ABSTRACT

A DIGITAL SIGNATURE AND WATERMARKING BASED AUTHENTICATION SYSTEM FOR JPEG2000 IMAGES

by
Rui Jing

In this thesis, digital signature based authentication system was introduced, which is able to protect JPEG2000 images in different flavors, including fragile authentication and semi-fragile authentication. The fragile authentication is to protect the image at code-stream level, and the semi-fragile is to protect the image at the content level.

The semi-fragile can be further classified into lossy and lossless authentication. With lossless authentication, the original image can be recovered after verification. The lossless authentication and the new image compression standard, JPEG2000 is mainly discussed in this thesis.

A DIGITAL SIGNATURE AND WATERMARKING BASED AUTHENTICATION SYSTEM FOR JPEG2000 IMAGES

by
Rui Jing

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 2005

APPROVAL PAGE

A DIGITAL SIGNATURE AND WATERMARKING BASED AUTHENTICATION SYSTEM FOR JPEG2000 IMAGES

Rui Jing

Dr. Yun-Qing Shi, Thesis Advisor Date
Professor, NJIT

Dr. Richard Haddad, Committee Member Date
Professor, NJIT

Dr. Sirin Tekinay, Committee Member Date
Associate Professor, NJIT

BIOGRAPHICAL SKETCH

Author: Rui Jing

Degree: Master of Science

Date: January 2005

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
New Jersey Institute of Technology, Newark, NJ, 2005

- Bachelor of Science in Electrical Engineering,
Southeast University, Nanjing, P.R.China, 1996

Major: Electrical Engineering

To my beloved family

ACKNOWLEDGEMENT

First, I would like to thank my advisor, Dr. Yun-Qing Shi for his encouragement and support. His guidance on choosing topics and solving problems has helped me significantly. I also appreciate his great patience in correcting the thesis.

I thank my other committee members, Dr. Richard Haddad and Dr. Sirin Tekinay for taking time out of their busy schedules to provide help and constructive criticism.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

The real world is becoming a digital world. It is well-known that the storage space that people need exceeds the capability of the development of hardware in semiconductor and computer. Data compression has been found to be necessary in many areas, especially, these applications relative to image or video.

For example, the FBI is digitizing the nation's fingerprint database at 500 dots per inch with 8 bits of grayscale resolution. A sample fingerprint image measuring is 768 x 768 pixels (= 589,824 bytes). At this rate, a single fingerprint card turns into about 10 MB of data! The FBI has been collecting fingerprint cards since 1924, and because (like most of us) they find it hard to throw things out, over the past 70 years their collection has grown to over 200 million cards occupying an acre of filing cabinets in the J. Edgar Hoover building back in Washington. This includes some 29 million records they examine each time they're asked to round up the usual suspects. Therefore, the new image compression standard is needed [1].

Compressing an image is significantly different than compressing raw binary data. Of course, general-purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties, which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area. Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is the case when binary data such as executables, documents etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable.

1.2 Outline of the Thesis

The main topics of this thesis are:

1. JPEG2000 still image compression standard
2. Digital watermarking
3. Authentication system for JPEG2000 images

The first topic is covered by Chapter 2 and 3, while the last two topics are covered by Chapter 4 and 5. The chapter outlines are as follows:

Chapter 2: The comprehensive standards of JPEG2000 from ISO 15444 or ITU-T Recommendation T.800 are being issued in twelve parts. JPEG2000 fundamental building blocks are introduced. DWT has replaced the DCT in JPEG 2000. Filter normalization is expressed by the DC gain of the low-pass analysis filter h_0, and the Nyquist gain of the high-pass analysis h_1. JPEG2000 organizes the compressed data from the codeblocks into packets or layers.

Chapter 3: To address this issue, JPEG 2000 Secured (JPSEC) or Part 8 of the standard is standardizing tools and solutions in terms of specifications. JPEG2000 protects the content including digital signatures, watermarking, encryption and scrambling. Main security marker has two kinds of tools: template protection tool and registration authority protection tool.

Chapter 4: Digital watermarks are designed to be completely invisible. Digital watermarking and steganography have the common feature: describing methods to embed information transparently into a carrier signal.

Chapter 5: It will describe the proposed authentication system for JPEG2000 that provides integrity protection and source authentication services for JPEG2000 images. As explained in previous section, fragile authentications protect images at code stream level. In our system, the semi-fragile authentication has four sections: feature extraction, Error Correction Coding (ECC) of feature, signature generation and data embedding.

The conclusion is presented by Chapter 6.

CHAPTER 2

AN OVERVIEW OF THE JPEG2000 STILL IMAGE COMPRESSION STANDARD

2.1 Introduction

The JPEG committee began to investigate possibilities for a new still image compression standard to serve current and future applications.

JPEG 2000 is a new image coding system, it uses state-of-the-art compression techniques based on wavelet technology. Its architecture should lend itself to a broad range of uses from portable digital cameras through to advanced pre-press, medical imaging and other key sectors.

The comprehensive standards of JPEG2000 from ISO 15444 or ITU-T Recommendation T.800 is being issued in twelve parts. JPEG 2000 refers to all parts of the standard: Part 1 (the core) is now published as an International Standard, five more parts (2-6) are complete or nearly complete, and four new parts (8-11) are under development [2]. The parts are:

- Part 1, Core coding system
- Part 2, Extensions (adds more features and sophistication to the core)
- Part 3, Motion JPEG 2000
- Part 4, Conformance
- Part 5, Reference software (Java and C implementations are available)
- Part 6, Compound image file format (document imaging, etc.)
- Part 7 has been abandoned

- Part 8, JPSEC (security aspects)
- Part 9, JPIP (interactive protocols and API)
- Part 10, JP3D (volumetric imaging)
- Part 11, JPWL (wireless applications)
- Part 12, ISO Base Media File Format (*common with MPEG-4*)

This chapter mainly discusses part 1 of JPEG 2000, which defines the core of JPEG 2000. These include the syntax of the JPEG 2000 codestream and the necessary steps involved in encoding and decoding JPEG 2000 images. The later parts of the standard are all concerned with extensions of various kinds, and none of them is essential to a basic JPEG 2000 implementation.

Part 1 also defines a basic file format called JP2. This allows metadata such as color space information to be included with a JPEG 2000 codestream in an interoperable way. JP2 uses an extensible architecture shared with the other file formats in the JPEG 2000 family defined in later parts of the standard.

2.2 JPEG2000 Fundamental Building Blocks

These components include pre-processing, Digital Wavelet Transformation (DWT), quantization, arithmetic coding (tier-1 coding), and bit-stream organization (tier-2 coding). This section will discuss each of these components in more details. The fundamental building blocks of a typical JPEG2000 encoder are shown in Figure 2.1.

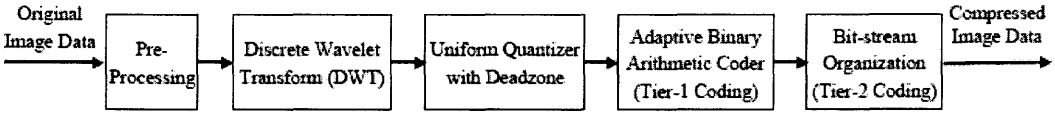
Original Image Data
Pre-Processing
Discrete Wavelet Transform (DWT)
Uniform Quantizer with Deadzone
Adaptive Binary Arithmetic Coder (Tier-1 Coding)
Bit-stream Organization (Tier-2 Coding)
Compressed Image Data

Figure 2.1 JPEG2000 fundamental building blocks.

2.2.1 Pre-processing

Images have different kinds of size, we usually need to process these images to a suitable shape in order to easily process these images in the following steps, we call it pre-processing, which is separable to several steps.

The first step, it is to partition the original image into rectangular and non-overlapping tiles of equal size. The size of the tile is different from each other, it can be as larger as the original image, or it can be smaller as one pixel. Every tile can be processed by their compression method.

The second step, in every component, unsigned values are level shifted by subtracting a fixed value of 2^{B-1} from every sample to make its value symmetric around zero. Signed sample valuesare not level shifted.

Finally, one of two transformation choices was utilized. One transformation is the irreversible color transformation (ICT), the other is the reversible color transform (RCT).

Irreversible Color Transformation:

The same pixel can also be represented by the three values Y', C_B, and C_R. Y' corresponds to the perceived brightness of the pixel, which is independent of the hue of the pixel. All the hue information (*chroma*) is held by the C_B and C_R values. These are also called the "color difference" channels, because C_B is related to (B' - Y') and C_R is related to (R' - Y').

Irreversible color transformation is identical to the traditional red-green-blue (RGB) to YC_bC_r color transformation and it is used in lossy coding area. The following is the definition of forward ICT:

where both transforms operate on the first three components of an image tile with the implicit assumption that these components correspond to RGB.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.500 \\ 0.500 & -0.41869 & -0.08131 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \qquad (2.1)$$

The inverse ICT is:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \times \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}. \qquad (2.2)$$

Reversible Color Transformation:

Reversible color transformation is a reversible integer-to-integer thansform, and it can be used for lossy and lossless coding. The following is both forward and inverse RCT:

$$Y = \lfloor (R + 2G + B)/4 \rfloor,\ U = R - G,\ V = B - G, \qquad (2.3)$$

$$G = Y - \lfloor (U + V)/4 \rfloor,\ R = U + G,\ B = V + G, \qquad (2.4)$$

2.2.2 The Discrete wavelet transformation (DWT)

In JPEG 2000, DWT has replaced the DCT. First, we consider one-dimension DWT, and then extend the concepts to two-dimensions DWT [3].

2.2.2.1 The One-dimension DWT. The forward one-dimension DWT can be defined as a continuous application of a pair of high-pass and low-pass filters, and then followed by a downsampling by a factor of 2. It shows in Figure 2.2. The analysis-bank consists of a pair of low-pass and high-pass filters, and downsampling by a factor of 2.

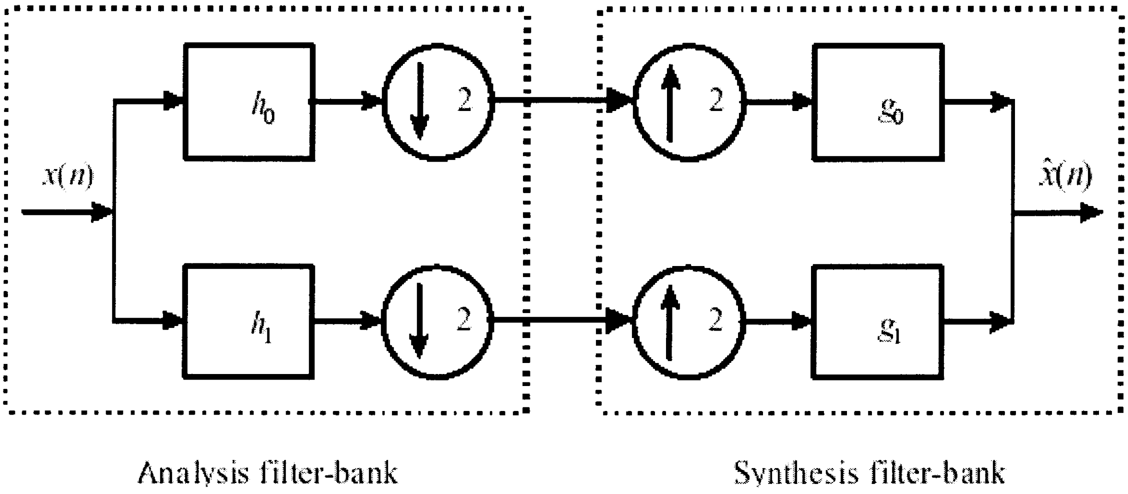

Figure 2.2 1-D, 2-band wavelet analysis and synthesis filter-bank.

A one-dimension singal x(n) passes the analysis filter-bank, which is reference to as a (5,3) filter-bank. A analysis low-pass filter is $h_0(n)$, such as, $h_0(n)$= (-1 2 6 2 -1)/8, which is a symmetric and has five integer taps. The analysis high-pass filter is $h_1(n)$, for example, $h_1(n)$= (-1 2 -1)/2, which is a symmetric and three integer taps. The reconstructed signal x' (n) will be recovered by the synthesis filter-bank. The synthesis low-pass filter is presented by $g_0(n)$, and the synthesis high-pass filter is expressed by $g_1(n)$ [3, 4].

If $h_0(n)$ and $h_1(n)$ have already been known, we must can find a suitable $g_0(n)$ and $g_1(n)$ that is suitable for receiver to decode the received signal. $h_0(n)$ and $h_1(n)$, $g_0(n)$ and $g_1(n)$ must satisfied Equation 2.5 and Equation 2.6.

$$H_0(z)G_0(z)+H_1(z)G_1(z)=2, \qquad (2.5)$$

$$H_0(-z)G_0(z)+H_1(-z)G_1(z)=0, \qquad (2.6)$$

Here, H_0 (z) is the Z-transform of h_0 (n), G_0 (z) is the Z-transform of g_0 (n), H_1 (z) is the Z-transform of h_1 (n), $G_1(z)$ is the Z-transform of g_1 (n).

The following is the typical filter-bank. Table 2.1 is the most well-known Daubechies (9,7) filter bank. Table 2.2 is the integer (5,3) filter-bank.

Table 2.1 Analysis and Synthesis High-pass Filter Taps for Floating Point Daubechies (9,7) Filter-bank

n	Low-pass, $h_0(n)$	Low-pass, $g_0(n)$
0	+0.602949018236360	+1.115087052457000
±1	+0.266864118442875	+0.591271763114250
±2	-0.078223266528990	-0.057543526228500
±3	-0.016864118442875	-0.091271763114250
±4	+0.026748757410810	

n	High-pass, $h_1(n)$
-1	+1.115087052457000
-2, 0	-0.591271763114250
-3, 1	-0.057543526228500
-4, 2	+0.091271763114250

n	High-pass, $g_1(n)$
1	+0.602949018236360
0,2	-0.266864118442875
-1, 3	-0.078223266528990
-2, 4	+0.016864118442875
-3, 5	+0.026748757410810

Table 2.2 Analysis and Synthesis Filter Taps for the Integer (5,3) Filter-bank

n	$h_0(n)$	$g_0(n)$
0	3/4	+1
±1	1/4	+1/2
±2	-1/8	

n	$h_1(n)$
-1	+1
-2, 0	-1/2

n	$g_1(n)$
1	+3/4
0, 2	-1/4
-1, 3	-1/8

2.2.2.2 The Two-dimension DWT. The one-dimension DWT was discussed in the last section. Now we will discuss the two-dimension DWT. The one-dimension can be easily extended to two-dimension by applying the filter-bank Figure 2.3 shows a 3-level, 2-D dyadic decomposition and the corresponding labeling for each subband. Each time, the original image can be [6] decomposed into four subimages after one wavelet decomposition. Here, 1HH, 1HL, 1LH, 1LL subband images were got after one-level wavelet decomposed. After the second-level wavelet decomposition, we decomposed the 1LL into other four subband images: 2HH, 2HL, 2LH, 2LL. After the third-level wavelet

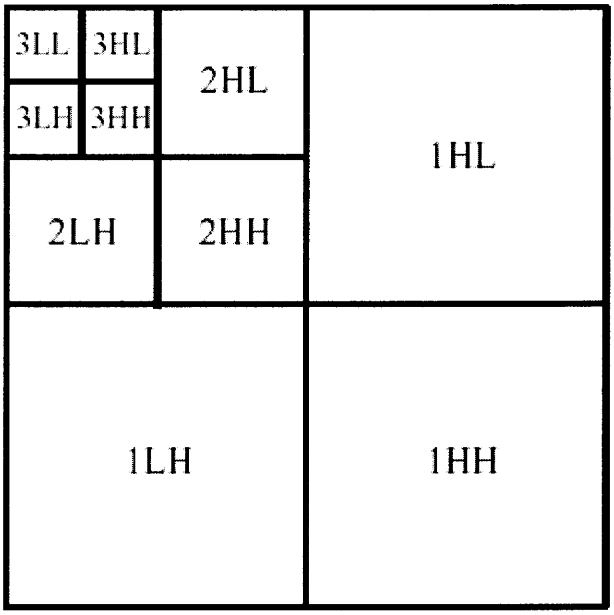

Figure 2.3 Two-dimensions, 3-level wavelet decomposition.

decomposition, we decomposed the 2LL subband image into 3HH, 3HL, 3LH, 3LL. At last, 1HH, 1HL, 1LH, 2HH, 2HL, 2LH, 3HH, 3HL, 3LH, 3LL subband images can be found. The subband label kHL, H means that a horizontal high-pass filter has been applied to the rows; the L means a vertical low-pass filter applied to the columns; k means the k-th level of DWT decomposition.

Figure 2.4 shows a two-dimension image that has three levels DWT composition using the (9, 7) filter-bank in Table 2.3. Most of the image energy is stored in the lower frequency subband.

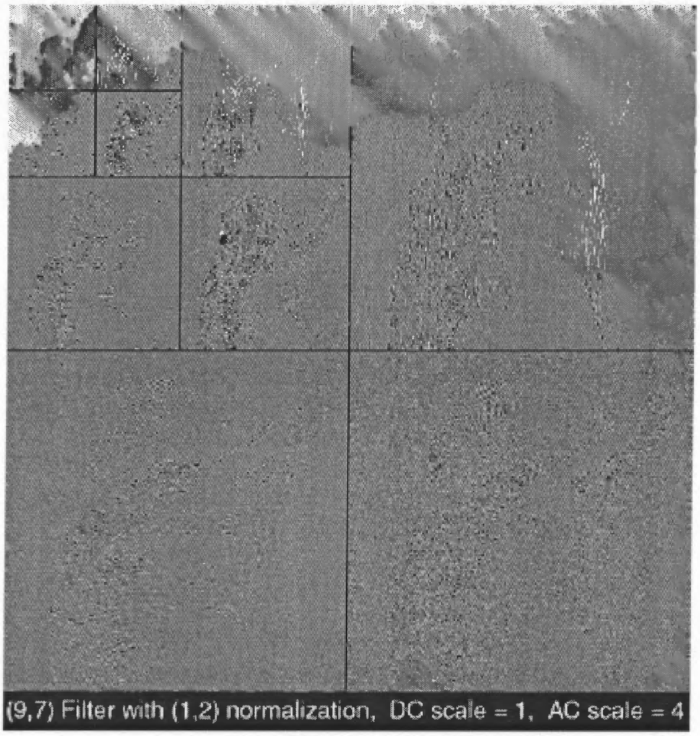
(9,7) Filter with (1,2) normalization, DC scale = 1, AC scale = 4

Figure 2.4 2-D, 3-level wavelet decomposition of Lena using the (9,7) filter-bank.

2.2.3 Filter Normalization

Filter normalization is expressed by the DC gain of the low-pass analysis filter h_0, and the Nyquist gain of the high-pass analysis h_1. The DC gain and the Nyquist gian of a filter h(n) are defined as [8]:

$$G_{DC} = \left|\sum_n h(n)\right|, \qquad G_{Nyquist} = \left|\sum_n (-1)^n h(n)\right|$$

The following table shows the (sqrt (2), sqrt (2)) Normalization and the (1,2) Normalization. The (sqrt (2), sqrt (2)) Normalization is adopted by Part 1 of JPEG 2000.

Table 2.3 L2-Norms of the DWT Subbands after 2-D, 3-level Wavelet Decomposition

	$(\sqrt{2}, \sqrt{2})$ Normalization		(1,2) Normalization	
Subband	(5,3) Filter-bank	(9,7) Filter-bank	(5,3) Filter-bank	(9,7) Filter-bank
3LL	0.67188	1.05209	5.37500	8.41675
3HL	0.72992	1.04584	2.91966	4.18337
3LH	0.72992	1.04584	2.91966	4.18337
3HH	0.79297	1.03963	1.58594	2.07926
2HL	0.79611	0.99841	1.59222	1.99681
2LH	0.79611	0.99841	1.59222	1.99681
2HH	0.92188	0.96722	0.92188	0.96722
1HL	1.03833	1.01129	1.03833	1.01129
1LH	1.03833	1.01129	1.03833	1.01129
1HH	1.43750	1.04044	0.71875	0.52022

2.2.4 Quantization

Quantization techniques generally compress by compressing a range of values to a single quantum value. By reducing the number of discrete symbols in a given stream, the stream becomes more compressible [5].

The quantization employed in JPEG2000 part 1 is nearly similar to JPEG. Figure 2.5 shows the quantization. The deadzone that can be seen has twice the quantizer step-size in part 1.

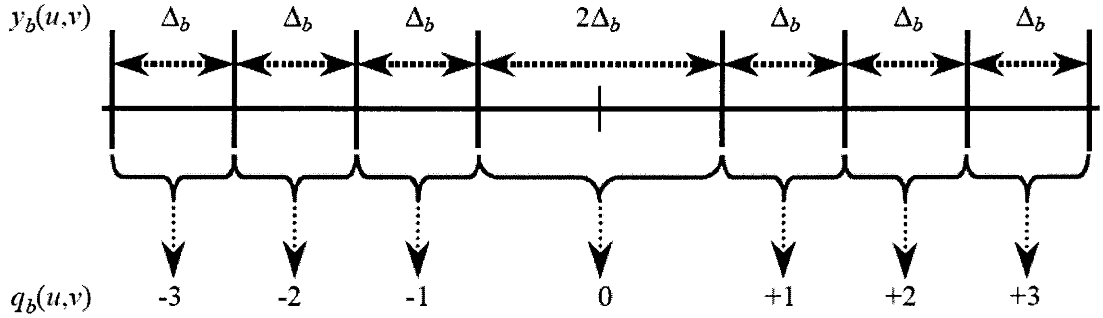

Figure 2.5 Uniform quantizer with deadzone with step-size Δb.

Quantization at the Encoder

Every subband, a basic quantizer step-size Δ_b is selected by the user and is used to quantize all the coefficients in that subband.

Inverse Quantization at the Decoder

When using irreversible (9,7) filter-bank, the reconstructed transform taps, $Rq_b(u,v)$ can express by:

$$Rq_b(u,v) = (q_b(u,v) + \gamma)\ \Delta_b\ ,\ \text{if}\ Rq_b(u,v) > 0\ ,$$

$$Rq_b(u,v) = (q_b(u,v) - \gamma)\ \Delta_b\ ,\ \text{if}\ Rq_b(u,v)\ \ 0\ ,$$

$$Rq_b(u,v) = \qquad 0 \quad ,\ \text{otherwise}\ ,$$

2.2.5 Tier-1 Coding

The coding has two parts: Tier-1 coding and Tier-2 coding. Figure 2.6 illustrates a simple example of the compressed data using Tier-1 coding. Next section we will discuss about the Tier-2 coding.

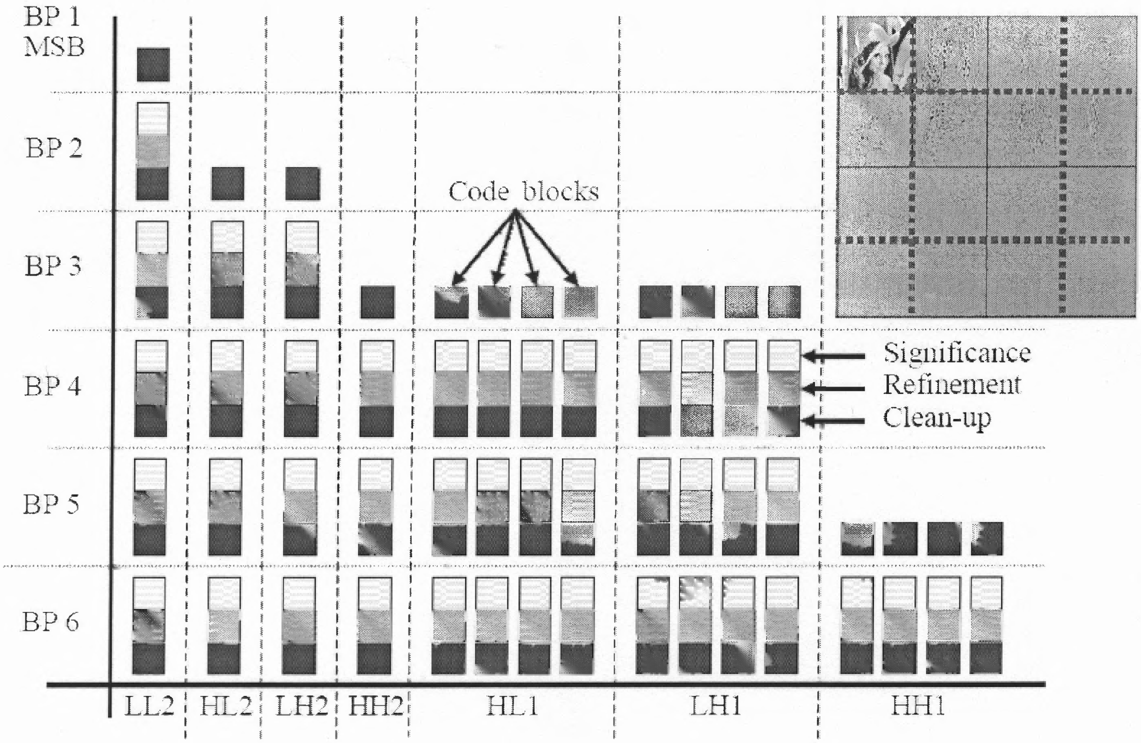

Figure 2.6 Example of compressed data associated with various sub-bitplane coding passes.

Bitplane encoding of wavelet taps has been used by several embedded wavelet coders such as WZW and SPIHT. In JPEG2000, each subband is encoded independent of the other subbands. And JPEG2000 uses a block coding paradigm in the wavelet domain as EBCOT (Embedded Block Coding with Optimized Truncation). Each subband is divided into small rectangular blocks, referred to as codeblocks. Figure 2.7 shows bitplanes coding of quantized wavelet taps. The symbols of the quantized coefficients are encoded into a bit-stream, the higher digital is MSB, the lower digital is LSB. A quantized wavelet tap is called insignificant if the quantizer index is still zero. When the non-zero bit is encoded, the tap becomes significant.

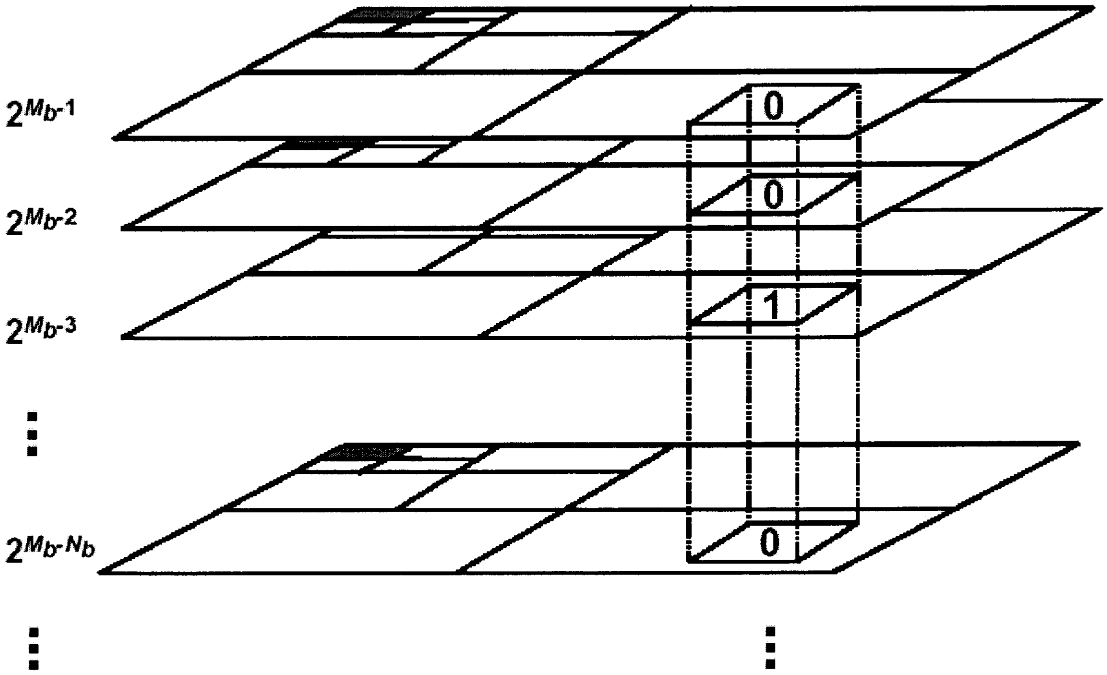

Figure 2.7 Bitplane coding of quantized wavelet coefficients.

Tier-1 coding is also called arithmetic coding of bitplane data. Huffman coding is an optimal prefix code for a given distribution that can be constructed by a algorithm discovered by Huffman. Arithmetic coding is different from Huffman coding. Arithmetic coding is mapped into a single codeword. This codeword is developed by recursive internal partitioning using the symbol probabilities, and the final codeword represents a binary fraction that points to the subinterval determined by the sequence. Arithmetic coding provides the compression efficiency, but only a single symbol is encoded at one time. Unlike Huffman coding, arithmetic coding does not require the development of new codewords each time the symbol probabilities change so that it is easy to adapt to the changing symbol probabilities.

Q-coder developed by IBM was one of the early practical implementations of adaptive binary arithmetic coding. Later on, we use QM-coder as entropy coder for JPEG 2000. Now, the JPEG 2000 committee chose a modified coder, MQ-coder.

Each bitplane is encoded in three sub-bitplane passed instead of encoding the entire bitplane in one coding pass. Figure 2.8 shows us the encoding of a single bitplane from a codeblock in three coding passes (labeled A, B, and C).

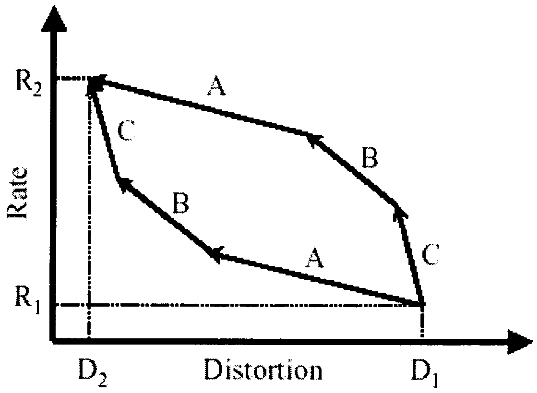

Figure 2.8 R-D path for optimal embedding.

The two coding paths ABC and CBA codes the same data in a different order, and they have the same rate-distortion points. However, their embedded performances are significantly different. The path CBA has more distortion than path ABC. There are three type passes: the first pass, known as the significance propagation pass; the second pass, referred to as refinement pass; the final pass, cleanup pass. In the following, It will describe each coding pass.

During the significance propagation pass, the insignificant taps that have the highest probability of becoming significant in current bitplane are encoded. Every sample that has one significant immediate neighbor at least, its significance state is updated when a tap is coded. The significance states of the eight neighbors can usually create 256 different contexts. Many of these contexts can be merged because of their similar probability estimates.

The magnitude bit of a taps that has become significant in a previous bitplane is arithmetic encoded with three contexts during the refinement pass. The refinement bits usually have an even distribution if the taps has not become significant in the previous bitplane.

In the cleanup pass, all the remaining taps in the bitplane are encoded as they have the lowest probability of becoming significant.

About the entropy coding options, 18 coding contexts in addition to a uniform context to the following assignment was used in the coding model. Contexts 0-8 are sued for significance coding during the significance propagation and cleanup passes, context 9 is used for run coding, contexts 10-14 are used for sign coding, and contexts 15-17 are used for the refinement pass.

2.2.6 Bit-stream Organization

In this section it will discuss the following concepts, components, tiles, subbands, resolution levels and codeblocks. These structures partition the image data into: (1) color channels (through components); (2) spatial regions (through tiles); (3) frequency region (through subbands and resolution levels); (4) space-frequency regions (through codeblocks). In JPEG 2000, It also provides an intermediate space-frequency structure known as a precinct. A precinct is a collection of spatially contiguous codeblocks from all subbands at a particular resolution level.

JPEG2000 organizes the compressed data from the codeblocks into packets or layers. The compressed bit-stream for each codeblock is distributed across one or more layers in the codestream. Figure 2.9 shows an example of codeblocks belonging to a precinct. The number of the codeblocks represents the order in which the coded data from the codeblocks will appear in a packet.

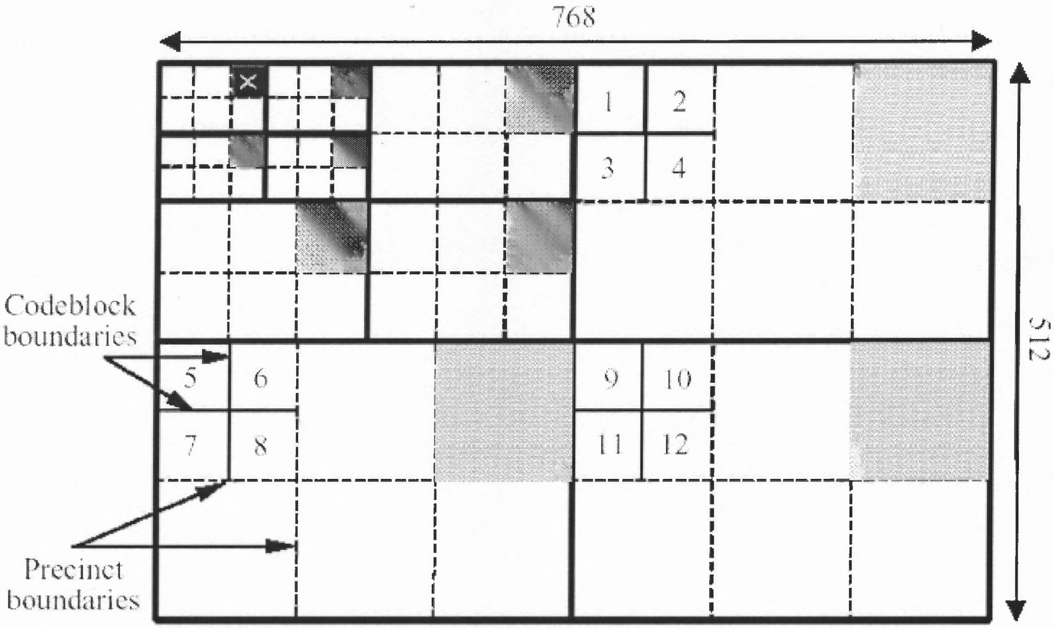

Figure 2.9 Examples of precincts and codeblocks.

The order where packets appear in the codestream is called the progression order. For a given tile, four parameters, component, resolution, layer, and position (precinct), are needed to identify a packet. There are only five progression orders in JPEG2000 Part 1, they are (1) layer-resolution-component-position progression; (2) resolution-layer-component-position progression; (3) resolution-position-component-layer progression; (4) position-component-resolution-layer; (5) component-position-resolution-layer.

2.3 Rate Control

Rate control is defined as the process of generating an optimal image for a target file size (bit-rate) and it is strictly an encoder issue [7]. The criterion for optimality can be based on mean squared error (MSE) between the original and reconstructed image, and visual distortion. A typical JPEG rate control algorithm starts with a basic q-table and modifies the q-table elements until the desired bit-rate is achieved. Rate control is divided into: rate control using explicit Q-table, and rate control using the EBCOT algorithm. EBCOT algorithm was proposed by Taubman. It is an efficient rate control method that achieves a desired rate with minimum distortion.

2.4 Performance Comparison of JPEG2000 Encoder Options

It compare with the effects of various coding on the coding efficiency for lossless compression. It is not very easy to compare the speed and implementation complexity of different coding, therefore, it will point out the relative speed advantages of certain options.

2.4.1 Reversible Color Transform

It is well known that it can improve the coding efficient by applying a color transform. Compared with the performance of the JPEG2000 algorithm for lossless coding compression between with applying the RCT and without the RCT in Table 2.4.

Table 2.4 Comparison of Lossless Bit-rates for Color Images with and without RCT

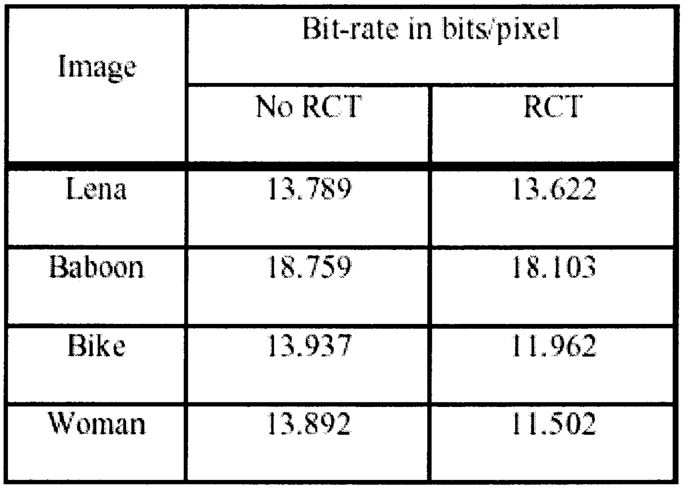

Image	Bit-rate in bits/pixel	
	No RCT	RCT
Lena	13.789	13.622
Baboon	18.759	18.103
Bike	13.937	11.962
Woman	13.892	11.502

It is used the 24-bit color version instead of 8 bit test images and the results are based on using the reversible (5,3) filter-bank. From the table we can save 0.16-2.39 bpp with applying the RCT transform.

2.4.2 Lossless Encoder Options

The lossless compression performance of the JPEG2000 standard as a function of tile size is summarized in Table 2.5. The lossless compression performance of the JPEG2000 standard as a function of number of decomposition levels is summarized in Tables2.6, The lossless compression performance of the JPEG2000 standard as a function of codeblock is summarized in Table 2.7. The lossless compression performance of the JPEG2000 standard as a function of lazy-parallel modes is summarized in Table 2.8.

Table 2.5 Comparison of Average Lossless Bit-Rates (bits/pixel) for Different Tile Sizes

No tiling	512 × 512	256 × 256	128 × 128	64 × 64	32 × 32
4.797	4.801	4.811	4.850	5.015	5.551

Table 2.6 Comparison of Average Lossless Bit-Rates (bits/pixel) for Different Number of Decomposition

5 levels	4 levels	3 levels	2 levels	1 level	0 levels
4.797	4.798	4.802	4.818	4.887	5.350

Table 2.7 Comparison of Average Lossless Bit-Rates (bits/pixel) for Different Codeblock Sizes

64 x 64	32 × 32	16 × 16	8 × 8
4.797	4.846	5.005	5.442

Table 2.8 Comparison of Average Lossless Bit-Rates (bits/pixel) for 'Lazy', 'Parallel' and 'Lazy-Parallel' Modes

Reference	Lazy	Parallel	Lazy-parallel
4.797	4.799	4.863	4.844

CHAPTER 3

JPSEC and Signal Syntax

3.1 Introduction

The development of the web and the advances in computer technology have produced a proliferation of digital media content that can be efficiently copied, processed and distributed at negligible cost, both for licit and illicit use. Security issues are therefore very important features in many imaging applications targeted by JPEG 2000.

To address this issue, JPEG 2000 Secured (JPSEC) or Part 8 of the standard is standardizing tools and solutions in terms of specifications in order to ensure the security of transaction, protection of contents (IPR), and protection of technologies (IP), and to allow applications to generate, consume, and exchange JPEG 2000 Secured bit streams.

The following examples are the applications addressed by JPSEC:

- Encryption: JPSEC will provide a flexible mechanism to allow for encryption of image content and metadata. This includes partial encryption of the latter, or encryption with different strengths.

- Data integrity: JPSEC will allow for data integrity verification. This includes semi-robust integrity verification, as well as mechanisms to optionally identify locations in the image content where the integrity is put into question.

- Conditional access: JPSEC will allow for conditional access to portions of an image or its associated metadata. For instance, a user could be allowed to view a low resolution (preview) of an image without being able to visualize a higher resolution.

- Ownership protection: JPSEC will allow for protection of the content owner rights (copyright). This includes ownership identification mechanisms robust to malicious attacks and non-malicious processing of the JPEG 2000 bitstream and/or the image it represent.

JPEG2000 protects the content including digital signatures, watermarking, encryption and scrambling. These techniques will be enabled in JPSEC by means of a registration authority. More specifically, all techniques have to be previously registered in a central repository, the registration authority, which uniquely identifies these techniques.

3.2 Normative Part

3.2.1 The Overview of JPSEC Framework

Figure 3.1 shows us the JPSEC framework. Central to the JPSEC framework is the JPSEC bitstream: this represents a secure JPEG 2000 image.

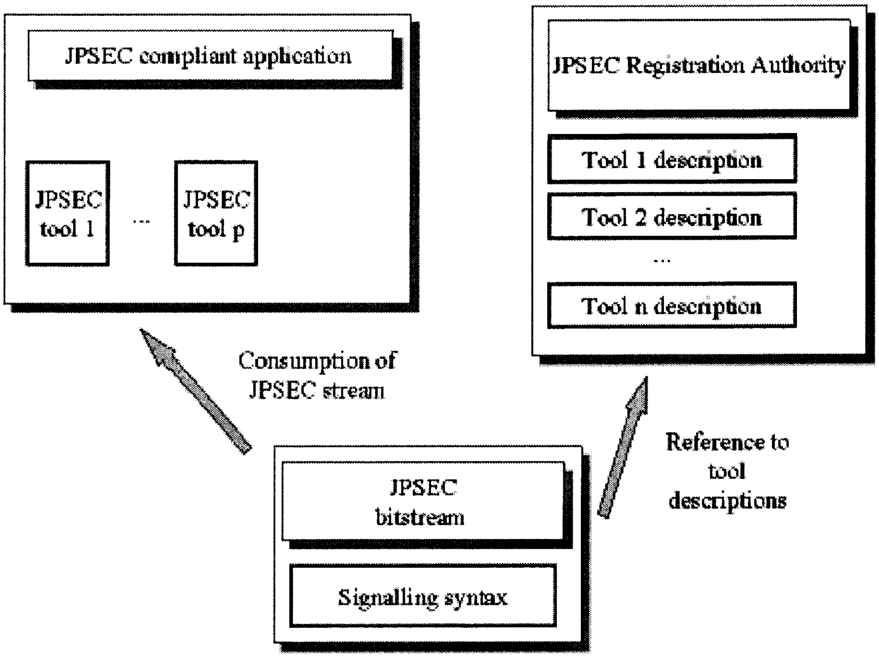

Figure 3.1 Overview of JPSEC framework.

JPSEC bitstreams can be created from an original image, from JPEG-2000 coded data, or from other JPSEC bitstream. Figure 3.2 shows the creation method of JPSEC bitstreams. In case A, the encoding and protection operations are performed at the same time, so the protection tool has access to the original image content. This may be important for security tools such as content authentication. In case B, the JPSEC bitstream is created from JPEG-2000 coded data. This may occur when performing encryption on a database of JPEG-2000 images. In case C, the JPSEC bitstream may be created from another JPSEC bitstream.

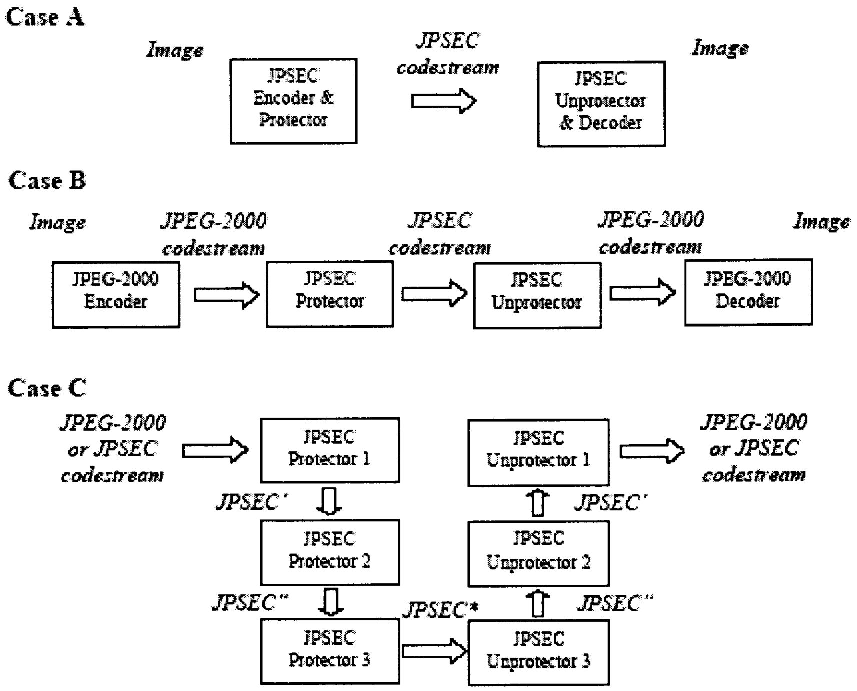

Figure 3.2 Creation and consumption modes of JPSEC content.

3.2.2 Main Security Marker

A SEC maker segment which is located in the main header packet is defined for presenting a flexible, simple syntax for signaling. The SEC marker segment can be define all of the required information fro JPSEC images. The SEC marker syntax is defined in Figure 3.3. The following is the description of SEC, Lsec, and Tool. SEC: Marker code, Table 3.1 shows the sizes and values of the symbols and parameters for the main security marker segment; Lsec: Length of marker segment in bytes; Tool(i): Parameters for protection tool i. If multiple protection methods are signalled, then a JPSEC decoder shall process each protection method in the order of appearance in the SEC marker segment.

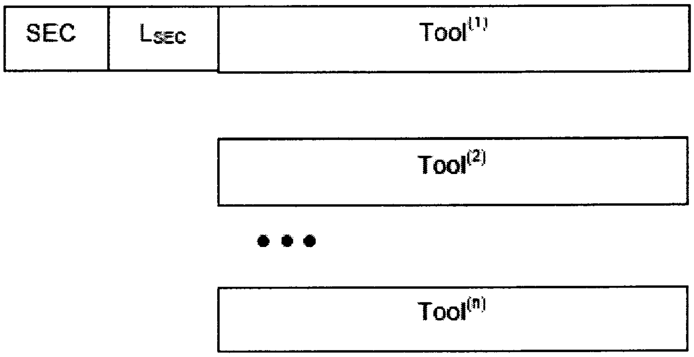

Figure 3.3 Main security marker syntax.

Table 3.1 Main Security Parameter Values

Parameter	Size (bits)	Values
SEC	16	0xFF94
Lsec	16	$2 - (2^{16}-1)$
Tool$^{(i)}$	Variable	See Sec. 5.3.2 and Sec. 5.3.3

Main security marker has two kinds of tools: template protection tool and registration authority protection tool.

Template protection tool syntax shown in Figure 3.4 and Table 3.2.

i	t	$L_{ZOI}^{(i)}$	$ZOI^{(i)}$	$L^{(i)}$	$T^{(i)}$	$PD^{(i)}$	$G^{(i)}$

Figure 3.4 Template Protection Tool syntax (t=0).

i: Tool instance index (can be used as a unique identifier).

t=0: Tool type. 0 indicates protection tool from protection method templates.

LZOI(i):Length of LZOI(i) in Bytes.

ZOI(i):Zone of influence for protection tool i.

L(i): Length of L(i) + T(i) + PD(i) + G(i) in Bytes.

T(i): Protection method template parameters for protection tool i.

PD(i): Processing domain for protection tool i.

G(i): Granularity for protection tool I.

Table 3.2 Template Protection Tool Parameter Values

Parameter	Size (bits)	Values
i	7	$0 - (2^7-2)$ 2^7-1, reserved
t	1	0
$L_{ZOI}^{(i)}$	16	$0 - (2^{16}-1)$
$ZOI^{(i)}$	Variable	See Sec. 5.4.
$L^{(i)}$	16	$0 - (2^{16}-1)$
$T^{(i)}$	Variable	See Sec. 5.5.
$PD^{(i)}$	Variable	See Sec. 5.6.
$G^{(i)}$	8	See Sec. 5.7.

The registration authority protection tool syntax shown in Figure 3.5 and Table 3.3. It can be widely used in a variety of security techniques, such as image data integrity, access control and rights protection methods.

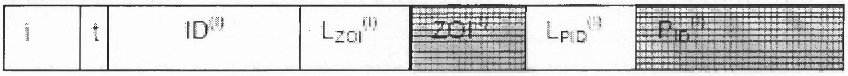

Figure 3.5 Registration Authority Protection Tool syntax (t=1).

i: Tool instance index (can be used as a unique identifier).

t=1: Tool type. 1 indicates protection tool from the registration authority.

ID(i): Registration authority ID number for protection tool i.

LZOI(i):Length of LZOI(i) + ZOI(i) in Bytes.

ZOI(i): Zone of influence for protection tool i.

LPID(i):Length of LPID(i) + PID(i) in Bytes.

PID(i): Parameters for protection method I.

Table 3.3 Registration Authority Protection Tool Parameter Values

Parameter	Size (bits)	Values
i	7	$0-(2^7-2)$ 2^7-1, reserved
t	1	1
$ID^{(i)}$	32	$0-(2^{32}-1)$
$L_{ZOI}^{(i)}$	16	$0-(2^{16}-1)$
$ZOI^{(i)}$	Variable	See Sec. 5.4
$L_{PID}^{(i)}$	16	$0-(2^{16}-1)$
$P_{ID}^{(i)}$	Variable	Defined by registration authority tool $ID^{(i)}$

3.3 JPSEC Applications

Figure 3.6 shows an overview of JPSEC applications in secure area. In these applications, the JPSEC application may be required to provide various security services for JPEG 2000, such as, confidentiality of image exchange, and authentication of image origin.

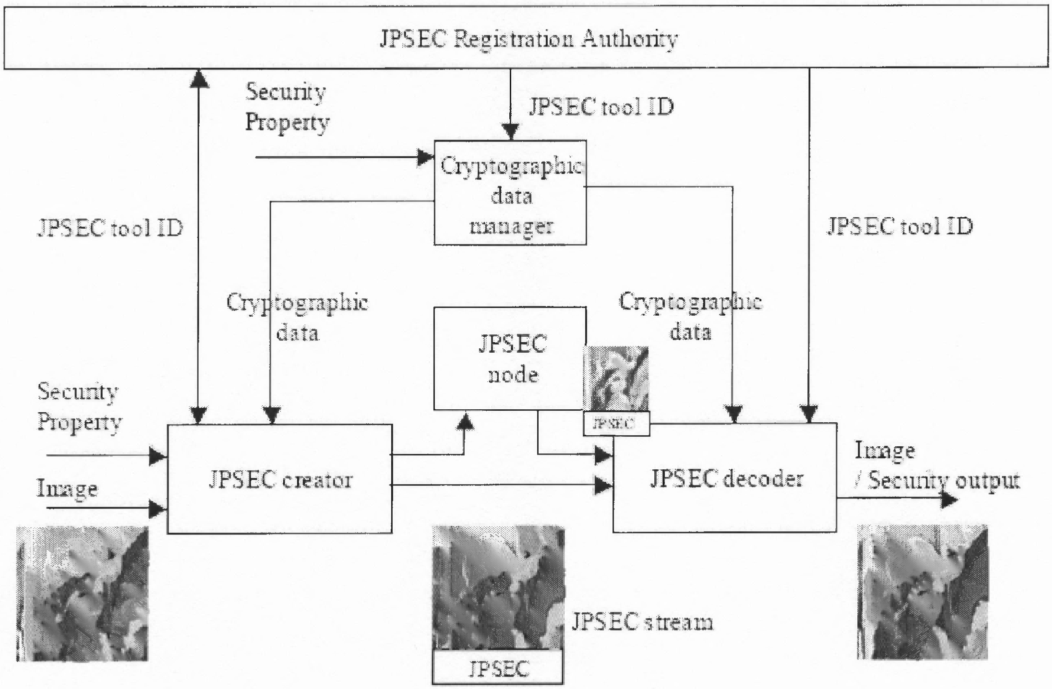

Figure 3.6 Overview of a secure JPEG 2000 image distribution application

The secure JPEG 2000 image distribution application is divided into the following three steps:

Step 1: A JPSEC stream is created by a JPSEC creator.

Step 2: The JPSEC stream is distributed through some JPSEC node or nodes

Step 3: The JPSEC stream received and rendered by a JPSEC decoder.

3.3.1 JPSEC Stream Creation

The creator creates the secure JPEG 2000 stream. This stream may be created from uncompressed data or from JPEG 2000 compressed data. A JPSEC creator applies various security methods, such as encryption, signature generation, and ICV (Integrity Check Value) generation to a given image data. To secure the image data the creator defines which Security Property is associated to the image. A "Security Property" includes the following attributes:

- Zone of Influence (coverage area of each protection method)
- Processing Domain (domain to be processed by each protection method)
- Granularity (unit of each protection method)
- Protection Method identification

3.3.2 JPSEC Stream Delivery

A JPSEC stream can be transferred to a JPSEC decoder directly via a network. It can also be transferred through a JPSEC node, which can apply various types of additional processing, such as a transcoding, to the JPSEC stream. When required by the security methods in the Security Property of the JPSEC stream, the JPSEC creator must distribute to the JPSEC decoder the corresponding cryptographic data through an independent ('secret') channel. This data can be managed either manually or automatically by a cryptographic data manager.

3.3.3 JPSEC Stream Rendering

A JPSEC stream is subject to a JPSEC decoder process according to the applied Security Property: this implies applying the appropriate unprotection methods, such as decryption and authentication. Further, for each security method, a JPSEC creator and JPSEC decoder may use various types of cryptographic data. As an output of the JPSEC decoder, a decrypted image data and/or security output, such as a verification result, is produced. A JPSEC creator, JPSEC decoder and cryptographic data manager may reference the JPSEC Registration Authority to obtain necessary processing instructions of a specific JPSEC tool ID.

CHAPTER 4

DIGITAL WATERMARKING

4.1 The Definition of Digital Watermarking

What is digital watermarking? As simply digital watermarking, a pattern of bits inserted into a digital image, audio or video file that identifies the file's copyright information (author, rights, etc.). The name comes from the faintly visible watermarks imprinted on stationery that identify the manufacturer of the stationery. The purpose of digital watermarking is to provide copyright protection for intellectual property that's in digital format.

Unlike printed watermarks, which are intended to be somewhat visible, digital watermarks are designed to be completely invisible, or in the case of audio clips, inaudible. Moreover, the actual bits representing the watermark must be scattered throughout the file in such a way that they cannot be identified and manipulated. And finally, the digital watermark must be robust enough so that it can withstand normal changes to the file, such as reductions from lossless compression algorithms.

It is not easy to satisfy all these requirements, however, there are a number of companies offering competing technologies. All of them work by making the watermark appear as noise - that is, random data that exists in most digital files anyway. To view a watermark, you need a special program that knows how to extract the watermark data. Watermarking is also called data embedding and information hiding.

Watermarking content owners turn into to be cyptograpy, which is probably the most common method of protecting digital content. It is certainly one of he best developed as a science. The content is encrypted before transmission, those who have purchased legal copies of the content can get a decryption key.

4.2 Steganography, Data Hiding and Watermarking

Digital watermarking and steganography have the common feature: describing methods to embed information transparently into a carrier signal. Digital watermarking does not hide the fact of secret information transmission from third parties, while steganography is a method, which establishes a covered information channel through point-to-point, connects. Watermarking also has the additional requirement of robustness against manipulations in order to remove the embedded information from the marked carrier object.

Data hiding is a term of steganography, it embeds data into digital media for the purpose of identification, annotation and copyright. There are several constraints to affect the process: the quantity of data to be hidden, the need for invariance of these data under conditions where a "host" signal is subject to distortions, such as, lossless compression, and the degree to which the data must be immune to interception, modification, or removal by a third person. We explore both traditional and novel techniques for addressing the data-hiding process and evaluate these techniques in light of three applications: copyright protection, tamperproofing, and augmentation data embedding.

Digital watermarking give us a method to embed information transparently into a carrier signal, this makes watermarking is suitable for applications in the area of the knowledge of a hidden message. A feature of digital watermarking is to hide the additional useful information, a certain perceptual threshold is required to allowing the insertion of additional information and hence distortions of the carrier signal without incurring unacceptable perceptual degradation of the original carrier signal. Watermarking system are therefore context specific, in other word, the algorithms must designed with respect to the media type of the data to be watermarked.

4.3 Digital Watermarking

4.3.1 History of Watermarking

The art of papermaking was invented in China a thousand years ago. In 1282, the paper watermarks appeared in Italy. The marks were made by adding thin wire patterns to the paper molds. The paper would be slightly thinner where the wire was and hence more transparent. People do not know the purpose of earliest watermarks. They may have been used for practical functions, or as a trademark to identify the paper maker.

Watermarking on paper made in Europe and American started to utilize clearly by the eighteenth. People use to record the date the paper and to indicate the sizes of original sheets as trademarks.

In 1990s, some companies were established to market watermarking products. The technology of the Verance Corporation was adopted into the first phase of SDMI (The Secure Digital Music Initiative), and the technology was used by Internet music distributors.

4.3.2 Features

Watermarking is different from other techniques in three key ways: First, watermarking is not perceptible. Second, watermarking is inseparable from the works in which they are embedded. Last, watermarking performs the same transformations as the works. These three keys make watermarking invaluable for some applications.

Watermarking was used in a wide kind of applications, such as broadcast monitoring, owner identification, proof of ownership, transaction tracking, authentication, copy control and device control. It can use digital watermarking to protect our rights in these areas.

There are many properties in the digital watermarking, we will introduce 10 main properties of digital watermarking: effectiveness, fidelity, and payload, blind, informed detection, false positive behavior, robustness, security, secret keys, and costs. The relative important of each property depends on the requirements of the application and the role the watermark will play. The first three are a number of features associated with a watermark embedding process. We then turn to features connected with detection: blind and informed detection, false positive behavior, and robustness. The next feature, costs is associated with both digital watermarking and digital watermarking detection.

4.3.3 Watermarking System

Many systems need to be evaluated so that we can get a robustness system. For the digital watermarking, we need to find a good algorithm to evaluate the digital watermarking system.

4.3.3.1 The Notion of "Best". The notion of "best" means that we want to have some idea of what makes one system better than another one before evaluating a digital watermarking system. Our evaluation must depend on the application if we are interested in using watermarking as some specific application.

4.3.3.2 Benchmarking. Benchmarking is the term of the system that can be laid out as a minimum requirement for a relative feature. All tests can be developed to measure if systems meet those requirements.

4.3.3.3 Scope of Testing. The tested Works should represent a typical range of applications if a system is being tested with no a specific application in mind.

4.3.4 Models of Watermarking

These models is divided into two wide group: first, models based on a view of watermarking as a method of communications; second, models based on geometric views of watermarking algorithm[4].

The traditional communications system model shows in Figure 4.1. Original signal, m, needs to be transmitted through a communications channel. This message is

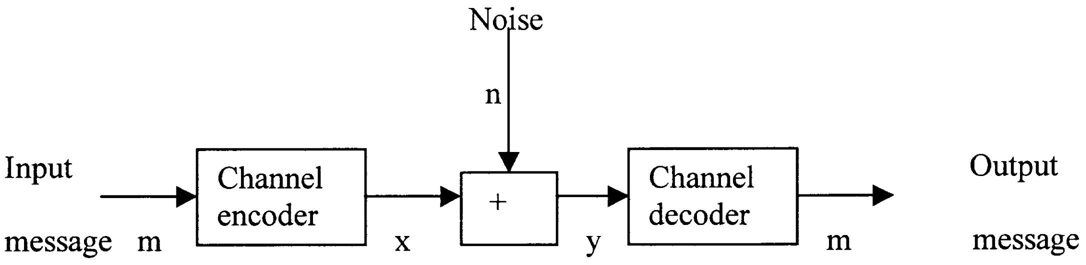

Figure 4.1 The model of communication system.

encoded by channel encoder, channel encoder makes each possible message into a code word so that message can be transmitted over the channel. The encoded message is usually defined as x. Then the encoded message was transmitted by physical channel, which can be cable, wireless, SDH, PDH, etc. It defied the main reason that causes the error code as noise. When encoded message is transmitted through physical channel, noises are added to the encoded message. It can produce a certain of error code. At the receiving end of the channel, we name the received signal, y. After that, the signal is sent to channel decoder, which inverts the encoding process to correct because the noises cause the transmission errors.

Secure transmission is very important for a real communication system, especially, the communication system is used for military communication. Prior to transmission, cryptography is used to encrypt a message, or cleartext, suing key. Then it transmit the encrypted message. The encrypted message can be decrypted using the same key to recovery the original message.

Spread spectrum communication is best illustrated with an example because of the signal jamming. Spread spectrum technology has two parts: FHSS (Frequency Hopping Spread Spectrum) and DSSS (Direct Sequence Spread Spectrum). FHSS is one of the earliest and simplest spread spectrum technologies. The pattern of hops from one frequency to another frequency is controlled by a key, called SN. The receiver and the transmitter must know the key. Spread spectrum also divides into SFHSS and FFHSS. Now slow FHSS technology can be used in commercial areas. Such as, the wireless route produced by P-COM company.

CHAPTER 5

AUTHENTICATION SYSTEM FOR JPEG2000 IMAGES

5.1 Introduction

Chapter 2, 3, 4 and almost all of literature focus on DWT, JPSEC and digital watermarking. This chapter will describe a digital signature based authentication system for JPEG2000 images that provide integrity protection and source authentication services for JPEG2000 images, including bitstream level and image content level. Therefore, it can satisfy the different requirements from users. Authentication system can be divided into two parts; one is fragile authentication, and the other is semi-fragile authentication. Fragile authentication is to protect the whole or the part of JPEG2000 bitstream. It will be not authenticated if any one bit modified, in this case, fragile authentication is suitable for an ideal environment where images are exchanged with no any distortion. And semi-fragile authentication is to protect the JPEG2000 image content. Because the fragile authentication is not easy to realize, we prefer the semi-fragile authentication that is more robust to incidental distortions than fragile does, such as, compression, transcoding and format conversion. The semi-fragile authentication embeds some information into original images, there are two methods for this applications: lossy mode and lossless mode. When lossy method was used, the image will be not recovered after a data was embedded into original image. However, with a lossless method, the data is embedded into the image, the image can be recovered.

The authentication systems for JPEG2000 images have a few applications in different areas, fragile authentication is suitable for an ideal environment where images are exchanged with no any distortion; especially, compare with fragile authentication, and the proposed semi-fragile authentication can identify the changed areas by the attacker when the image content is modified. Thus, the semi-fragile authentication has wide applications such as medical, remote imaging, image editing and desktop publishing. There is one common element, say, lowest authentication bit-rate (LABR), which will be use in a lot different authentication scenario, such as, fragile authentication, semi-fragile authentication that includes lossy mode and lossless mode. For instance, if a JPEG2000 image is protected as LABR of 0.5bpp (bit per pixel), any transcoded version of the image shall be not recovered as authentic as long as the bit-rate after transcoding is less than 0.5bpp.

We can have a look at Figure 5.1 [9], it give us one possible application scenario for image authentication in all kinds of environment, where different entities have different capabilities in terms of bandwidth, display size and processing power.

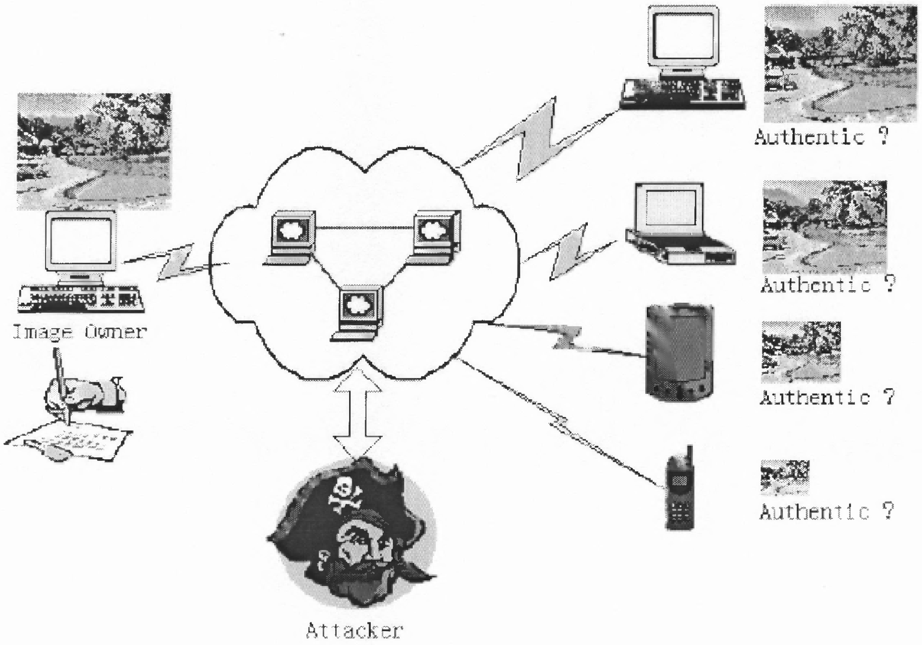

Figure 5.1 Image authentication in heterogeneous environment.

When the image is exchanged between different entities, various kinds of distortions are introduced. For instance, unreliable carrier (like wireless link) may introduce some bit errors; the image has to be transcoded or the image format has to be converted to suit the receiver's capability. These distortions are also referred as incidental distortions. There is another distortion called intentional distortion, also referred to as malicious modification introduced by attackers. Usually, incidental distortion does not change the image content, while intentional distortion does. Therefore, an ideal authentication solution should be robust to incidental distortion while being sensitive to intentional distortion.

5.2 Fragile Authentication

As explained in previous section, fragile authentications protect images at code stream level. It is an application of traditional crypto signature on JPEG2000 images, the bitstream to be protected is used to generate a digital signature, using the well-known crypto algorithm, i.e. Rivest, Shamir, and Adleman (RSA). The fragile authentication can use JPEG2000 code-stream. And it is able protect any layers, tiles, resolutions, precincts, codeblocks or any their combination. Although this solution provides secure protection against intentional distortion, it is not robust to aforementioned incidental distortions, as even a single-bit change will make the image unauthentic. The LABR is used to specify the most layers to be protected. Figure 5.2 shows us, when images are exchanged between different entities, no transcoding is necessary, it are able to use fragile authentication for protecting the images, in this case, it will not change.

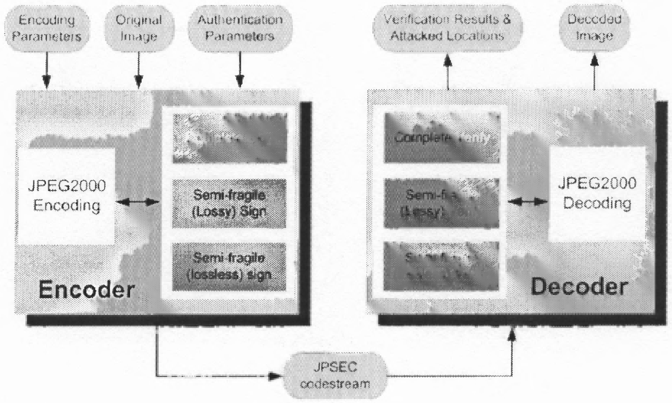

Figure 5.2 System overview of the authentication system.

5.3 Semi-fragile Authentication

The authentication system is composed of fragile authentication solution together with semi-fragile solution for authentication of JPEG2000 images. The semi-fragile authentication has wider applications than the fragile authentication does [9].

The semi-fragile authentication utilizes the crypto signature-based schemes, such as, DSA. In our system, the semi-fragile authentication has four sections: feature extraction, Error Correction Coding (ECC) of feature, signature generation and data embedding. Additionally, the proposed semi-fragile authentication can identify the changed areas that the image content is modified by the attacker, it helps to convince the receiver on the authentication results.

It is not like fragile authentication that aims on bitstream integrity, semi-fragile authentication focuses on image content. We propose a semi-fragile authentication technique that can distinguish common image processing operations from other malicious manipulations. The image will pass the verification process if the image content does not change or even though it has experienced incident alteration. The image content can be expressed by a set of "features", the features are extracted from the most significant bitplanes of wavelet coefficients. Here, we use the specified Lowest Authentication Bit-Rate (LABR) parameter, the bit-planes from which features are extracted is decided by the rate control process (EBCOT) of JPEG2000. Thus, if the image is transcoded to a bit-rate that is larger and equal than the LABR, the extract features will be kept invariant and they nearly express the image content.

The robustness is very important in natural, engineering, and social systems. How can we deal the minor change of features caused by with incidental distortion? We utilize Error Correction Coding (ECC) which is used to encode the extracted the features to generate a set of codewords. All of these codewords are connected into one string, which is used to generate a global signature by using traditional cryptographic algorithm such as RSA or DSA. Through these generated global signature, it can enhance the robustness of the features. There is one-to-one correspondence between the block where features are extracted, therefore, if the protected image is attacked, by decoding ECC block by block, the authentication system will be able to easily find the areas that is attacked.

Another method, Parity Check Bits (PCB) is used for image process, there are two reasons: one is to increase the robustness, it can help us to correct certain number of bit changes; another is to help us to locate the attacked areas when the image is modified by attacker.

5.4 Lossless Mode

The semi-fragile authentication is divided into two parts by data embedded methods: lossy mode (lossy authentication) and lossless mode (lossless authentication), which target at different applications. In this section, lossless mode will be introduced.

5.4.1 Crypto-algorithm

Crypto-algorithm: well-defined procedure or sequence of rules or steps, or a series of mathematical equations used to describe cryptographic processes such as encryption/decryption, key generation, authentication, signatures, etc. The followings are some mode of crypto-algorithms: Data Encryption Standard (DES) and triple DES, Rivest, Shamir, and Adleman (RSA), Elgamal, (CAST) and Secure Socket Layer (SSL).

5.4.2 Hash Algorithm

Theoretically, MD5 and SHA1 are algorithms for computing a 'condensed representation' of a message or a data file. The 'condensed representation' is of fixed length and is known as a 'message digest'. For instance, when you download or receive a file, you can use MD5 or SHA-1 to guarantee that you have the correct, unaltered file by comparing its hash with the original. You are essentially verifying the file's integrity.

SHA-1: The Secure Hash Algorithm (SHA) was developed by NIST and is specified in the Secure Hash Standard (SHS, FIPS 180). SHA-1 is a revision to this version and was published in 1994. It is also described in the ANSI X9.30 (part 2) standard. SHA-1 produces a 160-bit (20 byte) message digest. Although slower than MD5, this larger digest size makes it stronger against brute force attacks. MD5 was developed by Professor Ronald L. Rivest in 1994. Its 128 bit (16 byte) message digest makes it a faster implementation than SHA-1.

5.4.3 Digital Signature

This is a typical scenario in cryptography area. Bob has been given two keys. One of Bob's keys is called a Public Key; the other is called a Private Key. Susan, Doug, Pat are Bob's co-workers.

Bob's Public key is available to anyone who needs it, but he keeps his Private Key to himself. Keys are used to encrypt information. Encrypting information means "scrambling it up", so that only a person with the appropriate key can make it readable again. Either one of Bob's two keys can encrypt data, and the other key can decrypt that data. Susan can encrypt a message using Bob's Public Key. Bob uses his Private Key to decrypt the message. Any of Bob's coworkers might have access to the message Susan encrypted, but without Bob's Private Key, the data is worthless. With his private key and the right software, Bob can put digital signatures on documents and other data. A digital signature is a "stamp" Bob places on the data that is unique to Bob, and is very difficult to forge. In addition, the signature assures that any changes made to the data that has been signed cannot go undetected. To sign a document, Bob's software will crunch down the data into just a few lines by a process called "hashing". These few lines are called a message digest. Bob's software then encrypts the message digest with his private key. The result is the digital signature. Finally, Bob's software appends the digital signature to document. All of the data that was hashed has been signed.

Bob now passes the document on to Pat. First, Pat's software decrypts the signature (using Bob's public key) changing it back into a message digest. If this worked, then it proves that Bob signed the document, because only Bob has his private key. Pat's software then hashes the document data into a message digest. If the message digest is the same as the message digest created when the signature was decrypted, then Pat knows that the signed data has not been changed. Doug (our disgruntled employee) wishes to deceive Pat. Doug makes sure that Pat receives a signed message and a public key that appears to belong to Bob. Unbeknownst to Pat, Doug deceitfully sent a key pair he created using Bob's name. Short of receiving Bob's public key from him in person, how can Pat be sure that Bob's public key is authentic. It just so happens that Susan works at the company's certificate authority center. Susan can create a digital certificate for Bob simply by signing Bob's public key as well as some information about Bob. Now Bob's co-workers can check Bob's trusted certificate to make sure that his public key truly belongs to him. In fact, no one at Bob's company accepts a signature for which there does not exist a certificate generated by Susan. This gives Susan the power to revoke signatures if private keys are compromised, or no longer needed. There are even more widely accepted certificate authorities that certify Susan. Let's say that Bob sends a signed document to Pat. To verify the signature on the document, Pat's software first uses Susan's (the certificate authority's) public key to check the signature on Bob's certificate. Successful de-encryption of the certificate proves that Susan created it. After the certificate is de-encrypted, Pat's software can check if Bob is in good standing with the certificate authority and that all of the certificate information concerning Bob's identity has not been altered. Pat's software then takes Bob's public key from the certificate and uses it to check Bob's signature. If Bob's public key de-encrypts the signature successfully, then Pat is assured that the signature was created using Bob's private key, for Susan has certified the matching public key. And of course, if the signature is valid, then we know that Doug didn't try to change the signed content.

5.4.4 Implementation

The lossless embedding algorithm was used. The LABR is the key element that we want to use, a sub-band is selected according to the specified LABR. The larger LABR will be chosen, the higher resolution level we will use. In other words, if we choose larger LABR, we will have to choose higher resolution level; if we choose the smaller LABR, we will have to choose lower resolution level.

When the LABR is larger than or equals to 4 bpp, we can get level 5, HL of the subband for feature extraction; when the LABR is less than 4bpp and larger than or equals to 2 bpp, we can get level 4, HL of the subband for feature extraction; when the LABR is less than 2bpp, we can get level 3, HL of the subband for feature extraction.

The original image is 512 by 512. Here we choose LABR is 1.0, in order to get the feature extraction, we utilize 3HL. The code block in subband is 64 by 64. The feature is the result of "OR" operation of the most significant 3 bitplanes. Then, hash operation is applied to the feature, generating a hash value.

For the watermarking part, the first 18 bits of hash value from corresponding codeblock in level 3, HL subband is encoded with BCH (63, 18, 10), generating 63 bits codeword, which is to be embedded into the code block in level 3, LH. A 64 by 64 codeblock in 3LH subband is divided into 64 small blocks 8 by 8. Using the algorithm by Ni, each 8 by 8 block embedded with 1 bit from the codeword.

Registration Authority Protection Tool Syntax

Table 5.1 Parameters in ZoI and Protection Template for Lossless Semi-fragile Authentication Example

Parameter			Value	Size (bits)	Derived meaning
i			0	7	Instance index
t			1	1	RA protection syntax
ID			Xxxxxxxxxxx	32	ID to be assigned
L_{ZOI}			0x000B	16	Length of ZoI is 11 bytes
			ZoI syntax		
ND_{ZOI}			1	8	Number of zones is one.
Zone^{0}	DC_{ZOI}		0	1	Image-related description class
			100000	6	Only image region is specified
			0	1	BAS structure not extended
	P_{ZOI}	M_{ZOI}	0	1	Influenced
			0	1	Single item specified
			00	2	Rectangle mode
			01	2	I_{ZOI} uses 16-bits integer
			1	1	I_{ZOI} is described in two dimensions
			0	1	BAS structure not extended
		I_{ZOI}	100(decimal)	16	X coordinate of upper-left corner is 100
			100(decimal)	16	Y coordinate of upper-left corner is 100
			300(decimal)	16	X coordinate of lower-right corner is 300
			300(decimal)	16	Y coordinate of lower-right corner is 300

				Protection Template		
$L^{(1)}$				0x002D	16	Length of Protection template, PD and G is 45 bytes.
PM_{ID}				0000 0010	8	Authentication Template
T_{AUTH}	M_{AUTH}			0000 0010	8	Digital signature is used for authentication
	P_{AUTH}	M_{DS}		0000 0011	8	Digital signature method : DSA
		KT_{DS}	LK_{KT}	100000000	16	Length of key: 128 bits
			KID_{KT}	0000 0001	8	Public key used
			LKI_{KT}	0001 0000	16	Length of key information : 16 bytes
			KI_{KT}	xxxxxxxxx	Variable	A byte string containing public key
		SIG_{DS}		xxxxxxxxx	Variable	A byte string containing the signature
				Processing Domain		
PD				00	2	Processed in wavelet coefficients domain
				Private Parameters		
LABR				00000011	16	Lowest Authentication Bit-rate: 3 bpp
Threshold				00000101	8	The threshold value for lossless watermarking
Shuffle				00000010	8	The number of shuffling in order to embed watermark bits

5.4.5 Experimental Results

For the semi-fragile authentication with lossless mode, the image is encoded using JPEG2000 5 × 3 integer wavelet transformation (IWT) filter. Here, LABR is set to 1.0 bpp. SHA-1 is used for hashing and DSA is used for signature generation. Figure 5.3 shows the original image. The watermarked image is 1.0bpp, Figure 5.4 shows the watermarked image that has been compressed. Figure 5.5 shows the recovered image after verification. The image can be authenticated when passing by the authentication system. We have evaluated its performance in terms of False Positive Rate (FPR), False Negative Rate (FNR) and robustness. The FPR is defined as the percentage of those attacked images that can falsely pass the verification, while the FNR is the percentage of those authentic images that cannot pass the verification. In our experiment, the 20 test images are protected using our authentication system and each protected image was attacked with Adobe Photoshop in different ways. After that, the 20 protected images and the 160 attacked images are sent for verification. The FNR is always zero, but the FPR varies with the LABR. When the LABR is larger, the probability of undetected attack becomes smaller, because a larger LABR indicates high protection strength.

In summary, the LABR is a very important parameter that heavily influences the performance of the proposed authentication system in terms of the FPR, FNR and robustness of signature. In particular, when LABR is smaller, it has higher false positive rate and higher robustness, and vice versa.

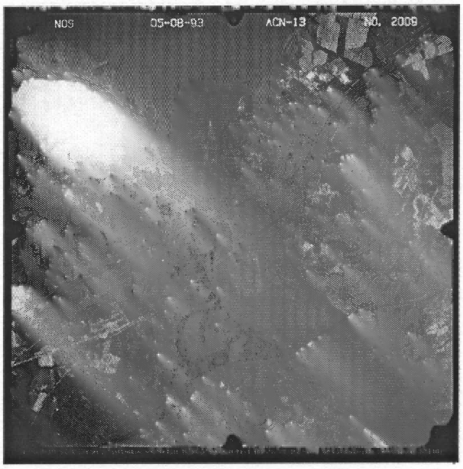

Figure 5.3 Original image.

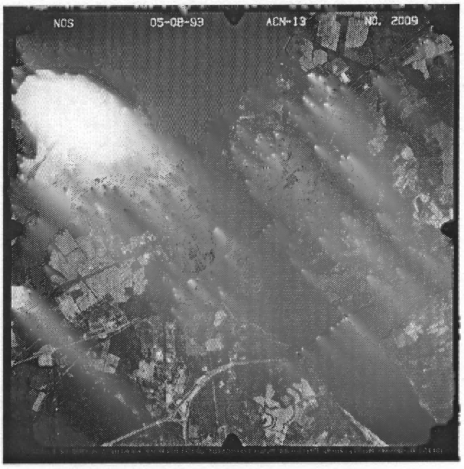

Figure 5.4 Watermarked image.

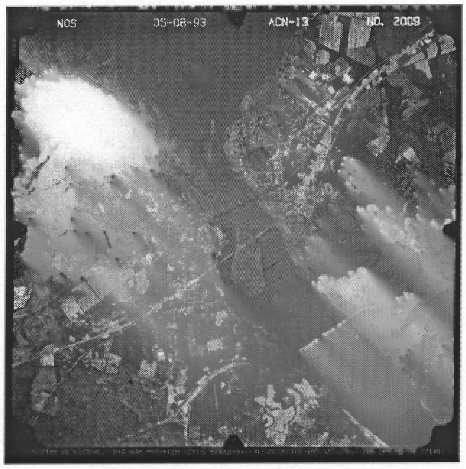

Figure 5.5 Recovered image.

CHAPTER 6

CONCLUSION

In this thesis, a digital signature and watermarking based authentication system for JPEG2000 images were described, which is able to protect JPEG2000 images in different flavors, including fragile authentication and semi-fragile authentication. The fragile authentication is to protect the image at code-stream level, and the semi-fragile is to protect the image at the content level.

Firstly, the new still image compression standard, JPEG2000, was studied.

Secondly, both semi-fragile authentication and fragile authentication were introduced, which give users more freedom to choose a proper type of authentication according to their requirements of the applications.

Finally, a new parameter called Lowest Authentication Bit Rate (LABR) to quantitatively control the security strength and robustness is introduced. With the lossless mode of semi-fragile authentication, the original image can be recovered exactly after verification if there is no alteration occurring to the stego-image.

REFERENCES

1. M. D. Adams and F. Kossentini, "Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis", IEEE Trans. Image Processing 9 (6) (June 2000) 1010-1024.

2. M. D. Adams, H. Man, F. Kossentini, and T. Ebrahimi, "JPEG2000: The Next Generation Still Image Compression Standard", ISO/IEC JTC1/SC29/WG1 N1734, June 2000.

3. Majid Rabbani and Rajan Joshi, "An Overview of the JPEG2000 Still Image Compression Standard", ISO/IEC JTC 1/SC 29/WG1N2233, July 9, 2001.

4. M. D. Adams and F. Kossentini, "A Software-Based JPEG-2000 Codec Implementation", IEEE Int. Conf. Image Processing, Vancouver, CA, September 2000.

5. C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 Still Image Coding System: An Overview", IEEE Trans. On Consumer Electronics 46 (4) (November 2000) 1103-1127.

6. C. Chrysafis and A. Ortega, "Line-Based, Reduced Memory, Wavelet Image Compression", IEEE Trans. Image Processing 9 (3) (March 2000) 378-389.

7. W. B. Pennebaker, J. L. Mitchell, G. G. Langdon Jr., and R. B. Arps, "An Overview of the Basic Principles of the Q-coder Adaptive Binary Arithmetic Coder", IBM J. Research Development 32 (6) (November 1988) 717-726.

8. J. D. Villasenor, B. Belzer, and J. Liao, "Wavelet Filter Evaluation for Image Compression", IEEE Trans. Image Processing 4 (8) (August 1995) 1053-1060.

9. Zhishou Zhang, Qibin Sun and Xiao Lin, "A Unified Authentication Framework for JPEG2000", ISO/IEC JTC1/SC29/WG1 N3074.