IEICE TRANS. INF. & SYST., VOL.E91–D, NO.5 MAY 2008

PAPER Special Section on Information and Communication System Security

Multi-Bit Embedding in Asymmetric Digital Watermarking without Exposing Secret Information*

Mitsuo OKADA[†], ***Student Member***, **Hiroaki KIKUCHI**[††], ***and*** **Yasuo OKABE**[†††], ***Members***

SUMMARY A new method of multi-bit embedding based on a protocol of secure asymmetric digital watermarking detection is proposed. Secure watermark detection has been achieved by means of allowing watermark verifier to detect a message without any secret information exposed in extraction process. Our methodology is based on an asymmetric property of a watermark algorithm which hybridizes a statistical watermark algorithm and a public-key algorithm. In 2004, Furukawa proposed a secure watermark detection scheme using patchwork watermarking and Paillier encryption, but the feasibility had not tested in his work. We have examined it and have shown that it has a drawback in heavy overhead in processing time. We overcome the issue by replacing the cryptosystem with the modified El Gamal encryption and improve performance in processing time. We have developed software implementation for both methods and have measured effective performance. The obtained result shows that the performance of our method is better than Frukawa's method under most of practical conditions. In our method, multiple bits can be embedded by assigning distinct generators in each bit, while the embedding algorithm of Frukawa's method assumes a single-bit message. This strongly enhances capability of multi-bit information embedding, and also improves communication and computation cost.

key words: *asymmetric digital watermarking, statistical digital watermarking, public-key encryption*

1. Introduction

The demand for contents security is increasing due to severe crime augmentation accompanying rapid development of information technology. All kinds of contents have become available in digital form, which might accelerate making of perfect copies of digital video, image, and music data. Despite the fact that an enormous number of those contents might be pirated for an illegal use, the copyright law had been the only enforceable protection against the crime till the technical protection mechanism such as information hiding was introduced. One of the major information hiding technique is a digital watermarking that makes copyright notice or some secret data concealed in the contents. The hidden information is used for claiming copyright, detecting tamper, and so forth.

The ideal form of digital watermark is the one in which hidden information should not be removed by any contents manipulations, the embedded contents should not be spoiled by embedding information, and hiding should not perceptually appear. However, the most critical issue of watermarking is its symmetric property, that is, exactly the same secret key is used for both embedding and extracting a message. In almost all of conventional watermarking algorithms, the secret key of modified pixels is exposed in extraction process. Hence, the risk of exposure of the secret in extraction process is not avoidable.

In the symmetric methods, verifiers need to be trustwealthy since they use a secret key to obtain an embedded message. On the other hand, the verifiers need not to be trustwealthy in asymmetric methods, since a secure encryption scheme should be able to extract the embedded message without exposing the secret information of which pixels have been modified. In other words, without an encryption, the secret information, the place of the modified pixels will be exposed at the extracting process. Therefore, malicious verifiers are easily able to obtain the secret information to remove the embedded message. The following asymmetric digital watermarking was proposed to resolve the problem.

Asymmetric watermark schemes applying a patchwork watermarking algorithm [3] and a homomorphic public-key algorithm for the secure detection have been developed by the following researchers. Patchwork watermark is one of the statistical digital watermarking schemers which may be one of the most robust methods because it embeds information in the skew of statistics of an image. Minematsu proposed a scheme [5] which applies patchwork watermark and Okamoto-Uchiyama encryption [6], but Furukawa who proposed a scheme [2] found a drawback that a secret key of the key authority may be exposed by malicious users. In order to resolve the drawback that allows an authorized verifier to detect the hidden message without revealing the secret information, Furukawa proposed the scheme based on [5] that uses patchwork watermark and Paillier encryption [4]. However, we found that the drawback in [2] is a heavy overhead of Paillier encryption, which makes the scheme inefficient. To resolve the issue, we propose a new scheme based on [2], which employs the modified El Gamal encryption instead of Paillier encryption.

The methods mentioned above are considered for only the case of single-bit information embedding. However, when applying for multi-bit information embedding, size of a ciphertext of the secret information increase along with the

Manuscript received July 25, 2007.
Manuscript revised December 16, 2007.

[†]The author is with the Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[††]The author is with the Dept. of Information Media Technology, School of Information Science and Technology, Tokai University, Hiratsuka-shi, 259–1292 Japan.
[†††]The author is with Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.

*The part of this work has been published at [1].

DOI: 10.1093/ietisy/e91–d.5.1348

number of bits to be embedded. That results in high communication cost.

Our contributions are (1) proposal of a new method showing a feasibility of our protocol, (2) evaluation of the performance of our protocol based on implementation, and (3) proposal of a new multi-bit embedding algorithm.

Main goal of our methodology is to prove the feasibility of the proposed protocol. To show the feasibility, we have selected a patchwork watermarking and the modified El Gamal encryption as an instance for implementation.

In this paper, after reviewing patchwork watermark and Furukawa's scheme in Sect. 2, we present a new watermarking scheme which allows secure detection of an embedded message and improves the efficiency in Sect. 3. In Sect. 4, we present a new method of expanding to multi-bit information embedding. In Sect. 5, we evaluate the performance of the proposed scheme based on our testbed implementation, which qualifies the scheme for secure watermark detection. We also estimate the efficiency of multi-bit embedding based on the single-bit embedding testbed.

2. Preliminarily

2.1 Statistical Watermarking

Patchwork watermarking, proposed in 1995 by Bender et al., embeds information in statistical value of contents [3]. In this method, embedding key is a seed of pseudo-random process which chooses a large number of pairs of pixels. The first pixel value of a pair is made slightly brighter and the second one is made slightly darker. This process is iterated for all pairs. Conceptually, the contrast between pixels of the pairs encodes some secret information.

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 watermarks that have a higher degree of robustness against attacks and image manipulations.

We describe a single-bit embedding process of patchwork watermark. First, we 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,

$$\lim_{n\to\infty} \bar{S_n} \to 0. \tag{1}$$

Figure 1 labeled as "Original Image" shows a distribution of differences in Lena (256×256 pixels, 256 gray scale levels), with $n = 10000$. At this experiment, we obtained $\bar{S_n} = 0.0121$, that satisfies the condition (1).

We describe an embedding process, how to hide a secret message ω into I. We 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

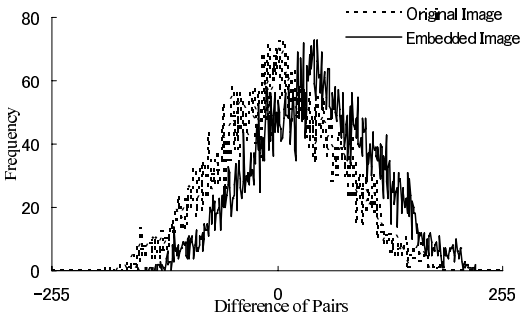


Fig. 1 Distributions of differences $(a_i - b_i)$ and $(a'_i - b'_i)$.

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

With the parameter of $\delta = 20$, the distribution of $(a'_i - b'_i)$ is shifted 40 to right as illustrated in Fig. 1. 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, we choose a'_i, and b'_i according to the random numbers, and then determine,

$$\omega = \begin{cases} 0 & \bar{S_n}' < \tau, \\ 1 & \bar{S_n}' \geq \tau, \end{cases} \tag{3}$$

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 Lena, we have $\bar{S_n}' = 40.0158$, which satisfies the condition of $\bar{S_n} \geq \tau = \delta = 20$.

2.1.1 Advanced Patchwork Watermarking Schemes

We introduce some of the advanced symmetric patchwork watermarking methods, [9], [10], and [11].

Arnold proposed the method [9] in which patchwork watermarking for an audio data in frequency domain is used and hypothesis testing has been made by power density function. Another his contribution is a scale-shift embedding, where multiplicative embedding as $a_i(1 + \delta)$, and $b_i(1 - \delta)$ is used instead of $a_i + \delta$, and $b_i - \delta$. Yeo and Kim proposed the Modified Patchwork Algorithm (MPA) [10] in which watermark is embedded in frequency domain. In MPA, a threshold is calculated based on sample means and sample variances and a sign function is used for embedding to improve the detection accuracy. This improvement enables the number of modified pairs, n, to be reduced to

around 50 samples. Yeo and Kim also proposed the Generalized Patchwork Algorithm (GPA) [11]. GPA applies the idea of additive and multiplicative embedding methods which provides flexibility in detection, and enhances robustness by maintaining a high accuracy of watermark detection.

These advanced symmetric patchwork watermark algorithms can be applied to our asymmetric watermarking protocol for the better performance in the embedding part. Any of the advanced techniques are convertible with our scheme, since an asymmetric part is independent from watermark algorithm. More details about improvement and performance are described in Sect. 5.7.

2.2 Cryptosystems

In this section, we review two cryptosystems, the modified El Gamal encryption, and Paillier encryption. They satisfy both an additive homomorphism and an indistinguishability denoted by IND[†].

The reason that we select these algorithms is that our proposed scheme requires the public-key algorithm satisfying both homomorphism and IND. IND is necessary since the only three kinds of plaintexts are encrypted in our proposed protocol. Otherwise, the plaintexts can be identified from the ciphertext. More details are described in Sect. 3.3. The relationship between public-key algorithms and properties is shown in Table 1. For our experiment, we select El Gamal and Paillier Encryption as representatives that have distinct difference in computation cost.

El Gamal encryption is a public key encryption algorithm. Customized version of El Gamal, called the modified El Gamal is used with patchwork watermark for our method. The security of the encryption relies on the difficulty of the discrete logarithm problem. Note that we adopt modified version of El Gamal encryption so that an additive homomorphism of ciphertext should be satisfied.

Let p and q be secure prime numbers and g be a generator of multiplicative group Z_p^*. The order of g is q. A public key y is defined by $y = g^x \bmod p$ where $x \in Z_q$ is a private key. A ciphertext of plaintext m, $E(m) = (c, d)$, is defined by $c = g^m y^r \bmod p$ and $d = g^r$. The decrypted ciphertext is obtained by $g^m = D(c, d) = c/d^x \bmod p$.

Paillier encryption is proposed in [4]. For the key generation phase, generate large prime numbers p, and q, and pick $g \in Z_{N^2}$ such that $\gcd(L(g^\lambda \bmod N^2), N) = 1$, where $N = pq$, $\lambda = \mathrm{lcm}(p-1, q-1)$. Note that public key is g, N and private key is p, q. For the encryption phase, let m be a plaintext to be encrypted, r be a random number chosen from Z_N, and E be an encryption function defined by

$$e = E(m) = g^m r^N \bmod N^2. \tag{4}$$

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

$$m' = D(e) = \frac{L(e^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N, \tag{5}$$

where $L(t) = (t-1)/N$.

Table 1 List of public-key algorithms.

Type of Encryptions	Homomorphism	IND	Computation Cost
Modified El Gamal	additive, (multiplicative)	YES	low
Paillier [4]	additive	YES	high
Okamoto-Uchiyama [6]	additive	YES	high
RSA	multiplicative	NO	low

2.3 Asymmetric Watermark Detection [5]

Minematsu proposed an asymmetric watermark scheme [5] in 2000. His scheme applies patchwork watermark detection by using homomorphic public-key encryption in order to detect watermark with exposing no secret information used in embedding process. A verifier possesses the embedded image and sends the image to the key authority for watermark verification either a watermark exists or not. As a homomorphic encryption algorithm, he uses Okamoto-Uchiyama encryption [6].

2.4 Secure Watermark Detection [2]

Furukawa proposed a secure patchwork watermark detection protocol by adopting Paillier encryption. The primal idea of this method is nearly same as [5]. However, the detection scheme is modified so that verifier can prove validity of results without revealing secret information. With proof of validity, the scheme prevents dishonest users from cheating a key authority.

In this protocol, detection is carried out by verifying a ciphertext which contains the indexes of the modified pixels. Due to its unique property of Paillier encryption, the watermark information is encoded as exponents of the ciphertext. In other words, the indexes of the modified pixels are never exposed to a verifier even after the extraction process is carried out.

An author defines threshold τ, and the number of pixels ℓ, and chooses random subsets A, $B \subset \{1, \ldots, \ell\}$. He also generates a pair of public key and private key of Paillier encryption. He then generates ciphertext $(e_1, \ldots, e_\ell)$ such that

$$e_i = \begin{cases} E[1] & \text{if } i \in A, \\ E[-1] & \text{if } i \in B, \\ E[0] & \text{otherwise.} \end{cases}$$

In watermark extraction scheme, a verifier who posses the embedded image $I' = (z_1, \ldots, z_\ell)$ computes $e = \prod_{i=1}^{\ell} e_i^{z_i}$, and sends e to a trusted key authority. In watermark detection process, the verifier identifies the watermark message ω as

$$\omega = \begin{cases} 0 & \text{if } D(e) < \tau, \\ 1 & \text{if } D(e) \geq \tau. \end{cases}$$

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

3. Proposed Scheme

3.1 Outline

To resolve the problem of the symmetric property of watermark system, our approach employs a concept of public-key encryption protocol to conceal the indexes of the modified pixels against the verifier. In order to assure trust between an author and a verifier, extraction process requires cooperation of a third party, who holds a private key of the modified El Gamal encryption.

A drawback of [2] is the heavy overhead of Paillier encryption, which is replaced by the modified El Gamal encryption in our scheme. Since patchwork watermark only needs to determine the sum of differences to be close to either 0 or $2n\delta$, it is possible to examine all possible messages, i.e., g^0, or $g^{2n\delta}$. Note that we examine $2n\delta$ (not 2δ as shown in Eq. (2)) in our scheme, because the division by n is not able to perform in modular arithmetic.

3.2 Model

In this section, we describe a model of our scheme using three entities, Alice, Bob, and Kevin, representing an author, a verifier, and a key authority. To simplify an explanation, we use a single-bit embedding.

Suppose that, Alice embeds information into the contents, Bob verifies the watermark, and Kevin generates a secret key sk and public key pk for the modified El Gamal encryption. Not only does interposal of the third party enhance the reliability of verification, but also prevent the author from cheating a verifier. Note that Kevin needs not to be fully trustworthy since he does not learn the embedding key, which is the index of modified pixels determined by Alice through out 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 an image I and I'. We illustrate our model in Fig. 2.

3.3 The Proposed Protocol

Kevin generates the modified El Gamal public key, $y = g^x \bmod p$, where 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.

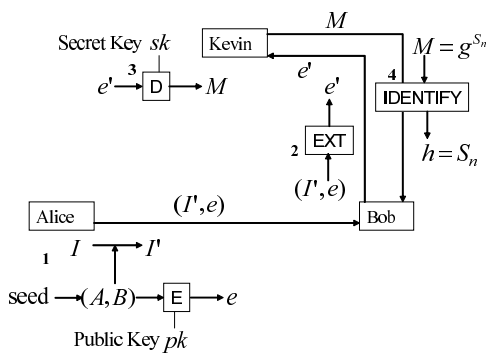


Fig. 2 The model of the proposed scheme.

STEP 1: (Embedding) Alice 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 $|A| = |B| = n$. She chooses δ and modifies pixels according to (A, B) in the image I to generate I' as

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

for $i = 1, \ldots, \ell$. Alice 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, Alice sends $I' = (z_1, \ldots, z_\ell)$ to Bob in conjunction with encrypted indexes $e = (c_1, \ldots, c_\ell, d_1, \ldots, d_\ell)$.

STEP 2: (Extracting) Bob computes ciphertext $e' = EXT(I', e) = (C, D)$ as follow;

$$\begin{aligned} C &= c_1^{z_1} c_2^{z_2} \cdots c_\ell^{z_\ell} = \prod_{i=1}^{\ell} g^{m_i z_i} y^{r_i z_i} \\ &= g^{\sum^{\ell} m_i z_i} y^{\sum^{\ell} r_i z_i} = g^{S_n} y^R, \\ D &= d_1^{z_1} d_2^{z_2} \cdots d_\ell^{z_\ell} = \prod_{i=1}^{\ell} g^{r_i z_i} = g^R, \end{aligned} \tag{8}$$

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 Kevin.

STEP 3: (Decrypting) Kevin uses his 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 Bob.

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

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

where τ is the threshold. To increase efficiency of our protocol, we only examine the range of 0 to $n\tau$ instead of examining all the way to $2n\tau$. Since we determine $\omega = 1$ if there is no value matching within the range, $h < n\tau$.

Note that, we use the sum of difference, h to form Eq. (9) instead of the average $\bar{S_n}$ in Eq. (3). In other words, Eq. (9) is equivalent to Eq. (3).

Note that we assume that a watermark message ω is embedded for I'. 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 (a message is not embedded).

4. Multi-Bit Embedding

We study k-bit information ω embedding technique with consideration of reducing the communication cost in a naive method, and a distinct generators method.

4.1 Naive Method

Assume that embedding k-bit message in naive method. Randomly choose sets of indexes for the pixels to be modified, $(A_1, B_1), \ldots, (A_k, B_k)$, and iterate the embedding process describe in Sect. 3.3 for k times. The embedded image is $I' = (z_1, \ldots, z_\ell)$ where

$$z_i = \begin{cases} x_i + \delta & \text{if } i \in A_1 \cup \cdots \cup A_k, \\ x_i - \delta & \text{if } i \in B_1 \cup \cdots \cup B_k, \\ x_i & \text{otherwise.} \end{cases} \tag{10}$$

We estimate the communication cost of the naive method for a particular case when $k = 3$ bit is embedded to the image of $\ell = 65536$. Two 1024-bit[†] ciphertexts are generated by the modified El Gamal encryption for each of ℓ pixels. The total amount of ciphertext e is $1024 \cdot 2\ell k = 50$ Mbyte. Generally, for k-bit embedding, the size of e increases k times, which results in increasing processing cost linear to k. Hence, the naive method is less efficient for both communication and computation cost.

4.2 Distinct Generators Method

Let $g_1, \ldots, g_k$ be k independent generators of order of q in Z_q. The k distinct generators, having a unique multiplicative group $\langle g_1 \rangle = \langle g_2 \rangle = \cdots = \langle g_k \rangle$, are assigned for plaintext $(A_1, B_1), \ldots, (A_k, B_k)$. In this method, we use ℓ plaintexts $m_1, \ldots, m_\ell$ assigned for ℓ pixels as

$$m_i = g_1^{\mu_{i,1}} g_2^{\mu_{i,2}} \cdots g_k^{\mu_{i,k}} \bmod p,$$

where

$$\mu_{i,j} = \begin{cases} +1 & \text{if } i \in A_j, \\ -1 & \text{if } i \in B_j, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

To encrypt the plaintext, we can apply any homomorphic public-key algorithm such as the modified El Gamal encryption, $E(m_i) = (c_i, d_i) = (m_i \cdot y^{r_i}, g^{r_i} \bmod p)$, or Paillier encryption, $E(m_i) = m_i r^N \bmod N^2$. Note that plaintext is not g^{m_i}.

Watermark extraction method depends on public-key algorithm as follows.

1. The modified El Gamal encryption. Since $D(e') = g^{m_i}$ is obtained by decryption, we need to identify m_i to given g^{m_i} as follows. For all possible combination $S_1 = 1, \ldots, 2n\delta, \ldots, S_k = 1, \ldots, 2n\delta$, we examine
$$D(e') \stackrel{?}{=} g_1^{S_1} \cdot g_2^{S_2} \cdots g_k^{S_k} \bmod p, \tag{12}$$
until we find out at least one $(S_1^*, \ldots, S_k^*)$ that satisfies Eq. (12). Note that the identification of $(S_1^*, \ldots, S_k^*)$ involves large complexity of $O(n^k)$. Hence, the method is limited within a small number of k and n. The estimation of the uniqueness of identification is studied in Sect. 4.3.
2. Paillier encryption. It is not necessary to identify m_i from the decrypted message. Instead, we find $(S_1^*, \ldots, S_k^*)$ that satisfies
$$D(e') \stackrel{?}{=} S_1 + \phi S_2 + \cdots + \phi^{k-1} S_k \bmod N \tag{13}$$
where ϕ is a constant such that $\phi > 2n\delta$ and $g_j = g^{\phi^j} \bmod N$. In extraction phase, each bit can be obtained by dividing $D(e')$ by ϕ.

We summarize the two methods in Table 2. The proposed method is efficient in communication cost since the size of ciphertext $|e|$ is independent from the number of bit, k. However, computational overhead in extracting exists.

Table 2 Comparison of methods.

	Naive Method	Distinct Generator Method
Image I'	$z_1, \ldots, z_\ell$	$z_1, \ldots, z_\ell$
The Size of e (Communication cost)	$1024 \cdot 2 \cdot \ell k$	$1024 \cdot 2 \cdot \ell$
Extraction	Iterate STEP2 and 3 for k times	Execute STEP 2 and 3 once and then identify the embedded information from all possible combination 2^k.

4.3 Uniqueness of Identification

There exists at least one tuple $(S_1^*, \ldots, S_k^*)$ such that $D(e') = g_1^{S_1^*} \cdots g_k^{S_k^*}$ because of the definition of e' in Eq. (8).

Given $D(e')$, however, $(S_1, \ldots, S_k)$ may not be always determined uniquely, i.e., there may be false tuple $(S_1', \ldots, S_k')$ such that $g_1^{S_1'} \cdots g_k^{S_k'} = g_1^{S_1^*} \cdots g_k^{S_k^*} = D(e')$, and $S_1' \neq S_1^*, \ldots, S_k' \neq S_k^*$. Hence, we estimate a risk of failure in identifying $(S_1, \ldots, S_k)$.

[†]We assume the size of ciphertext is 1024-bit, which is considered to be secure enough in many secure applications, e.g., digital signature, and SSL. We use the same size for Paillier encryption to simplify comparison of those two encryptions.

Lemma 1: Let $g_1, \ldots, g_k$ be independent generators of multiplicative group $\langle g \rangle$ of order q. Assume g^S is uniformly distributed over Z_p^*. Given $x \in \langle g \rangle$, a randomly selected tuple $(S_1, \ldots, S_k) \in Z_q^k$ satisfies $x \stackrel{?}{=} {g_1}^{S_1} \cdots {g_k}^{S_k}$ with probability of $1/q$.

(Proof) Since ${g_i}^{S_i}$ belongs to $\langle g \rangle$ for all $i = 1, \ldots, k$, ${g_1}^{S_1} \cdots {g_k}^{S_k}$ belongs to $\langle g \rangle$ as well. From the assumption of a uniform distribution, we have the lemma. (Q.E.D.)

Theorem 1: When an identification scheme examines every tuple $(S_1, \ldots, S_k)$, where $S_i = 0, \ldots, 2n\delta$, $i = 1, \ldots, k$, correct tuple $(S_1, \ldots, S_k)$ is always examined. The expected value of the number of false tuple $({S_1}', \ldots, {S_k}')$ in the scheme is $(2n\delta)^k/q$, which is negligible when $(2n\delta)^k \ll q$.

(Proof) Since there are $2n\delta$ possibilities for each S_i, the scheme examines total of $(2n\delta)^k$ Bernoulli trials with probability of success $1/q$ from Lemma 1. Hence, the number of success (i.e., false tuple,) is distributed accounting to a binomial distribution, whose expected value is given $(2n\delta)^k/q$. (Q.E.D.)

Theorem 1 states that honest Alice can always find correct tuple, and a probability that she finds distinct (false) tuple is negligibly small. Hence, even if Alice is malicious, it is almost infeasible to find the false tuple to cheat Bob. Any third party other than Alice can not perform the test for identifying valid tuple because the indexes of pixels where watermark is embedded are encrypted with a public key. Therefore, the scheme is secure against any malicious party.

In a practical environment, $|q|$ is of hash function, e.g., 2^{160} for SHA1. With the parameters $n = 1165 \approx 2^{10}$ and $\delta = 2$, we can embed $k = 3$-bit message for a portion such that $(2n\delta)^k = 2^{12k} = 2^{36} < 2^{160}$, with 2^{-124} false tuples expected.

4.4 Technique for Remaining Low Deterioration

In order to maintain quality of contents for embedding, we consider following technique. First one is avoiding duplication of selecting A_i and B_j between different bit layer to reduce false positive at extraction. Denote by i and j is the indexes of each bit layer. Second, for the case of duplicity of A_i and A_j or B_i and B_j, we apply avoidance of the δ modification since one modification affects for the both bit layers. For example, assuming pixels belong to $i \in A_1 \cup A_2$, we can accumulate modified pixels as $c_i = g_1 g_2 \cdot y^{r_i}$, that is, and we let $z_i = x_i + \delta$ which affects both g_1 and g_2. To see the effect, we illustrate SNR in Fig. 3 which multi-bit information is embedded.

5. Evaluation

5.1 Security

In this section, we show the security of patchwork watermark. The security relies on the following facts. 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. 5, 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 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.

Figure 4 illustrates an example of a single-bit information being embedded into Lena (256×256 pixels, 256 gray scale levels) with the parameters of $n = 2053$, and $\delta = 3$. The SNR for Fig. 4 is 50.6 [dB] which is considered to be acceptable.

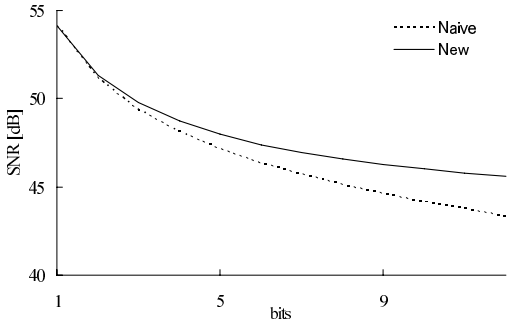


Fig. 3 SNR of the multi-bit embedded images.

Fig. 4 Embedded images.

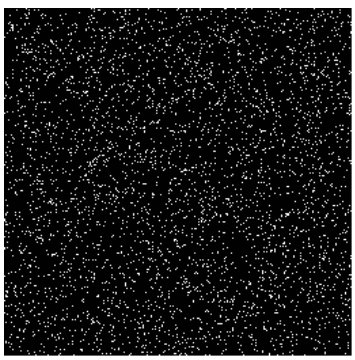


Fig. 5 Distribution of A and B.

Fig. 6 8-bit embedded.

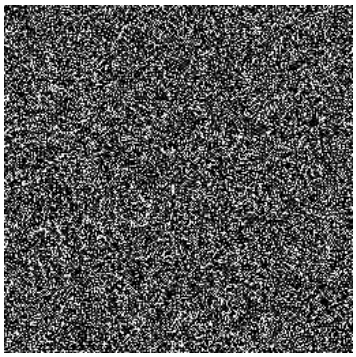


Fig. 7 Distribution of A, B for 8 bit embedded image.

For multi-bit embedding, the naive method is used as shown in Fig. 6, where 8-bit message is embedded with the same parameters. SNR for the image is 46.5767 [dB], and Fig. 7 shows distribution of A, B. Therefore, we can conclude that it is hard to retrieve the embedded message from given image I' as well as ordinary patchwork algorithm.

We discuss security of the modified El Gamal encryption and robustness against manipulation attacks. From given image I' and ciphertext e, Bob learns nothing about embedding key (A, B), under the assumption of difficulty of discrete logarithm problem. From given ciphertext (C, D) sent from Bob, Kevin knows neither the image I' nor (A, B), which has been accumulated into the ciphertext.

5.2 Optimal Parameter

In this section, we discuss an optimal parameter δ 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 3, and the optimal δ given n is obtained from Fig. 8. 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 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,

$$\begin{aligned} \mathrm{SNR} &= 10 \cdot \log_{10} \frac{255^2}{\mathrm{MSE}^2} \\ &= 10 \cdot \log_{10} \frac{255 \cdot 255}{1/\ell \sum (x_i - z_i)^2}, \end{aligned} \tag{14}$$

where MSE is the mean-square error between I and I'. Lena of 256×256 pixels is used for this test with the parameters in Table 3. Figure 10 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. Figure 9 illustrates how SNR of the image varies for the image size ℓ, where single-bit is embedded and $n = 2053$ pixels are manipulated.

Table 3 Parameters for δ determination.

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

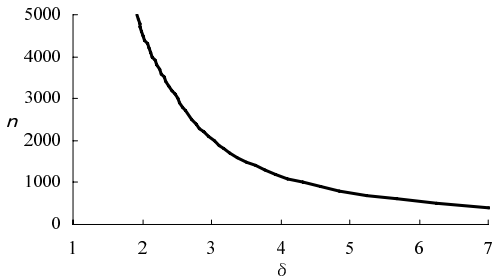


Fig. 8 Optimal δ distribution.

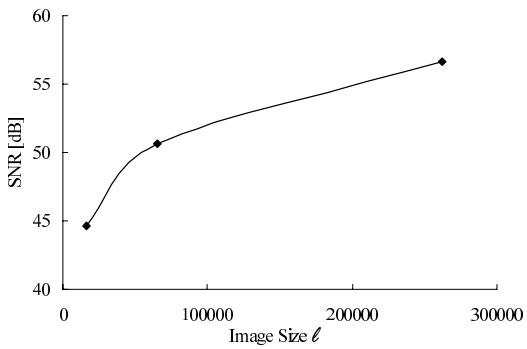


Fig. 9 SNR for different image size ℓ.

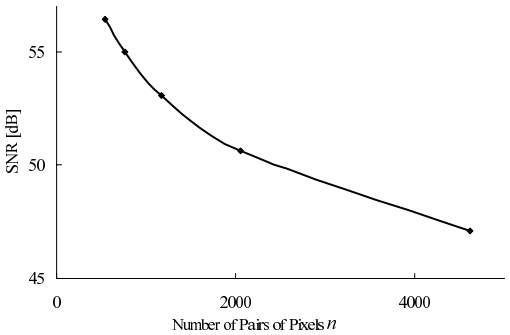


Fig. 10 The relation between the number of modified pairs of pixels n and SNR.

Table 4 Implementation environment.

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

5.3 Implementation System

In order to estimate a total performance of the proposed scheme, we implemented watermark embedding and extracting process for gray scale images in C. Cryptographic computations are implemented in Java. Environment specifications are described in Table 4.

5.4 Performance for Embedding

We use Lena as a host image I in three different sizes; $\ell = 64\times64$, 128×128, and 256×256 pixels to perform embedding, encrypting, decrypting, and extracting processes.

5.4.1 Watermark Embedding Scheme

Embedding processing time for image size ℓ is illustrated in Fig. 11, which is performed in C. Time consumption increases proportionally to image size ℓ. Processing cost in embedding k-bit message using the proposed method only increases k times of multiplications since all processes regarding to k are only involved in a plaintext m and μ which is either $+1$ or -1. Hence, costs for encryption and communication cost are independent from k. We estimate processing time for k-bit message embedding in Fig. 12, where the $\ell = 256\times256$ pixels image is used with $n = 2053$.

5.4.2 Ciphertext e Generation in the Modified El Gamal Encryption

A single 1024-bit the modified El Gamal encryption and decryption time are 0.104 [s], and 0.077 [s], respectively. Whereas, those of Paillier encryption are 3.303 [s], 2.127 [s].

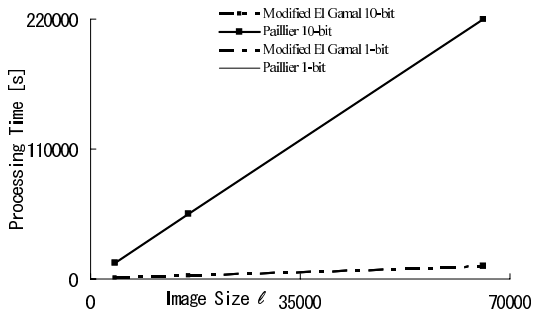


Fig. 11 Comparison of processing time for 1 bit and 10 bit embedding.

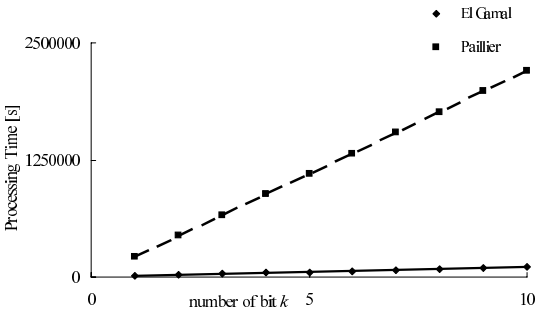


Fig. 12 Processing time for k-bit embedding.

Table 5 Processing time for generating ciphertext e.

Image Size ℓ	64×64	128×128	256×256
The modified El Gamal [s]	654.840	2620.57	10496.0
Paillier [s]	13530.3	55038.6	220155

Table 6 Processing time for watermark verification.

Image Size ℓ	64×64	128×128	256×256
The modified El Gamal [s]	5.68	22.07	88.52

The generation of e takes time in proportion to the number of pixels ℓ, shown in Table 5.

5.4.3 Processing Time for Watermark Verification

Watermark verification process, second step of proposed protocol is performed by Bob, and is supposed to be linear to the size of images. The samples of time consumption with respect to ℓ, 64×64, 128×128, and 256×256 are taken in Table 6.

5.4.4 Processing Time for Total Watermark Verification

Bob needs to send ciphertext to Kevin and requests him to perform decryption, which is independent from the size of an image. Total time required for the whole verification process including identification process with respect to the number of n pairs is shown in Table 7, and Fig. 13.

The reason that the watermark detection time takes so long is that it includes the process of identifying the value by testing all possible numbers as described in Eq. (9). The time consumed for iteration is liner to the number of pairs,

Table 7 Total verification processing time.

Image Size ℓ	64×64	128×128	256×256
Average Processing Time [s]	11.562	26.875	93.876
Standard Deviation	0.8277	0.8356	0.7750

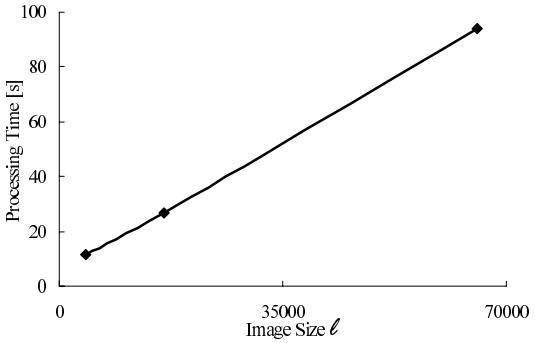


Fig. 13 Processing time for total watermark verification.

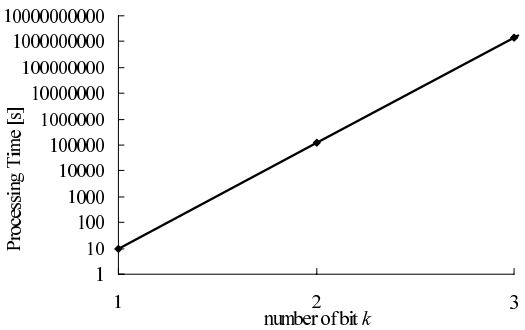


Fig. 14 Processing time for watermark verification for multi-bit.

n. While the decryption process in Sect. 3.3 (STEP 3), only takes 0.077 seconds.

To illustrate increment of computation cost at k-bit message extraction process, we show processing time in terms of k in Fig. 14. A processing cost for verification increases exponentially for k. From the experimental result in this testbed, we can estimate the processing time for k-bit as

$$E_k = T \times \frac{(2n\delta)^k}{2} = (2n\delta)^{k-1} E_1,$$

where T is a constant time for one exponentiation modulo (p).

5.5 Robustness against Noise Addition and JPEG Compression Attacks

We evaluate the robustness of patchwork watermarking against attack of "Add Noise", and "JPEG Compression" using StirMark [7], [8]. We have used I' originated from Lena (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, we applied extracting process with the parameter of $\tau = 3$ for all attacked images I'.

In JPEG compression attack, we confirmed verification successfully up to 80% of JPEG quality level as shown in Fig. 15. In Add Noise attack, we confirmed success as shown in Fig. 16, and Table 8. The noise level represents that of normalized from 0 to 100 such that 0 gives an identity function and 100 gives a complete random image. In the figure, we indicate the threshold level of $\tau = 3$ by which watermark extractions are confirmed.

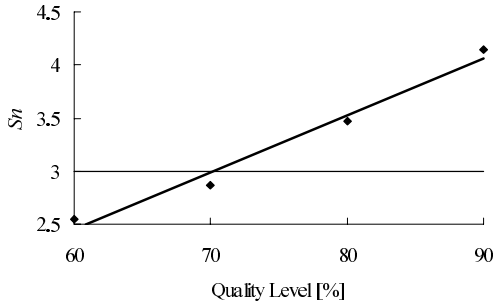


Fig. 15 JPEG compression attack.

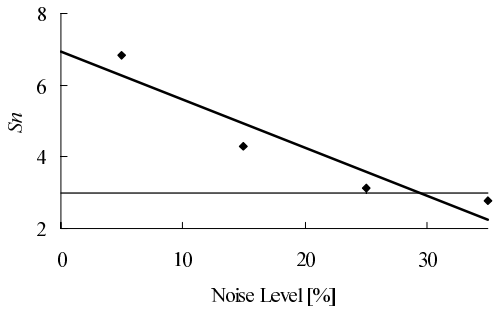


Fig. 16 Add noise attack.

Table 8 Add noise attack.

Noise Level[%]	5	15	25	35
S_n	6.8378	4.3064	3.1173	2.7681

5.6 Comparison between Furukawa's Method and the Proposed Scheme

Essential difference between Furukawa's scheme [2] and the proposal scheme comes from the cryptographical primitives, that is, the modified El Gamal and Paillier encryption. Figure 17 shows the processing time of extracting phase in the modified El Gamal and Paillier encryptions. We examine processing time for all cases in Table 3. Each of cases is provided average of ten samples of different seeds. The values used to plot in Fig. 17 are shown in Table 9.

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 we only needs to perform exact one decryption to extract watermark.

Supposing the processing time follows linearly to n as

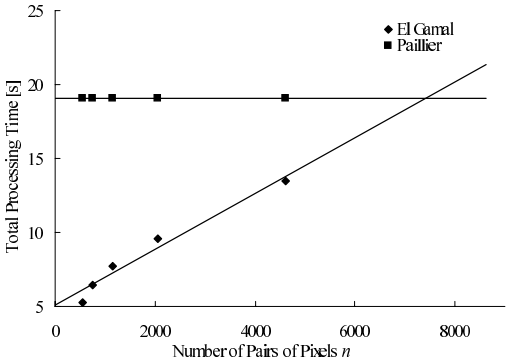


Fig. 17 Processing time of proposed scheme and that of [2].

Table 9 Processing time in watermark detecting.

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

illustrated in Fig. 17, Paillier processing time would crosses over that of the modified El Gamal at $n^* = 7403$. From the result, we can say that our scheme is superior to Furukawa's method [2] 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, such that Eq. (12) holds, which feasibility is stated in Sect. 5.6. 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 Sect. 5.2 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 $n^* = 7403$. Therefore, we claim that there are many cases that are appropriate to El Gamal encryption.

We consider the processing time for multi-bit embedding based on single-bit testbet result. For the case of the modified El Gamal encryption, extraction processing time v increases up to v^k for k bit. On the other hand, the extraction processing cost for Paillier encryption increases for the time of seeking which identify the result from the combination of k^2.

5.7 Improvement with Advanced Patchwork Watermark [9], [10], and [11]

We consider about an improvement of performance when applying the advanced patchwork watermark embedding algorithms. The improvement of the modified El Gamal with [11] is shown in Table 10, where three combinations of encryptions and watermark algorithms [3], and [11] are estimated in terms of performance and robustness.

Table 10 Estimation of improvement with [11].

Asymmetric Watermark	n	Verification Time	JPEG Robustness
Paillier with [3]	1000	19.11	70%
El Gamal with [3]	1000	6.9	70%
El Gamal with [11]	30	5.1	30%

According to [11], it is able to reduce n from approximately 1000 to 30 and improve robustness against JPEG compression attack. As for the decrement of n, we estimate the extraction time of the combination of [11] with the modified El Gamal to be approximately 5.1 seconds based on the experimental data in Fig. 17. This result shows that [11] with the modified El Gamal is about four times faster than [3] with Paillier when n is 1000.

As for JPEG robustness, we compare the three combinations in Table 10 that satisfies about 95% of detection accuracy. The detection accuracy of Paillier with [3] and El Gamal with [3] is shown in Fig. 15 and [11] with El Gamal can be referred in Table 1 in [11] where 95.8% of DR, and 0.5% of BER is shown in the location-shift method. The comparison indicates the fact that [3] with El Gamal only endures for approximately 70% of JPEG compression to detect a massage with 95% accuracy, while [11] endures up to 30% of JPEG compression with about 95% of detection accuracy. Note that, the difference between frequency and spatial domain still need to be discussed.

6. Conclusions

An asymmetric digital watermarking protocol provides secure detection by hybridizing statistical watermarking and homomorphic public-key encryption. The protocol enables reducing the risk, exposure of secret information in an extraction process by encrypting the secret information. Following our contributions are based on the asymmetric watermark protocol applying patchwork watermark and the modified El Gamal.

(1) We have shown the feasibility of proposed asymmetric digital watermarking scheme.

(2) Our experiment shows that the modified El Gamal encryption is superior to Paillier encryption under the certain condition, that is, our protocol is more efficient than [2] for the case when the number of the modified pairs of pixels is $n < 7403$. The experimental results also show that detection processes take time proportional by the size of images. For instance, the size of 256×256 takes approximately 93 seconds for detection.

(3) We have proposed a new algorithm for multi-bit embedding by providing independent generators. This method achieves an efficient embedding by the constant size of ciphertext.

We will proceed to enhance reliability in watermark detection. This problem can be solved by applying an error correction coding technology. We adopt our method to high resolution images, 32-bit levels to contribute digital contents security.

References

[1] M. Okada and H. Kikuchi, "Asymmetric digital watermark detection without secret of modified pixels," Workshop of Information Security and Application (WISA 2005), LNCS, vol.3786, pp.165–178, 2005.

[2] J. Furukawa, "Secure detection of watermarks," IEICE Trans. Fundamentals, vol.E87-A, no.1, pp.212–220, Jan. 2004.

[3] W. Bender, D. Gruhl, and N. Morimoto, "Technique for data hiding," SPIE, vol.2020, pp.2420–2440, 1995.

[4] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," EUROCRYPT '99, LNCS, vol.1525, pp.223–238, 1999.

[5] K. Minematsu, "On a secure digital watermark detection protocol using patchwork watermarking," International Symposium on Information Theory and its Applications (ISITA), no.12A, pp.673–676, 2000.

[6] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," EUROCRYPT '98, LNCS, vol.1403, pp.308–318, 1998.

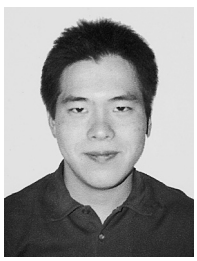
[7] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," Information Hiding, Second International Workshop (IH '98), LNCS, vol.1525, pp.219–239, 1998.

[8] F.A.P. Petitcolas, "Watermarking schemes evaluation," IEEE Signal Process., vol.17, no.5, pp.58–64, 2000.

[9] M. Arnold, "Audio watermarking: Features, applications, and algorithms," IEEE International Conference on Multimedia and Expo, pp.1013–1016, 2000.

[10] I.-K. Yeo and H.J. Kim,"Modified patchwork algorithm: A novel audio watermarking scheme," IEEE Trans. Speech Audio Process., vol.11, no.4, pp.381–386, 2003.

[11] I.-K. Yeo and H.J. Kim, "Generalized patchwork algorithm for image watermarking," ACM Multimedia Systems, vol.9, pp.261–265, 2003.



Mitsuo Okada received B.S. from Eastern Oregon University, U.S.A in 2002, and M.S. from Tokai University, Japan in 2005. He is currently a Ph.D. candidate at the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Japan. His research interests are in contents security and any inovative technology. He is a student member of the Information Processing Society of Japan (IPSJ).

Hiroaki Kikuchi received B.E., M.E. and PhD. degrees from Meiji University, Japan in 1988, 1990 and 1994. After he worked in Fujitsu Laboratories Ltd. from 1990 through 1993, he joined Tokai University, Japan in 1994. He is currently a Professor in Department of Information Media Technology, School of Information Science and Technology, Tokai University. He was a visiting researcher of computer science, in Carnegie Mellon University in 1997. His main research interests are fuzzy logic, cryptographical protocol, and network security. He is a member of IPSJ, the Japan Society for Fuzzy Theory and Systems (SOFT), IEEE and ACM.

Yasuo Okabe received the M.E. from Department of Information Science, Kyoto University in 1988. From 1988 he was an Instructor of Faculty of Engineering, from 1994 he was an Associate Professor of Data Processing Center, and from 1998 he was an Associate Professor of Graduate School of Informatics, Kyoto University. He is now a Professor of Academic Center for Computing and Media Studies, Kyoto University. Ph.D in Engineering. His current research interest includes Internet architecture, ubiquitous networking and distributed algorithms. He is a member of IPSJ, the Institute of Systems, Control and Information Engineers (ISCIE), Japan Society for Software Science and Technology (JSSST), IEEE, ACM and European Association for Theoretical Computer Science (EATCS).