We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



6

Privacy-Secure Digital Watermarking for Fair Content Trading

Mitsuo Okada
Kyoto University
Japan

1. Introduction

This chapter describes a privacy-secure digital watermarking scheme for fair content trading against cybercrime on digital content piracy and privacy leakage. Conventional digital watermarking schemes are effective only for providers since privacy of a client is not concerned though content is protected. Provider's security as well as client's security need to be considered to enhance security level of privacy management. Blind and Pseudo-blind watermarking schemes utilizing encryption and media processing respectively are the solutions to protect client's security.

In conventional digital watermarking, embedding and extracting are carried out by the same providers. Therefore, a malicious provider could provide fake extracted results to a client. Verification result is considered as trustworthy evidence under the assumption that a provider is trustworthy since most of the providers were owned by enterprises which has relatively high social credibility. However, anyone is able to become a provider these days along with development of technologies. Therefore, improper providers which have insufficient level of knowledge or skill would manage client's privacy information. In other words, a user has to own his/her risk on the privacy information.

Privacy secure watermarking techniques, blind watermarking (Iwamura et al., 1997; Okada et al., 2008) and pseudo-blind watermarking (Okada et al., 2009) are able to protect both content and privacy for the sake of both providers and clients. Blind watermarking based on cryptography and watermarking blinds up content by encryption against a provider to conceal content information. However, the downside is incompatibility of cryptography and watermarking in terms of robustness of watermark and processing cost of cryptography. Pseudo-blind watermarking is another approach that uses media processing to blinds up content instead of using cryptography which enhances compatible with watermarking. The pseudo-blind watermarking provides better performance in robustness and processing cost potentially since media process is compatible to watermarking. The technical detail, features, performance evaluations of both schemes are described in this chapter based on our experimental results.

1.1 History of digital content

Analog content has been alternatively replaced by digital content such as picture, music, movie and book. Analog and digital has completely different features. For example, in

analog content such as painting on a canvas, only similar replica or picture which is obviously different from the original piece can be generated instead of making perfect copy. Alteration of content is also easily identified from the original one. On the other hand, digital content can be easily duplicated without any degeneration. Analog content is recorded on physical media while digital content is only data which can be output by a monitor or speaker. Therefore, digital content can be easily distributed to thousands of people at once through the Internet.

The features of digital content, easiness of duplication, alteration and distribution are practical in terms of productivity. For instance, manufacturing and delivering movie through the Internet is much easier and inexpensive than distributing packaged DVD sold in retail stores. Clients also get benefit because they don't have to go to the store, confirm stocks and almost no space is needed to store the purchased content. However, digitalizing content involves many issues regarding to illegal use such as piracy.

In addition to content protection, privacy of a purchaser also needs to be considered. Purchasers (clients) are able to purchase analog content without exposing privacy such as who bought what kinds of content if payment is made by cash. However, purchasing content through the Internet requires user registration containing privacy information which may involve privacy leakage.

1.2 Risk of digital content piracy

Recently, with a rapid development of IT infrastructure, all kinds of digital content can be purchased through the Internet such as music, image, movie, and book as shown in Fig.1. However, an enormous amount of digital content might have been pirated since they can be easily duplicated and distributed through the Internet. In fact, an amount of distributed analog audio content such as CD was peaked out in 1998 toward decreasing as shown in Fig.2 where quantities of distributed content is shown. Note that bars labeled as "CD" in the figure show the amount of CD sold in the store while the other bars show amount of digital music data based on downloaded counting. Decreasing may be because of illegal file sharing using file sharing applications. Napster which had been used to share music content might have been accelerated piracy in 1999. After that, P2P applications such as WinMX Winny and Cabos had been alternatively used for illegal file sharing. Increasing online distribution of music content indicates importance of content protection against piracy of audio data. Because of the property of digital content, much unintentional crime might have been occurring. For example, many people share files illegally using P2P applications without conscious of guilty.

Cryptography or information hiding is digital content protection techniques against piracy. Cryptography encrypts entire content to protect content, but it will be exposed as no protection when decrypted. For example, movie delivered by CATV is encoded by scrambling and then a set top box decodes the encrypted content to play the content. The content is secure as long as encrypted, but once it is decrypted, it would be exposed as no protection (Fig. 3).

Another effective protection technique is digital watermarking that makes some secret data concealed in content. The hidden information, watermark is used for copyright protection, tamper detection, covert communication, source tracking of leakage, and so forth. The ideal form of watermark for copyright claiming is the one in which watermark should not be removed by any manipulations, the watermarked content (endorsed content) should not be

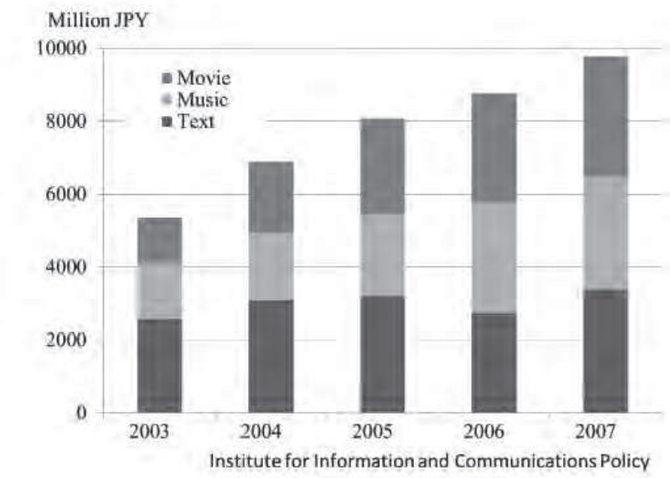


Fig. 1. Distribution Amount of Digital Content

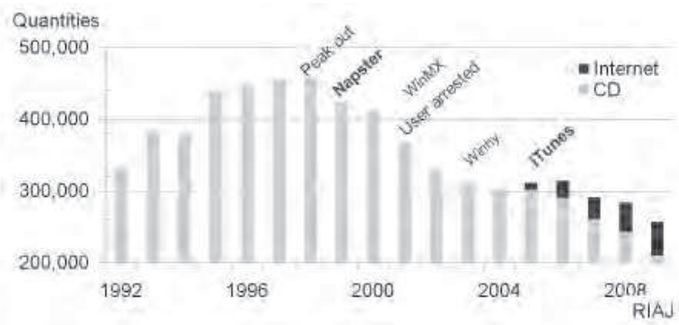


Fig. 2. Distribution Amount on Analog Music Content

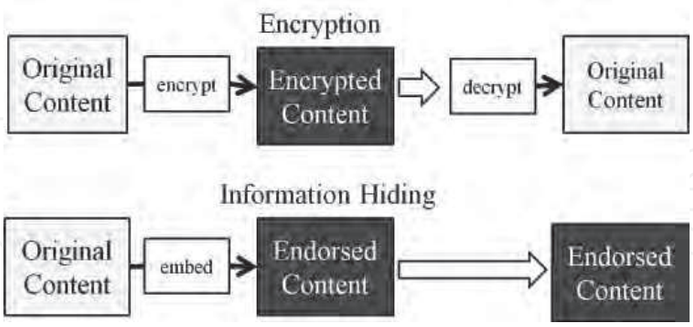


Fig. 3. Protection Techniques

degenerated by embedding watermark and embedded watermark should not perceptually appear.

Two significant issues in the conventional watermarking schemes need to be considered, that is, the provider's security as well as client's security needs to be considered for secure content trading. The issues are fairness and privacy of a client on content trading.

The demand to protect privacy information of client is increasing along with the increment of privacy leakage. Moreover, in a conventional watermarking scheme, a client can be excused from the responsibility of piracy as long as information leakage is technically possible by a provider since the provider also possesses the same delivered watermarked content. In order to resolve the problem, a fair and privacy-secure content trading framework needs to be urgently provided.

2. Content protection techniques

Protection techniques, cryptosystems and information hiding techniques are introduced in this section.

2.1 Cryptosystems

Cryptography protects content entirely by encryption. The content is secure as long as it is encrypted, but once decrypted, the content would be insecure. Cryptography is mainly classified as common key encryption and public key encryption. The former one uses the same key for encryption and decryption in which calculation cost is low, but insecure since the key is exposed when delivering it to a reviver. The latter one uses different keys for encryption and decryption respectively. An encryption key cannot be used for description which enhances security and usability. For example, assume Alice (sender) encrypts data, and Bob (receiver) decrypts it. In the common key encryption, Bob must deliver the encryption key to Alice in a strictly secure method. In other words, a key may be tapped by malicious party during the delivering process. In public key encryption, Bob prepares a pair of public key and secret key which are used for encryption and decryption respectively. Alice uses the public key obtained from Bob to encrypt data and then deliver the encrypted data to Bob. Bob decrypts the encrypted data using the secret key. The public key for encryption cannot be used for decryption and a secret key for decryption is only possessed by Bob. Hence, even though the encryption key is tapped, ciphertext cannot be decrypted.

2.1.1 Public key encryption

A public key cryptography such as RSA and El Gamal is originally proposed in 1976 by Diffie and Hellman. It has advantage in usability, but processing cost is heavy compare to the common key encryption. El Gamal (El Gamal, 1985) and Paillier encryption (Paillier, 1999) which can be adapted to the watermarking schemes are described.

2.1.1.1 El Gamal encryption

El Gamal is proposed in 1982 by El Gamal in which the security relays on the difficulty of the discrete logarithms problem. The asymmetric watermarking (Okada et al., 2008), related work of blind watermarking (Iwamura et al., 1997) uses the modified El Gamal which is customized version of El Gamal. The detail is described below.

STEP 1:(Preliminary) Bob generates a large prime number p and then finds generator g. Multiplicative group of order q on Z_p^* is also figured out. Next, determine $x \in Z_p$ and

then calculates $y = g^x \bmod p$ where

$$\begin{cases} x & \text{secret key,} \\ y, g, p & \text{public key.} \end{cases}$$

The public key needs to be shared in prior to trading.

STEP 2:(Encryption) Alice generates ciphertext $E(m) = (c, d)$ by generating a random number $r \in_u Z_q$ and then encrypts the message m using the public key as

$$\begin{cases} c = g^m y^r \bmod p, \\ d = g^r, \end{cases}$$

and then sends the ciphertext to Bob.

STEP 3:(Decryption) Bob decrypts ciphertext (c, d) received by Alice using the secret key x as

$$g^m = D(c, d) = c/d^x \bmod p$$

to obtain m.

2.1.1.2 Paillier encryption

Paillier encryption (Paillier, 1999) is another homomorphic encryption which can be used with watermarking. In a key generation phase, two large prime numbers p, q are generated. $g \in Z_{N^2}$ is selected such that $\gcd(L(g^\lambda \bmod N^2), N) = 1$ where $N = pq$, $\lambda = \text{lcm}(p-1, q-1)$. Note that a public key is g, N and a private key is p, q. For the encryption phase, let m be plaintext to be encrypted, r be a random number chosen from Z_N, and $E(\cdot)$ be an encryption function defined by

$$e = E(m) = g^m r^N \bmod N^2. \tag{1}$$

For decryption phase, the decrypted ciphertext m' is obtained by

$$m' = D(e) = \frac{L(e^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N \tag{2}$$

where $L(t) = (t-1)/N$ and $D(\cdot)$ is decryption function.

The modified El Gamal and Paillier cryptography satisfy both an additive homomorphism and an indistinguishability denoted by IND [1] which are requirement to be utilized with watermarking. IND is necessary since the only three kinds of plaintexts $(-1, 0, 1)$ would be encrypted in an asymmetric watermarking protocol. Otherwise, the plaintexts can be identified from the ciphertext. The relationship between public-key algorithms and their properties are shown in Table 1.

2.2 Information hiding

Information hiding, in particular watermark has been used for verification in which banknotes are embedded such as the one in paper currency. Watermark can be extracted by anyone,

[1] A cryptosystem is secure in terms of indistinguishability if a ciphertext of given randomly chosen message m_0 or m_1 cannot be identified by any adversary.

Cryptography	Homomorphism	IND	Computation Cost
Modified El Gamal	additive, (multiplicative)	YES	low
Paillier (Paillier, 1999)	additive	YES	high
Okamoto- Uchiyama (Okamoto et al., 1998)	additive	YES	high
RSA	multiplicative	NO	low

Table 1. List of Public-key Algorithms

but difficult to regenerate it. Information hiding had been used for covert communication techniques such as military, diplomacy, spy and so forth that enable to conceal the existence of confidential communication. For example, a sender uses special ink to embed invisible secret message in blank space in the letter which disappears after a certain amount of time to ensure security. The secret message can be extracted only by an authorized receiver who has special liquid. This conceals existence of communication to the others.

Information hiding in digital format embeds secret message in digital content such as images, music, movies, text and so forth. The message is embedded by adding noise-like signals in content. Since the signals are so weak, only a watermark extraction program can recognize it. Basic watermarking models and features are summarized in Fig. 4.

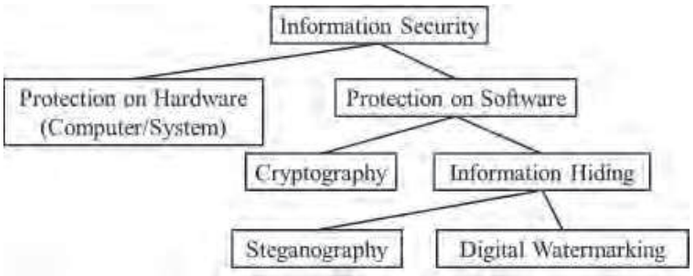


Fig. 4. Classification of Security Techniques

2.2.1 Digital image

Before we get into the watermarking technique, we overview a digital image which is composed of a huge number of small dots called "pixels" standing for a picture cell. These dots represent brightness to form an image. The brightness of Red, Green and Blue known as the three primal colors represents colors of an image. For example, *Lenna* shown in Fig. 25 is composed of 512×512 dots in column and row respectively and 8-bit, 256 levels of brightness.

An index color image such as TIFF or GIF uses less color valuation for smaller data size than full color images. Therefore, they are mostly used for website where quick response is required. Gray-scale images composed of single color valuation, black-white is smaller in file size which are used for surveillance cameras where color information is not needed. Binary images represented in either black or white are used for copy machines and FAX because they were not capable of processing color information when they were invented.

2.2.2 Frequency domain

In this section, we describe frequency which is indispensable on compression for multimedia data. Frequency is very familiar term for audio data. Image can be also represented in

frequency component. For example, hair parts in *Lenna* which are considered as complicated area contains high frequency component while flat area such as skin area contains low frequency components.

Multimedia data is often compressed by using frequency component since complicated area where information omitting is hardly recognized can be effectively selected. For example, little noise in hard and noisy parts in music is hardly recognized. Compression formats in images are JPEG, GIF and TIFF. In audio, MP3 and WAV and in movie, MPEG2, MPEG4 and MKV are the major ones.

Frequency domain is more effective for watermark to be embedded because the one embedded in spatial domain somewhat damaged when compressed. However, watermark is hardly influenced if embedded in frequency domain.

An example of watermark embedding process is briefly described below as shown in Fig.5. Following descriptions show rough embedding techniques in spatial domain using x, y coordinates. Watermark is embedded by modifying brightness in either odd or even number. For example, if $\omega = 0$, the value would be changed to odd number. Otherwise, it would be changed to even numbers. Small change in brightness is perceptually unnoticeable. This method is fragile against manipulations, but embedding in frequency domain provides more robust watermark.

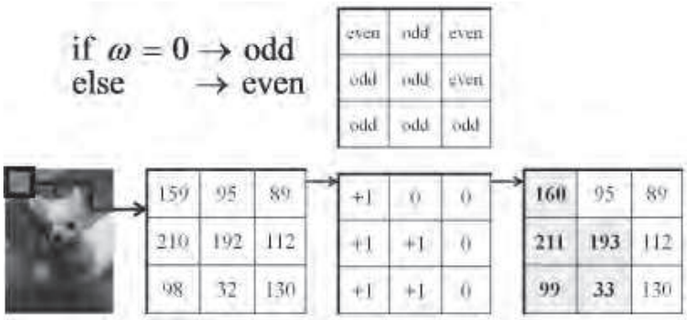


Fig. 5. An Example of Watermark Embedding Procedure

2.2.3 Digital watermark scheme

Digital watermark protects digital content by embedding imperceptive message such as serial numbers, a client ID and copyright notice into the content. The watermark is hardly removed from the watermarked content. If the pirated content had been found, the provider would extract the message to claim the copyright. Watermark is required to be imperceptive and robust against manipulation attacks which is in trade off because embedding robust watermark causes more degradation. If imperceptiveness is increased, watermark becomes fragile.

2.2.4 Potential attacks

Watermark could be removed by manipulations such as compression, re-sizing, rotation and so forth. Stirmark Benchmark Tool is a common benchmark tool for robustness of watermark. Stirmark applies various manipulations to a watermarked image based on media processing

and then output attacked images. Robustness is verified by extracting watermark from those attacked images. Alternatively, this tool is called "Dewatermarker" indicating watermark remover. Stirmark covers all major attacks as shown in Table 2.

Methods	Detail
EmbedTime	Overwrite watermark
AddNoise	Noise addition
JPEG	JPEG compression
MedianCut	Smoothing without change of size and channel
ConvFilter	Modification of low-pass, sharpness, histogram
RemoveLines	Remove lines in vertical and horizontal
Cropping	Crop a part of an image
Rescale	Re-scaling
Rotation	Rotating
Affine	Twisting an image horizontally and vertically

Table 2. Attacks in StirMark Benchmark Tools

2.2.5 Tamper detection using fragile digital watermark

Fragile watermark is used for integrity check. If watermark had been damaged, it would indicate that noise or alteration had been added to the watermarked content. Cropping detection techniques based on fragile watermark (Lin et al., 2000) has been proposed in which alternated area can be detected if watermark extracting is failed from the area.

2.2.6 Steganography

Steganography is used for covert communication by hiding message in the dummy content as watermark. For example, assume Alice wants to send strictly confidential message to Bob. Alice embeds very fragile watermark to dummy content. If the content had been attempted to extract watermark while sending, the watermark must be disappear to protect message. In this method, increasing message (watermark) length without degeneration is required.

2.2.7 Digital fingerprinting

In a fingerprinting technique, a provider embeds a unique client ID for every client. If pirated content had been found, a provider would extract watermark from the content to track the source. This deters illegal re-distribution or unintentional leakage. In this technique, only few message for a user ID needs to be embedded robustly against manipulations.

2.3 Extended version of watermarking

2.3.1 Statistical watermarking

Patchwork watermarking (Bender et al., 1995), one of the statistical watermarking, embeds message in statistical value of contents. In this method, an embedding key is a seed of pseudo-random process which chooses a large number of pairs of pixels. Brightness values in the pairs are made slightly brighter and darker for all pairs. Conceptually, the contrast between pixels of the pairs encodes some secret information.

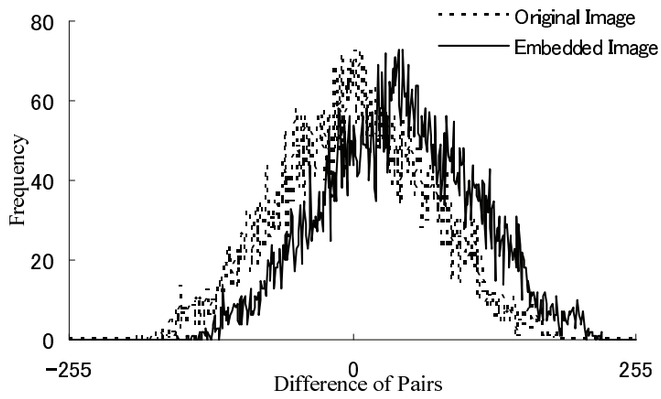


Fig. 6. Distributions of Differences $(a_i - b_i)$ and $(a'_i - b'_i)$

The extraction is carried out by finding the same pairs of the pixels chosen in the embedding process and analyzing the difference of their brightness values for all pairs. This provides invisible watermark that has a higher degree of robustness against attacks and image manipulations.

A single-bit embedding process of patchwork watermark is described below. First, choose a large number of pairs from an original image I and then obtain difference in each pair. Let a, b be the first and second pixel of a pair, and S_n be the sum of $(a_i - b_i)$ for n pairs, i.e.,

$$S_n = \sum_{i=1}^{n} (a_i - b_i).$$

Let $\bar{S_n}$ be an expected value defined by $\bar{S_n} = S_n/n$. Note that $\bar{S_n}$ approaches 0 as n increases,

$$\lim_{n\to\infty} \bar{S_n} \to 0. \tag{3}$$

A distribution of differences in *Lenna* (256 × 256 pixels, 256 gray scale levels) with $n = 10000$ is shown in Fig. 6 ("Original Image"). At this experiment, a statistical value of an original image would be $\bar{S_n} = 0.0121$, that satisfies the condition (3).

An embedding process, hiding a secret message ω into I is described. First, choose a seed of pseudo-random sequence to assign two pixels (a_i, b_i) for n pairs. Next, to generate an embedded image I', we modify the assigned pixels as, $a'_i = a_i + \delta$, and $b'_i = b_i - \delta$, for $i = 1, \ldots, n$, where δ is a constant that governs robustness of the watermark. Note that the expected value $\bar{S_n}'$, an average of sum of the difference of the embedded image I', approaches 2δ as

$$\bar{S_n}' = \frac{1}{n}\sum_{i=1}^{n} (a_i + \delta) - (b_i - \delta) = \frac{1}{n}\sum_{i=1}^{n} (a_i - b_i) + 2\delta = 2\delta. \tag{4}$$

with the parameter of $\delta = 20$, the distribution of $(a'_i - b'_i)$ is shifted 40 to right as illustrated in Fig. 6. Hence, as δ goes larger, accuracy of detection increases, and as δ goes smaller, the risk of a false detection increases. To extract the hidden message ω, choose a'_i, and b'_i according to

the random numbers, and then determine,

$$\omega = \begin{cases} 0 & \bar{S_n}' < \tau, \\ 1 & \bar{S_n}' \geq \tau, \end{cases} \tag{5}$$

where τ is a threshold. The optimal threshold is given as $\tau = \delta$ to equalize the false positive and false negative. In the sample image *Lenna*, statistical value is $\bar{S_n}' = 40.0158$, which satisfies the condition of $\bar{S_n} \geq \tau = \delta = 20$.

3. Secure watermarking technique

Consideration of purchaser's privacy is another important issue for fair and secure content trading. Blind and Pseudo blind schemes are solutions to enhance the privacy protection. The big picture of the secure content trading based on three-way communication by interposing TTP (trusted third party) are introduced below (Fig. 7) which are non-blind, blind, pseudo-blind watermarking techniques (Table 3).

The information to be protected is privacy of a client such as who purchased what kind of content. Even though, TTP is trustworthy, there is demand that a client doesn't want to expose privacy information.



Fig. 7. Big Picture of Secure Content Protection Techniques

3.1 Non-blind watermarking

TTP is interposed for watermark embedding and pseudonymous user verification which is the simplest way to protect privacy against a provider (Fig. 8). However, this scheme doesn't fulfill the needs since content is exposed to TTP. A client doesn't want to expose unnecessary information to TTP even though TTP is trustworthy.

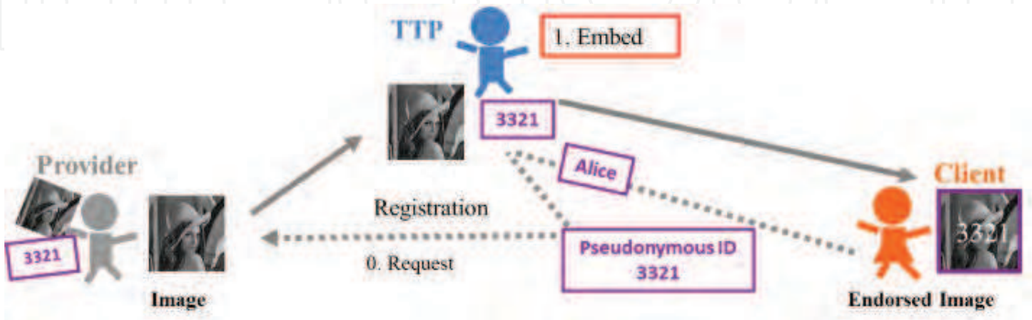


Fig. 8. Non-blind Watermarking

3.2 Blind watermarking, combination of homomorphic encryption and watermark

Blind watermarking blinds up content by encryption against TTP. In this scheme, watermark is able to be embedded even though the content is encrypted. Therefore, TTP cannot obtain information of content. However, it is inefficient in terms of watermark strength and processing cost.

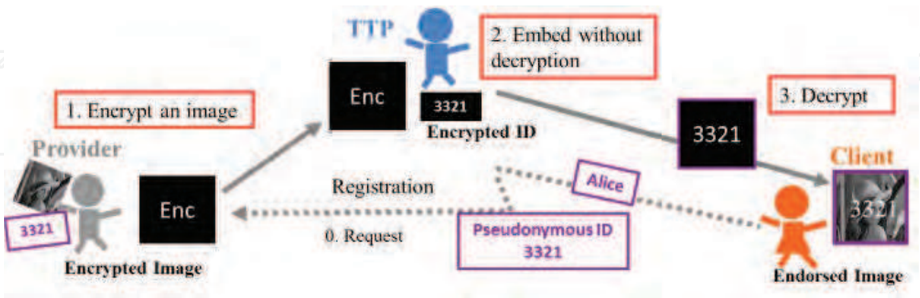


Fig. 9. Blind Watermarking

3.3 Pseudo-blind watermarking, combination of media processing and watermark

Pseudo-blind watermarking (Okada et al., 2009) is an alternative method of blind watermarking which is as secure as blind watermarking and as practical as a non-blind watermarking. The pseudo-blind watermarking partially scrambles the content so that content is blinded against TTP. At the same time, watermark is well embedded since scrambling is designed to preserve feature of the content such as edge of recoded subject where watermark is embedded. Hence the embedded watermark has sufficient level of robustness. A prototype of a content trading system based on the pseudo-blind method has been designed and implemented, and the performance of the pseudo-blind watermarking is evaluated on the system.

The scheme is briefly described below (Fig.10). In prior to trading, a client obtains a pseudonymous ID from TTP. The client requests content by using the ID. If verified, the provider decomposes requested content into two pieces. One of which is blinded by media processing which contains sufficient amount of image feature. Another one is the counterpart. The former one is sent to TTP for watermark embedding and latter one is delivered to the client. At this point, the provider has no information to profile a client because of pseudonymity. Next, TTP embeds watermark into blinded piece (endorse piece) and then delivered to the client. At this point, TTP has no clue as to what kind of content has been traded due to blindness. Finally, the client integrates those decomposed pieces to obtain complete endorsed image. Hence, the client can obtain a complete endorsed image without exposing the privacy information.

4. Verification and performance evaluation

4.1 Performance summary of blind method

Asymmetric watermarking (Pfitzmann et al., 1996) is one of the related works of blind watermarking. Fundamental problems of the blind schemes are specified with our

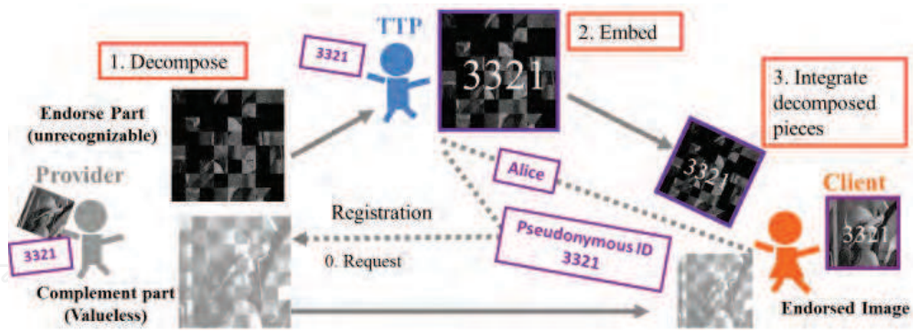


Fig. 10. Pseudo-blind Watermarking

	Non-blind	Blind	Pseudo-blind
Compatibility of watermark and blinding method	applicable to any watermark algorithms	applicable to certain watermark algorithms	applicable to any watermark algorithms
Privacy against TTP	No protection	Blinded by encryption	Blinded by Media processing (scrambling)
Watermark strength	Robust	Fragile	Robust

Table 3. Comparison of Three-way Communication Content Trading

implemented results (Okada et al., 2008) that uses El Gamal encryption and patchwork watermarking.

Suppose that, a provider embeds watermark into content, a client verifies watermark, and TTP generates a secret key sk and public key pk for the modified El Gamal encryption. Not only does interposal of TTP enhances the reliability of verification, but also prevents a provider from cheating a client. Note that TTP needn't to be fully trustworthy since it does not obtain the embedding key, which is the index of modified pixels determined by a client throughout the embedding process.

Let $I = (x_1, \ldots, x_\ell)$ be an original image, $I' = (z_1, \ldots, z_\ell)$ be an embedded image, and ℓ be the number of pixels in I and I'. An asymmetric watermarking scheme is illustrated in Fig. 11.

4.1.1 The asymmetric protocol

TTP generates the modified El Gamal public key, $y = g^x \bmod p$, where a secret key is x. Let EXT be conversion function in the second step, and $IDENTIFY$ be a function to obtain ω at the final step, respectively.

STEP1:(Embedding) A client generates random numbers by giving a seed to pseudo-random generator, and obtains subsets A and B of set of indexes $\{1, 2, \ldots, \ell\}$ such that $A \cap B = \phi$ and $|A| = |B| = n$. The client chooses δ and modifies pixels according to (A, B) in the image I to generate I' as

$$z_i = \begin{cases} x_i + \delta & \text{if } i \in A, \\ x_i - \delta & \text{if } i \in B, \\ x_i & \text{otherwise,} \end{cases} \tag{6}$$

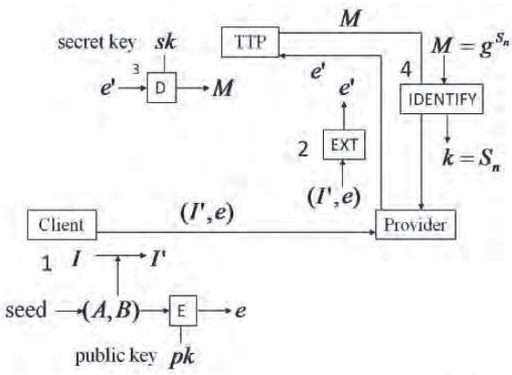


Fig. 11. The Model of the Asymmetric Digital Watermarking

for $i = 1, ..., \ell$. A client computes e, a ciphertext of (A, B) as $e = (c_1, \ldots, c_\ell, \ d_1, \ldots, d_\ell)$, where $c_i = g^{m_i} y^{r_i}$, $d_i = g^{r_i} \bmod p$,

$$m_i = \begin{cases} 1 & \text{if } i \in A, \\ -1 & \text{if } i \in B, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

and r_i is random numbers of Z_q, for $i = 1, \ldots, \ell$. Finally, a client sends $I' = (z_1, \ldots, z_\ell)$ to the provider in conjunction with encrypted indexes $e = (c_1, \ldots, c_\ell, d_1, \ldots, d_\ell)$.

STEP2:(Extracting) The provider computes ciphertext $e' = EXT(I', e) = (C, D)$ as follow;

$$C = c_1^{z_1} c_2^{z_2} \cdots c_\ell^{z_\ell} = \prod_{i=1}^{\ell} g^{m_i z_i} y^{r_i z_i} = g^{\sum^{\ell} m_i z_i} y^{\sum^{\ell} r_i z_i} = g^{S_n} y^R, \tag{8}$$

$$D = d_1^{z_1} d_2^{z_2} \cdots d_\ell^{z_\ell} = \prod_{i=1}^{\ell} g^{r_i z_i} = g^R,$$

where $R = \sum_{i=1}^{\ell} r_i z_i \bmod q$, and S_n is the sum of difference in patchwork watermark scheme, i.e., $S_n = 2n\delta$ and then sends e' to TTP.

STEP3:(Decrypting) TTP uses its private key x to decrypt $e' = (C, D)$ as $M = D(e') = C/D^x = g^{S_n}$ and then sends back the decrypted text M to the provider.

STEP4:(Identifying) The provider identifies exponent h of M as $IDENTIFY(M)$ such that $M = g^h$ by testing all possible $h = 1, 2, \ldots, n\tau$. Statistically h is distributed around $2n\delta$, which is much smaller than q, and thus able to be identified. The hidden message ω is obtained according to

$$\omega = \begin{cases} 0 & \text{if } h < n\tau, \\ 1 & \text{if } h \geq n\tau, \end{cases} \tag{9}$$

where τ is the threshold. Determine $\omega = 1$, if there is no value matching within the range, $h < n\tau$. Sum of difference, h to form Eq. (9) instead of the average $\bar{S_n}$ in Eq. (5) is used. Note that Eq. (9) is equivalent to Eq. (5).

In other words, $\omega = 0$ does not mean that watermark is not embedded. Difference whether $\omega = 0$ or none can be examined by adopting some optional techniques. One example is that,

we assign $\zeta = -1(\omega = 0); 1(\omega = 1)$ as

$$z_i = \begin{cases} x_i + \delta\zeta & \text{if } i \in A, \\ x_i - \delta\zeta & \text{if } i \in B, \\ x_i & \text{otherwise,} \end{cases}$$

which is based on Eq. (6). The above modification provides three conditions such as $\omega = 0$, $\omega = 1$, or none (message is not embedded).

4.1.2 Security

In this section, the security of patchwork watermark is described. First, the embedding key A and B, the indexes of the modified pixels are uniformly distributed over $\{1, \ldots, \ell\}$. The distribution of (A, B) is illustrated in Fig. 12, where white dots represent (A, B). Hence, it is almost impossible to attack to determine (A, B) in I' without the knowledge of the embedding key. Second, the property that the original image is not required in an extraction process improves security against watermark removal due to a leakage of the original image. Third, since the brightness of some of the pixels has slightly changed, the difference is hardly perceptible.



Fig. 12. 1-bit Embedded Images and Distribution of A, B

Fig. 12 illustrates an example of a single-bit information being embedded into *Lenna* (256×256 pixels, 256 gray scale levels) with the parameters of $n = 2053$, and $\delta = 3$. The SNR for Fig. 12 is 50.6[dB] which is considered to be acceptable.

4.1.3 Optimal parameter

In this section, an optimal parameter δ is described in the sense that the least number of δ with an accuracy of 95% succeeds in detection.

Let σ' be standard deviation of n samples of $(a_i - b_i)$, and σ be standard deviation of the average value $\bar{S}_i$. Noting the well-known relation of variances, $\sigma = \sigma'/\sqrt{n}$, we can predict true σ from the sampled σ'. Hence, variance of average S_n decreases as n increases. In other words, an accuracy of S_n increases along with the increment of n. In order to achieve 95% confidence for detection, under an assumption of normal distribution, the embedded image should be shifted by at least 2σ which is identical to δ.

The parameters, average of S_n, μ, standard deviation σ, and optimal δ with respects to n are demonstrated on Table 4, and the optimal δ given n is obtained from Fig. 13. Note that the false positive of 5% with the following δ is not sufficient to practical use. In order to make an

image more robust, δ could be increased taking consideration of subjective evaluation. For

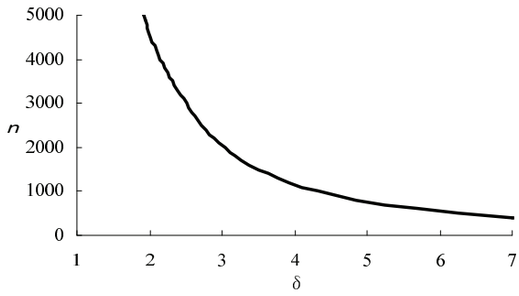


Fig. 13. Optimal δ Distribution

n	μ	σ'	σ	δ
4613	0.8847	67.4449	0.4769	2
2053	1.9206	67.9670	1.5000	3
1165	-0.4335	68.2865	2.0007	4
757	-1.3805	68.8136	2.5011	5
539	-2.0260	69.7601	3.0048	6

Table 4. Parameters for δ Determination

the sake of determination of δ, we study the relation between the number of modified pairs of pixels n and quality of an image, which is estimated by means of Signal to Noise Ratio defined by,

$$\mathrm{SNR} = 10 \cdot \log_{10} \frac{255^2}{\mathrm{MSE}^2} = 10 \cdot \log_{10} \frac{255 \cdot 255}{1/\ell \sum (x_i - z_i)^2}, \tag{10}$$

where MSE is the mean-square error between I and I'. An image *Lenna* of 256×256 pixels is used for this test with the parameters shown in Table 4. Fig. 15 indicates no significant difference between $n = 2053$ and $n = 4613$. This implies the parameter of $n > 2053$, which is $\delta = 3$, is the optimal choice to prevent the embedded image from being spoiled, under the condition that SNR is almost the same. Fig. 14 illustrates how SNR of the image varies for the image size ℓ, where single-bit is embedded and $n = 2053$ pixels are manipulated.

4.1.4 Implementation system

In order to estimate a total performance of asymmetric schemes is described below. Watermark embedding and extracting process for gray scale images are implemented in C, and cryptographic computations are implemented in Java. Environment specifications are described in Table 5. An image *Lenna I* (256 × 256 pixels) with a parameter of $n = 2053$ is used as a host image.

Based on our implementation, we have estimated embedding time and extracting time. Description and decryption time of a single bit embedding based on the 1024-bit modified El Gamal are 0.104 [s], and 0.077 [s], respectively. Those of Paillier encryption are 3.303[s] and 2.127[s].

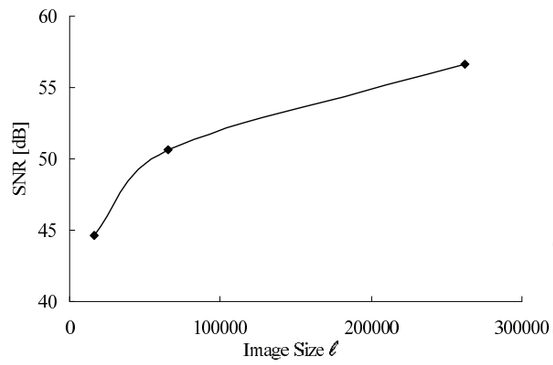


Fig. 14. SNR for Different Image Size ℓ

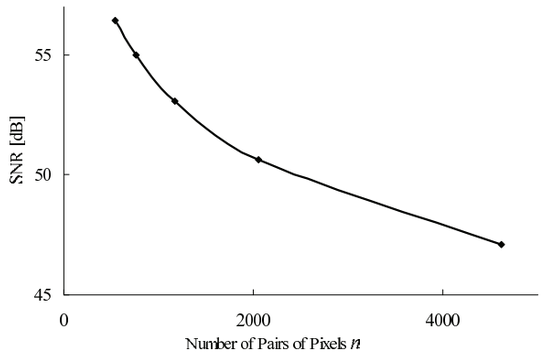


Fig. 15. The Relation between the Number of Modified Pairs of Pixels n and SNR

Detail	Specification
CPU	Xeon 2.3GHz
OS	Redhat 9.0, Linux 2.4.20
Memory	1GB
Encryption Algorithms	1024-bit the modified El Gamal, 1024-bit Paillier
Programming Languages	J2SDK 1.4.2, gcc 3.3.3

Table 5. Implementation Environment

4.2 Robustness against noise addition and JPEG compression attacks

The robustness of patchwork watermarking against attacks of "Add Noise" and "JPEG Compression" using StirMark (Petitcolas, 2000) are evaluated. I' originated from *Lenna* (256×256 pixels, 256 gray scale levels), with the parameters of $n = 2053$, $\delta = 3$, and $\bar{S}'_n$=6.9547. With this sample image, the parameter of τ=3 for all attacked images I' is applied on extraction process.

In JPEG compression attack, watermark has been successfully extracted up to 80% of JPEG quality level as shown in Fig. 16. Evaluation result in Add Noise attack is shown in Fig. 16.

The noise level represents that of normalized from 0 to 100 such that 0 gives no noise while 100 gives a complete random image.

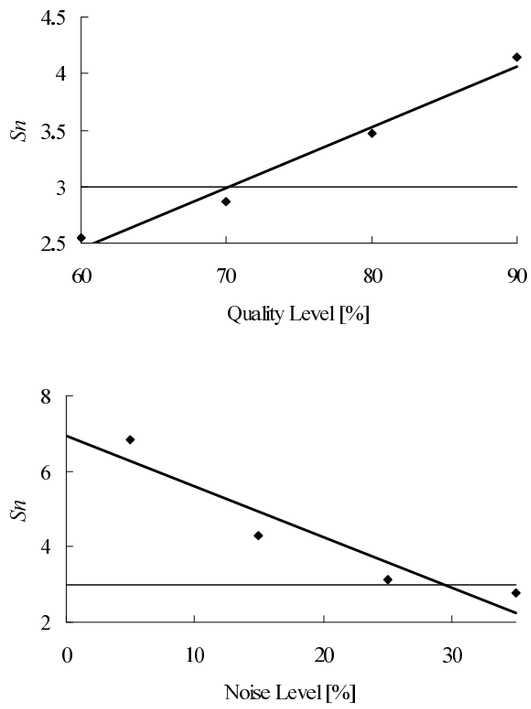


Fig. 16. Robustness on JPEG Compression and Add Noise Attacks

4.2.1 Comparison between Furukawa's method and the proposed scheme

Essential difference between Furukawa's scheme (Furukawa, 2004) and the proposal scheme comes from the cryptographical primitives, that is, the modified El Gamal and Paillier encryption. Fig. 17 shows the processing time of an extracting phase in the modified El Gamal and Paillier encryptions. Processing time for all cases is evaluated. Each of cases is provided average of ten samples of different seeds. The values used to plot in Fig. 17 are shown in Table 6.

For the modified El Gamal encryption, the processing time includes decrypting and identifying process, whereas Paillier encryption includes only decrypting process. The processing time of the modified El Gamal increases proportionally to n while processing time of Paillier encryption remains the same since only single decryption process is needed to extract watermark.

Supposing the processing time follows linearly to n as illustrated in Fig. 17, Paillier processing time would crosses over that of the modified El Gamal at $n^* = 7403$. This result shows that the scheme (Okada et al., 2008) is superior to Furukawa's method (Furukawa, 2004) with the condition when n is less than or equal to n^*.

For the modified El Gamal encryption, it is necessary to examine all possible numbers, which feasibility is stated in section 4.2.1. Whereas, brute force analysis is not necessary in Paillier

encryption since exponent can be figured out. Thus, processing cost is the same as encoding value of base ϕ in Paillier encryption.

We recall that as n increase, the detection accuracy improves, but the quality of the image becomes low. According to the section 4.1.3 where we studied the optimal n and δ in terms of SNR, efficient embedding n is estimated between the number of approximately, 2000 to 5000, which is less than threshold $n^* = 7403$.

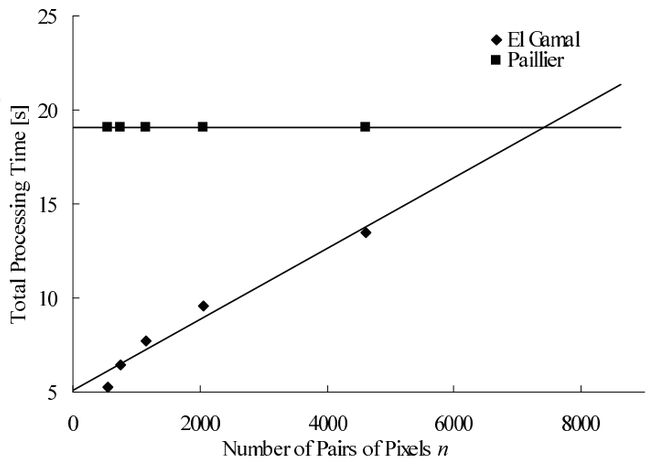


Fig. 17. Processing Time of Proposed Scheme and that of (Furukawa, 2004)

n	539	757	1165	2053	4613
Proposed scheme (the modified El Gamal)	5.279	6.475	7.697	9.590	13.47
Furukawa's scheme (Paillier)	19.11	19.11	19.11	19.11	19.11

Table 6. Processing Time in Watermark Detecting

4.3 Performance summary of pseudo-blind method

In this section, a basic model of practical privacy-secure image trading system based on pseudo-blind watermark is presented. Abstract of implemented system is described with an illustration in Fig.19. Image decomposition and watermark algorithms can be substituted according to the requirement of content trading. The details of image decomposition, embedding process, and integration process are described in Step 1 through 3 respectively in Fig. 19. For the implementation, an image of 512×512 pixels and 256 gray levels, 11-bit of ID, and 15-bit ω which includes codeword for ECC are used.

4.3.1 Procedures in the prototype

The procedure of the prototype which mainly contains 3 steps is described below.

4.3.1.1 Verification procedure

A client obtains a pseudonymous ID ID through registration page provided by TTP and then get verified by a provider. The provider verifies ID in cooperation with TTP by sending ID to TTP. If verified, TTP returns the verification result to the provider.

At this point, TTP possesses the client name and the anonymous ID while the provider only possesses the anonymous ID and purchasing history. Therefore, TTP has difficulty to profile what kind of image has been purchased while the provider has no clue as to who the client is.

4.3.1.2 Purchasing procedure

The purchasing procedures is described below (Fig. 18). Assume that the client had been successfully verified anonymously. A client selects an image. Trading procedure which contains image decomposition and watermark embedding process are executed and then two decomposed images are generated. The client receives these two images, an endorse piece and complement piece from TTP and the provider respectively. The images are integrated to be a complete endorsed image as described later on.

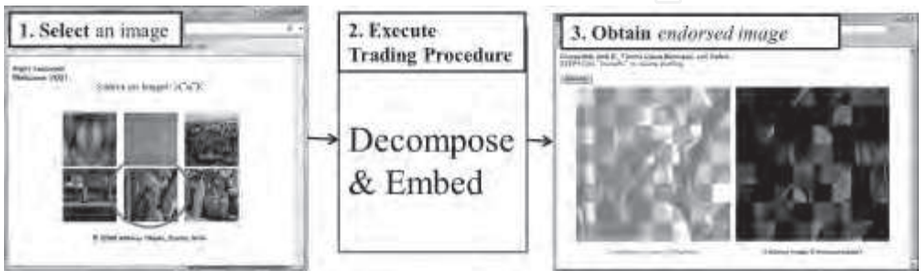


Fig. 18. Purchasing Procedure

4.3.1.3 Trading procedure

The trading procedure is briefly described below. The following instructions can be referred to Fig. 19. We assume that the client has selected an image.

1. A provider receives HTTP post from a client which contains *ID* and information of selected image such as image ID.
2. When the provider receives the data, the selected image is decomposed into a complement piece (I_c) and an endorse piece I_e as $(I_c, I_e) = DCMP(I)$. I_c is allowed to be accessed by the client, whereas I_e is allowed to be accessed by TTP. I_e is number of small $bc \times bc$ pixels of blocked images, $(I_{e_1}, \ldots, I_{e_{bn}}, bn = (Col/bc \times Raw/bc))$ as shown in the figure. In this implementation, $bn = 64 = (512/64 \times 512/64)$ of small blocked images are generated from a 512×512 pixels image.
3. The provider returns HTML content as the response to the client. The HTML content contains links to I_c and I_e. Former one is a single link to I_c while the latter one contains multiple links to small blocked images $(I_{e_1}, \ldots, I_{e_{64}})$. The provider generates a shuffle key *psk* in order to send the small blocks to TTP at random order. I_c and *psk* is sent directly to the client. I_e is sent to TTP at random order.
4. When TTP receives the blocked images, $(I_{e_1}, \ldots, I_{e_{64}})$ from the provider, TTP embeds watermark ω into the blocked images and then the images are forwarded to the client.
5. The client obtains randomly shuffled I_e from TTP and I_c and *psk* from the provider by accessing the links in the HTML content. Finally, the client integrates two images together.

The final step is generating a complete endorsed image by the client as $I' = INTG(I_c, I'_e)$ where $INTG(\cdot)$ is an image integration function.

4.3.2 Technical detail of the prototype

This scheme is mainly composed of verification, trading, and image integration procedures. In the verification procedure, a client obtains pseudonymous ID (*ID*) from TTP. The client begins trading using *ID*. A provider verifies *ID* in cooperation with TTP. If verified, the provider decompose I into an endorsed part I_e and a complement part I_c and then sends HTML content to the client (Step 1 in Fig.19). A client accesses the links to obtains I_c and shuffle key *psk* from a provider, and then sends request for I_e. I_e is composed of number of divided $zCol \times zRow$ images. As soon as TTP receives the request, TTP obtains divided I_e in which ω will be embedded (Step 2 in Fig.19). The client receives endorsed parts I'_e from TTP.

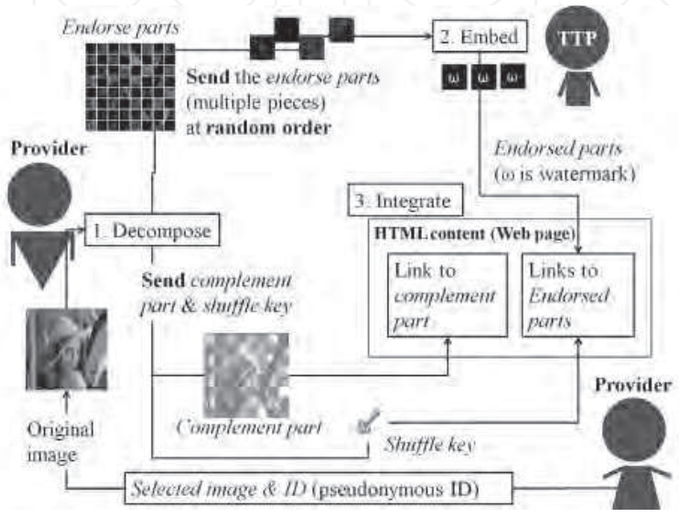


Fig. 19. Trading Procedure

4.3.3 Image decomposition

Image decomposition procedure (step 2 in Fig.19) is described below (Fig.22).

Step 1

FreQuency Decomposition extracts complicated area in an image where watermark is effectively embedded such as edge of recorded subject using high-pass filtering function $FQD(\cdot)$. $FQD(\cdot)$ should be applied to an original image in order to extract correct high-pass component. In other words, if other decomposition elements had been applied before $FQD(\cdot)$, noise or block border would affect high-pass component extraction. A high-pass filtered image is generated as $I_H = FQD(I)$. Next, the counterpart is generated as $I_L = SUB(I, FQD(I))$. $SUB(\cdot)$ is subtraction operation of pixel by pixel of two input images. For example, it subtracts pixels of $FQD(I)$ from those of I. Watermark is effectively embedded in the complicated area of an image since small manipulation is hardly noticed by human eyes. For example, brightness modification of single dot in hair area is almost impossible to recognize the difference, but the modification in skin area is easily recognized. Even though I_H is hardly recognized, detail of an entire figure in the image is somewhat visible. Furthermore, main component of an image remains in the counterpart I_L which may causes re-distribution of it.

Step 2

Block Division ($BRD(\cdot)$), breaks up entire image detail since original condition in I_H may be easily profiled from a high-pass filtered image. $BRD(\cdot)$ is a function which divides an image into $zCol \times zRow$ pixels and outputs two patterns of block-check images as $(I_{Ha}, I_{Hb}) = BRD(I_H)$. The divided blocks are composed of image elements and blank elements sorted alternatively as shown in Fig. 20. In this implementation, the image is divided into square blocks $Z_1, \ldots, Z_\eta$ which is effective universally. η is the total number of blocks in which 64 ($64 = \eta = 512/64 \times 512/64$) blocks such that $Z_1, \ldots, Z_{64}$ is generated where $zCol = zRow = 64$.

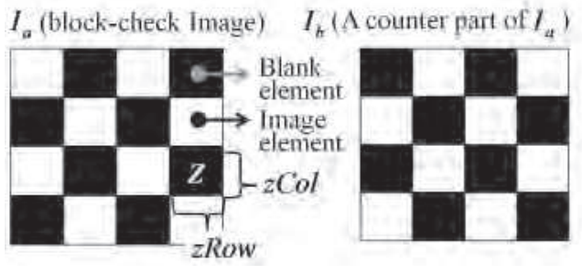


Fig. 20. Block-check Image Generation

Step 3

In order to make a valueless image, Invisible Masking function $IVM(\cdot)$ is used to add noise as $I_{Ln} = IVM(I_L)$ so that the client has no incentive to redistribute I_c without receiving I_e. $IVM(\cdot)$ adds up brightness values as noise in $nCol \times nRow$ pixels for all area $N_j, (1 \leq j \leq (Col/nCol \times Row/nRow))$. Pseudo-random value rnd_j used for noise (block noise) is generated based on minimum brightness of N_j because an input image I_L should be generated by simply summing up the brightness values of block noise image I_{Ln} and the counterpart $SUB(I_L, I_{Ln})$ (Fig. 21). For example, assume the brightness in N_j in I_L is 120,96,209,58, rnd_j in I_{Ln} is 200, the brightness to be assigned to counterpart $SUB(I_L, I_{Ln})$ would be (120-200),(96-200),(209-200),(58-200). Note that since negative integer is invalid for brightness, the pseudo-random value needs to be generated within the range of $1, \ldots, 58$ in order to avoid underflow and overflow when summing up two values together. In this implementation $nCol = nRow = 4$ is used. Block noise should be effective for this case since block noise is able to well conceal recorded subjects compare to every pixel wise noise. If the pixel is large, the subject is concealed well, but it affects watermark embedding due to reduction of feature in the image.

Step 4

Generate the other parts of block-check images as $(I_{Lna}, I_{Lnb}) = BRD(I_{Ln})$. The image element of which will be replaced with the blank parts in I_{Ha}. I_{Lnb} will be used.

Step 5

Block Integration ($BI(\cdot)$) integrates image elements in I_{Lnb} and I_{Ha} as $I_{LnbHa} = BI(I_{Lnb}, I_{Ha})$ which contains frequency components and noise components.

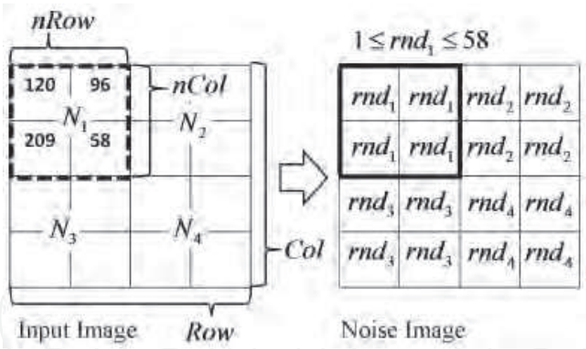


Fig. 21. Noise Generation

Step 6

Generate a complement part $I_c = SUB(I, (I_{LnbHa}))$.

Step 7

Shuffle blocks in I_{LnbHa} by using block shuffling function $RS(\cdot)$ to generate an endorse part I_e and a key psk for reversing shuffle as $(I_e, psk) = RS(I_{LnbHa})$.

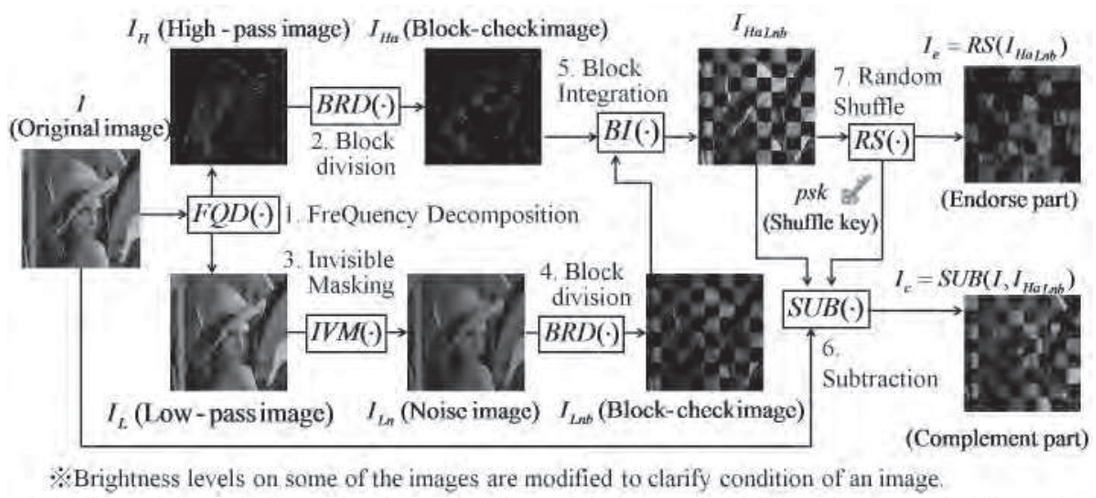


Fig. 22. Image Decomposition Procedure

4.3.4 Watermark embedding

In this section, embedding process is described (Step 2 in Fig. 19). As soon as TTP receives a request from a client, TTP obtains blocked images $I_e = Z_1, \ldots, Z_\eta$ at random to embed ω.

In this prototype, we apply a watermark algorithm that embeds watermark in frequency domain. I_e contains two types of blocks, a high-pass filtered block Z_h and a noise block Z_n. This embedding, effective for high-pass filtered blocks, is applied to Z_h.

Parameters for embedding are summarized below. Let robustness of watermark is δ, pairs of coefficients used for embedding are a_i, b_i, bit string of ω is ω_i, the index of a_i, b_i and ω_i is

$i = 1, \ldots, q$, bit length of ω is q, redundancy (number of Y to be modified) of ω is γ. In this implementation, $q = 15$, $\gamma = 30$, $\delta = 20$, $bCol = bRow = 8$ (ths size of Y_ℓ) is used.

First, finds out small blocks Y_ℓ in a blocked image that contains complicated area where watermark is effectively embedded. In this implementation, standard deviation is used to estimate complexity because standard deviation σ of brightness tends to be large in the complicated area in general. Ordinary images shown in the section 4.4 are used in this implementation. Hence, the block Y_ℓ which contain large σ provides better detection accuracy of watermark. First, divide Z into area Y_ℓ which is $bCol \times bRow$ pixels. Find σ from every Y_ℓ. Next, find Y_ℓ that satisfies $\sigma > \tau$. τ, threshold for complexity, is the average value of σ in all Y_ℓ in this implementation.

Embedding procedure is described below (Fig. 23). Select q pairs of DCT coefficient $a_1, \ldots, a_q, b_1, \ldots, b_q$ from the selected area Y_ℓ to embed watermark $\omega_1, \ldots, \omega_q$.

a_i is selected from low to middle frequency domain, b_i is selected from middle to high frequency domain. For embedding $\omega_i = 0$, the coefficients are modified as $a_i < b_i$, and for $\omega_i = 1$, these are modified as $a_i \geq b_i$. If $\omega_i = 0$ and the selected coefficients are $a_i < b_i$, then the coefficients are modified to satisfy as $a_i' = a_i - \delta, b_i' = b_i + \delta$. Otherwise ($a_i \geq b_i$), the coefficients are modified as $a_i' = b_i - \delta, b_i' = a_i + \delta$. If $\omega_i = 1$ and $a_i \geq b_i$, $a_i' = a_i + \delta$, $b_i' = b_i - \delta$. Otherwise ($a_i \leq b_i$), they are modified as $a_i' = b_i + \delta$, $b_i' = a_i - \delta$. Apply the above modification to all $i = 1, \ldots, q$.

If δ is large, watermark would be robust, but the image would be degenerated. If δ is small, an image get less degenerated, but watermark would be fragile.

Adding δ causes overflow when integrating two images. However, if the pixels would be larger than 255, we make the brightness in 255.

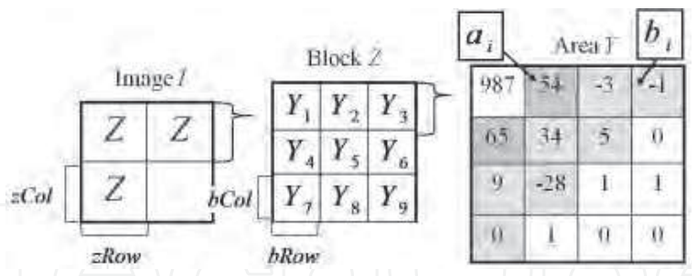


Fig. 23. Coefficients Selection in a block Y

Iterate the above process to all $Y_1, \ldots, Y_\gamma$ which satisfy $\sigma > \tau$. Note that, if γ is large, watermark can be robust, but the image would be degenerated.

Apply the above procedure for all high-pass blocks Z to generate endorsed blocks $Z_1', \ldots, Z_{\eta/2}'$. Note that total number of Z' is $\eta/2$ since high-pass blocks Z and noise blocks Z exist the equal amount in I' in this implementation.

An extraction method is described below. Extraction requires the information on modified coefficients. Deploy the endorsed image I' into frequency domain as embedding procedure. First, divides the image into Z and then extract $\omega = \omega_1, \ldots, \omega_q$ by examining the condition of DCT coefficients $(a_1', b_1'), (a_2', b_2'), \ldots, (a_q', b_q')$ in $Y_1', \ldots, Y_\gamma'$ in every Z' respectively. Next,

extracts ω from $Y'_1, \ldots, Y'_\gamma$ and then apply this process for all $Z'_1, \ldots, Z'_{\eta/2}$ to take an average of extracted bit stream.

4.3.5 Obtaining endorsed image

In the final step (Step 3 in Fig. 19), a complete endorsed image I' is generated by integrating I_c and I'_e which can be obtained by tracing links in the HTML content. The client obtains I_c and psk from a provider, and $I'_e = RS(I'_{LnbHa})$ from TTP.

I' is generated by following process. Reverse shuffle by $I'_{LnbHa} = RS^{-1}_{psk}(I'_e)$ and then combine with I_c as $I' = SUM(I_c, I'_{LnbHa})$ where $SUM(\cdot)$ is function that sum up brightness values of two input images. Note that a provider cannot obtain I' illegally because verification is required to obtain I'_e.

4.4 Evaluation

Perceptual and robustness evaluations are shown in this section. The former one shows perceptual condition of decomposed images and a watermarked image. In the latter one, robustness of watermark is shown. The environment used in this implementation is summarized in Table 7.

Detail	Specification
CPU	Intel Xeon E5345 2.33GHz
Memory	4GB RAM
OS	Fedora 10
DCMP, *EMB*	Matlab2009a
Web interface *INTG*	HTML, PHP

Table 7. Environment

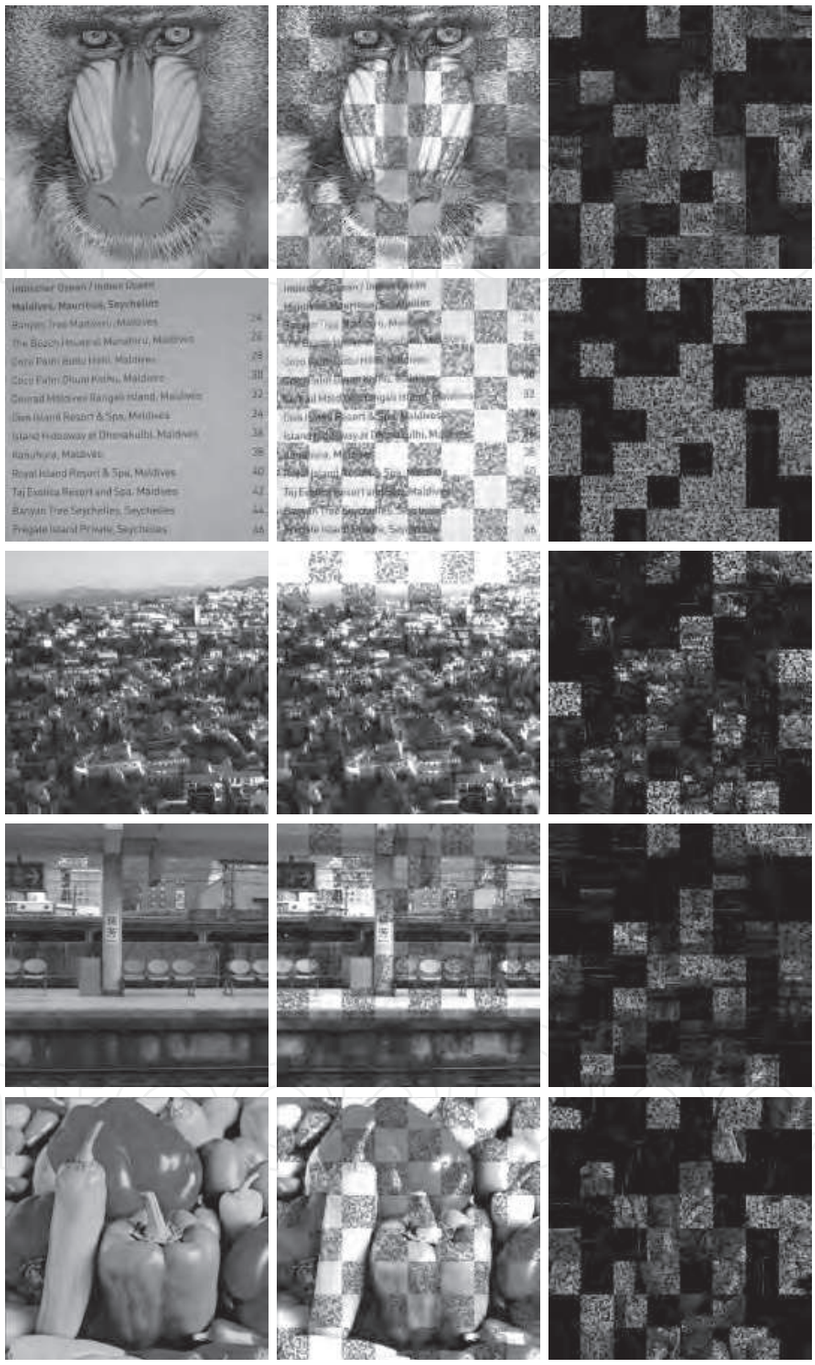
Attacks	Description	Total Attacks	Levels	Succeed
AFFINE	Affine transform	8	$1, 2, \ldots, 8$	None
CONV	Gaussian filtering	2	$1, 2$	All
CROP [%]	Cropping	3	$25, 50, 75$	None
JPEG[%]	JPEG compression	7	$20, 30, \ldots, 80$	$30, \ldots, 80$
MEDIAN	Median cut	4	$3, 5, 7, 9$	3
NOISE[%]	Add noise	8	$10, 20, \ldots, 80$	None
RESC [%]	Rescale	6	$50, 75, 90, 125, 150, 200$	All
RML [lines]	Remove lines	9	$10, 20, \ldots, 100$	All

Table 8. Parameters of StirMark and Evaluated Results

4.4.1 Perceptual evaluation

Perceptual evaluation for decomposition using various types of images is shown in Fig. 24 (From top to bottom; *Baboon*, *Book*, *Granada*, *Kyufun*, and *Peppers*). Note that *Baboon* and *Peppers* are provided by USC SIPI database. The other images are prepared by the authors.

An watermarked image I' is shown in Fig. 25 in which high strength-level of watermark has been applied to show distinct embedding effects. Therefore, I' is heavily degenerated.



Original Image I Complement piece I_c Endorse piece I_e

Fig. 24. Various Types of Output Images

4.4.2 Robustness evaluation of watermark using StirMark benchmark

Parameters on StirMark used in this implementation is listed in Table 8. For example of AFFINE, 8 levels of affine transformed images are generated. Robustness is examined by extracting ω from transformed images.

Evaluation results are shown below. Watermark is detected from 24 images out of 47 attacked images as shown in Table 8, labeled as "Succeed." We also show how a watermarked image is affected by decomposition. We have compared robustness of two watermarked images in which combination of embedding and decomposition and the one without decomposition have been applied. The latter one, the one without decomposition shows 31/47 cases are successfully extracted. The comparison of the two methods is shown in Fig. 26. Black lines show robustness of a watermarked image embedded with decomposition, and gray lines show the one without decomposition. The experimental results provide effective evidence showing that robustness of a watermarked image is little affected by decomposition.

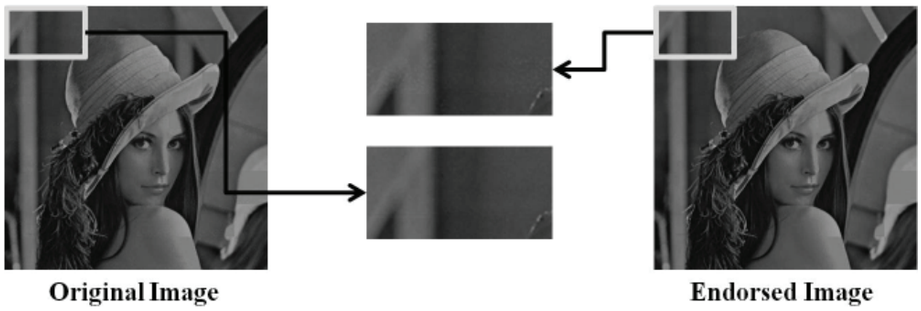


Fig. 25. Original Image I and Endorsed Image I'

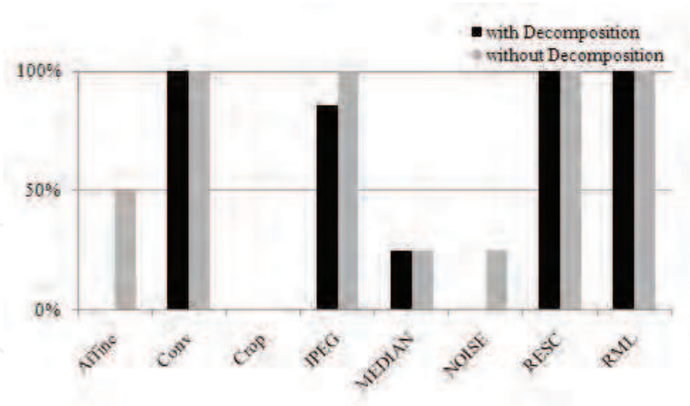


Fig. 26. Robustness of Watermark

5. Conclusion

Our primal consideration is that most of the security applications or tools are difficult to use in which special skill, knowledge and tools are needed. However, majority of the people is not capable of understanding programming or special mean of computer term. Although, the Internet becomes popular commodity, security tool is way behind to be commodity. Practical

and secure watermarking is urgently needed. A fair and secure digital content trading scheme that protects both provider's and client's security is introduced in this chapter.

Blind watermarking based on cryptography and watermarking is one of the effective techniques in which performance evaluation is introduced based on our implementation. This satisfies higher level of security at heavy processing cost because cryptography is not compatible with watermarking. Performance evaluation which shows feasibility of blind watermarking is introduced.

Pseudo-blind watermarking (Okada et al., 2009) which uses media processing instead of cryptography is an alternative method of the blind watermarking. This scheme enhances security and compatibility of watermarking. Performance evaluation is also introduced. This scheme is able to resolve the problems of blind watermarking which are robustness and processing cost.

Even though security tools have been developed, most of them are still difficult to use for ordinary people. However, our concern is that providing user-friendly security tools enables to enhance entire security level because if more people use security tools, entire security level would be increased rather than providing absolute security level to only certain people who has high literacy to use security tools. One of our future work (Okada et al., 2011), intuitive watermark extraction is designed for people who have no knowledge nor skill is proposed toward user-friendly digital watermarking.

6. References

Ackerman, M.S. (1999). Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences, *ACM Press*, pp.1-8

Acquisti, A.(2004). Privacy in Electronic Commerce and the Economics of Immediate Gratification, *ACM Electronic Commerce Conference (EC'04), ACM Press*

Bender, W.; Gruhl, D.; Morimoto, N.(1995). Techniques for Data Hiding, *SPIE*, Vol.2020, pp.2420-2440

El Gamal, T.(1985). A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Trans. on Information Theory*, Vol.IT-31, No.4, pp.469-472

Furukawa, J.(2004). Secure Detection of Watermarks, *IEICE Trans.*, Vol.E87-A, No.1, pp.212-220

Iwamura, K.; Sakurai, K; Imai, H. (1997). Blind Fingerprinting. *Technical report of IEICE. ISEC*, Vol.97, pp. 63-74

Lin, E.; Podilchuk, C. ; Delp, E.(2000). Detection of Image Alterations Using Semi-fragile Watermarks, *SPIE-3971*

Okada, M.; Kikuchi, H.; Okabe, Y. (2008). Multi-Bit Embedding in Asymmetric Digital Watermarking without Exposing Secret Information. *IEICE (The Institute of Electronics, Information and Communication Engineers)*, Vol.E91-D, No.5, pp.1348-1358

Okada, M.; Okabe, Y.; Uehara, T.(2009). A Privacy-Secure Content Trading System for Small Content Providers Using Semi-Blind Digital Watermarking. *The 2009 International Workshop on Forensics for Future Generation Communication environments (F2GC) in conjunction with CSA2009*, Vol.CFP0917F-PRT Vol.2, pp.561-568

Okada, M.; Okabe, Y. ; Uehara, T. (2010). A Web-based Privacy-Secure Content Trading System for Small Content Providers Using Semi-Blind Digital Watermarking, *Annual IEEE Consumer Communications and Networking Conference (IEEE-CCNC2010)*

Okada, M.; Matsuyama, S. ; Hara, Y.(2011). User-friendly Digital Watermark Extraction Using Semi-transparent Image, *8th Annual IEEE Consumer Communications and Networking Conference (IEEE-CCNC2011)*

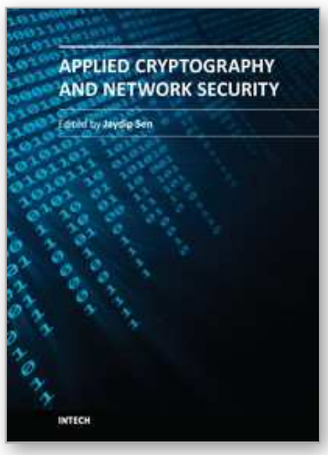
Okamoto, T. ; Uchiyama, S.(1998). A New Public-key Cryptosystem as Secure as Factoring, *EUROCRYPT'98*, pp.308-318

Paillier, P. (1999). Public-key Cryptosystems based on Composite Degree Residuosity Classes, *EUROCRYPT'99*, pp.223-238

Petitcolas, F.A.P. (2000). Watermarking Schemes Evaluation, *IEEE Signal Processing*, Vol.17, No.5, pp.58-64

Pfitzmann, B. ;Schunter, M. (1996). Asymmetric Fingerprinting, *EUROCRYPT'96 LNCS*, Vol.1070, pp.84-95

IntechOpen



Applied Cryptography and Network Security

Edited by Dr. Jaydip Sen

ISBN 978-953-51-0218-2

Hard cover, 376 pages

Publisher InTech

Published online 14, March, 2012

Published in print edition March, 2012

Cryptography will continue to play important roles in developing of new security solutions which will be in great demand with the advent of high-speed next-generation communication systems and networks. This book discusses some of the critical security challenges faced by today's computing world and provides insights to possible mechanisms to defend against these attacks. The book contains sixteen chapters which deal with security and privacy issues in computing and communication networks, quantum cryptography and the evolutionary concepts of cryptography and their applications like chaos-based cryptography and DNA cryptography. It will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Mitsuo Okada (2012). Privacy-Secure Digital Watermarking for Fair Content Trading, Applied Cryptography and Network Security, Dr. Jaydip Sen (Ed.), ISBN: 978-953-51-0218-2, InTech, Available from: http://www.intechopen.com/books/applied-cryptography-and-network-security/privacy-secure-digital-watermarking-for-fair-content-trading

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen