國立臺灣大學電機資訊學院資訊網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

# 適用於電子出版系統的雲計算智財權保護機制

# Cloud-based IPRs Protection Mechanism for the E-publishing System

李旻倫

Ming-Lung Lee

指導教授：吳家麟 博士

Advisor: Ja-Ling Wu, Ph.D.

中華民國111年7月

July, 2022

doi:10.6342/NTU202203199

# 國立臺灣大學碩士學位論文
# 口試委員會審定書

## 適用於電子出版系統的雲計算智財權保護機制
## Cloud-based IPRs Protection Mechanism for the E-publishing System

本論文係李旻倫君（學號 R08944042）在國立臺灣大學資訊網路與多媒體研究所完成之碩士學位論文，於民國一百一十一年七月六日承下列考試委員審查通過及口試及格，特此證明。

口試委員：

_____（簽名）
（指導教授）

_____ _____

_____ _____

所　長：_____



doi:10.6342/NTU202203199

# 致謝

非常感謝我的指導教授吳家麟老師總是給我很大的探索空間，且不管做什麼都很有耐心指導且站在學生的角度去幫助完成研究。老師的處事精神及學術熱情也為我們的榜樣，真的很感激我的碩班生涯由老師所指導，讓我求學過程中成長許多。另外也非常感謝家人的鼎力支持才能讓我心無旁騖地完成學業。我的實驗室同學們雖然因為疫情不常見面，但彼此的反饋和支持也是不可或缺的動力。儘管因為疫情造成許多波折，但因為有身邊每個人的支持才能得以順利完成，感謝每個幫助我的人。



doi:10.6342/NTU202203199

# 摘要

在現代商業模式運作中，創作者產製的數位產品更傾向於部分免費開放給目標客群，目的是為了產生流量和知名度，事先試用產品也能獲取消費者的信任和增加購買意願。創作者再以提供有價值的內容或服務來獲取收益，這已經是現今非常常見的收益模式，因此在本研究中我們將過去的雲端買賣方浮水印協定改進成更適合這套商業模式的機制。引入雲端固然可帶來相當的利益，但是龐大的流量同樣對服務提供者帶來的巨大的挑戰：造成頻寬負荷、計算效率、儲存成本的急遽增長。因此，我們在雲端買賣方浮水印協定的基礎上利用了客戶端浮水印和量化指標調變兩項技術去改善上述問題。基於該協定原有的特性，利用雲端作為買賣方的軟硬體資源的提供者，而且同時也作為一個公平、安全和體驗良好的交易平台。此外，其功能還包括有推播賣方的數位產品和方便消費者選購產品等。換言之，我們提出的電子出版系統是一個對於現代商業模式有效率且平衡各方權益的機制。

關鍵字

同態加密、雲端買賣方浮水印協定、量化指標調變、安全的客戶端浮水印、數位智慧財產權



doi:10.6342/NTU202203199

# Abstract

In modern business model, digital products produced by creators tend to be freely and partially open to target customers for getting notice. The purpose is to generate and increase traffic and popularity. Trial product distribution in advance can also enhance consumers' experiences and increase their willingness to buy. Creators then get revenue by selling valuable content or services to customers. This is a widespread revenue model today. In this work, we enhanced the previous Cloud-based Buyer-Seller Watermarking Protocols[4] to produce a more suitable mechanism for the Cloud-based E-publishing business model. Besides benefits brought by cloud, the considerable traffic, rapidly increasing bandwidth burden, computing efficiency, and storage costs also challenge service providers. The techniques of Secure Client-Side Watermarking and Quantization Index Modulation[7] are adopted to release the above-mentioned problems. Based on the characteristics of the pre-described protocol, the cloud is treated as a software and hardware resources provider for supporting buyers and sellers. In other words, a system operated under the proposed mechanism can act as a fair, secure, and user-friendly trading platform. It is our belief that our proposed mechanism is helpful to establish an efficient, secure, and rights balanced system for E-publishing.

Keywords:

Homomorphic encryption, Cloud-based Buyer-Seller Watermarking Protocol, Quantization Index Modulation, Secure Client-side Watermarking, Digital Intellectual Property Rights.



doi:10.6342/NTU202203199

# Contents





doi:10.6342/NTU202203199

doi:10.6342/NTU202203199

# List of Figures





doi:10.6342/NTU202203199

# Chapter 1

# Introduction

With the popularity of multimedia devices, everyone can be a creator and a consumer of digital contents at the same time. Thanks to the rapid progress in hardware technology and network speed, digital content's production, distribution, the time-spent and production costs have been shortened and reduced in an all-around way. The above facts boomed up the development of digital trading markets. Nevertheless, too easy in circulating digital contents challenges the protection of intellectual property rights and reduces the value in-return expected by normal people.

Digital fingerprinting[2] technology has been proposed to deal with the problem of the malicious distribution of digital contents. For example, a seller inserts a digital fingerprint (a unique watermark) into the content. Once malicious buyers illegally distribute the content, the seller can trace the distributor and take legal actions based on the embedded fingerprint. However, this seller-oriented approach raises the buyer's rights problem, where the buyer can claim that the digital fingerprint was placed by a malicious seller, thereby refuting the authenticity of the evidence. Therefore, the buyer-seller watermarking protocol[6, 8, 9, 14] is proposed to fix this problem. The core idea is to insert a watermark that the seller does not know about to prevent malicious sellers from placing dishonest watermarks. But the associated efficiency and costs are not ideal; therefore, the willingness to adopt is limited



doi:10.6342/NTU202203199

The Cloud-based Buyer-Seller Watermarking Protocol[4] targets on this problem. It includes the role of an another third-party cloud, which shares the burdens of computing, communication, and storage. Generally, constructing these infrastructures on the cloud is an ideal and low-cost approach. The protocol proposed in [4] is also scalable and reliable and can handle a large number of transactions at the same time. Furthermore, because of embracing the homomorphic encryption, the cloud can directly embed watermarks in the encrypted domain, so the digital content and the watermark leakaging problems are avoided. However, computing in the encrypted domain is quite time-consuming, even though the cloud is assumed to have enterprise-level computing power, this heavy-loading problem cannot be ignored. Moreover, the protocol uses the public key of each buyer to encrypt the content sold, which means that the cloud needs to serve each buyer peer-by-peer. The associated peer-by-peer connections bring high communication and computing costs which do not fit the commonly used Internet broadcasting scenarios. Nowadays, buyers are generally used to previewing products before purchasing. Encrypted content is unavailable in this circumstance, this shortage dramatically reduces the willingness to purchase, also.

In order to conquer the above problems, our proposed mechanism adopts the “Freemium” model. It refers to the client-side watermarking technology[1][5][10][20] and redistributes some of the burden of the embedding operation to the client. At the same time, encrypting all digital content is not necessary. Instead, use the extraction of valuable information as the key, place the basic version of the content in the public space for everyone to browse, and use the watermark-embedded key for sale. In this way, the embedding task on the encryption domain is performed on the key, which shortens the system's response time. The smaller amount of required data is also more convenient for transmission and storage. In addition, the Improved Spread Spectrum-based watermarking scheme (ISS-WS)[3] used in the Cloud-based BS watermarking protocol involves many floating-point operations, which is



doi:10.6342/NTU202203199

time-consuming in the Pailliar-based realization. So we use Quantization Index Modulation(QIM)[7] to shorten the embedding time significantly. Because of the characteristics of QIM, it performs very well under JPEG compression and is suitable for network environments where images are generally stored in JPEG-compressed format.

Our proposed mechanism can be applied to any digital content, such as videos, photos, sounds, etc. According to the type of digital content, QIM can be replaced by the appropriate watermark embedding method for the digital content. And the seller needs to determine which part is the valuable content for sale. Then, the proposed mechanism will also work well on the new digital content. However, to focus on the mechanism, we use an image as an example of digital content in the following.



doi:10.6342/NTU202203199

# Chapter 2

# Preliminary and Related work

## 2.1 Notations

- m : message
- $m_l$ : l-th bit of message
- $E()$ : Encryption function of Paillier cryptosystem
- $D()$ : decryption function of Paillier cryptosystem
- $E(m)$ : ciphertext of $m$
- $s^l$ : l-th spread code
- $s^l_\mu$ : μ-th bit of l-th spread code
- $U_1$ : Buyer's reference pattern
- $U_2$ : Seller's reference pattern
- $u_{1,i}$ : i-th bit of Buyer's reference pattern
- $u_{2,i}$ : i-th bit of Seller's reference pattern
- $I$ : cover image
- $x_i$ : i-th DCT coefficient of a block in a cover image
- $x_i$' : i-th modified DCT coefficient of a block
- X : a vector of DCT coefficients of a block in a cover image
- X' : a vector of modified DCT coefficients of a block
- $w_{1,i}$ : i-th bit of Buyer's watermark



doi:10.6342/NTU202203199

- $w_{2,i}$ : i-th bit of Seller's watermark
- $W_1$ : Buyer's watermark
- $W_2$ : Seller's watermark
- W(I) : watermarked image
- $W^2(I)$ : embed two watermarks image

## 2.2 Paillier Homomorphic Cryptosystem

Two major features of the Paillier cryptosystem[11] are homomorphic encryption and semantic security. As the encryption function is additively homomorphic[10], the homomorphic addition of plaintexts can be described as

$$D(E(m_1) \times E(m_2) \bmod n^2) = m_1 + m_2 \bmod n \quad , \qquad (2.1)$$

while and the homomorphic multiplication of plaintexts is represented by

$$D(E(m)^k \bmod n^2) = km \bmod n \quad . \qquad (2.2)$$

To simplify computation, we define the following three operators. The multiplication, "•", between two encrypted messages $m_1$ and $m_2$ is defined as the computation of

$$D(E(m_1) \bullet E(m_2)) = D(E(m_1) \times E(m_2) \bmod n^2) = m_1 + m_2 \bmod n \quad . \qquad (2.3)$$

The addition, "⊕", between the encrypted message m and a constant k can be represented as

$$D(E(m) \oplus k) = D(E(m) \times E(k) \bmod n^2) = m + k \bmod n \quad . (2.4)$$

And the multiplication, "⊗", between the encrypted message m and a constant k is

$$D(k \otimes E(m)) = D(E(m)^k \bmod n^2) = km \bmod n \quad . \qquad (2.5)$$



doi:10.6342/NTU202203199

## 2.3 Lookup-Table Based Encryption

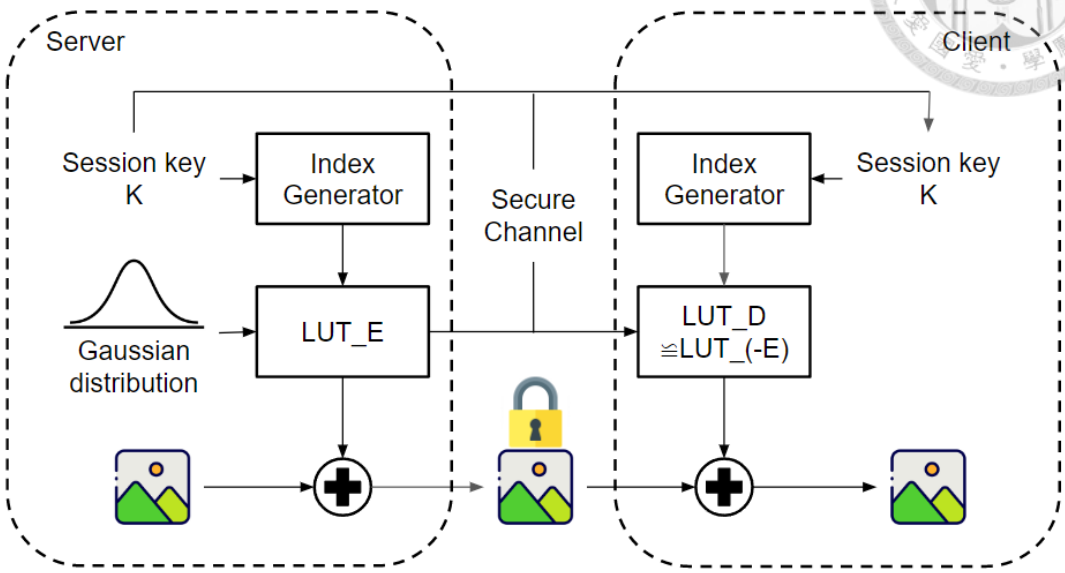



Figure 1: The model of an LUT based encrytion scheme[1][17].

Server generates an lookup table (LUT), LUT_E for encryption, and a pirate watermarking, LUT_D, for the client . We assume the entry values of E and $W_k$ are i.i.d. random variables drawn from zero-mean Gaussian distributions with variances $\sigma_E^2$ or $\sigma_W^2$, respectively. Additionally, server generates session key K, which is a seed for generating entry indexed. After having indexes of LUT_E for all pixels of the host image, sever adds the corresponding entry values taken from LUT_E to the corresponding pixel values for blinding the original pixel values. In order not to be confused with the next encryption method, we call this action “blind”. The Decryption LUT_D is actually equivalent to the LUT_(-E). Thus, Server sends LUT_D and session key K to the client through trusted channel, and the client can decrypt the image.



doi:10.6342/NTU202203199

## 2.4 Client-side Embedding Method

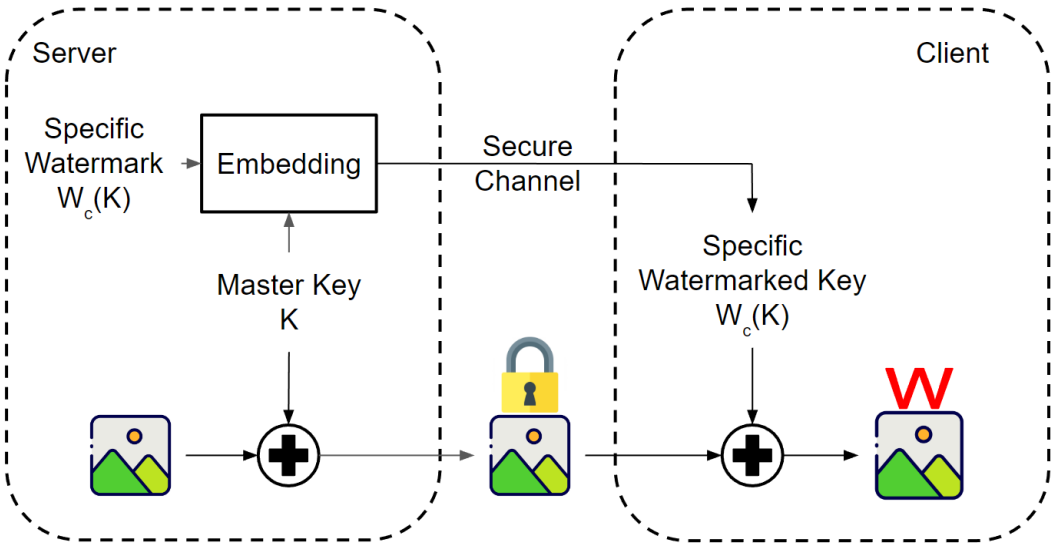



Figure 2: The model of a Client-side Embedding scheme[1][5][10][20].

The basic concept of the client-side watermark method[1][5][10][20] is that the server encrypts the file, and the step of embedding the watermark is delayed to the client-side. More detail is described as follows:

Server encrypts the digital content using his master key and delivers the encrypted copy to each buyer via a public channel. Server then embeds a specific watermark into the key for each buyer and sends the watermarked keys to buyers respectively through a secure channel. Because watermarked key hosts buyer-related information, the watermark will persist in the decrypted content when a buyer uses his specific watermarked key to decrypt the ciphertext. According to the above application scenario, the communication cost decreases, due to the relatively small cost of transmitting watermarked keys. Additionally, each buyer downloads the same encrypted content, so the cost of distributing encrypted



doi:10.6342/NTU202203199

content can be effectively saved by some existing broadcasting techniques, such as Content Delivering Network (CDN), Peer to Peer (P2P), and so on. The associated computational cost is also diminished because embedding the watermark into a key is much lighter than an entire digital content.

## 2.5 Spread Spectrum Watermarking Using Quantization Index Modulation

### 2.5.1 Embedding process

Our watermark embedding process bases on the paper [7]. In the spread spectrum-based watermarking technique, an L-bit message $m = (m_1, m_2, \cdots, m_L)$ is spread by using spread codes, where $m_l \in \{+1, -1\}$, and $l = 1, 2, \cdots, L$. Since the watermarks provide large redundancy, in the marking space, some errors can be corrected. In general, the watermarks are added into an image directly. A spread code $s^l = (s^l_1, s^l_2, \cdots, s^l_N)$ is generated from a pseudo random number consisting of -1 and +1. By using the l-th spread code for message bit $m_l$, the μ-th watermark bit $w^l_\mu$ is generated by

$$w^l_\mu = m_l s^l_\mu, \ \mu = 1, 2, \ \ldots, \ N, \qquad (2.5)$$

where spread code $s^l_\mu = \pm 1$. In the case of an image, we divide the image into segments. Then, a segment is divided into 8 × 8-pixel blocks. A watermark bit is embedded into one block. For message length L and spread code length N, LN blocks are used to embed the watermarks. An LN bit watermark is sequentially embedded in the segment. Each 8 × 8 pixels block is then transformed by DCT. Each bit of the watermark is embedded into a fixed



doi:10.6342/NTU202203199

position (1, 1) in the DCT domain. Quantization Index Modulation(QIM) [7] is a method that quantizes DCT coefficients. The quantized coefficient is multiples of the quantization step size Δ and is indexed by using a watermark bit {0, 1}. Therefore QIM is a blind watermarking scheme. To embed by QIM, the watermark bit $\omega_\mu^l \in \{1, -1\}$ is converted to bipolar ∈ {0, 1}, which is given by

$$w_\mu^l = \frac{1}{2}(w_\mu^l + 1). \tag{2.6}$$

The embedded DCT coefficient $\tilde{C}^1_\mu$ is given by

$$x'^l_\mu = 2\Delta \left\lfloor \frac{x_\mu^l}{2\Delta} - \frac{w_\mu^l}{2} + 0.5 \right\rfloor + \Delta w_\mu^l, \tag{2.7}$$

where ⌊x⌋ stands for the floor function, which returns the largest integer not greater than x. Δ is the quantization step size. In our work, the value of Δ is adopted in the Standard JPEG quantization table[15]. Both the sender and receiver know the same value. For the best performance on the quality of the image, The optimal step size Δ can be determined by improved perceptual models[21].

By the Standard JPEG quantization table[15], the watermark embedder can assume the degree of JPEG compression on the host image. Therefore, the value of the quantization table given the assumed degree is used in the embedding process. After embedding the watermarks in DCT coefficients, pixel values are obtained through inverse DCT. Following the above process, all blocks are operated in the same way.

### 2.5.2 Decoding Process

Since the watermarks are embedded into the (1, 1) position in the DCT domain, the watermark $\hat{x}_\mu^l$ candidates are also $\hat{w}_\mu^l$ extracted from the same position. Let the value of the DCT coefficient be . The extracted value ∈ {0, 1} is obtained by



doi:10.6342/NTU202203199

$$\hat{w}_{\mu}^{l} = \left\lfloor \frac{\left|\hat{x}_{\mu}^{l}\right|}{\Delta} + 0.5 \right\rfloor mod\ 2, \qquad (2.8)$$

where Δ is the quantization step size. From the extracted value $\hat{w}_{\mu}^{l} \in \{0, 1\}$, the watermark candidate $\hat{w}_{\mu}^{l} \in \{+1, -1\}$ becomes

$$\hat{w}_{\mu}^{l} = (-1)^{\hat{w}_{\mu}^{l}}. \qquad (2.9)$$

From the estimated watermark $\hat{w}_{\mu}^{l}$ and spread code $s^{l}_{\mu}$, the estimated L bit message $\hat{m}_{l}$ is given by

$$\hat{m}_{l} = sgn\left(\frac{1}{N}\sum_{\mu=1}^{N} s_{\mu}^{l}\hat{w}_{\mu}^{l}\right), \qquad (2.10)$$

where sgn(x) stands for the signum function

$$sgn(x) = \begin{cases} +1 & , x \geq 0 \\ -1 & , x < 0 \end{cases}. \qquad (2.11)$$



doi:10.6342/NTU202203199

## 2.6 Cloud-based Buyer-Seller Watermarking Protocol

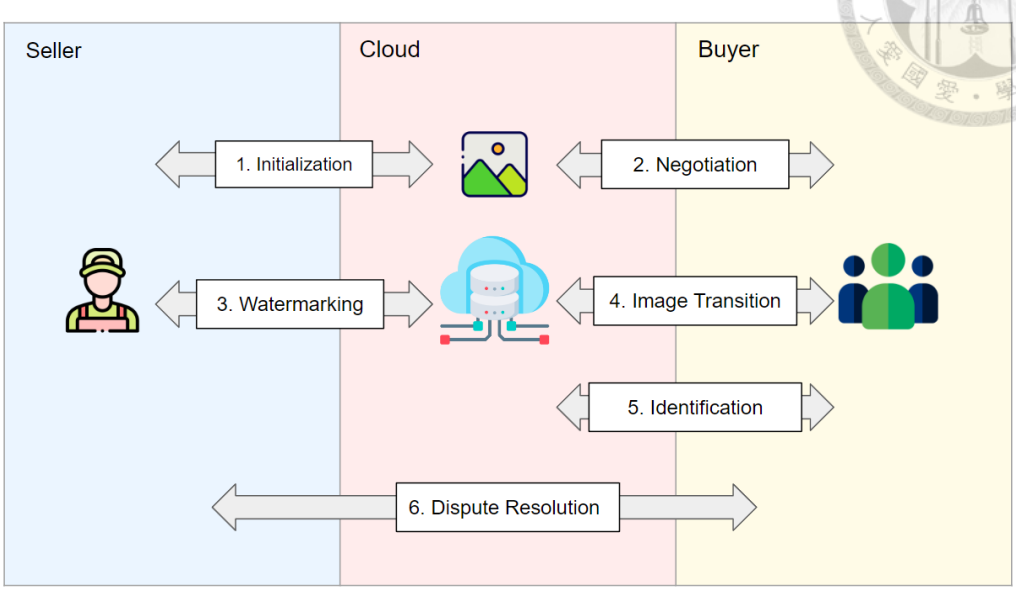



Figure 3: The model of Cloud-based Buyer-Seller Watermarking Protocol presented in[4].

In the original paper[4], there are Cloud-based Buyer-Seller Watermarking Protocol-1 and Cloud-based Buyer-Seller Watermarking Protocol-2. Here we detail the Cloud-based Buyer-Seller Watermarking Protocol-2, named CSBP-2.

### 2.6.1 Initialization Protocol

Seller performs block Discrete Cosine Transform (DCT) on an image, getting DCT(I). Then, Seller blinds DCT(I) by using LUT_E (cf. Section 2.3), and uploads blinded DCT(I) to the cloud. For simple presentation, we denote blinded DCT(I) as ||DCT(I)||. Additionally, both the seller and buyer sides negotiate for appropriate watermarking parameters.

### 2.6.2 Negotiation Protocol

First, Buyer generates a public key and a private key in the Paillier cryptosystem.



doi:10.6342/NTU202203199

Buyer encrypts a watermark determined by himself/herself, and sends the encrypted watermark and the public key to Cloud, then Cloud returns a retrieving address back and stores all the used certificates and signatures in a selling record table.

### 2.6.3 Watermarking Protocol

Cloud passes the public key to Seller and waits for inputs. Seller should return $E(W_{Seller})$, E(LUT -E), session key K, reference pattern sets $u_1$ and $u_2$, which are necessary in the embedding. In the next step, Cloud encrypt ||DCT(I)|| to E(||DCT(I)||), and unblinds it in the encryption domain to E(DCT(I)). Afterward, for embedding two watermarks, Spread-Spectrum based Watermarking Schemes, called ISS-WS, is introduced, and formulated as

$$x' = x + \sum_{\beta=1}^{2} (\alpha w_{\beta,i} - \lambda \frac{\langle x, u_\beta \rangle}{\langle u_\beta, u_\beta \rangle}) u_\beta, \qquad (2.12)$$

where α is the watermark strength, $u_\beta$ is the β-th reference pattern, and $b_{\beta,i}$ is the i-th bit of the β-th watermark. Notice that $u_1$ and $u_2$ had better be orthogonal to each other, or there are be some errors in the extraction process and here λ is set to 1. Cloud embeds the i-th bit of the watermark to the i-th block. In the Pailliar homomorphic encryption scheme, by the additive homomorphism property, the corresponding formula to (2.12) would be

$$\prod_{m=1}^{n} E(x'_j) = \prod_{m=1}^{n} E(x_j) \bullet \prod_{\beta=1}^{2} \left( \prod_{m=1}^{N} E(w_{\beta,i})^{\alpha u_{\beta,j}} \bullet \prod_{k=1}^{N} E(-x_k)^{u_{\beta,k} u_{\beta,j}} \right), \quad (2.13)$$

where n is the length of the reference pattern u. Because division is unavailable in Paillier cryptosystem, in implementation, both sides of equation (2.13) will be multipled by n. By equation (2.11), Performing ISS-WS in the encrytion domain is doable. The results of the embedding operation can be denoted as $E(W^2(DCT(I)))$.



doi:10.6342/NTU202203199

### 2.6.4 Image Transmission Protocol

When the embedding process is completed, Cloud stores E($W^2$(DCT(I))) in the retrieving address, so Buyer would retrieve it from this address, which is assigned in the negotiation sub-protocol. Now Buyer can decrypt E($W^2$(DCT(I))) to $W^2$(DCT(I)) by private key, and do inverse DCT to obtain the watermarked image, $W^2$(I).

### 2.6.5 Identification Protocol

If a watermarked image was distributed illegally, the Seller can extract the embedded watermark from it with reference pattern U. For extracting the i-th bit of $W_1$, we compute the correlation, as defined in eq.(2.14), with $U_1$. That is,

$$\frac{\langle X', U_1\rangle}{\langle U_1, U_1\rangle} = \frac{\left\langle X + (\alpha w_{1,i} - \lambda \frac{\langle X, U_1\rangle}{\langle U_1, U_1\rangle})U_1 + (\alpha w_{2,i} - \lambda \frac{\langle X, U_2\rangle}{\langle U_2, U_2\rangle})U_2, U_1 \right\rangle}{\langle U_1, U_1\rangle}. \quad (2.14)$$

Since $U_1$ and $U_2$ are designated to be orthogonal, we have $\langle U_1,U_2\rangle$=0. Hence, eq.(2.14) will be reduced to

$$\frac{\langle X', U_1\rangle}{\langle U_1, U_1\rangle} = \frac{\langle X, U_1\rangle}{\langle U_1, U_1\rangle} + \frac{\langle \alpha w_{1,i} U_1, U_1\rangle}{\langle U_1, U_1\rangle} - \lambda \frac{\langle X, U_1\rangle}{\langle U_1, U_1\rangle} \frac{\langle U_1, U_1\rangle}{\langle U_1, U_1\rangle}. \quad (2.15)$$

And we can figure out that the righthand-side of eq.(2.15)'s first term and third term can be eliminated. Thus, we have

$$\frac{\langle X', U_1\rangle}{\langle U_1, U_1\rangle} = \alpha w_{1,i}. \quad (2.16)$$

Likewise, we can do it again for extracting $W_2$, with $u_2$. Thus, the steps of Identification sub-protocol can be described as follow so:

For privacy preserving extraction, Seller permutes $W^2$(DCT(I)) with $K_{PB}$ blockwise, also $u_1$, $u_2$ and DCT coefficients with $K_{PW}$ intrablock. Notice that the watermark W and the image



doi:10.6342/NTU202203199

blocks share the same key, $K_{PW}$. Seller sends the permuted $W^2(DCT(I))$, $u_1$, and $u_2$ to Cloud. Using $u_1$ and $u_2$, Cloud extracts $W_1$ and $W_2$ in a permuted order, and then returning them to Seller. Therefore, Seller can permute $W_1$ and $W_2$ back with $K_{PB}$.

### 2.6.6 Dispute Resolution Protocol

The arbiter needs $W_1$ and trading records given by Seller for judging. Seller sends $u_1$ to Cloud for asking trading records. With them, Cloud performs matching, extracts the trading records, and returns the trading records to Seller. However, Cloud stores all the trading records of Buyers. In order not to let the Cloud recognize the exact Buyer from the records, Seller should randomly generate some obfuscated records in a superset of the true selling records. This action will confuse Cloud about the identities of the Buyers.



doi:10.6342/NTU202203199

# Chapter 3

# Proposed Mechanism

Our proposed mechanism originates from the buyer-seller watermarking protocol[6, 8, 9, 14]. The protocol introduces a homomorphic cryptosystem[11], so the seller can embed the watermark without knowing the buyer's watermark. It solves the buyer's rights problem. However, what comes with it is that the seller has to bear the computational burden of embedding operations in the encrypted domain. As the number of users increases, the bandwidth requirements will also increase.

Based on the above results, Cloud-based Buyer-Seller Watermarking Protocol(CBSP) [4] tries to eliminate the burden on the seller. Therefore, this protocol adds the role of Cloud among Buyer and Seller, which is responsible for the embedding and extraction of the watermark. Because the cloud is usually assumed to have enterprise-level hardware equipment and strong scalability, it can provide stable and low-cost communication services in addition to providing computing resources. The cloud help handle the technical problems that buyers and sellers face.

Nevertheless, the cloud does not have unlimited resources, especially with the exponential growth of users. It is necessary to consider each buyer's service response time and usage cost. In addition, to prevent leakage of the image's value, the seller needs to blind the image with LUT_E(cf. Section 2.3) before uploading it to the cloud. After receiving the buyer's request, the blinded image is decrypted with LUT_D(cf. Section 2.3) in the Pailliar encryption domain and embedded watermarks. The response time of the buyer's request will be too long.



doi:10.6342/NTU202203199

Therefore, we refer to the multi-resolution scheme[12, 13]. We replace the encrypted image with the low-resolution image (that is, the degraded image in figure 3), which saves blind and unblind steps in the encryption domain. Moreover, users can download the low-resolution version for free, which brings an additional advantage: buyers can preview the image first and increase their willingness to purchase. We only encrypt the high-resolution partition (that is, the residual image in figure 3). Then, we refer to the Client-side embedding method[1][5][10]. We embed the watermark in the residual image, so there is no need to encrypt and transmit the entire image, reducing the computation time and communication cost and shortening the time to respond to buyer requests. In addition, because of the additively homomorphic characteristics of the Pailliar homomorphic cryptosystem, the choices of available watermarking schemes are limited. CBSP chose the Improved Spread Spectrum based Watermarking Scheme(ISS-WS) [3]. However, ISS-WS involves vector dot product, which includes operations among different DCT coefficients and reference patterns. The number of operations is proportional to the length of vectors. Furthermore, the DCT coefficients are floating-point numbers. About the operation between floating-point numbers in the encryption domain, the exponent of the two floating-point numbers must be adjusted to be the same before the addition operation. It brings extra computation. To avoid the above weakness, we adopt Quantization Index Modulation(QIM) [7]. This method only needs to add an integer to a DCT coefficient to complete the watermark embedding. Another advantage is that the DCT coefficients will be quantized before encryption, so all the DCT coefficients in the encryption domain are integers. Hence, almost all embedding operations in the encryption domain are integer additions. This feature makes QIM very suitable in the Pailliar homomorphic cryptosystem. Another feature of QIM is that it is robust against JPEG compression, which is very suitable for the current network environment where JPEG images are the mainstream. The above features are the reasons why we choose QIM.



doi:10.6342/NTU202203199

## 3.1 Notations

- Buyer : one of the customers of Seller
- Seller : the digital content owner
- Cloud : the trading platform and the cloud service provider
- $I$ : cover image
- $I_D$ : degraded image
- $I_R$ : residual image
- $x_i$ : i-th DCT coefficient of a block in a cover image
- $x_i$' : embedded i-th DCT coefficient of a block
- X : a vector of DCT coefficients of a block in a cover image
- X' : a vector of modified DCT coefficients of a block
- $w_i$ : i-th bit of watermark $W$
- $\tilde{w}_i$ : modified i-th bit of watermark $W$
- $W_1$ : Buyer's watermark
- $W_2$ : Seller's watermark
- $s^l$ : l-th spread code
- $s^l_\mu$ : μ-th bit of l-th spread code
- $S_1$ : spread codes for embedding Buyer's watermark
- $S_2$ : spread codes for embedding Seller's watermark
- $\alpha^l_\mu$ : μ-th bit of l-th approximation code
- W(I) : watermarked image
- $W^2(I)$ : embed two watermarks image
- E() : Encryption function of Paillier cryptosystem
- D() : decryption function of Paillier cryptosystem



doi:10.6342/NTU202203199

- E(x) : the ciphertext of x
- H(W) : the hash value of watermark W
- L : length of the watermark
- N : the number of DCT coefficients in a block for embedding a bit of the watermark

## 3.2 Procedures of the Proposed Mechanism

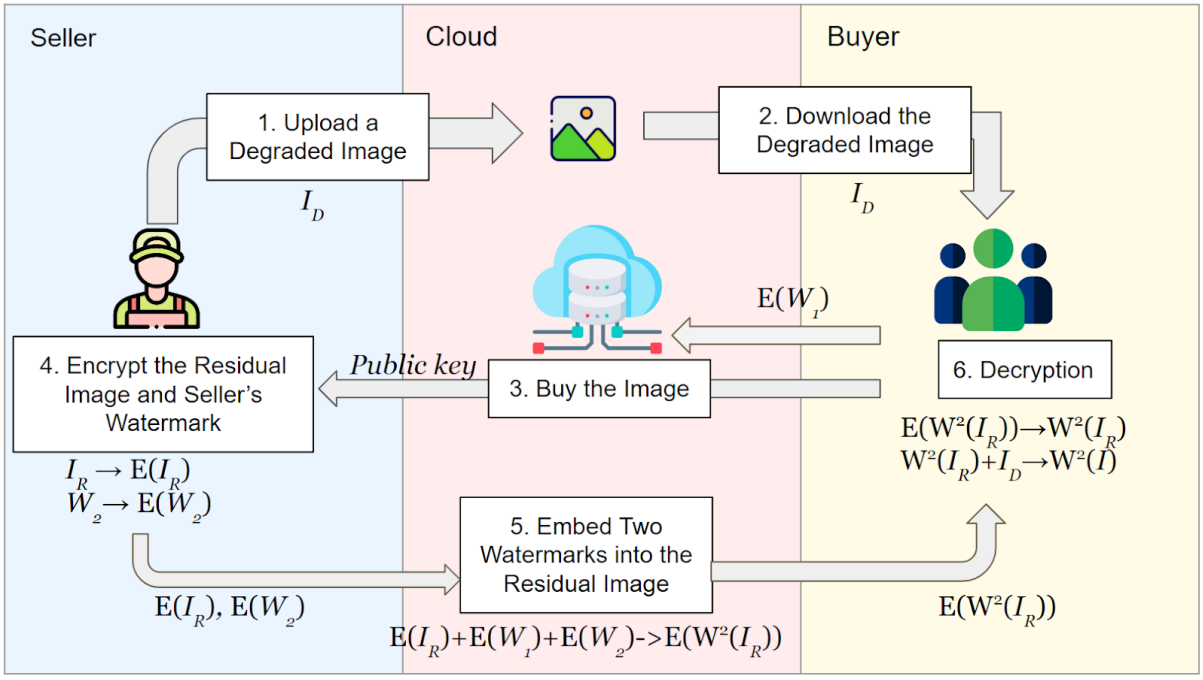



Figure 4: The Procedure of The proposed E-trading mechanism for an image among three involved parties: the seller, the buyer, and the cloud.

### 3.2.1 Upload a Degraded Image

Seller converts an image into DCT(I) using DCT transform with block size 8. Thus, Seller extracts specific DCT coefficients as a residual image, called DCT($I_R$). Other DCT coefficients are converted back to the spatial domain to obtain a degraded image, called $I_D$. $I_D$



doi:10.6342/NTU202203199

is uploaded to Cloud as a display image for Buyer to browse. If the Buyer wants to purchase the original image, a follow-up transaction procedure is required.

### 3.2.2 Download the Degraded Image

When the Buyer decides to purchase the image, it downloads $I_D$ and restores it later.

### 3.2.3 Buy the Image

Buyer will first use Pailliar cryptosystem to generate a public key, $K_p$, and a secret key, $K_s$, and then submit a trading request to the Cloud. The trading request contains an encrypted watermark $E(W_1)$, the hash value of watermark $H(W_1)$, and $K_p$. After receiving the transaction request, Cloud passes $K_p$ to Seller and maintains the information about Buyer and the hash value $H(W_1)$ in the trading records table.

### 3.2.4 Encrypt the Residual Image and Seller's Watermark

First, $DCT(I_R)$ is quantized by the quantization step size $2\Delta$, getting $DCT(Q(I_R))$. The strategy of choosing $\Delta$ is based on an improved perceptual model. Here, we choose the value of $\Delta$ from the standard JPEG quantization table[15]. Three involved parties share the same table. Seller encrypts the quantized residual image and Seller's watermark with $K_p$, obtaining $E(DCT(Q(I_R)))$ and $E(W_2)$, respectively. Besides them, spread codes $S_1$, $S_2$, and the hash value $H(W_2)$ are sent to Cloud.

### 3.2.5 Embed Two Watermarks into the Residual Image

The image is divided into many 8X8 blocks, a block embeds a bit of a watermark. Now, Cloud has $E(DCT(Q(I_R)))$, $E(W_1)$, $E(W_2)$, $S_1$, and $S_2$. In this step, Cloud performs QIM in the encryption domain. Based on the experiment result(cf. Appendix), embedding positions



doi:10.6342/NTU202203199

are (5, 0) and (6, 0) respectively in the DCT domain. To succinctly describe the embedding process, we just detail the embedding one of watermark in (5, 0) in the following:

A length N spread code $s^l = (s^l_1, s^l_2, \ldots, s^l_N)$ is generated, where $s^l_n \in \{-1, 1\}$. Notice that, a spread code covers a length N list of DCT coefficients in a block. N can be regarded as bandwidth in embedding a bit. And a length L watermark $W=(w_1, w_2, w_3, \ldots, w_L)$ is generated, where $w_l \in \{-1, 1\}$. Given the l-th bit of watermark $w_l$, then the encrypted spread code $s^l_n$ , $E(\tilde{\omega}^l_\mu)$ is computed by

$$E(\tilde{w}^l_\mu) = E(w_l) \otimes s^l_\mu, \quad \mu = 1, 2, 3, \ldots, N \tag{3.1}$$

To do the binary-to-bipolar mapping, we apply

$$E(\tilde{w}^l_\mu) = \frac{1}{2} \otimes (E(\tilde{w}^l_\mu) \oplus 1), \tag{3.2}$$

which converts $\tilde{\omega}_\mu{}^l \in \{-1, 1\}$ to $\tilde{\omega}_\mu{}^l \in \{0, 1\}$. This conversion is required for QIM. To improve the quality of the image, the usage of approximation code can make the DCT coefficient of the image modified by QIM closer to its original counterparts. In the QIM method, if a quotient of the DCT coefficient divided by the quantization table is odd, the embed bit is 1. Hence, the approximation code decides whether the odd quotient is upper or lower than the even quotient. The product of the approximation code $\alpha^l_\mu$ and $E(\tilde{\omega}^l_\mu)$ can be represented as

$$E(\tilde{w}^l_\mu) = \alpha^l_\mu \otimes E(\tilde{w}^l_\mu) \tag{3.3}$$

, where $\alpha^l_\mu \in \{-1, 1\}$. And the final step is embedding the watermark in the DCT coefficient, which is given by

$$E(Q(x^l_\mu)') = E(Q(x^l_\mu)) \bullet (\Delta \otimes E(\tilde{w}^l_\mu)) \tag{3.4}$$

Aggregating all the above steps, the overall formula can be represented as

$$E(Q(x^l_\mu)') = E(Q(x^l_\mu)) \bullet (\Delta \otimes \alpha^l_\mu \otimes \frac{1}{2} \otimes (E(w_l) \otimes s^l_\mu) \oplus 1)), \quad \mu = 1, 2, 3, \ldots, N \tag{3.5}$$



doi:10.6342/NTU202203199

where N=1 in our implementation. Next, the second watermark embedding is conducted following the above process again. After the embedding process is finished, $E(W^2(I_R))$ is stored in the retrieving address, so Buyer can download the data from this address.

### 3.2.6 Decryption

After the Buyer gets $E(Q(W^2(I_R)))$, Buyer can decrypt it with the private key, and dequantize it. Finally, Buyer merges the watermarked residual image $W^2(I_R)$ with the degraded image $I_D$ to restore the watermarked original image $W^2(I)$.

### 3.2.7 Identification

If the watermarked image was distributed illegally, Seller performs matching by detecting the watermark $W_2$ with the spread codes $S_2$. In an implementation, Seller changes $S_2$ during a period, so there are many $S_2$s corresponding to different sets of trading records. If a certain $S_2$ extracts $W_2$ successfully, Seller could locate the set of trading records. In the set, each record corresponds a $S_1$. By using $S_1$, many "$W_1$" are extracted. Then, $H(W_2)$ and $H(W_1)$ are sent to Cloud due to the request for trading records. Cloud performs matching and returns the corresponding trading record, which includes Buyer's identity.

### 3.2.8 Dispute Resolution

In court, Seller can prove his/her IPRs based on the watermark $W_2$. Besides, $W_1$ and the trading records are necessary for judging. The hash values in the trading records, $H(W_1)$ and $H(W_2)$, validate the watermarks' authenticity.



doi:10.6342/NTU202203199

# Chapter 4

# Introduction to Implementation of Proposed Mechanism

We have implemented this mechanism. The following will demonstrate the interface and operation flow of the demo program.

**Open the folder**

First, download the file named *Thesis.zip* and unzip it. The file path of the code, executable file, and images for illustrating is *Thesis/Demo* .

**Execute the File**

There are two executing methods. One is using the python compiler.

Instructions:

*python Seller.py*

*python Cloud.py*

*python Buyer.py*

The executing environment is

*python 3.9.9*

*numpy 1.22.2*

*phe 1.4.0*

*pyQt5 5.15.6*

*opencv-python 4.5.5.62*



doi:10.6342/NTU202203199

Also, you can directly execute the following instructions under Win10:

*Seller.exe*

*Cloud.exe*

*Buyer.exe*

**● Upload a Degraded Image**

Choose Seller's window, then the seller will see three tabs upside, Publish, Response, and Detect, which are three phases the seller will go through. On Publish page, the seller should select an image, then the image shows up on the left segment. After pressing the "Separate" button, the seller will see the degraded image show up on the right segment, and press the "Publish" button to send it to Cloud.

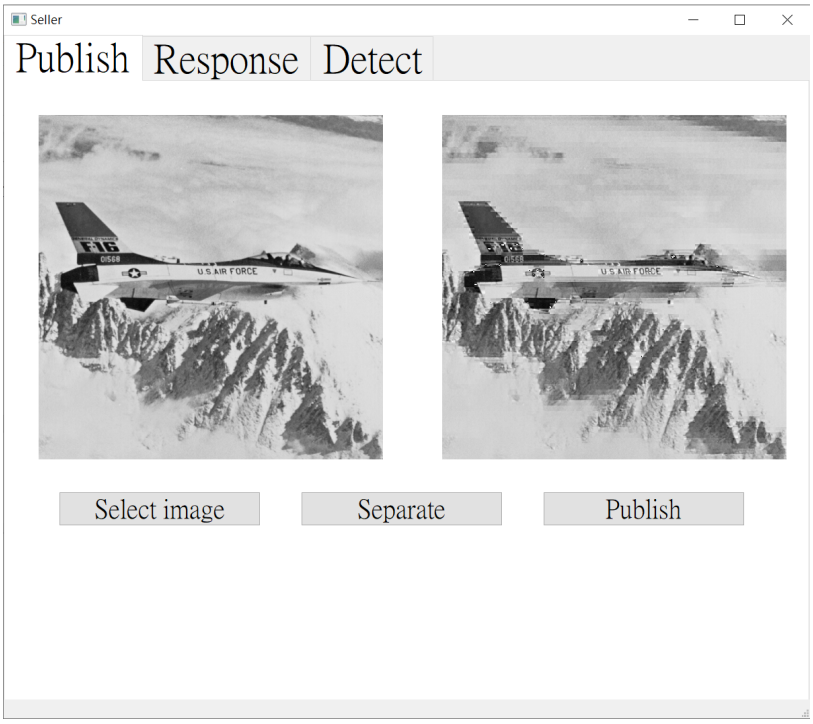



Figure 5: The snapshot of upload a degraded image.



doi:10.6342/NTU202203199

● **Check if the Image Had Been Published onto Cloud**

Choose Cloud's window and press the "Update" button, the degraded image will display on the right side. There is no buyer now.

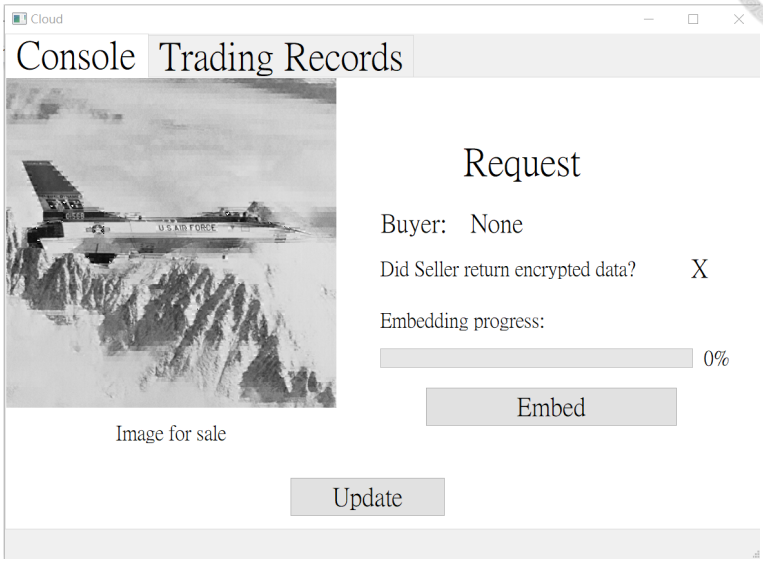



Figure 6: The snapshot of checking if the image had been published onto Cloud.

● **Download the Degraded Image**

Next, choose Buyer's window, also press the "Update" button, it should display the degraded image.


Figure 7: The snapshot of downloading the degraded image.



doi:10.6342/NTU202203199

● **Buy the Image**

If the buyer wants to buy the image, he/she clicks the “User Setting” tab, going to the next interface like follow snapshot. The buyer should generate his/her pailliar keypair and watermark, and press the “Buy” button for transmitting the encrypted watermark and public key to the cloud. After the above operations, the buyer gets the retrieving address “./Encrypted_image”.

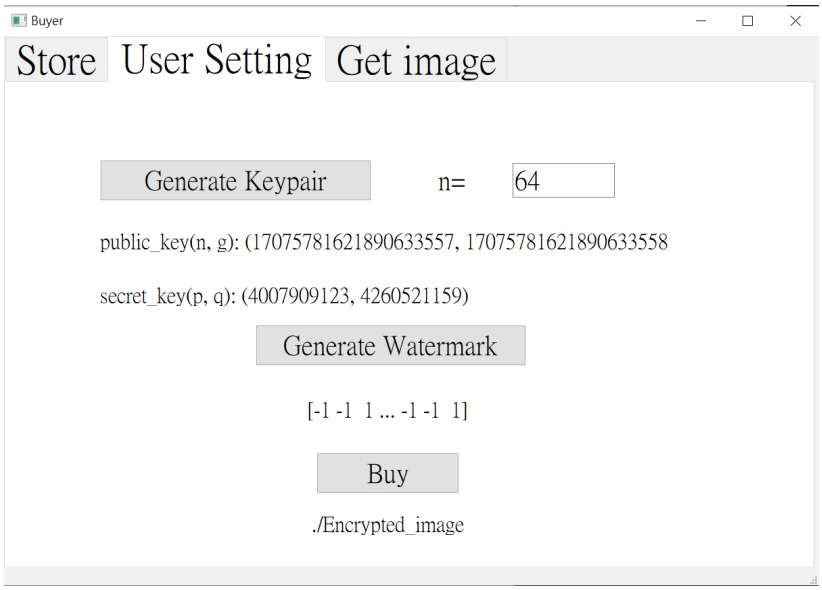



Figure 8: The snapshot of generating the buyer’s keypair and watermark.

The cloud can check the request from the buyer by clicking the “Update” button. If the request arrives, the buyer’s public key show after the “Buyer” text. Meanwhile, the request is passed to the seller.



doi:10.6342/NTU202203199

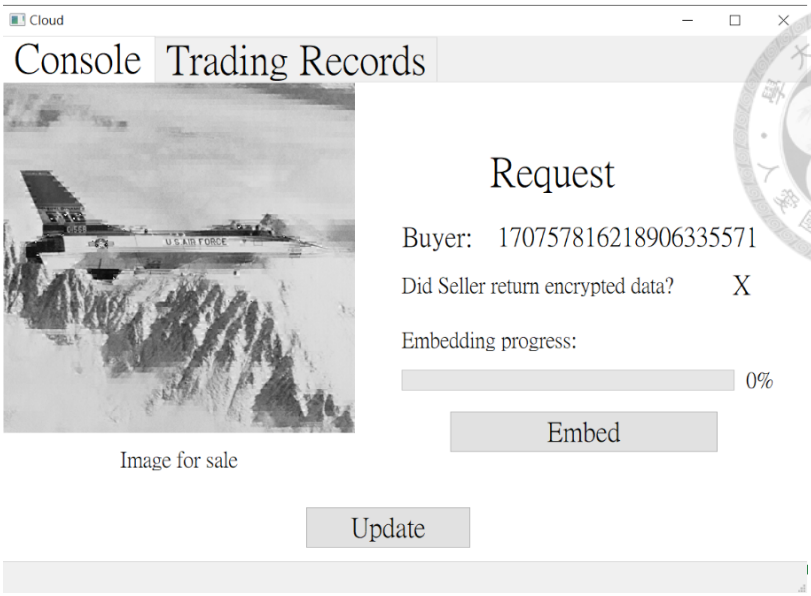
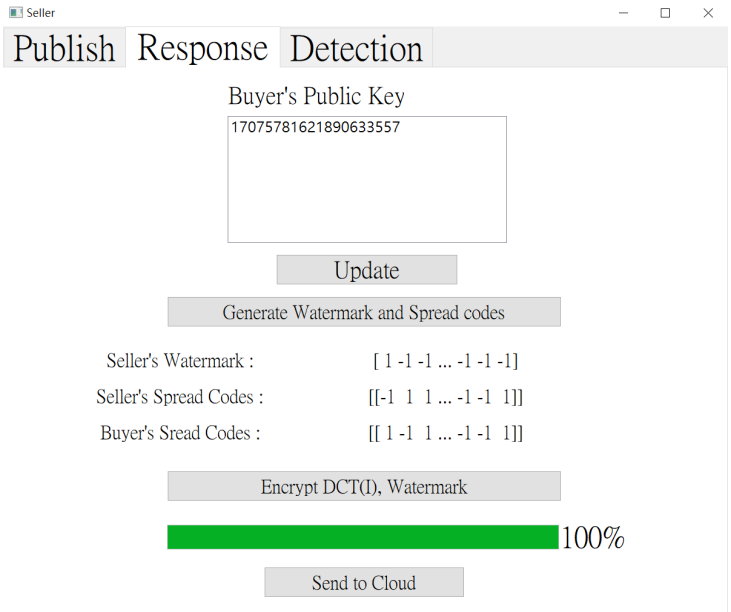



Figure 9: The snapshot of checking if the cloud receives the request from the buyer.

● **Encrypt the Residual Image and Seller's Watermark**

Click the "Response" tab upside. On this page, click the "Update" button, and the buyer's public key will show in the box, which means the seller receives the request from the buyer. Next, the seller should generate his/her watermark and spread codes, which is for embedding two watermark. After that, the seller clicks the "Encrypt DCT(I), Watermark" and "Send to Cloud".






doi:10.6342/NTU202203199

Figure 10: The snapshot of the seller handle the request from the buyer.

● **Embed Two Watermarks into the Residual Image**

The cloud can check whether the request had been finished by clicking the “Update” button. If the request had been finished, the cloud can click the “Embed” button to perform the embedding process. Then, the encrypted watermarked residual image will be stored in the retrieving address known by the buyer.

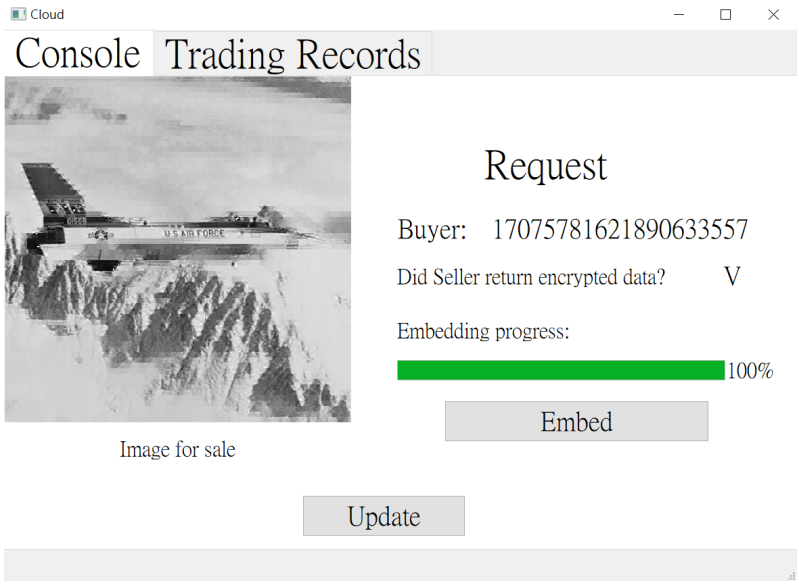



Figure 11: The snapshot of embedding watermarks into the residual image.

● **Decryption**

Lastly, after clicking the “Restore” button, the buyer retrieves the encrypted watermarked residual image, decrypts it, and restores the original image with the residual image.



doi:10.6342/NTU202203199

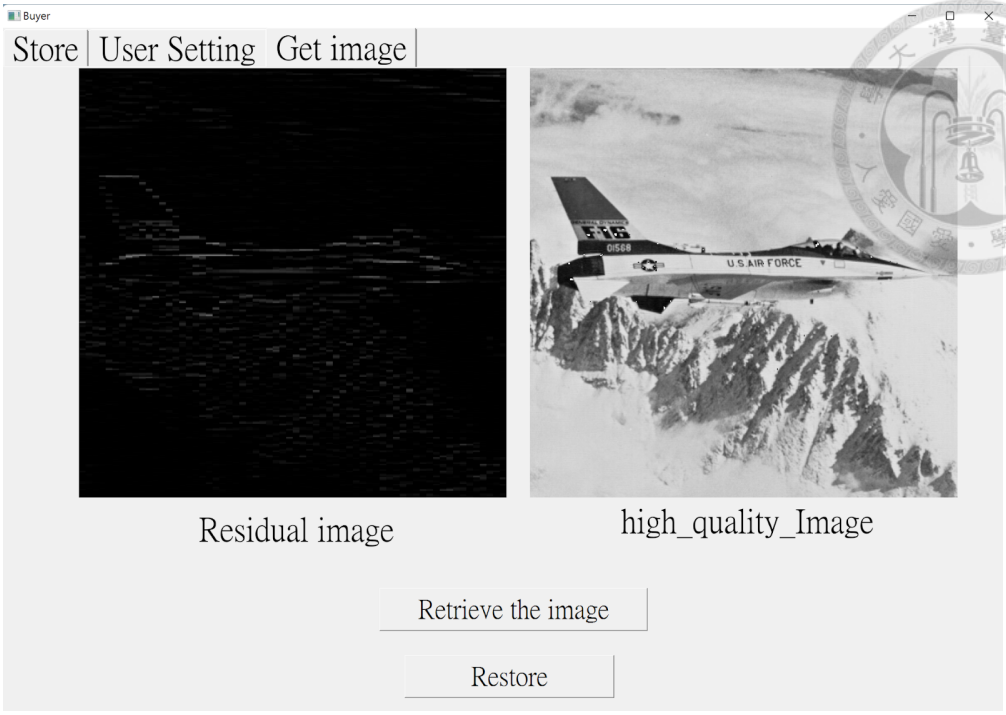



Figure 12: The snapshot of restoring the original image.

● **Identification**

When the watermarked image was illegally distributed, the seller can move to the "Detection" page, like the following snapshot. The seller selects the suspicious image, and clicks the "Extract Watermark". The buyer's watermark, the seller's watermark within the image, and the hash value of them will show up on the downside. Hence, this information can be used to search the trading records on the cloud side.



doi:10.6342/NTU202203199

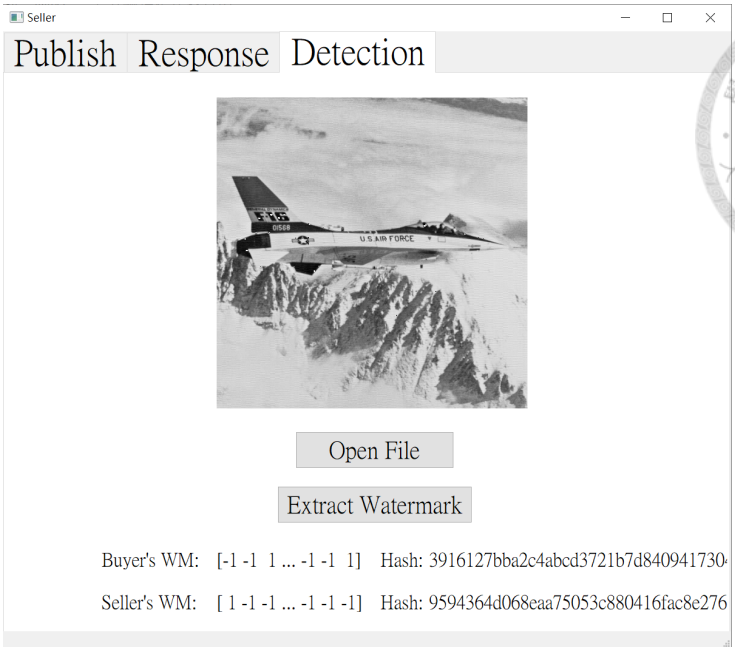



Figure 13: The snapshot of detecting two watermarks.

On the “Trading Records” page of the Cloud window, Clicking the “Update” button can check the trading records. With the hash value of the seller’s watermark and the Buyer’s watermark, the cloud can perform matching in the following table. When the trading record is found, the information about the buyer in the record is also known by the seller.

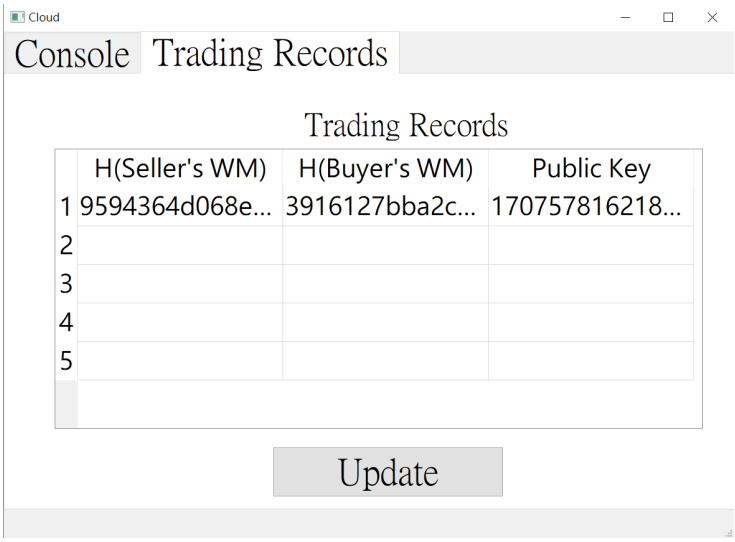



Figure 14: The snapshot of trading records table on the Cloud window.



doi:10.6342/NTU202203199

# Chapter 5

# Analyses and Experiments

In this section, we discuss the proposed mechanism's improvements in efficiency, robustness, and security, as compared with those of the CBSP-2(cf. section III.3).

## 5.1 Analysis of the Proposed Mechanism

Similar to CBSP-2, we regard the cloud as an E-commerce platform. It acts as a platform for publishing digital contents for creators while also sharing the burdens of data storage and computing of the creators. In short, the efficiency and robustness of the data and keys interchanging protocol and the embedding method are improved.

A noticeable change is that the encrypted images stored in Cloud are now presented in a degraded quality. It not only allows buyers to preview the digital content and is more in line with modern business models but also reduces the encryption and decryption cost of the Cloud. By referring to the clientside watermarking, Cloud does not need to use individual public keys for each buyer to encrypt images but only needs to encrypt and transmit the residual images. The ratio of the residual image is determined by the seller. If one over fourth of the host image is retained as the residual image, then the cost of computing, storage, and bandwidth reduce to one-fourth. In addition, we propose a more suitable watermark embedding method, QIM, which improves the efficiency of the embedding process in the encrypted domain. In the experiment(cf. Figure 14), The QIM's embedding time is 3.62



doi:10.6342/NTU202203199

seconds, which is about one-ten than ISS-WS. That is an acceptable computation time for a usual buyer.

## 5.2 Efficiency Analyses

### 5.2.1 Notations

- $O_{Enc}$ : an operation of encrypting a coefficient
- $O_{int}$ : an operation between encrypted integer and (encrypted)integer
- $O_{float}$ : an operation between encrypted float and (encrypted)integer or float
- $O_{enc}$ : an operation of encrypting a number
- $S_{Img}$ : the size of the host image
- $S_{R\_Img}$ : the size of the residual image, which is empirically less than one-fourth $S_{Img}$.

### 5.2.2 Quantization Index Modulatiom method[7]

QIM is used in our mechanism. First, because QIM does not involve a vector inner product operation, a bit can be embedded within an addition operation on a DCT coefficient. Compared with ISS-WS, embedding a bit requires the inner product of the DCT coefficient vector and the reference pattern. QIM is much simpler. Second, QIM method quantizes the DCT coefficients before performing the embedding operation, so all DCT coefficients become integers, which is a great property in the Pailliar homomorphic cryptosystem. The reason is that floating-point operation requires extra computation in the cryptosystem. Before adding two floating-point numbers, it takes many integer additions to keep the exponents of the two floating-point numbers the same. Only if this premise is achieved can the floating-point operation be done. QIM rarely uses floating-point operation. According to the formula,



doi:10.6342/NTU202203199

$$E(Q(x_\mu^l)') = E(Q(x_\mu^l)) \bullet (\Delta \otimes \alpha_\mu^l \otimes \frac{1}{2} \otimes (E(w_l) \otimes s_\mu^l) \oplus 1)), \quad \mu = 1, 2, 3, \ldots, N \tag{4.1}$$

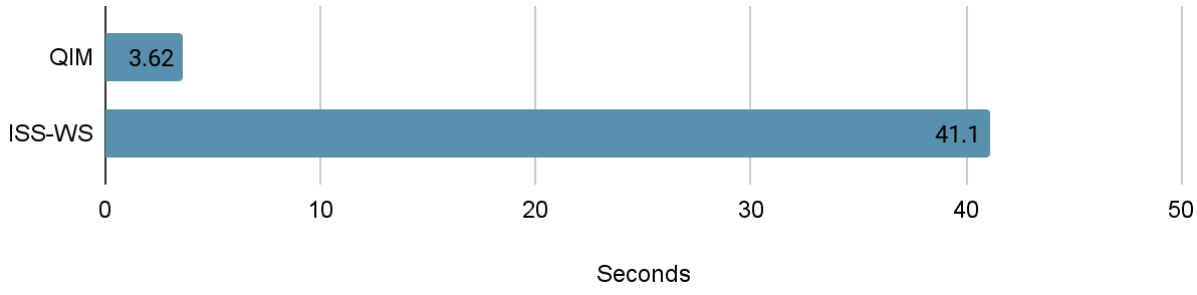
In order to compare on the same basis, bandwidth is set as 1, which means N is 1. Embedding one bit in a block needs $5O_{int}+1O_{float}$. In the implementation, because of two watermarks are needed, the time complexity is $10O_{int}+2O_{float}$.

### 5.2.3 Improved Spread-Spectrum based Watermarking Schemes[3]

ISS-WS is used in previous protocol[4](as mentioned in Section III-E). The formula in the encrytion domain is :

$$\prod_{m=1}^{n} E(x_j') = \prod_{m=1}^{n} E(x_j) \bullet \prod_{\beta=1}^{2} \left( \prod_{m=1}^{N} E(w_{\beta,i})^{\alpha} u_{\beta,j} \bullet \prod_{k=1}^{N} E(-x_k) u_{\beta,k} u_{\beta,j} \right) \tag{4.2}$$

To be on the same condition, bandwidth is set as 2, which means N is 2, and empirically, set **α** as 1, while embedding two bits in a block. So the time complexity is $6O_{int}+16O_{float}$, or after optimizing the order of operations, it would be $11O_{int}+10O_{float}$. But it is still more time consuming than QIM. Besides the time spent in embedding process, CBSP-2 must encrypt all pixels of the host image and unblind them in the encryption domain. The total time complexity is $O_{enc}S_{Img}+O_{float}S_{Img}+11O_{int}+10O_{float}$. However, In our proposed machism, the total time complexity is $O_{enc}S_{R\_Img}+10O_{int}+2O_{float}$.






doi:10.6342/NTU202203199

Figure 15: The histogram of QIM and ISS-WS embedding time.

Figure 14 shows the average embedding time of 4096 bits buyer's watermark and seller's watermark running on Intel i5-4300U CPU. The result is taken from an average of 100 rounds experiment.

## 5.3 Secure Analyses

### 5.3.1 Identification

A potential security hole was that any seller could ask Cloud for anyone's trading records. It is a serious security issue. That is, the seller must have $H(W_{Seller})$ and $H(W_{Buyer})$. $H(W_{Seller})$ proves the seller's IPRs and $H(W_{Buyer})$ proves that the one distributed the image. But there are two cases after the seller obtains the trading records. First, the trading record contains direct information about the malicious buyer, then the seller takes the Dispute Resolution. Second, a buyer registers an account in anoymous, only a little information about the Buyer leaves. The seller can entrust the reconnaissance agency to investigate, like the solution for internet criminals.

### 5.3.2 Watermarks leakage

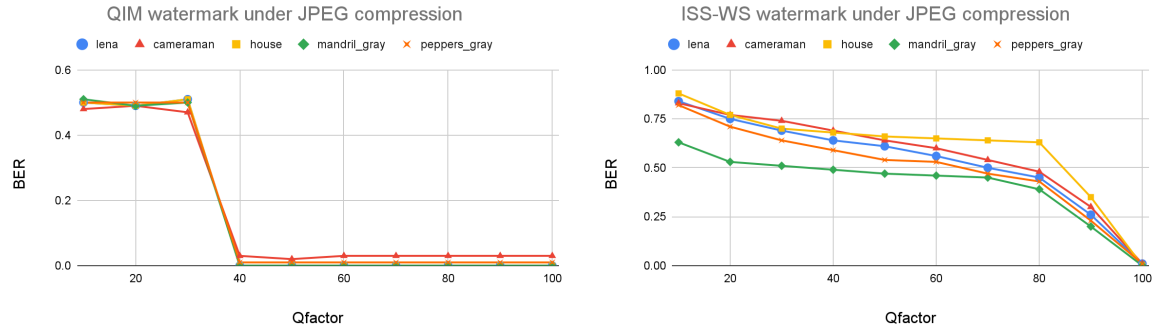
Our mechanism uses hash value of watermarks instead of the value of watermarks to search trading records. This method prevents the cloud from learning about the watermark content.



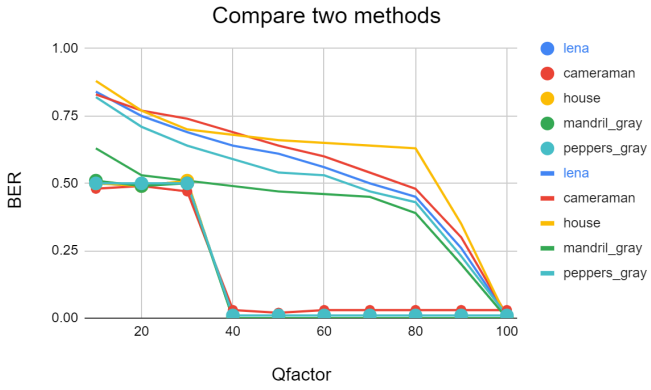
doi:10.6342/NTU202203199

## 5.4 Robustness Analyses

Images are usually distributed over the Internet in JPEG-compressed format, so the robustness of watermarks under JPEG compression is essential. We did the following experiments to verify the robustness of the two methods.




(a) (b)




(c)

Figure 16: Performance of two watermark embedding methods under JPEG compression. (a) bandwidth=2, quantization step Δ is set as Qfactor 50; (b) bandwidth=4; (c) Compare the above two methods with the same five images. The circle represents QIM's results, and the simple line represents ISS-WS's results.

For reducing bias from the variations of different images, we test the five images, and the result of trends are similar. Based on the experimental results(cf. Appendix), the optimal



doi:10.6342/NTU202203199

parameters of QIM are bandwidth=2, Quality factor=50, and 20~21th DCT coefficients. Notice that, Quality factor(Qfactor) is a parameter for controlling the quantization step on JPEG compression. Given a Quality factor, the specific quantization table can be generated accordingly[15]. BER is the abbreviation of bit error rate, which indicates the percentage of bits that have errors relative to the total number of bits received in a transmission. According to Figure 15(a), the BER is close to zero when Qfactor is larger than 40, it matches the anticipation. Figure 5(b) shows that QIM outperforms the ISS-WS method under JPEG compression.



doi:10.6342/NTU202203199

# Chapter 6

# Conclusion

This thesis proposes a more suitable mechanism for modern business models. It provides a security guarantee by using the buyer-seller protocol[6, 8, 9, 14] and the ability to protect the rights of both sides. It also successfully combines the efficiency of client-side watermarking. Additionally, choosing QIM that computes faster in the encryption domain successfully breaks through the time bottleneck of the entire mechanism. This change makes service response time shortened enough(cf. figure14) to be accepted by normal buyers. However, there are still many issues to be discussed, such as if Cloud has a spread code that can extract the watermark, will it affect the security? And how to extend the results of this mechanism to other digital contents, such as audio and documents, many cases still need to be addressed.



doi:10.6342/NTU202203199

# Bibliography

[1] Celik, M. U., Lemma, A. N., Katzenbeisser, S., & Van Der Veen, M. Lookup-Table-Based Secure Client-Side Embedding for Spread-Spectrum Watermarks. IEEE Transactions on Information Forensics and Security, 3(3), 475-487, 2008.

[2] Kuribayashi, M., & Tanaka, H. Fingerprinting protocol for images based on additive homomorphic property. IEEE Transactions on Image Processing, 14(12), 2129-2139, 2005.

[3] Malvar, H. S., & Florencio, D. A. F. Improved spread spectrum: a new modulation technique for robust watermarking. IEEE Transactions on Signal Processing, 51(4), 898-905, 2003.

[4] Peng, Y.-J., Hsieh, Y.-C., Hsueh, C.-W., & Wu, J.-L. Cloud-based buyer-seller watermarking protocols. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, 2017, 2017-08-01.

[5] Sun, J.-H., Lin, Y.-H., & Wu, J.-L. Secure Client Side Watermarking with Limited Key Size. In MultiMedia Modeling (pp. 13-24). Springer International Publishing, 2015.

[6] Chin-Laung Lei, Pei-Ling Yu, Pan-Lung Tsai and Ming-Hwa Chan, "An efficient and anonymous buyer-seller watermarking protocol," in IEEE Transactions on Image Processing, vol. 13, no. 12, pp. 1618-1626, Dec. 2004, doi: 10.1109/TIP.2004.837553.



doi:10.6342/NTU202203199

[7] Yamamoto, T., & Kawamura, M. Method of Spread Spectrum Watermarking Using Quantization Index Modulation for Cropped Images. IEICE Transactions on Information and Systems, E98.D(7), 1306-1315, 2015.

[8] N. Memon and P. Wong, “A buyer-seller watermarking protocol,” IEEE Trans. Image Process., vol. 10, no. 4, pp. 643–649, Apr. 2001.

[9] Alfredo Rial, Mina Deng, Tiziano Bianchi, Alessandro Piva, and Bart Preneel, “A provably secure anonymous buyer-seller watermarking protocol”, IEEE Trans. Inf. Forensics Security, vol. 5, no. 4, Dec. 2010.

[10] T. Bianchi and A. Piva, "Secure Watermarking for Multimedia Content Protection: A Review of its Benefits and Open Issues," in IEEE Signal Processing Magazine, vol. 30, no. 2, pp. 87-96, March 2013, doi: 10.1109/MSP.2012.2228342.

[11] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Advances in Cryptology—EUROCRYPT 1999 (Lecture Notes in Computer Science, no. 1592). New York: Springer, 1999, pp. 223–238.

[12] A. M. C. Correia, J. C. M. Silva, N. M. B. Souto, L. A. C. Silva, A. B. Boal and A. B. Soares, "Multi-Resolution Broadcast/Multicast Systems for MBMS," in IEEE Transactions on Broadcasting, vol. 53, no. 1, pp. 224-234, March 2007, doi: 10.1109/TBC.2007.891705.



doi:10.6342/NTU202203199

[13] Jia Liu and A. Annamalai, "Multi-resolution signaling for multimedia multicasting," IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004, 2004, pp. 1088-1092 Vol. 2, doi: 10.1109/VETECF.2004.1400189.

[14] Zhang, Jun, Weidong Kou, and Kai Fan. "Secure buyer–seller watermarking protocol." IEE Proceedings-Information Security 153.1 (2006): 15-18.

[15] Cogranne, Rémi. "Determining JPEG image standard quality factor from the quantization tables." arXiv preprint arXiv:1802.00992 (2018).

[16] A. Adelsbach, U. Huber, and A.-R. Sadeghi. Fingercasting—joint fingerprinting and decryption of broadcast messages. In *Information Security and Privacy*, pages 136–147. Springer, 2006.

[17] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen. Secure embedding of spread spectrum watermarks using look-up-tables. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–153–II–156, April 2007.

[18] M. Khan, V. Jeoti, A. Malik, and M. Khan. A joint watermarking and encryption scheme for dct based codecs. In *Communications (APCC), 2011 17th Asia-Pacific Conference on*, pages 816–820, Oct 2011.



doi:10.6342/NTU202203199

[19] T. Farah, H. Hermassi, R. Rhoouma, and S. Belghith. Watermarking and encryption scheme to secure multimedia information. In *Computer and Information Technology (WCCIT), 2013 World Congress on*, pages 1–5. IEEE, 2013.

[20] A. Piva, T. Bianchi, and A. De Rosa. Secure client-side st-dm watermark embedding. *Information Forensics and Security, IEEE Transactions on*, 5(1):13–26, March 2010.



doi:10.6342/NTU202203199

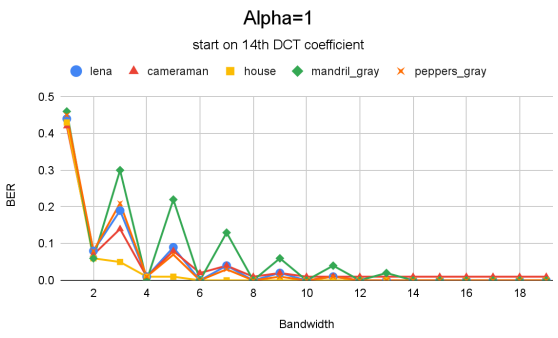
# Appendix

Notice that the position of DCT coefficients is located by Zig-Zag scanning order in the following experiments.

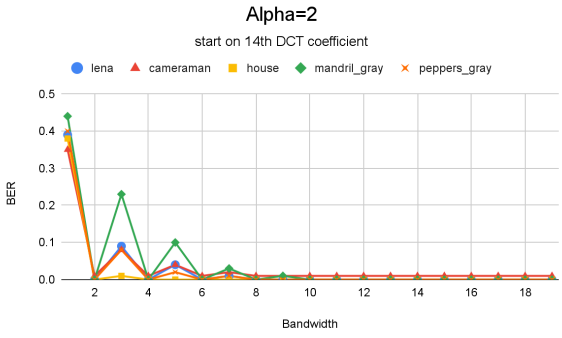
A. ISS-WS[3, 4]

The goal of the following experiments is that we want to search for the optimal parameters in ISS-WS, making the lowest Bit Error Rate(BER) and the hightest Peak signal-to-noise ratio(PSNR).

1) Measuring performance in Bit Error Rate(BER), The coefficients in the high-frequency domain may be cut off by the JPEG quantizer and they are not suitable to be used for embedding the given constraint. On the other hand, modifications in the low frequencies may cause visible diffusion in the image. Empirically, The most suitable frequencies to be embedded are those in the middle-frequency range. Hence, 14th~33th DCT coefficients are candidates. We tested in $\alpha$=1~4 in different images.



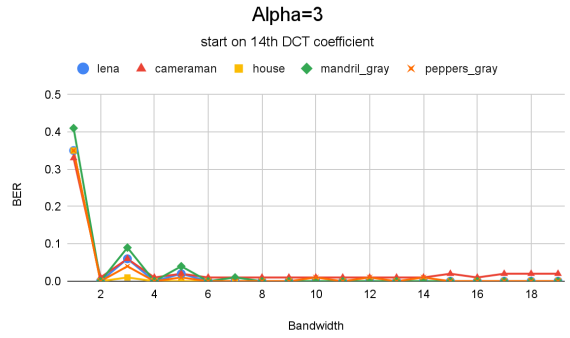

(a)



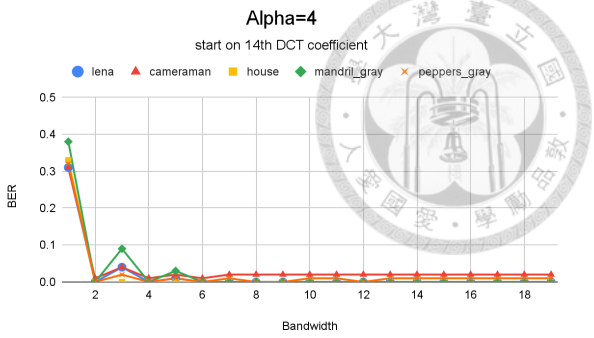

(b)



doi:10.6342/NTU202203199

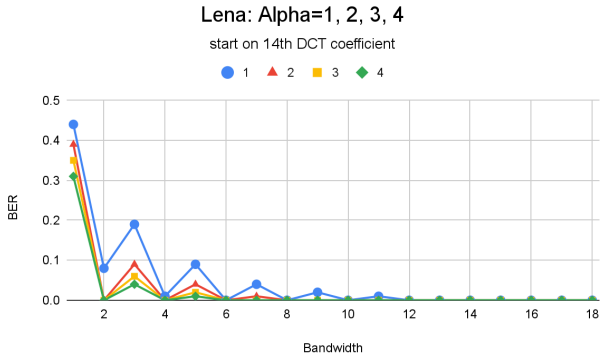



(c)




(d)

In all images, an observation is that the local optimum occurs when the bandwidth is an even number.




(e)

Figure 17: (a) Alpha=1; (b) Alpha=2; (c) Alpha=3; (d) Alpha=4; (e) Alpha=1, 2, 3, 4 in testing Lena.

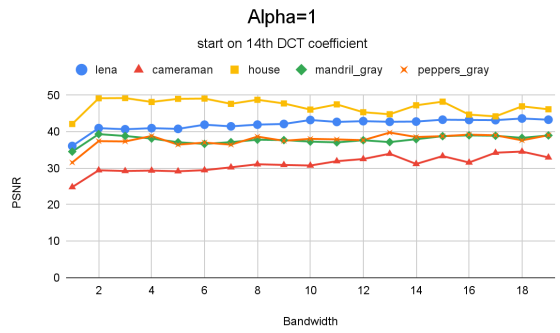
Drawing four lines associated with $\alpha$=1~4 in testing Lena, in the even number of bandwidth, the relatively good performance is occurred the local optimal in all tested alpha values. Hence, as shown in the above figure, $\alpha$=1 and bandwidth=4 may be the optimal parameter.

2) Measuring performance in Peak signal-to-noise ratio(PSNR), PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of the image. For short, PSNR is a metric to measure the fidelity of the image.

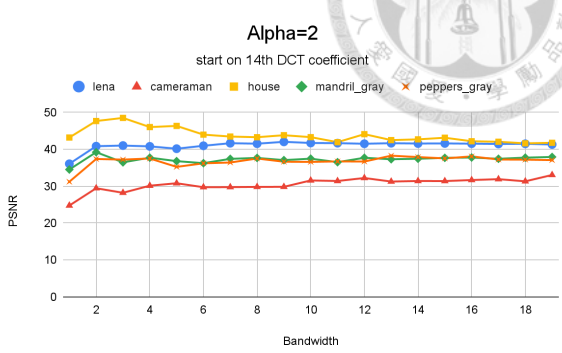


doi:10.6342/NTU202203199

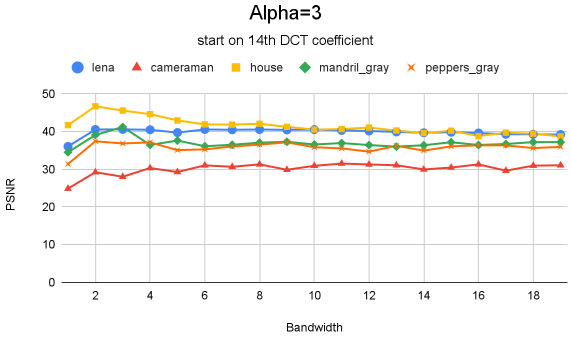
Bandwidth is the number of DCT coefficients for embedding a bit within a block. We tested in $\alpha$=1~4 in different images.



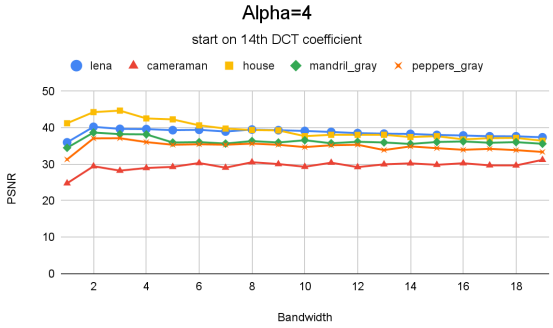

(a)



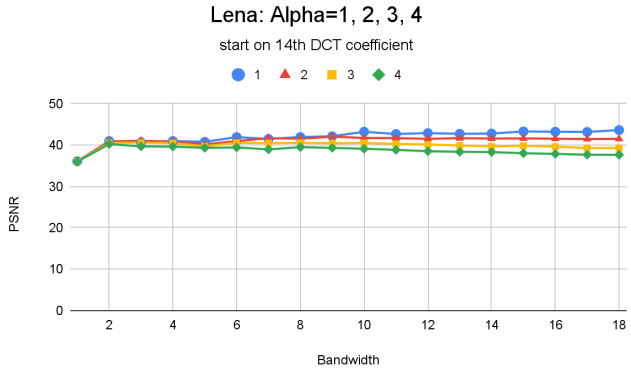

(b)




(c)




(d)




(e)

Figure 18: (a) Alpha=1; (b) Alpha=2; (c) Alpha=3; (d) Alpha=4 in testing five different images. (e) Alpha=1, 2, 3, 4 in testing Lena.

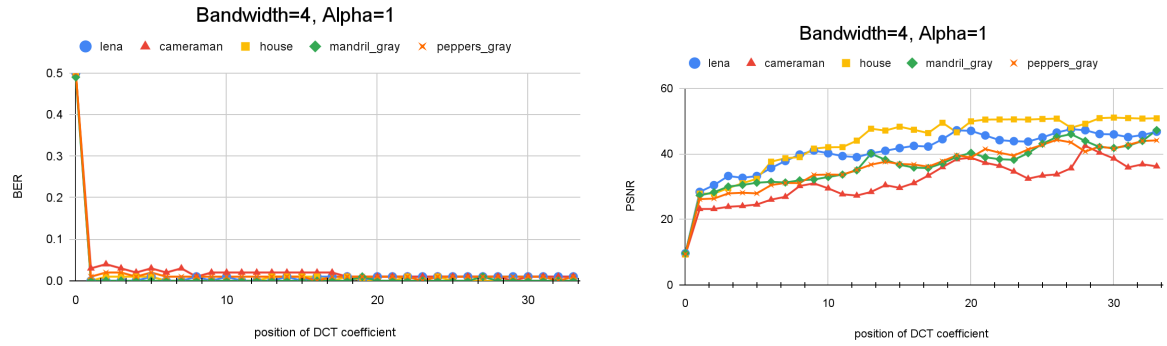


doi:10.6342/NTU202203199

In this experiment, we want to find out at what alpha and bandwidth the relatively good PSNR is obtained

According to the above chart, bandwidth is not a significant factor, and PSNR slightly decreases when $\alpha$ increases, so we continue the above decision $\alpha$=1 and bandwidth=4 is the optimal choice.

3) Fixing $\alpha$=1 and bandwidth=4, search for the best position to embedding measured by performance in BER and PSNR.




(a) (b)

Figure 19: (a) Measuring in BER; (b) Measuring in BER in testing five different images.

In the result of Section A-1&2, the optimal parameters are $\alpha$=1 and bandwidth=4, and here we want to find the best embedding position. The x-axis in Figure 18 is the starting position embedded a bit given $\alpha$=1 and bandwidth=4. Observing Figure 18(a), when the embedded position is larger than 18th, BER almost approaches zero. Observing Figure 18(b), in the 20th position, PSNR almost approaches the highest value in Lena and house and is the local optimum in other images. Hence, The 20th~23th DCT coefficients are our choice.

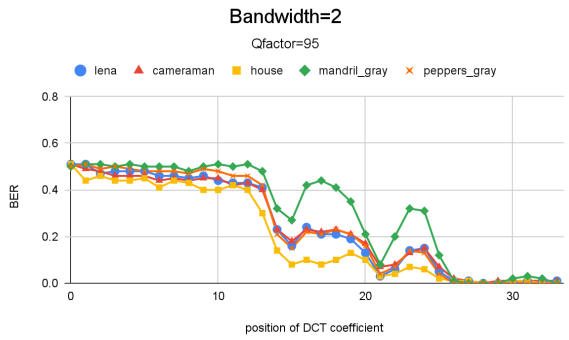
B. QIM[7]



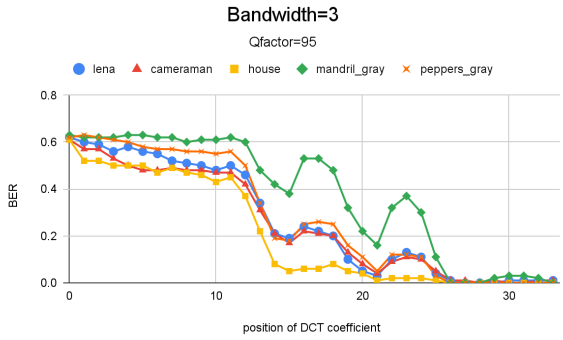
doi:10.6342/NTU202203199

As the ISS-WS's experiments, The goal of the following experiments is that we want to search for the optimal parameters in QIM, making the lowest Bit Error Rate(BER) and the highest Peak signal-to-noise ratio(PSNR).

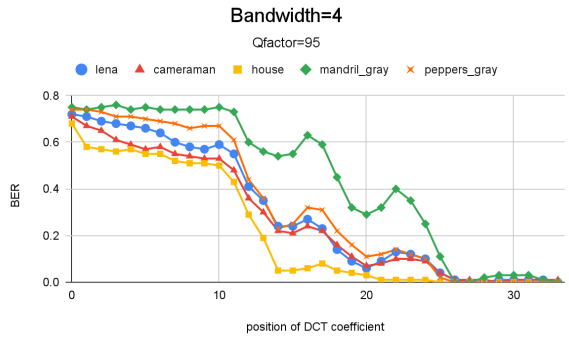
1) Measuring performance in Bit Error Rate(BER), we tested in bandwidth=2~5 in different images. The x-axis in Figure 19 is the starting position embedded a bit.



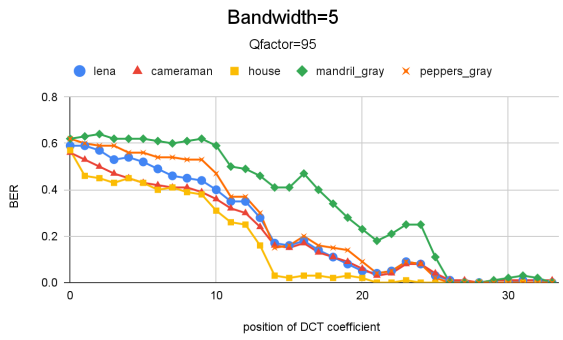

(a)



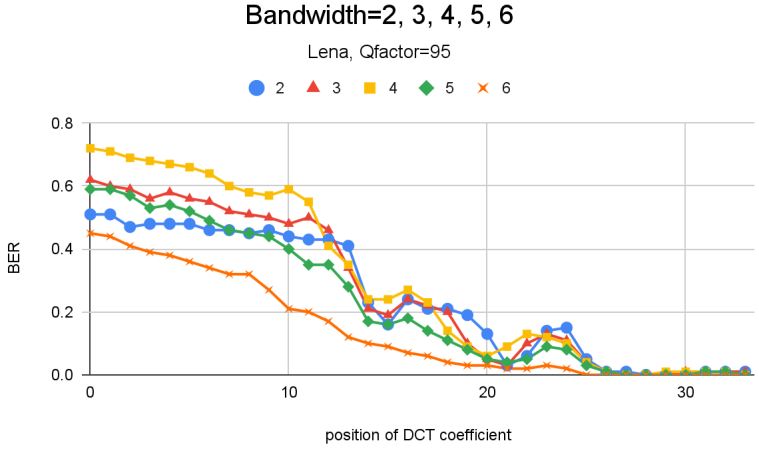

(b)




(c)




(d)



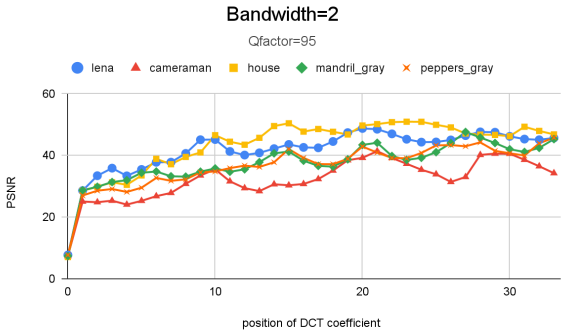



doi:10.6342/NTU202203199

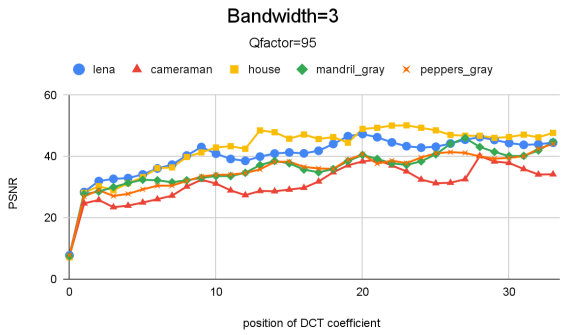
(e)

Figure 20: (a) Babdwidth=2; (b) Babdwidth=3; (c) Babdwidth=4; (d) Babdwidth=5 in testing five different images. (e) Babdwidth=2, 3, 4, 5, 6 in testing Lena.

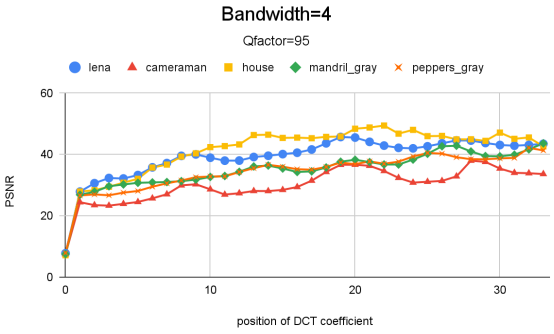
The 28th position is optimal, but the 20th position also has acceptable performance. When Bandwidth is 2 or 6, BER decreases observably.



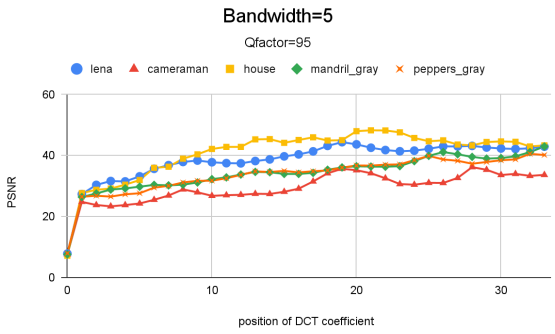

(a)




(b)



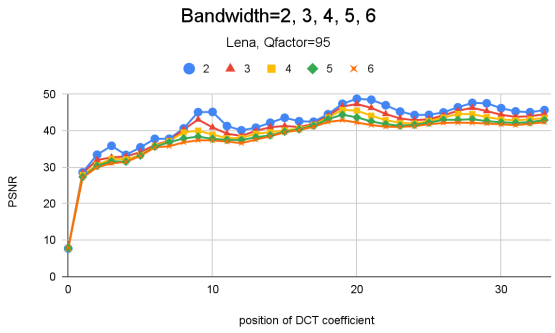

(c)




(d)



doi:10.6342/NTU202203199




(e)

Figure 21: (a) Babdwidth=2; (b) Babdwidth=3; (c) Babdwidth=4; (d) Babdwidth=5 in testing five different images. (e) Babdwidth=2, 3, 4, 5, 6 in testing Lena..

As expected, the best PSNR performance is in Bandwidth=2. According to the above results, Embedding in 20th DCT coefficient and bandwidth=2 are reasonable parameters.



doi:10.6342/NTU202203199