= \bar{I}_{Aw} \oplus W, \\ I &= (\bar{I}_{Aw} \ominus W_A^{pri} \oplus W). \end{aligned} \quad (5.6.22)$$

From Eq. 5.6.22 it can be seen that by using his fake original $\bar{I}_{Aw}$, Bob can show the presence of his watermark W both in Alice's watermarked version I_{Aw} and Alice's original image I . Bob can therefore accuse Alice that I_{Aw} and I are indeed his copies of the watermarked image I_{Aw} . Similarly, Alice can also show the presence of her watermark W_A^{pri} in both $\bar{I}_{Aw}$ and I_{Aw} . Craver *et al.* [93] have termed such a scheme as *invertible*, which can actually lead to an ownership deadlock. To solve this problem, Craver *et al.* [93] suggested the following:

1. Hash the original image I to generate a seed S .

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

2. The seed S is used by a fixed pseudorandom number generator to generate the watermark W_s .
3. The watermarked image, $I_w = I \oplus W_s$.

Now an attacker cannot generate a fake original because generating a fake original requires subtraction of a watermark that is derived from the original itself. Ramkumar and Akansu have shown that this method is still invertible [92]. Their proposed attack requires about 10^9 trials to find the watermark for which the probability of error in watermark detection is bounded by 10^{-9} . This indicates that the attack is computationally feasible. The authors in [92] have also proposed some remedies to ensure non-invertibility. With the proposed protocol, however, an invertible watermark attack does not seem possible because the watermarks are issued by the CTA. Suppose Bob wants to create a fake original $\bar{I}_{Aw}$ by subtracting a watermark W from I_{Aw} . With the proposed scheme, Bob does not have the liberty to choose W because the watermarks are issued by the CTA. Suppose the CTA issues Bob the watermark W_B^{pri} . Can Bob subtract W_B^{pri} from I_{Aw} to get $\bar{I}_{Aw}$ and then claim $\bar{I}_{Aw}$ to be his fake original? To answer this question, refer to Step 1 of the watermarked image generation and distribution protocol. Before Bob can request for a private watermark from the CTA, he is required to send the hash of his original image to the CTA. Since Bob can only obtain his fake original $\bar{I}_{Aw}$ once he subtracts W_B^{pri} from I_{Aw} , he cannot perform Step 1 of the watermarked image generation and distribution protocols.

The security of the proposed scheme against invertibility from another perspective will now be examined. Bob calculates $H(I_{Aw})$ and sends it to the CTA along with a request for a private watermark. Suppose CTA generates a private watermark W_B^{pri} and sends the following watermark certificate to Bob.

$$\begin{aligned} Cer_B &= \{X_B, S_{CB}(X_B, T_2)\}, \\ X_B &= \{W_B^{pri}, S_C(W_B^{pri}, H(I_{Aw}), T_2), T_2\}, \\ S_{CB}(X_B, T_2) &= E_{K_B^{pub}}(E_{K_C^{pri}}(H(X_B, T_2))). \end{aligned} \tag{5.6.23}$$

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

It is very realistic to assume that the time stamp $T_2 > T_1$, since Bob can only send I_{Aw} after Alice generates I_{Aw} (using Cer_A with the time stamp T_1). If Bob subtracts W_B^{pri} from I_{Aw} to get a fake original $\bar{I}_{Aw}$ can he accuse Alice in a court of law that I_{Aw} and I are actually his copies of the watermarked image I_{Aw} ? Now a protocol is presented that will prevent Bob from claiming such false ownership. The protocol proceeds as follows:

1. Alice and Bob present their original images I and $\bar{I}_{Aw}$ along with their respective watermark certificates Cer_A and Cer_B to the judge.
2. The judge will check whether the identities of Alice and Bob are associated with their respective watermark certificates Cer_A and Cer_B . In addition, the judge will also check the authenticity and integrity of Cer_A and Cer_B . For example in case of Alice, the identity of Alice and the authenticity and integrity of Cer_A are verified as follows:
 - (a) Alice will first decrypt $S_{CA}(X_A, T_1)$ with her private key K_A^{pri} . The result is then further decrypted with the CTA's public key K_C^{pub} to get $H(X_A, T_1)$.

$$H(X_A, T_1) = D_{K_C^{pub}}(D_{K_A^{pri}}(S_{CA}(X_A, T_1))). \quad (5.6.24)$$

Note:

The CTA obtains $S_{CA}(X_A, T_1)$ in Step 4, Section 5.5.1 by first encrypting $H(X_A, T_1)$ with CTA's private key and then with Alice's public key. Therefore, $S_{CA}(X_A, T_1)$ has to be first decrypted with Alice's private key and then with the CTA's public key to get back $H(X_A, T_1)$. However, in a real world scenario, Alice cannot release her private key to the judge as this may compromise the security of the public key cryptosystem. As an alternative, she can decrypt $S_{CA}(X_A, T_1)$ with her private key and present the result to the judge. The judge can then again encrypt this result with Alice's public key. If the judge is able to get back $S_{CA}(X_A, T_1)$, the judge will be convinced that Alice

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

has correctly performed the decryption. Using CTA's public key, the judge then further decrypts the decrypted result given by Alice to get $H(X_A, T_1)$. This is possible because in the RSA public key cryptosystem, used in this chapter, if the public key is used for encryption, then the corresponding private key is used for decryption and vice versa.

(b) Retrieve X_A from Cer_A and T_1 from X_A . Hash X_A and T_1 to get $\overline{H(X_A, T_1)}$. If $\overline{H(X_A, T_1)} = H(X_A, T_1)$, the following will be proved:

- i. Cer_A has been assigned to Alice by the CTA.
- ii. S_{CA} has been generated by the CTA at the instant T_1 and it has not been tampered by Alice or any one else.

3. The judge will then check whether the hash of Alice's and Bob's supplied original images along with their respective private watermarks and time stamp provided in their watermark certificates match with CTA's signatures.

For example, in case of Alice, the judge will do the following:

(a) Decrypt $S_C(W_A^{pri}, H(I), T_1)$ with the CTA's public key to get $H(W_A^{pri}, H(I), T_1)$

$$H(W_A^{pri}, H(I), T_1) = D_{K_C^{pub}}(S_C(W_A^{pri}, H(I), T_1)). \quad (5.6.25)$$

(b) Using W_A^{pri} and T_1 from Cer_A and the original image I supplied by Alice (Step 1), calculate the hash, $\overline{H(W_A^{pri}, H(I), T_1)}$. If $\overline{H(W_A^{pri}, H(I), T_1)} = H(W_A^{pri}, H(I), T_1)$, the judge will be convinced that the image I that Alice presented as her original is indeed the image that was used by Alice to obtain the watermark certificate Cer_A .

4. The judge will also check the visual similarity/correlation between I and I_{Aw} .

5. Similarly, for Bob, the judge will perform the following steps:

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

- (a) Decrypt $S_C(W_B^{pri}, H(I_{Aw}), T_2)$ with CTA's public key to get $H(W_B^{pri}, H(I_{Aw}), T_2)$:

$$H(W_B^{pri}, H(I_{Aw}), T_2) = D_{K_C^{pub}}(S_C(W_B^{pri}, H(I_{Aw}), T_2)). \quad (5.6.26)$$

- (b) Using W_B^{pri} and T_2 from Cer_B and the original image $\bar{I}_{Aw}$ supplied by Bob (Step 1), calculate the hash, $\overline{H(W_B^{pri}, H(\bar{I}_{Aw}), T_2)}$. It is interesting to note that Bob obtained Cer_B using I_{Aw} . However, the fake original that Bob has presented in Step 1 is $\bar{I}_{Aw}$. Since $H(I_{Aw}) \neq H(\bar{I}_{Aw})$, therefore, $\overline{H(W_B^{pri}, H(\bar{I}_{Aw}), T_2)} \neq H(W_B^{pri}, H(I_{Aw}), T_2)$. Therefore, Bob's claim for false ownership will not be successful. To make the scheme invertible, Bob has to tweak the image $\bar{I}_{Aw}$ to get another image $\tilde{I}_{Aw}$ such that:

- i. $H(\tilde{I}_{Aw}) = H(I_{Aw})$.
- ii. There should be a high visual similarity/correlation between I_{Aw} and $\tilde{I}_{Aw}$.

Finding an image $\tilde{I}_{Aw}$ which satisfies condition (i) requires breaking the security of the cryptographic hash function. In case, if the hash function used is SHA1, there will be 2^{160} different combinations from which only one will satisfy condition (i). This is computationally difficult.

5.6.3 Watermark Removal Attack

Consider a scenario in which Bob performs some operation on Alice's watermarked image I_{Aw} to get another image, $\hat{I}$ such that the strength of Alice's watermark W_A^{pri} in $\hat{I}$ is significantly reduced to an extent that W_A^{pri} is rendered undetected in $\hat{I}$. Next Bob sends $H(\hat{I})$ to the CTA along with a request for a genuine watermark. Suppose the CTA sends Bob the watermark certificate:

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

$$\begin{aligned}
 Cer_B &= \{X_B, S_{CB}(X_B, T_2)\}, \\
 X_B &= \{W_B^{pri}, S_C(W_B^{pri}, H(\hat{I}), T_2), T_2\}, \\
 S_{CB}(X_B, T_2) &= E_{K_B^{pub}}(E_{K_C^{pri}}(H(X_B, T_2))).
 \end{aligned} \tag{5.6.27}$$

Again, it is realistic to assume that the time stamp $T_2 > T_1$, since Bob can only obtain $\hat{I}$ after Alice generates I_{Aw} (using Cer_A with the time stamp T_1). Bob watermarks $\hat{I}$ with W_B^{pri} to get a watermarked image, $\hat{I}_{Bw}$. If Alice accuses Bob that $\hat{I}_{Bw}$ originated from I_{Aw} , can Bob prove his fake ownership of $\hat{I}_{Bw}$? Despite the fact that W_A^{pri} cannot be detected in $\hat{I}_{Bw}$, a protocol that shall prevent Bob from his false claim will be shown. For this protocol to work, Alice should have a copy of her watermarked image I_{Aw} . Now the main steps of this protocol are outlined as:

1. Alice and Bob present their original images I and $\hat{I}$ along with their respective watermark certificates Cer_A and Cer_B to the judge.
2. As discussed in Step 2 (Section 5.6.2), the judge will check whether the identities of Alice and Bob are associated with their respective watermark certificates Cer_A and Cer_B . The judge will then check the authenticity and integrity of Cer_A and Cer_B . In addition, the judge will also check whether the hash of Alice's and Bob's supplied original images along with their respective private watermarks and time stamp provided in their watermark certificates match with the CTA's signatures.
3. The judge will then check the visual similarity between (I, I_{Aw}) and $(\hat{I}, \hat{I}_{Bw})$. Since the underlying watermarking scheme is invisible, there should be a very high visual similarity/correlation between the original image and its watermarked version. For a particular claimant, if visual similarity/correlation is not found between the original image and its watermarked version, then that person will not be considered as a candidate for true ownership. It should be noted that for Alice's claim to be true, there should be a high visual similarity/correlation between $\hat{I}_{Bw}$ and I_{Aw} .

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

4. The judge will then check for the presence of Alice's and Bob's private watermarks W_A^{pri} and W_B^{pri} in their respective watermarked images, I_{Aw} and $\hat{I}_{Bw}$.
5. In this specific illustration, W_A^{pri} will be detected in I_{Aw} with a high correlation. Similarly W_B^{pri} will also be detected in $\hat{I}_{Bw}$ with a high correlation. The judge will then decide about the actual owner by looking at the time stamp in Alice's and Bob's watermark certificates. Since Alice got Cer_A before Bob, therefore, $T_1 < T_2$. As a result, Alice will be considered the true owner and Bob shall fail to prove his false claim of ownership.

5.7 Security Analysis

In this section, the security analysis of proposed scheme against tampering of the watermark certificate, or the original image is presented. An attack called the *Time Stamp Attack* is introduced to show that the scheme can resist such an attack. In addition, it will be shown that it is computationally infeasible for an attacker to modify or tamper the contents of the watermark certificate or the original image. As discussed in Section 5.6.3, in case of watermark removal attack, the true seller will be identified by looking at the CTA's time stamp provided in the watermark certificate. Consider again the scenario discussed in Section 5.6.3. A question that arises is whether it is possible for Bob to watermark Alice's watermarked image I_{Aw} using a watermark obtained from the CTA at time T_0 ($T_0 < T_1$). If Bob somehow manages to launch this attack, he will be considered as the true seller in case of an ownership dispute with Alice. We call this attack as the Time Stamp Attack (TSA) and show that the proposed set of copyright protocols can prevent Bob from launching such an attack. To demonstrate this fact, assume that for some image J that Bob possesses, CTA issues a watermark certificate Cer_{B0} to Bob at time T_0 . Let the watermark certificate Cer_{B0} be defined as

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

$$Cer_{B0} = \{X_{B0}, S_{CB0}(X_{B0}, T_0)\} \quad (5.7.28)$$

$$X_{B0} = \{W_{B0}^{pri}, S_C(W_{B0}^{pri}, H(J), T_0), T_0\} \quad (5.7.29)$$

$$S_{CB0}(X_{B0}, T_0) = E_{K_C^{pub}}(E_{K_C^{pri}}(H(X_{B0}, T_0))) \quad (5.7.30)$$

Suppose, later in time, Alice watermarks her original image I using the private watermark W_A^{pri} obtained from the CTA's watermark certificate Cer_A given by Eq. 5.5.14 to generate a watermarked image I_{Aw} . Let us assume that Bob gets a copy of Alice's watermarked image I_{Aw} , and some how removes W_A^{pri} from I_{Aw} to get another image $\hat{I}$. He then watermarks $\hat{I}$ using the private watermark W_{B0}^{pri} obtained from Cer_{B0} to get a watermarked image I_{Bw0} . Now suppose, there is an ownership dispute between Alice and Bob over the image I_{Bw0} . Alice accuses Bob that I_{Bw0} originated from I_{Aw} . The question that arises is whether Bob can prove his fake ownership. The protocol discussed in Section 5.6.3 can be used to decide about the actual seller of I_{Bw0} . As discussed in Section 5.6.3, before checking the presence of Alice's and Bob's private watermarks in the respective watermarked images I_{Aw} and I_{Bw0} , the judge will check the following:

1. Association of Alice and Bob with their respective watermark certificates.
2. Authenticity and integrity of Cer_A and Cer_{B0} .
3. The perceptual similarity/correlation between (I, I_{Aw}) , $(\hat{I}, \hat{I}_{Bw0})$ and (I_{Aw}, I_{Bw0}) .
4. The authenticity and integrity of the claimant's supplied original image and its respective digital signature by the CTA in the watermark certificate.

Proposition 1

To prove his fake ownership of I_{Bw0} , Bob has to find a fake original image P such that the following conditions are satisfied:

1. There should be no perceptual difference between the image P and I_{Bw0} and further the normalized correlation coefficient between P and I_{Bw0} should approach unity.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

2. The fake original image P , the watermark and the time stamp in the watermark certificate Cer_{B0} should satisfy the following:

$$H(W_{B0}^{pri}, H(P), T_0) = H(W_{B0}^{pri}, H(J), T_0). \quad (5.7.31)$$

Proof

As discussed in Section 5.6.3, in case of Bob, the judge will check the perceptual similarity/correlation between I_{Bw0} and the image that Bob will present to the judge as his original. Since Bob has obtained his watermark certificate Cer_{B0} before he had the knowledge of I_{Aw} , it is reasonable to assume that the original image J that Bob had used to get Cer_{B0} will be different from I_{Aw} . Therefore, Bob has to fool the judge by presenting a fake image P that is visually similar to I_{Bw0} instead of J . This proves Condition 1. Condition 2 can be proved by proving Eq. 5.7.31. Since Bob had provided $H(J)$ to the CTA for obtaining Cer_{B0} , therefore, the CTA's digital signature will be,

$$S_C(W_{B0}^{pri}, H(J), T_0) = E_{K_C^{pri}}(H(W_{B0}^{pri}, H(J), T_0)). \quad (5.7.32)$$

During the ownership verification protocol discussed in Section 5.6.3, the judge will perform the following:

$$H(W_{B0}^{pri}, H(J), T_0) = D_{K_C^{pub}}(S_C(W_{B0}^{pri}, H(J), T_0)). \quad (5.7.33)$$

To keep the CTA's digital signature intact, Bob's fake original P should satisfy Eq. 5.7.31.

Proposition 2

It is computationally difficult for Bob to find an image P that satisfies Eq. 5.7.31.

Proof

One way that Bob may use to start finding a fake image P is to make small changes in I_{Bw0} until Eq. 5.7.31 is satisfied. The changes made in I_{Bw0} should

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

be such that P looks visually similar to I_{Bw0} . However, for Bob to be successful, he has to try all unique combinations of the hash functions before he can get one that satisfies Eq. 5.7.31. In addition, the changes that Bob makes to get P from I_{Bw0} should still keep a high visual similarity/correlation between P and I_{Bw0} . Finding an image P that satisfies Eq. 5.7.31 requires breaking the security of the cryptographic hash function. In case, the hash function used is SHA1, there are 2^{160} different combinations from which only one will satisfy Eq. 5.7.31. This is computationally difficult.

Remark

Let us assume a hypothetical scenario in which Bob is successful in finding a fake original P that satisfies Eq. 5.7.31. However, to generate an authentic watermark certificate, Bob will have to forge the digital signature of the CTA to generate the second element of the watermark certificate. This is computationally difficult as it implies breaking the RSA public key cryptosystem. Therefore, the cryptographic security of the hash function and the RSA cryptosystem ensures the security of the proposed watermarking protocols against TSA and any other attack that involves any modification or fake insertion of the original image (whose hash was presented to the CTA for obtaining the watermark certificate) or the private watermark present in the watermark certificate.

5.8 Improvement in the Key Size and System's Robustness to JPEG Compression

In this section, we show the improvement in the key size of the ICA public watermark and robustness in the public watermark extraction. In [124], Dan Yu and Sattar have used spatial domain approach for watermark embedding and extraction. Due to spatial domain, the size of the host image and the watermark needs to be the same. If the size of the watermarked image is, for example, 256×256 pixels, then the technique used in [124] would require a public-watermark key of size $2 \times 256 \times 256$ pixels or 128 kB. However, using the wavelet domain

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

watermarking approach presented in this chapter, if the wavelet coefficients are rounded off to 10-bit integer, the size of the public-watermark key would be 1280 bytes. We feel that this is a considerable improvement.

We now show the robustness improvement to JPEG compression of our proposed wavelet domain approach as compared to the spatial domain approach proposed in [124]. Figure 5.9 shows the sections of the spatial public watermarks. We have used the Cameraman image shown in Fig. 5.3a to embed these watermarks in the spatial domain according to the method described in [124]. The watermarked image had a PSNR of 47dB. To test the robustness of spatial domain watermark embedding against JPEG compression, the watermarked image was JPEG compressed with quality factors of 100 and 75. Figure 5.10 shows the extracted public watermarks. As compared to the extracted watermark in Fig. 5.10a, the extracted watermark shown in Fig. 5.10b has been quite degraded because of the increase in JPEG compression ratio.

We now compare this result with our proposed wavelet domain approach for watermark embedding. The same host image (Fig. 5.3a) is used in this case. Figure 5.11 shows the two segments of the wavelet domain public watermark. These watermarks were embedded in the wavelet domain according to the method described in Section 5.3.2. The watermarked image had a PSNR of 47dB. The watermarked image was then JPEG compressed with quality factors of 100 and 75. Watermark was extracted using the method outlined in Section 5.4.1. The results of watermark extraction are shown in Fig. 5.12. The watermark pattern is clearly visible in both the cases. Comparing this result with the result obtained in Fig. 5.10, we conclude that the proposed wavelet domain method for watermark embedding is quite robust to JPEG compression as compared to the spatial domain method used in [124].

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.9: Public watermarks. (a) Segment 1 of the public watermark. (b) Segment 2 of the Public Watermark.

Figure 5.10: Watermark extraction in the spatial domain. (a) Extracted public watermark at JPEG QF=100. (b) Extracted public watermark at JPEG QF=75.

Figure 5.11: Wavelet domain public watermarks. (a) Segment 1 of the public watermark. (b) Segment 2 of the public watermark.

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

Figure 5.12: Watermark extraction in the wavelet domain. (a) Extracted public watermark at JPEG QF=100. (b) Extracted public watermark at JPEG QF=75.

5.9 Conclusion

A watermarking algorithm alone cannot completely address the complex issues involved in copyright protection. For a reliable and secure watermarking scheme, it is necessary that the watermarking algorithm being used is well integrated with a secure protocol. In this chapter, a secure watermarking scheme that is aimed at addressing some of the important issues in copyright protection is proposed. Specifically, three issues of copyright protection have been focused, that is, buyer-seller identification, copyright infringement and verification of ownership. By buyer-seller identification, it is meant that a successful watermark extraction at the buyer's end will reveal identities of the buyer and seller of a watermarked image. In case of copyright infringement, the proposed scheme will enable the seller to identify the specific buyer from whom an illegal copy of a watermarked image has been originated and further prove this fact to a third party. By verification of ownership, it is meant that the seller of a watermarked image should be able to prove his or her genuine ownership in case of multiple ownership claims. In conventional watermarking schemes, the seller of a watermarked image embeds some information, for example a logo, in a cover image to associate his/her ownership with the image. But the question which arises is that how can some kind of information (i.e. a watermark) be associated with a particular person? To resolve this issue, there has to be some legal binding of the extracted water-

A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection

mark with the identity of the person who claims to be the buyer/seller of the watermarked image. This issue has been addressed by associating the identities of the buyer/seller with their respective public-private key pairs. In a real world situation, Public Key Infrastructure (PKI) provides a framework in which a public-private key pair is associated with the identity of a person, for example, by issuing digital certificates [127]. With this idea, a secure watermarking scheme that uses public-private keys and digital signatures to bind the identities of the buyer/seller with the watermarked image has been devised. Using public-private keys and digital signatures, the proposed scheme enables a buyer to prove that he/she has legally purchased a watermarked image from a specific seller. In case of copyright infringement, the cryptographic primitives allow a seller to identify the specific buyer from whom an illegal copy has been originated and further prove this fact to a third party. Issues pertaining to multiple ownership claims have also been addressed by using watermarks and their respective time-stamped digital signatures issued from a certified trusted authority. The security analysis shown in Section 5.7 demonstrates the security of the proposed scheme against tampering of the original image, the time stamp or the watermarks. The cryptographic security of the hash function and the RSA public key cryptosystem ensure the security of proposed watermarking scheme against such attacks.

Although the spread spectrum technique used for embedding the private watermark is quite robust, however, it is a non-blind technique. For example, in case of copyright infringement and ownership verification, the legal seller is required to produce the original image to resolve the ownership dispute. This may not be desirable in some applications as it potentially leaks the secrecy of the original un-watermarked image. Using a blind watermarking technique for private watermark extraction seems a good solution. However, issues related to ambiguity in dispute resolving or ownership deadlocks due to weaknesses in watermarking algorithms or the cryptographic protocols have to be thoroughly investigated.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

The tremendous growth in the field of information technology has made it very easy for individuals to share, sell and purchase multimedia contents like image, video or audio over insecure networks like the Internet. Along with the boom in information technology, the computing power has also increased drastically, thus making it possible to implement complex signal processing algorithms on desktop PCs. As a result, powerful multimedia processing softwares are now available that can be used for advanced filtering, addition or removal of contents, sophisticated animations, etc. There is also a negative side of this technology. Tampering in a digital media can now be done very easily in a way that is difficult to trace. Forging of digital documents, illegal distribution of multimedia contents like images, movies, games, etc over the Intranet or the Internet has become very easy. These threats have created huge losses in revenues and have triggered researchers around the world to come up with multimedia security technologies that can counter them. As a result, we now see several emerging technologies like digital watermarking, fingerprinting, steganography, multimedia authentication and encryption.

Developing security technologies for multimedia applications is quite different from conventional security paradigms covered by cryptography. A multimedia en-

Conclusion and Future Directions

vironment has several issues like lossy compression, filtering operations to improve the quality of the content, geometric transformations, like resizing the content, etc. This makes the task of designing multimedia security algorithms very much challenging. Further, for different applications, different level of security is desired. For example, the level of security required to encrypt an image that is to be transmitted from a military satellite to the respective ground station is much more than the security required to encrypt an image or a video on a cable TV for home entertainment. Hence, designing a security scheme for a multimedia application needs to be done keeping in view its very intrinsic mechanics, the application scenario and the nature of non-malicious distortion that the multimedia content may undergo during transmission or storage. To balance security along with these extra set of requirements is some thing that we feel is quite challenging. This was the basic motivation that has been the primary driving force towards the research work done in this thesis.

This thesis was geared towards addressing three closely related problems pertaining to multimedia security, i.e., encryption, authentication and copyright protection. Specifically, encryption, authentication and copyright protection algorithms are developed for digital images. The basic theme behind the proposed encryption and authentication algorithms is to make them robust against content preserving operations and at the same time provide sufficient security. This is a drift from standard cryptographic techniques in which there is no robustness requirement. Chapter 1 and Chapter 2 define the fundamental goals of multimedia security and present a comprehensive review about state-of-the-art encryption, authentication and copyright protection technologies for digital images.

In Chapter 3, a JPEG compression tolerant image encryption scheme has been proposed. A unique feature of this scheme is that the cryptosystem is an independent entity and is not an integral part of any compression algorithm like for example, JPEG. The proposed image encryption scheme uses the actual image data in a file to perform encryption and decryption. Due to this reason, if the encrypted image is stored as a BMP file, for example, decryption is possible

Conclusion and Future Directions

no matter the encrypted file is converted into some other format, for example JPEG, GIF, TIFF, etc. Further, the proposed scheme has the capability to recover the plaintext-image from the cipher-image even if the cipher-image data is compressed using JPEG lossy compression. The distortion in the recovered plaintext-image will increase if the JPEG QF is decreased. The security analysis carried out reveals that the proposed scheme can withstand typical cryptographic attacks that are used to test an encryption scheme.

In Chapter 4, a wavelet-based hashing scheme is proposed that is designed to simultaneously address core multimedia issues; robustness, security and tamper detection. The experimental results and analysis presented in the chapter show that the proposed scheme generates a hash that is very sensitive to change in the key and can be effectively used to detect tampering. In addition, the hash values are less sensitive to JPEG compression, low pass and high pass filtering that are content preserving operations. To reduce the size of the hash, a quantization scheme has also been proposed. It is believed that the information used by the quantization algorithm at the receiver's end does not leak information required to estimate the hash or the secret key. The experimental results reveal that balancing robustness and security is indeed very challenging. Increasing the robustness parameter has a negative impact on the security of the hash function. Likewise, increasing the security level will make the hash sensitive to non-malicious distortion that does not alter the content of multimedia data.

The basic motivation that was instrumental in undertaking the research work presented in Chapter 5 is due to the work of Craver and his co-workers towards developing counterfeit watermarking scheme [93]. The fundamental focus of the chapter was to secure the underlying watermarking scheme so that disputes arising between the seller and the buyer of a watermarked image can be resolved. A watermark is a piece of data that does not have association to anyone. It is with the help of a proper protocol, the watermarking technology can be used to solve real-life problems. Specifically, three issues of copyright protection have been focused, that is, buyer-seller identification, copyright infringement and verifica-

tion of ownership. The proposed algorithms and protocols have the capability to address several important issues in image watermarking such as buyer-seller identification, copyright infringement, resolving issues related to multiple watermark attacks, invertible watermark attacks and watermark removal attacks.

6.2 Future Directions

Image Encryption

1. The technique proposed in Chapter 3 performs encryption and decryption in the spatial domain. A disadvantage of the proposed technique is high computational cost and an increase in the memory requirement as the size of the image increases. To address this issue, it would be interesting to investigate energy compaction transform domain techniques, for example, DCT. Some initial work in this regard has been done by the author whose reference can be found in the author's list of publication.
2. Another interesting direction in the area of image encryption is to develop encryption and decryption algorithms that can work with print and scanned documents. This technology can be very useful in developing very low cost access control systems. An access control card can simply be made by first taking the digital photo of the person and then encrypting it. The encrypted pattern can then be printed on a piece of paper that will be attached to the ID card. When the person wants to enter the specific security zone, he/she will put the card on the scanner that will read the image, and decrypt it. The decrypted image can either be verified manually by comparing it with the actual face of the person, or it can be done using an automatic face recognition system. This solution will just require a high speed scanner and a printer. The challenge is to devise such an encryption scheme that can withstand the RST problem and distortion in the pixels during the print and scan operation.
3. Nearly all image encryption schemes proposed in the literature are sym-

metric, i.e. the same key is used at the encryption and decryption side. This means that the key is exposed once it is used. It will be a good break through if public-key image encryption schemes can be devised. Anyone with the public-key can encrypt an image, but it can only be decrypted using the corresponding private key.

Image Hashing

1. It is very difficult to come up with a single hashing scheme that can solve all the security issues and still have robustness to withstand non-malicious manipulations. Further, if the scheme is robust and has the capability to detect tampering, it should be free from all security loop holes that may help an attacker to create a forgery. For practical deployment of robust hash functions in multimedia authentication, we need such a scheme. Security is of paramount importance when we talk about systems that actually work in real-life scenarios. If a scheme has excellent robustness properties but is weak from the security's perspective, its practical deployment is not possible. If the multimedia community wants to see a robust hashing scheme working on gadgets, then they need to devise and standardize some algorithm that can perform in real-life systems. A collaborative research is required to formulate such a scheme.
2. Another important issue in designing a robust hash function is to come up with a hash function that meets the security and robustness requirements and in addition, its size should be a few hundred bits. Although the hash function proposed by Xie *et al.* [55] generates such a hash, however, its tamper detection capability is hampered by the fact that the hash bits are calculated by taking the mean of a block. There can be two perceptually different blocks with the same mean value. Hence there is a need to come up with a hash-based authentication scheme that generates a hash of very small size, say, 200-600 bits and still provide robustness to non-malicious operations, good tamper detection capability and a high level of security

against various attacks.

3. Another interesting research direction is to use the multimedia hash function towards generating a secret key derived from a biometric image that may be a fingerprint, for example. Some initial work in this regards was done by the author as mentioned in the author's list of publications. If such a robust hash function is developed, then it may be possible to come up with practical biometric-based digital signature systems.

Copyright Protection Schemes

1. There are many copyright protection schemes, including the one proposed in this thesis that uses a certification authority. It will be interesting to devise copyright protection schemes that do not require certification authority. When people work with multimedia data from across the planet, then practical implementation of copyright protection algorithms requiring a certification authority seems difficult.
2. In the area of watermarking for copyright protection and fingerprinting, a number of algorithms and protocols with good robustness and security characteristics have been proposed. However, we still do not find any standard that specifies how to use this technology in actual systems. Serious collaborative research is required in this area to come up with a single or a set of techniques that can be used practically.

Author's Publications

Book Chapters

1. Fawad Ahmed and M. Y. Siyal, A Secure and Robust Wavelet-Based Hashing Scheme for Image Authentication, *Advances in Multimedia Modeling*, LNCS 4352, pp. 51-62, 2007. (paper presented at MMM2007, Singapore, 2007)
2. M. Y. Siyal and Fawad Ahmed, Security Services and Issues in WLANs, *Handbook of Wireless Local Area Networks*, CRC Press, USA, 2005.

Journal Papers

1. Fawad Ahmed, M. Y. Siyal and Vali Uddin Abbas, "A Secure and Robust Hash-Based Scheme for Image Authentication," *Signal Processing*, Vol. 90, Issue 5, 2010.
2. Fawad Ahmed, Farook Sattar, M. Y. Siyal and Dan Yu, "A Secure Watermarking Scheme for Buyer-Seller Identification and Copyright Protection," *EURASIP Journal on Applied Signal Processing*, Vol. 2006, pp. 1-15.
3. Fawad Ahmed and M. Y. Siyal, "A Novel Approach for Regenerating a Private Key using Password, Fingerprint and Smart Card," *Information Management & Computer Security*, Vol. 13, No. 1, pp. 39-54, Emerald Group Publishing Limited, UK, 2005.

Under review

1. Fawad Ahmed, M. Y. Siyal and Vali Uddin Abbas, "A JPEG Compression Tolerant Image Encryption Scheme", *Signal Processing: Image Communication*.

Conclusion and Future Directions

Conference Papers

1. Fawad Ahmed, M. Y. Siyal and Vali Uddin Abbas, "A Perceptually Scalable and JPEG Compression Tolerant Image Encryption Scheme", accepted: PSVIT 2010, Singapore.
2. Fawad Ahmed, M. Y. Siyal and Vali Uddin Abbas, "A Robust Image Encryption Scheme using State-Feedback Control," *IEEE Int. Conf. Computer, Control & Communication*, Karachi, Pakistan, 2009.
3. Fawad Ahmed and M. Y. Siyal, "Image Authentication using Soft Hashing Technique," *Int. Conf. Information, Communication and Signal Processing*, Singapore, 2007.
4. Fawad Ahmed and M. Y. Siyal, "A Robust and Secure Signature Scheme for Video Authentication," *IEEE Int. Conf. Multimedia and Expo*, China, July, 2007.
5. Fawad Ahmed and M. Y. Siyal, "A Novel Hashing Scheme for Image Authentication," *Innovation in Information Technology*, Dubai, 2006.
6. Fawad Ahmed and M. Y. Siyal, "A Secure and Robust DCT-Based Hashing Scheme for Image Authentication," *IEEE Int. Conf. Communication Systems*, Singapore, 2006.
7. Fawad Ahmed and M. Y. Siyal, "A Secure and Robust Hashing Scheme for Image Authentication," *Int. Conf. Information, Communication and Signal Processing*, 2005, Thailand.
8. Fawad Ahmed and M. Y. Siyal, "A Hybrid-Watermarking Scheme for Asymmetric and Symmetric Watermark Extraction," *9th IEEE Multi-Topic Conference*, Karachi, Pakistan, 2005.
9. M. Y. Siyal and Fawad Ahmed, "A Biometric-Based Scheme for Enhancing Security of Cryptographic Keys," *Proc. IEEE TENCON*, Vol. B, pp. 407-410, Thailand, Nov 2004.

Conclusion and Future Directions

10. M. Y. Siyal and Fawad Ahmed, "A Novel Approach for Generating Digital Signature using Fingerprint, Password and Smart Card," *Proc. Int. Workshop on Frontiers of Information Technology*, Islamabad, Pakistan, 2003.

Bibliography

- [1] Announcing the ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standards Publication 197, 2001. (<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>)
- [2] R. A. Mollin, An Introduction to Cryptography, CRC Press, Boca Raton, FL., 2006.
- [3] S. A. Vanstone, A. J. Menezes, and P. C. Oorschot, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL., 1996.
- [4] B. Schneier, Applied Cryptography, John Wiley & Sons, Inc., USA, 1996.
- [5] W. Zeng, H. Yu and C. -Y Lin, Multimedia security technologies for digital rights managements, Elsevier Inc. , 2006.
- [6] B. Furht and D. Kirovski, Multimedia Security Handbook, CRC Press, Boca Raton, FL., 2005.
- [7] S. Lian, Multimedia content encryption: techniques and applications, CRC Press, Boca Raton, FL., 2008.
- [8] I. J. Cox, M.L. Miller, and J.A. Bloom, Digital Watermarking, Morgan Kaufmann Publishers, Inc., San Francisco, 2001.
- [9] M. Yang, N. Bourbakis and S. Li, "Data-Image-Video Encryption," *IEEE Potentials*, pp. 28-34, 2004.
- [10] R. Radhakrishnan, Z. Xiong and N. Memon, "On the Security of the Visual Hash Function," *Security and Watermarking of Multimedia Contents V*,

BIBLIOGRAPHY

- Edward J. Delp III, Ping Wah Wong (Eds)*, SPIE Vol. 5020, pp. 644-652, 2003.
- [11] C.Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 11, No. 2, pp. 1453-168, 2001.
 - [12] Y. Matias and A. Shamir, "A video scrambling technique based on space filling curves," *Proc. on Advances in Cryptology - CRYPTO'87*, Lecture Notes in Computer Science, Vol. 293, pp. 398-417, 1987.
 - [13] CENELEC (European Committee for Electrotechnical Standardization). Access control system for the MAC/packet family, *EUROCRYPT*, European Standard EN 50094. Brussels, 1992.
 - [14] J. Scharinger, "Kolmogorov systems: Internal time, irreversibility and cryptographic applications," *Proc. of the AIP Conference on Computing Anticipatory Systems*, Vol. 437, Ed. D. Dubois, Woodbury, NY, 1998.
 - [15] S. Li, C. Li, G. Chen and N. G. Bourbakis, "A general cryptanalysis of permutation-only multimedia encryption algorithms," *IACRs Cryptology ePrint Archive*, Report 2004/374, 2004.
 - [16] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. Journal*, Vol. 28, pp. 656-715, 1949.
 - [17] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," *Proc of the ACM Multimedia*, pp. 219-229, Boston, MA., 1996.
 - [18] L. Qiao and K. Nahrstedt, "Comparison of MPEG encryption algorithm," *Int. Journal on Computers and Graphics*, Vol. 22, No. 4, pp. 437-448, 1998.
 - [19] G. A. Spanos and T. B. Maples, "Performance study of a selective encryption scheme for the security of networked, real-time video," *Proc. Int. Conf. Computer Communication and Networking*, 1995.

BIBLIOGRAPHY

- [20] T. Lookabaugh and D. C. Sicker, "Selective Encryption for Consumer Applications," *IEEE Communication Magazine*, pp. 124-129, 2004.
- [21] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Processing*, Vol. 48, No. 8, pp. 2439-2451, 2000.
- [22] A Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE. Trans. Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, 1996.
- [23] C. -P Wu and C. -C. J. Kuo, "Design of integrated multimedia compression and encryption system," *IEEE. Trans. Multimedia*, Vol. 7, No. 5, pp. 828-839, 2005.
- [24] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," *Proc. 6th ACM Int. Conf. on Multimedia*, 1998.
- [25] C. Kuo and M. Chen, "A new signal encryption technique and its attack study," *Proc. IEEE Int. Carnahan Conf. on Security Technology*, pp. 149-153, 1991.
- [26] J. Wen, M. Muttirell and M. Severa, "Access control of standard video bitstreams," *Proc. Int. Conf. Media Future*, Italy, 2001.
- [27] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Trans. Image Processing*, Vol. 15, No. 7, 2006.
- [28] M. V. Droogenbroeck and R. Benedett, "Techniques for selective encryption of uncompressed and compressed images," *Proc. Advanced Concepts for Intelligent Vision Systems*, Ghent, 2002.
- [29] M. Podesser, H. -P. Schmidt and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," *IEEE Nordic Signal Processing Symposium*, pp. 90-97, Norway, 2002.

BIBLIOGRAPHY

- [30] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Trans. Multimedia*, Vol. 1, No. 1, pp. 118-129, 2003.
- [31] A. Said, "Measuring the strength of partial encryption schemes," *Proc. IEEE Int., Conf. Image Processing*, Vol. 2, pp. 1126-1129, 2005.
- [32] M. Grangetto, E. Enrico and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Trans. multimedia*, Vol. 8, No. 5, pp. 905-917, 2006.
- [33] P. W. Moo and X. Wu, "Resynchronization properties of arithmetic coding," *Proc. IEEE Int. Conf. Image Processing*, 1999.
- [34] D. V. D. Ville, W. Phillips, R. V. de Walle and I. Lemanhieu, "Image scrambling without bandwidth expansion," *IEEE Trans. Circuits and Systems Video Technology*, Vol. 14, No. 6, pp. 892-897, 2004.
- [35] A. D. Wyner, "An analog scrambling scheme which does not expand bandwidth, Part I: Discrete time," *IEEE. Trans. Information Theory*, Vol. 25, No. 3, pp. 261-274, 1979.
- [36] S. Li, C. Li, K. -T. Lo and G. Chen, "Cryptanalysis of an image scrambling scheme without bandwidth expansion," *IEEE Trans. Circuits and System Video Technology*, Vol. 18, No. 3, pp. 338-349, 2008.
- [37] A. M. Alattar and G. I. A. -Regib, "Evaluation of selective encryption techniques for secure transmission of MPEG-Compressed bit-streams," *IEEE Int. Symposium on Circuits and Systems*, Vol. 4, pp. 340-343, 1999.
- [38] A. Pommer and A. Uhl, "Selective encryption of wavelet-packet encoded image data: efficiency and security," *Multimedia System*, Vol. 9, pp. 279-287, 2003.
- [39] W. Puech, J. M. Rodrigues and A. G. Bors, "Analysis and cryptanalysis of a selective encryption method for JPEG images," *8th Int. Workshop Image Analysis for Multimedia Interactive Services*, 2007.

BIBLIOGRAPHY

- [40] D. Engel and A. Uhl, "An attack against image-based selective bitplane encryption," *Proc. Int. Conf. Image Processing*, Vol. II, pp. 141-144, 2007.
- [41] D. Engel, E. Pschernig and A. Uhl, "An analysis of lightweight encryption schemes for fingerprint images," *IEEE Trans. Information Forensics and Security*, Vol. 3, No. 2, 2008.
- [42] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, Vol. 1, No. 3, pp. 6-21, 2001.
- [43] Y. Mao and G. Chen, Chaos-based image encryption, in *Handbook of Geometric Computing*, pp. 231-265, Springer Berlin Heidelberg, 2005.
- [44] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation and Chaos*, Vol. 16, No. 8, pp. 2129-2151, 2006.
- [45] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physics Letters A*, Vol. 64, pp. 821-824, 1990.
- [46] J. Yen and J. Guo, "A new chaotic key-based design for image encryption and decryption," *Proc. IEEE Int. Symposium Circuits and Systems*, Vol. 4, pp. 49-52, Geneva, Switzerland, 2000.
- [47] S. Li and X. Zheng, "Cryptanalysis of a chaotic image encryption method," *IEEE Int. Symposium Circuits and Systems*, Scottsdale, Vol. 2, pp. 708-711, 2002.
- [48] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation and Chaos*, Vol. 8, No. 6, pp. 1259-1284, 1998.
- [49] S. Li, G. Chen and X. Zheng, Chaos-based encryption for digital images and video, B. Furht and D. Kirovski Eds., *Multimedia Security Handbook*, Ch. 4, pp. 133-167, CRC Press, 2004.

BIBLIOGRAPHY

- [50] G. Álvarez, F. Montoya, M. Romera and G. Pastor, "Cryptanalysis of a discrete chaotic cryptosystem using external key," *Physics Letters A*, Vol. 306, pp. 334-339, 2003.
- [51] G. Hu, Z. Feng and R. Meng, "Chosen ciphertext attack on chaos communication based on chaotic synchronization," *IEEE. Trans. Circuits and Systems*, Vol. 50, No. 2 pp. 275-279, 2003.
- [52] S. Li, C. Li, G. Chen and K. -T. Lo, "Cryptanalysis of RCES/RSES Image Encryption Scheme," *J. of Systems and Software*, Vol. 81, No. 7, pp. 1130-1143, 2008.
- [53] G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 4, pp. 905-910, 1993.
- [54] M. Schneider and S. F. Chang, "A content based digital signature for image authentication," *Int. Conf. Image Processing*, Lausanne, Switzerland, pp. 227-230, 1996.
- [55] L. Xie , G.R. Arce G. R. and R.F. Graverman, "Approximate message authentication codes," *IEEE Trans. Multimedia*, Vol. 3, No. 2, pp. 242-252, 2001.
- [56] R. F. Gravemen and K. Fu, "Approximate message authentication codes," *Proc. 3rd Annual Fedlab Symp. Advanced Telecommunications Information Distribution*, Vol. 1, College Park, MD, 1999.
- [57] D.-C. Lou and J.-L. Liu, "Fault resilient and compression tolerant signature for image authentication," *IEEE Trans. Consumer Electronics*, Vol. 46, No. 1, pp. 31-39, 2000.
- [58] M. P. Queluz, "Content-based integrity protection of digital images," *Security and Watermarking of Multimedia Contents*, Ping Wah Wong; Edward J. Delp; Eds. Vol. 3657, pp. 85-93, 1999.

BIBLIOGRAPHY

- [59] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," *Proc. Int. Conf. Image Processing*, pp. 435-438, Chicago, USA, 1998.
- [60] J. Fridrich and M. Goljan, "Robust Hash Functions for Digital Watermarking," *Proc. Int. Conf. Information Technology: Coding and Computing*, pp. 178-183, 2001.
- [61] R. Radhakrishnan, N. Memon, "On the Security of the Digest Function in the SARI Image Authentication System," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 12, No. 11, pp. 1030-1033, 2002.
- [62] T. Uehara and R. Safavi-Naini, "On (In) security of a robust image authentication method," *Y.-C. Chen et al. Eds, LNCS 2532*, pp. 1025-1032, 2002.
- [63] Q. Sun and S.-F. Chang, "A Robust and Secure Media Signature Scheme for JPEG Images," *Journal of VLSI Signal Processing*, 41, pp. 305-317, 2005.
- [64] C.S. Lu, H.-Y. M. Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme," *IEEE Trans. on Multimedia*, Vol. 5, No. 2, pp. 161-173, 2003.
- [65] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust Image Hashing," *Proc. Int Conf. Image Processing*, 2000.
- [66] F. Lefebvre, J. Czyz, and B. Macq, "A Robust Soft Hash Algorithm for Digital Image Signature," *Proc. Int. Conf. on Image Processing*, Vol. 2, pp. 495-498, Spain, 2003.
- [67] V. Monga and B. L. Evans, "Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs," *IEEE Trans. on Image Processing*, Vol. 15, No. 11, 2006.

BIBLIOGRAPHY

- [68] A. Swaminathan, Y. Mao and Min Wu, "Image Hashing Resilient to Geometric and Filtering Operations," *IEEE 6th Workshop on Multimedia Signal Processing*, pp. 355-358, 2004.
- [69] V. Monga and M.K. Mihcak, "Robust Image Hashing Via Non-Negative Matrix Factorizations," *IEEE Int. Conf. on Acoustic Speech and Signal Processing*, pp. 225-228, 2006.
- [70] X. Lv and Z. J. Wang, "Fast Johnson-Lindenstrauss transform for Robust and secure image hashing," *10th IEEE Workshop on Multimedia and Signal Processing*, Cairns, pp. 725-729, Australia, 2008.
- [71] A. Swaminathan, Y. M. Mao and M. Wu, "Image hashing resilient to geometric and filtering operations," *IEEE 6th Workshop on Multimedia Signal Processing*, pp. 355-358, Italy, 2004.
- [72] C.-S. Lu and C.-Y. Hsu, "Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication," *Multimedia Systems*, Vol. 11, No. 2, 159-173, 2005.
- [73] C.-S. Lu, S.-W. Sun, C.-Y. Hsu and P.-C. Chang, "Media hash-dependent image watermarking resilient against both geometric attacks and estimation attacks based on false positive-oriented detection," *IEEE Trans. Multimedia*, Vol.8, No. 4, 668-685, 2006.
- [74] C. Deng, X. Gao, D. Tao and X. Li, "Geometrically invariant watermarking using affine covariant regions," *Int. Conf. Image Processing*, pp. 413-416, USA, 2008.
- [75] I. J. Cox, J. Kilian, F. T. Leighton, and T. G. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, Vol. 6, No. 12, pp. 1673-1687, 1997.
- [76] T. Furon, I Venturini and P. Duhamel, "A Unified Approach of Asymmetric Watermarking Schemes," *Security and Watermarking of Multimedia Contents, III, SPIE*, 2001.

BIBLIOGRAPHY

- [77] T. Furon and P. Duhamel, "An Asymmetric Watermarking Method," *IEEE Transactions on Signal Processing*, Vol. 51, No. 4, pp. 981-995, 2003.
- [78] J.J. Eggers, J.K Su, and B. Girod, "Public Key Watermarking by Eigenvectors of Linear Transforms," *European Signal Processing Conference*, 2000.
- [79] H. Choi, K. Lee, and T. Kim, "Transformed Key Asymmetric Watermarking System," *IEEE Signal Processing Letters*, Vol. 11, No. 2, pp. 251-254, 2004.
- [80] M.L. Miller, "Is Asymmetric Watermarking Necessary or Sufficient?," *Proceedings of the European Signal Processing Conference*, EUSIPCO 2002, France, 2002.
- [81] P.W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification," *IEEE Trans. Image Processing*, Vol. 10, No. 10, pp. 1593-1601, 2001.
- [82] T. Chen, C. Huang, T. Chen and C. Liu, "Authentication of lossy compressed video data by semi-fragile watermarking," *Int. Conf. Image Processing*, Singapore, 2004.
- [83] E. Delp and R. Wolfgang, "A watermark for digital images," *Int. Conf. Image Processing*, pp. 219-222, Switzerland, 1996.
- [84] W. Zhu, Z. Xiong and Y. Q. Zhang, "Multiresolution watermarking for images and video: a unified approach," *Proc. Int. Conf. Image Processing*, Vol. 1, pp. 465-468, 1998.
- [85] M. Ramkumar, A. N. Akansu and A. A. Alatan, "A robust data hiding scheme for images using DFT," *Proc. Int. Conf. Image Processing*, Vol. 2, pp. 211-215, 1999.
- [86] J. J. K. O. Ruanaidh, W. J. Dowling and F. M. Boland, "Phase watermarking of digital images," *Proc. Int. Conf. Image Processing*, Vol. 3, pp. 239-242, 1996.

BIBLIOGRAPHY

- [87] F. Hartung, J. K. Su, and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counterattacks," *SPIE Conference on Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 147-158, 1999.
- [88] I. J. Cox and J.-P. M.G. Linnartz, "Some general methods for tampering with watermarks," *IEEE Journal on Selected Areas of Communications*, Vol. 16, No. 4, pp. 587-593, 1998.
- [89] M. Costa, "Writing on dirty paper," *IEEE Trans. Information Theory*, Vol. 29, No. 3, pp. 439-441, 1983.
- [90] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Information Theory*, Vol. 47, No. 4, 1423-1443, 2000.
- [91] N. Memon and P. W. Wong, "A Buyer-Seller Watermarking Protocol," *IEEE Transactions on Image Processing*, Vol. 10, No. 4, pp. 643-649, 2001.
- [92] M. Ramkumar and A.N. Akansu, "A Robust Protocol for Proving Ownership of Multimedia Content," *IEEE Transactions on Multimedia*, Vol. 6, No. 3, pp. 469-478, 2004.
- [93] S. Craver, N. Memon, B.-L. Yeo and M.M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks and Implications," *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 4, pp. 573-586, 1998.
- [94] S. Katzenbeisser, "On the design of copyright protection protocols for multimedia distribution using symmetric and public-key watermarking," *12th International Workshop on Database and Expert Systems*, pp. 815-819, 2001.
- [95] F. Frattolillo, "Watermarking protocol for web context," *IEEE Trans. Information Forensics and Security*, Vol. 2, No. 3, 2007.
- [96] C. -L. Lei, P. -L. Yu, P. -L. Tsai and M. -H Chan, "An efficient and anonymous buyer-seller watermarking protocol," *IEEE Trans. Image Processing*, Vol. 13, No. 12, 2004.

BIBLIOGRAPHY

- [97] T. -H. Chen and G. Horng, "A lightweight and anonymous copyright-protection protocol," *Computer and Standards*, Vol. 29, pp. 229-237, 2007.
- [98] Shujun Li, "Analyses and New Designs of Digital Chaotic Ciphers," Ph.D. Thesis, Xian Jiaotong University, China, 2003.
- [99] S. Sudharsanan, "Shared key encryption of JPEG color images," *IEEE Trans. on Consumer Electronics*, Vol. 51, Issue 4, pp. 1204-1211, 2005.
- [100] J. M. Rodrigues, W. Puech and A.G. Bors, "Selective Encryption of Human Skin in JPEG Images," *Int. Conf. Image Processing*, Atlanta, GA, pp. 1981-1984, 2006.
- [101] B. Yang, C.-Q. Zhou, C. Busch and X.-M. Niu, "Transparent and perceptually enhanced JPEG image encryption," *Int. Conf. Digital Signal Processing*, Santorini-Hellas, pp. 1-6, 2009.
- [102] X. Niu, C. Zhou, J. Ding and B. Yang, "JPEG encryption with file size preservation," *Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, Harbin, pp. 308-311, 2008.
- [103] L. Krikor, S. Baba, T. Arif and Z. Shaaban, "Image encryption using DCT and stream cipher," *European Journal of Scientific Research*, Vol. 32, No. 1, pp. 47-57, 2009.
- [104] D. C. Lay, *Linear algebra and its applications*, Pearson Education, 2009.
- [105] C. D. Meyer, *Matrix analysis and applied linear algebra*, SIAM, 2000.
- [106] H. Andrews, C. L. Patterson, "Singular value decomposition and digital image processing," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-24, No. 1, pp. 26-53, 1976.
- [107] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE. Trans. multimedia*, Vol. 4, No. 1, pp. 121-128, 2002.

BIBLIOGRAPHY

- [108] M. Ghaderpanah and A. B. Hamza, "Secure copyright protection of digital images using nonnegative matrix factorization," *23rd Biennial Symposium on Communication*, pp. 344-347, 2006.
- [109] M. -S. Koh, E. R. -Marek and C. Talarico, "A novel data dependent multimedia encryption algorithm secure against chosen-plaintext attacks," *Int. Conf. Multimedia and Expo*, pp. 1527-1530, 2007.
- [110] D.C. Montgomery and G.C. Runger, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, Inc., Third edition, 2004.
- [111] R.E. Walpole and R.H. Myers, *Probability and Statistics for Engineers and Scientists*, Maxwell Macmillan International Editions, 1990
- [112] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw Hill, 2nd Edition, 1987.
- [113] W. Li, Y. Yuan and N. Yu, "Passive detection of doctored JPEG image via block artifact grid extraction," *Signal Processing*, Article in press.
- [114] F. Ahmed and M. Y. Siyal, "A secure and robust wavelet-based hashing scheme for image authentication," *Advances in Multimedia Modeling*, LNCS 4352, 51-62, 2007.
- [115] F. Ahmed and M. Y. Siyal, "Image authentication using soft hashing technique," *Proc. 26th Int. Conf. Information, Communication and Signal Processing*, pp. 1-5, Singapore, 2007.
- [116] B. Coskun and N. Memon, "Confusion/Diffusion capabilities of some robust hash functions," *Proc. 40th Annual Conf. on Information Sciences and Systems*, pp. 1188-1193, Princeton NJ, 2006.
- [117] W. Stallings, *Cryptography and Network Security: Principles and Practices*, Pearson Education, Inc., Chapter 6, pp. 191-194, 2006.

BIBLIOGRAPHY

- [118] S. Katzenbeisser, “On the Integration of Cryptography and Watermarks,” *International Workshop on Digital Watermarking*, Springer Lecture Notes in Computer Science, Vol. 2939, pp. 50-60, 2003.
- [119] R. L Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public- Key Cryptosystems,” *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.
- [120] A. Hyvarinen, “Survey on Independent Component Analysis,” *Neural Computing Surveys*, Vol. 2, pp. 94–128, 1999.
- [121] T.-W. Lee, *Independent Component Analysis—Theory and Applications*, Kluwer Academic Publishers, 1998.
- [122] I. T. Jolliffe, *Principal Component Analysis*, Springer Verlag, 2002.
- [123] S. E. Noel and H. H. Szu, “Multimedia Authenticity with ICA Watermarks,” *Proceedings of SPIE, Wavelet Applications VII*, Vol. 4056, pp. 175-184, 2000.
- [124] D. Yu and F. Sattar, “A New Blind Watermarking Technique Based on Independent Component Analysis,” *International Workshop on Digital Watermarking*, Springer Lecture Notes in Computer Science, Vol. 2613, pp. 51-63, 2003.
- [125] F. Mintzer and G.W. Braudaway, “If One Watermark is Good, Are More Better?,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2067-2069, 1999.
- [126] J.-F. Cardoso, “High-Order Contrasts for Independent Component Analysis,” *Neural Computation*, Vol. 11, pp. 157-192, 1999.
- [127] S. Burnett and S. Paine, *RSA Security’s Official Guide to Cryptography*, Osborne/McGraw-Hill, CA, 2001.
- [128] N. Koblitz, *Algebraic Aspects of Cryptography, Algorithms and Computation in Mathematics*, Vol. 3, Springer, 1998.