An Efficient Buyer-Seller Watermarking Protocol Based on Composite Signal Representation

Mina Deng
IBBT-COSIC, K.U.Leuven
Kasteelpark Arenberg 10
B-3001 Leuven-Heverlee, Belgium
mina.deng@esat.kuleuven.be

Tiziano Bianchi
Dept. Electronics and Telecommunications
University of Florence
Via S. Marta 3, 50139 Firenze, Italy
tiziano.bianchi@unifi.it

Alessandro Piva
Dept. Electronics and Telecommunications
University of Florence
Via S. Marta 3, 50139 Firenze, Italy
alessandro.piva@unifi.it

Bart Preneel
IBBT-COSIC, K.U.Leuven
Kasteelpark Arenberg 10
B-3001 Leuven-Heverlee, Belgium
bart.preneel@esat.kuleuven.be

ABSTRACT

Buyer-seller watermarking protocols integrate watermarking techniques with cryptography, for copyright protection, piracy tracing, and privacy protection. In this paper, we propose an efficient buyer-seller watermarking protocol based on homomorphic public-key cryptosystem and composite signal representation in the encrypted domain. A recently proposed composite signal representation allows us to reduce both the computational overhead and the large communication bandwidth which are due to the use of homomorphic public-key encryption schemes. Both complexity analysis and simulation results confirm the efficiency of the proposed solution, suggesting that this technique can be successfully used in practical applications.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce; E.2 [**Data Storage Representations**]: Object representation

General Terms

Security, Performance

1. INTRODUCTION

Today's rapid development of multimedia technology resulted in a number of security issues including copyright protection, traitor tracing, authentication and identification. At the same time, more attention has been paid to privacy protection for users in emerging multimedia applications. In order to meet these needs, digital watermarking and fingerprinting protocol has experienced a surge in research activities over the last decade, and a variety of elegant watermarking protocols have been proposed [24, 23, 3], allowing the content provider to embed seller's information in a distributed content to preserve copyright, or buyer's information to identify copyright violators. Traditional watermarking schemes assume that content providers are trustworthy such that they would never distribute content illegally and always perform the watermark embedding honestly. However, in practice, such assumptions are not fully established. As a consequence, the watermark tracing mechanism is discredited, because a malicious seller may benefit from framing an innocent buyer or a guilty buyer may repudiate the fact of copyright infringements by invoking the possibility of framing by the seller. It is against this background that buyer-seller watermarking protocols were introduced, as a cross-disciplinary application, combining cryptography with watermarking to ensure copyright protection, security and privacy for both the content provider and the customer simultaneously. The cryptographic and watermarking requirements that a secure buyer-seller watermarking protocol is expected to fulfil are outlined in Section 2.

In the literature, the first known buyer-seller watermarking protocol was introduced by Memon and Wong [19] using homomorphic cryptosystems to embed watermark in the encrypted domain. In a typical setting, the content provider and the customer perform a protocol and both generate only part of the watermark, and this ensures the watermarked content delivered to the buyer is unknown by the seller, the unwatermarked original content is unavailable to the buyer, and none of them have access to the embedded watermark. Some of the successors were proposed as an extension and variation to [19] including [18, 12, 16].

However, a common problem of the aforementioned approaches is that they do not focus on the actual embedding of the watermark in a specific multimedia content. This is a classical scenario where cryptographic techniques should be applied together with signal processing techniques. In such a scenario, the availability of signal processing modules that work directly on encrypted data would be of great help to satisfy the security requirements.

Signal processing in the encrypted domain (s.p.e.d.) is a new field of research aiming at developing a set of specific tools for processing encrypted data to be used as building blocks in a large class of applications [14]. As to buyer-seller watermarking protocols, the literature offers few examples of s.p.e.d. oriented approaches. In [17], a basic amplitude quantization-based scheme based on an additively homomorphic cryptosystem has been proposed for embedding the watermark in the encrypted domain, which has been adapted to more robust watermarking techniques in [26]. However, such techniques require processing each content feature as a sep-

MM&Sec'09, September 7–8, 2009, Princeton, New Jersey, USA.

arate encryption, which leads to a high computational complexity since it introduces a huge expansion factor between the original signal sample and the encrypted one. To the best of our knowledge, there is no solution in the literature addressing both the security issues stemming from the protocol and the efficiency issues related to the actual embedding of the watermark in the encrypted domain.

As an extension of the previous work [2, 12], we have proposed a secure buyer-seller watermarking protocol based on homomorphic public-key encryption with an efficient watermark embedding method in the encrypted domain using the composite signal representation. Addressing the security and efficiency issues, our contribution of this paper is twofold:

Avoid double watermark insertion. Double watermark insertions, required by the most predecessors, may cause a degradation of the final quality of the distributed contents. When applied independently, the second watermark could confuse or discredit the authority of the first watermark, thus acting as an actual "ambiguity attack" [10]. That is avoided by designing a unique watermark, composed of the buyer's secret watermark, the seller's secret watermark, and a transaction index.

Efficient watermark embedding. The existing s.p.e.d. watermark embedding schemes are reviewed under a unifying framework and combined with a composite signal representation [2] that permits to represent several features of the content in a single encryption. Several composite embedding strategies are proposed, which demonstrate the practical feasibility of the protocol.

2. REQUIREMENTS OF SECURE BUYER-SELLER WATERMARKING PROTOCOL

2.1 Cryptographic Requirements

In the following are the main requirements of a secure buyer-seller protocol as recognized in the cryptographic literature [3, 12].

Correctness: All protocols should terminate successfully whenever its players are honest (no matter how other players behaved in other protocols).

Traceability: A copyright violator should be able to be traced and identified.

Non-framing: Nobody can accuse an honest buyer.

Non-repudiation: A guilty buyer cannot deny his responsibility for a copyright violation caused by him.

Dispute resolution: The copyright violator should be identified and adjudicated without him revealing his private information, e.g. private keys or watermark.

Anonymity: A buyer's identity is undisclosed until he is judged to be guilty.

Unlinkability: Nobody can determine whether the different watermarked contents are purchased by the same buyer or not.

2.2 Signal Processing Requirements

Here we try to identify the signal processing requirements of a buyer-seller protocol in order to provide a realistic watermarking system. Some of these refer to common watermarking requirements [20]; others are specific to s.p.e.d. applications.

Robustness: The watermark should be correctly decoded after common signal processing operations such as compression, filtering, noise addition, desynchronization, cropping, insertions, mosaicing, and collage.

Security: An adversary should not be able to obtain any secret information about watermark embedding, such as permutations of coefficients, quantization dithering, etc. With such information an adversary could reverse the watermarking algorithm, which is assumed public, and completely remove the watermark.

Collusion resistance: A limited number of adversaries should not be able to remove the watermark by comparing or composing their differently watermarked copies.

Perceptual quality (Transparency, Fidelity): Watermark embedding should not cause perceptual degradation of the host signal, according to the type of digital medium.

S.p.e.d. compatibility: Watermark embedding should be tailored to the particular encrypted domain representation of the content.

S.p.e.d. complexity: The complexity of signal processing in the encrypted domain should be sustainable.

3. PRIMITIVES

3.1 Cryptographic primitives

3.1.1 Privacy Homomorphism

An encryption scheme is said to be *homomorphic* if the encryption function E satisfies

$$\forall m_1, m_2 \in \mathcal{M} : E(m_1 \odot_{\mathcal{M}} m_2) = E(m_1) \odot_{\mathcal{C}} E(m_2)$$

for some operators $\odot_{\mathcal{M}}$ in the plain domain $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in the encrypted domain $\mathcal{C}$.

Homomorphic cryptosystems can be classified as two groups, namely the ones whose security relies on the *"decisional composite residuosity assumption" (DCRA)*, and the ones of the ElGamal class based on *"decisional Diffie-Hellman assumption" (DDH)*. The strongest security level a privacy homomorphism can reach is *IND-CPA*, instead of *IND-CCA2*. For instance, the *ElGamal* cryptosystem [13] are multiplicative privacy homomorphism. The *Paillier* cryptosystem [21], and *Paillier*'s generalization by Damgård-Jurik [11] are additive privacy homomorphism.

3.1.2 Group Signature

Group signatures [4], enable group members, each with his/her own private signature key to produce signatures on behalf of the group. Group signature groups can either be static or dynamic, and dynamic groups allow to update group members with time. The security properties of static and dynamic group signature schemes are formalized in [1] as follows:

Anonymity allows group members to create signatures anonymously, such that it is hard for an adversary, not in possession of the group manager's opening key to recover the identity of the signer.

Traceability permits the signer's anonymity to be revoked by the group manager in case of misuse, and ensures that no colluded group members can create unverifiable signatures, or signatures that can't be traced back to some member of the coalition.

Non-frameability requires that no adversary can produce a signature that an honest opener would attribute to a user unless the latter indeed produced it.

3.2 Watermarking primitives

3.2.1 Dither Modulation

Dither modulation techniques belong to the class of data hiding schemes defined informed embedding or host-interference rejecting methods [8], where the watermarking problem is viewed as one of communications with side information at the encoder. Within this class of methods, Quantization Index Modulation (QIM) [7] and Rational Dither Modulation (RDM) [22] are widely employed due to their good performance. Such methods hide signal-dependent watermarks using as embedding rule the quantization of some content features. In our scheme, the extension of such technique to watermark embedding in the encrypted domain is considered [17, 26].

The simplest example of such techniques is a binary Dither Modulation (DM) with uniform scalar quantizers: in this realization, we assume that $\mathbf{w}$ is a binary vector, and that each bit of $\mathbf{w}$, say w_i, determines which quantizer, chosen between two uniform scalar quantizers, is used to quantize a single scalar host feature x_i. Two codebooks $\mathcal{U}_0$ and $\mathcal{U}_1$ associated respectively to a bit value $w=0$ and $w=1$ are built as:

$$\begin{aligned} \mathcal{U}_{\delta,0}^{\Delta} &= \{u_{0,k}\} = \{k\Delta+\delta, k\in\mathbb{Z}\}, \\ \mathcal{U}_{\delta,1}^{\Delta} &= \{u_{1,k}\} = \{k\Delta+\Delta/2+\delta, k\in\mathbb{Z}\}, \end{aligned} \tag{1}$$

where Δ is the quantization step and δ is the dithering value.

Watermark is embedded by applying to the feature x either the quantizer Q_0 associated to $\mathcal{U}_0$, or the quantizer Q_1 associated to $\mathcal{U}_1$, depending on the to-be-hidden bit value $w=\{0,1\}$:

$$Q_{\delta,w}^{\Delta}(x) = \arg\min_{u_{w,k}\in\mathcal{U}_{\delta,w}^{\Delta}} |u_{w,k}-x| \tag{2}$$

where $u_{w,k}$ are the elements of $\mathcal{U}_{\delta,w}^{\Delta}$. By letting y indicate the marked feature, we have $y = Q_{\delta,w}^{\Delta}(x)$.

3.2.2 Composite Signal Representation

Composite representation of signals [2] permits to group several signal samples into a single word and to perform basic linear operations on them. This representation has been proposed to solve the problems related to the data expansion from the plaintext to the encrypted representation of signals, due to the use of cryptosystems operating on very large algebraic structures. Composite signal representation allows to speed up linear operations on encrypted signals via parallel processing and to reduce the size of the whole encrypted signal. In our scheme, composite signal representation is used to reduce the size of the digital content (image) before watermark embedding in the encrypted domain.

Let us consider an integer valued signal $a_n\in\mathbb{Z}$, satisfying $|a_n|\leq Q$, where Q is a positive integer. Given a pair of positive integers β, R, the *composite* representation of a_n of order R and base β is

$$a_{C,k} = \sum_{i=0}^{R-1} a_{i,k}\beta^i, \;\; k=0,1,\ldots,M-1 \tag{3}$$

where $a_{i,k}$, $i=0,1,\ldots,R-1$ indicate R disjoint subsequences of the signal a_n.

If $\beta > 2Q$ and $\beta^R \leq N$, it can be shown [2] that the composite representation $a_{C,k}$ takes no more than N distinct values. Thanks to this property, $a_{C,k}$ can be represented over $\mathbb{Z}_N$ without losing information. Moreover, as long as the aforementioned hypotheses hold, several kinds of linear processing can be applied directly to the composite representation of the signal, allowing for a parallel processing of the original signal samples.

4. PROPOSED PROTOCOL

The proposed buyer-seller watermarking protocol involves four players: the seller $\mathcal{A}$, the buyer $\mathcal{B}$, the trustworthy *CA*, and an arbitrator $\mathcal{J}$. In this section, we elaborate on the three subprotocols. First, in the registration protocol, $\mathcal{B}$ registers at the *CA* before the purchase. Second, in the watermark generation and insertion protocol, $\mathcal{B}$ purchases a digital content from a media distributer $\mathcal{A}$. Third, in the identification and arbitration protocol, enables $\mathcal{A}$ to identify the copyright violator, with the collaboration of the $\mathcal{J}$ and the *CA*. We assume the *CA* is trustworthy and a secure *Public Key Infrastructure* is well deployed such that each party has a certified public and private key pair. For consistency, we assume the digital content is a still image, although the protocol can be applied to other multimedia format. As an illustration, we follow the formal definition of dynamic group signatures of Bellare et al. [1].

4.1 Registration Protocol

The registration protocol performed between the buyer $\mathcal{B}$ and the *CA* is depicted in Fig.1.

1. The *CA* executes the *group-key generation* algorithm `GKg` to produce the group public key gpk, the issuer key ik, and the opener key ok.
2. $\mathcal{B}$ begins with the *user-key generation* algorithm `UKg` to obtain a public and private key pair (upk_B, usk_B).
3. To join the group, $\mathcal{B}$ generates a key pair (sk_B, pk_B), signs pk_B with usk_B, and sends (pk_B, sig_B) to the issuer. If sig_B is verified, the issuer issues a certificate of pk_B and $\mathcal{B}$'s identity B. Then (pk_B, sig_B) are stored in a registration table as $reg[B]$.
4. Upon receiving $cert_B$, $\mathcal{B}$ generates his private group signature key gsk_B from the tuple $(B, pk_B, sk_B, cert_B)$, where B denotes $\mathcal{B}$'s identity.

4.2 Watermark Generation and Embedding Protocol

The protocol can be executed multiple times for multiple transactions between the seller $\mathcal{A}$ and the buyer $\mathcal{B}$, as depicted in Fig.2. $\mathcal{A}$ and $\mathcal{B}$ first need to negotiate a purchase agreement *ARG* on rights and obligations as well as the specification of the digital content X.

1. $\mathcal{B}$ first generates a one-time anonymous key pair (pk_B^*, sk_B^*). Then $\mathcal{B}$ applies the *group signing* algorithm `GSig` to create a signature μ to pk_B^* with his group signature key gsk_B and the group public key gpk, as $\mu = \texttt{GSig}(gpk, gsk_B, pk_B^*)$.
2. Next, $\mathcal{B}$ computes an key escrow cipher $E_{esc} = E_{pk_{CA}}(sk_B^*)$ to recover sk_B^* from the *CA* in case of disputes. Then $\mathcal{B}$ (as the prover) and $\mathcal{A}$ (as the verifier) engage in a zero knowledge proof ZKP_1, in order to assure $\mathcal{A}$ that the ciphertext E_{esc} is valid without compromising the encrypted message, which is $\mathcal{B}$'s private key sk_B^*.
3. $\mathcal{B}$ generates the buyer's secret watermark as a n-bit number $W_B = \{w_{B_1}..w_{B_n}\}$ where $w_{B_i}\in\{0,1\}$, in compliance with the features of X for robustness, and encrypts W_B bit-by-bit with his public key pk_B^* as $ew_{B_i} = E_{pk_B^*}(w_{B_i})$. The encrypted watermark is presented as $ew_B = \{ew_{B_1}..ew_{B_n}\}$. After this, for the correctness of the embedding and the successive detection a zero-knowledge proof ZKP_2 has to be performed, such that the buyer proves to the seller that the given ciphertext ew_{B_i} can be decrypted to a bit (i.e., the plaintext is either 1 or 0), without revealing any secret information. An alternative strategy could consider to neglect this step, confiding either in the ability of the watermark detector to reveal such fingerprint artifacts or in the fact that values different from $(0,1)$ will significantly degrade the content during the embedding process.
4. $\mathcal{B}$ sends $(pk_B^*, \mu, ARG, ew_B, E_{esc})$ as m with his signature $s = \texttt{Sig}_{sk_B^*}(m)$ to $\mathcal{A}$.
5. After $\mathcal{A}$ performed the *group signature verification* algorithm `GVf` to verify $\mathcal{B}$'s group signature μ with gpk and $\mathcal{B}$'s signature s with pk_B^*. $\mathcal{A}$ then generates the seller's secret watermark W_A and an index ϕ to locate the current transaction record in $Table_A$. Let $W_{AB} = W_A \oplus W_B$, $W = W_{AB} + \phi 2^n$. W consists of the n-bit W_{AB} and the ℓ-bit ϕ. W can be decomposed into $\ell+n$ binary numbers, with $w_i\in\{0,1\}$, satisfying $W = \sum_{i=0}^{n+\ell-1} w_i 2^i$. The watermark embedding can be considered as a function which takes the encrypted watermark bits $\mathcal{E}(w_i)$ and the content X as input, and returns the encrypted watermarked content $\mathcal{E}(Y)$ as output, where $\mathcal{E}(\cdot)$ denotes $E_{pk_B^*}(\cdot)$. The encrypted watermark can be computed in the encrypted domain as

$$\mathcal{E}(W) = \{\mathcal{E}(\phi_1),..,\mathcal{E}(\phi_l)\}||\{\mathcal{E}(w_{AB_1}),..,\mathcal{E}(w_{AB_n})\} \tag{4}$$

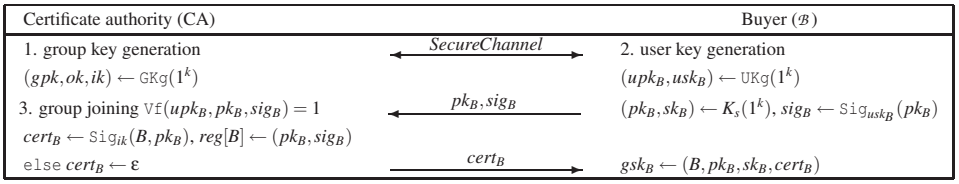


Figure 1: The registration protocol performed between the buyer $\mathcal{B}$ and the certificate authority *CA*.

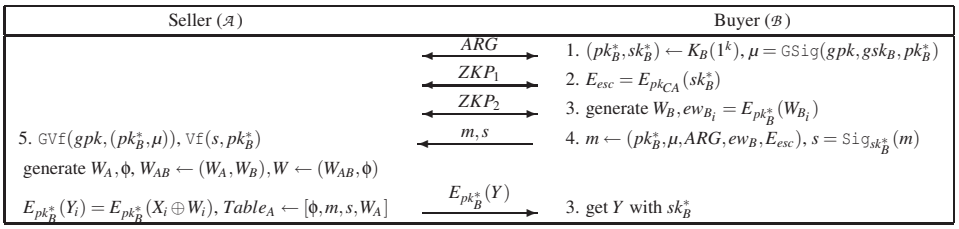


Figure 2: The watermark generation and embedding protocol performed between the seller $\mathcal{A}$ and the buyer $\mathcal{B}$.

where

$$\mathcal{E}(w_{AB_i}) = \mathcal{E}(w_{A_i} \oplus w_{B_i}) = \begin{cases} \mathcal{E}(w_{B_i}) & w_{A_i} = 0 \\ \mathcal{E}(1) \cdot \mathcal{E}(w_{B_i})^{-1} & w_{A_i} = 1 \end{cases} \quad (5)$$

Note that $||$ denotes concatenation, and $\oplus$ denotes watermark embedding operation.

6. $\mathcal{A}$ stores (ϕ, m, s, W_A) in $Table_A$, and delivers the encrypted watermarked content $E_{pk_B^*}(Y)$ to $\mathcal{B}$. As a result, $\mathcal{B}$ obtains the watermarked content Y with a decryption $D_{sk_B^*}(E_{pk_B^*}(Y))$.

4.3 Identification and Arbitration Protocol

The identification and arbitration protocol, performed among the seller $\mathcal{A}$, the judge $\mathcal{J}$, and the *CA*, is depicted in Fig. 3.

1. Once a pirated copy Y' of X is found, $\mathcal{A}$ extracts the watermark U from Y' and retrieves the most significant ℓ bits of U as an index ϕ' to search in $Table_A$, by choosing the ϕ from $Table_A$ most correlated with ϕ'. $\mathcal{A}$ provides the collected information to $\mathcal{J}$.

2. $\mathcal{J}$ verifies the buyer's signature s with the provided key pk_B^*. If verified, $\mathcal{J}$ sends the key escrow cipher E_{esc} to the *CA*. Otherwise, the protocol halts.

3. The *CA* decrypts E_{esc} to recover the suspected buyer's private key $sk_B^* = D_{sk_{CA}}(E_{esc})$, and sends encryption $E_{pk_J}(sk_B^*)$ back to $\mathcal{J}$.

4. $\mathcal{J}$ recovers $sk_B^* = D_{sk_J}(E_{pk_J}(sk_B^*))$, $W_B = D_{sk_B^*}(ew_B)$, and calculates W_{AB} from W_A and W_B. $\mathcal{J}$ then extracts the watermark U' from Y and retrieve the n least significant bits of U' as W'_{AB}. If W'_{AB} and W_{AB} match with a high correlation, the suspected buyer is proven to be guilty. Otherwise, the buyer is innocent. Note that until now, the buyer has stayed anonymous.

5. $\mathcal{J}$ sends a court order to the *CA*, which executes the *group signature open* algorithm `Open` with its opener key *ok* and the registration table *reg* to retrieve the identity B with a claim proof τ.

6. $\mathcal{J}$ verifies B and τ with the *group signature judging* algorithm `Judge`. If verified, $\mathcal{J}$ closes the case and announces that the buyer $\mathcal{B}$ with identity B is guilty. Otherwise, the protocol halts.

4.4 Zero Knowledge Proofs

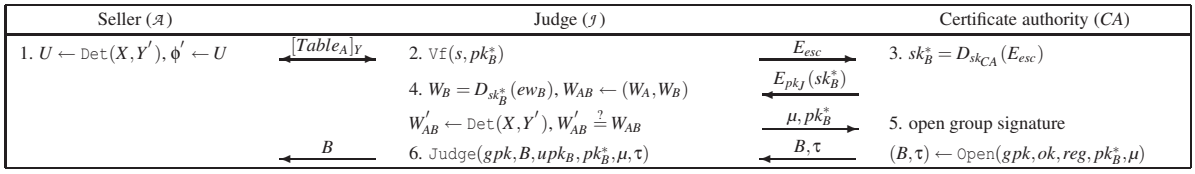
The additive homomorphic cryptosystem used to encrypt the buyer's and the seller's watermark is Paillier's cryptosystem [21], and the encryption is $\mathcal{E} : \mathbb{Z}_n \to \mathbb{Z}_{n^2}^*$. Randomly Choose two large prime numbers p and q, independent of each other. Compute $n = pq$ and $\lambda = lcm(p-1, q-1)$; select a random integer g where $g \in \mathbb{Z}_{n^2}^*$, and ensure that n divides the order of g by checking the existence of the modular multiplicative inverse: $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$, where function L is defined as $L(u) = \frac{u-1}{n}$. The public key is (n, g), the private key is λ. For encryption, let m be the plaintext message where $m \in \mathbb{Z}_n$, select random r where $r \in \mathbb{Z}_n^*$, and compute ciphertext as $c = g^m \cdot r^n \mod n^2$. For decryption, the ciphertext $c \in \mathbb{Z}_{n^2}^*$, and compute the plaintext as $m = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n$.

4.4.1 *Zero knowledge proof for fair encryption of private keys ZKP_1*

In our protocol, the buyer (as the prover $\mathcal{P}$) needs to convince the seller (as the verifier $\mathcal{V}$) that given the ciphertext $E_{esc} = E_{pk_{CA}}(sk_B^*)$ is an encryption of some value related to his private key, e.g., the factorization of the modulus n, without revealing any secret information; and the trusted third party *CA* is able to recover the buyer's private key, with the encryption E_{esc} and *CA*'s private key. Indeed, the buyer's Paillier public key is $n = pq$ and g, and his Paillier private key is $\lambda = lcm(p-1, q-1)$ which is equivalent to the factorization of the modulo n. The statistical zero knowledge proof ZKP_1 contains two building blocks as follows:

ZKP_A: Prove the correctness of the public key setup Due to the fact that the key pair (pk_B^*, sk_B^*) is self-generated by the buyer, it is essential to first prove the public key is correctly setup and n is the product of two large primes. That is to prove that the committed value is related to the private key, and the quantity committed to is the factorization of an RSA modulus. We follow the statistical zero-knowledge protocol by Camenisch et al. [5], proving that a modulus n is the product of two safe primes, i.e., primes p and q such that $(p-1)/2$ and $(q-1)/2$ are primes as well.

ZKP_B: Prove the correctness of the private key encryption Two candidate schemes seems to fit our setting, namely the verifiable encryption by Camenisch et al. [6] and the fair encryption of RSA keys by Poupard et al. [25]. Despite the claim of [6] that [25] may overlook the fact that the underlying encryption scheme provides security against chosen ciphertext attacks, we decide to employ Poupard's scheme due to its efficiency of zero knowledge proofs. Please refer to [25] for the encryption scheme of the buyer's private key and the proof of its correctness.



Note: $\mu = \mathtt{GSig}(gpk, gsk_B, pk_B^*)$

Figure 3: The copyright violator identification and arbitration protocol performed among $\mathcal{A}$, $\mathcal{J}$, and the *CA*.

4.4.2 *Zero knowledge proof for bit encryption ZKP_2*

The following round should be repeated m times, where m is the bit length of the buyer's watermark. The buyer (as $\mathcal{P}$) needs to prove to the seller (as $\mathcal{V}$) that a given ciphertext C is an encryption of a bit, but the seller doesn't know which one is encrypted exactly. In other words, the buyer needs to prove that the given encryption $E(w_i)$ is either $E(1)$ or $E(0)$, namely $ZKP\{w_i : E(w_i) \bigwedge (w_i \in \{0,1\})\}$. Our proof protocol is based on the zero knowledge proof by Damgård and Jurik [11]. As explained above, Paillier encryption is $E(i) = g^i \cdot r^n \mod n^2$, and it can be seen a specialized form of the Damgård-Jurik cryptosystem. Given ciphertext c and two candidate plaintexts $w_1 = 1$ and $w_2 = 0$, $\mathcal{P}$ and $\mathcal{V}$ both compute $u_1 = cg^{-w_1} \mod n^2$ and $u_2 = cg^{-w_2} \mod n^2$. It is easy to see that the proof is equivalent to convincing $\mathcal{V}$ that either u_1 or u_2 is a n-th residue modulo n^2. We assume that $\mathcal{P}$ knows an n-th root u_1, and $\mathcal{M}$ is the honest-verifier simulator for the n-th residue modulo n^2 protocol. The honest-verifier zero knowledge proof consists of two building blocks, namely to prove a value is n-th residue modulo n^2 and a value is 1-out-of-2 n-th residue modulo n^2, elaborated in [11]. To construct four-round perfect zero-knowledge proofs of knowledge based on honest-verifier zero knowledge proofs, we refer to the framework introduced by Cramer, Damgård, and MacKenzie [9].

5. SECURE WATERMARK EMBEDDING

The buyer-seller protocol described in the previous section requires that a vector of encrypted bits to be embedded in a digital media through a suitable watermarking scheme. We will name a watermarking scheme with such capabilities a secure watermark embedding scheme. Secure watermark embedding schemes based on dither modulation techniques and homomorphic cryptosystems have been proposed in [17, 26]. In the following, the aforementioned techniques are reviewed under a unifying framework and combined with the composite signal representation in order to provide an efficient implementation.

Let us assume that a vector of host features $\mathbf{x}$ has been extracted from the original content and denote a generic feature as x_i. The corresponding watermarked features using a scalar binary dither modulation can be expressed as

$$y_i = f(x_i, \mathbf{x}) + w_i \cdot \Delta(x_i, \mathbf{x}) \tag{6}$$

where $f(x_i, \mathbf{x})$ and $\Delta(x_i, \mathbf{x})$, denoting respectively a suitable function of the original feature and a signal dependent quantization step, change according to the chosen embedding technique. Namely, standard QIM is obtained by choosing

$$f(x_i, \mathbf{x}) = Q^{2\Delta}_{\delta_i,0}(x_i)$$
$$\Delta(x_i, \mathbf{x}) = \Delta \cdot \mathrm{sgn}(x_i - Q^{2\Delta}_{\delta_i,0}(x_i))$$

distortion compensated QIM (DC-QIM) is obtained as

$$f(x_i, \mathbf{x}) = Q^{2\Delta}_{\delta_i,0}(\alpha x_i) + (1-\alpha)x_i$$
$$\Delta(x_i, \mathbf{x}) = \Delta \cdot \mathrm{sgn}(\alpha x_i - Q^{2\Delta}_{\delta_i,0}(\alpha x_i))$$

and rational dither modulation (RDM) is obtained as

$$f(x_i, \mathbf{x}) = Q^{2\Delta}_{\delta_i,0}\left(\frac{x_i}{\mu(\mathbf{x})}\right)\mu(\mathbf{x}, i)$$
$$\Delta(x_i, \mathbf{x}) = \Delta \cdot \mathrm{sgn}\left(\frac{x_i}{\mu(\mathbf{x}, i)} - Q^{2\Delta}_{\delta_i,0}\left(\frac{x_i}{\mu(\mathbf{x}, i)}\right)\right)\mu(\mathbf{x}, i)$$

where $\mathrm{sgn}(x) = x/|x|$, α is a constant in $[0,1]$ and $\mu(\mathbf{x}, i)$ is a suitable function of the features around x_i [22, 26].

The watermarked features in (6) are not suitable for processing through a homomorphic cryptosystem, since they are represented as real values. An integer valued watermarked feature is then obtained as

$$\begin{aligned} z_i &= \lceil f(x_i, \mathbf{x}) \cdot Q \rfloor + w_i \cdot \lceil \Delta(x_i, \mathbf{x}) \cdot Q \rfloor \\ &= f_Q(x_i, \mathbf{x}) + w_i \cdot \Delta_Q(x_i, \mathbf{x}) \end{aligned} \tag{7}$$

where $\lceil \cdot \rfloor$ is the rounding function and Q is a scale factor that can be adjusted according to the required precision. By assuming an additively homomorphic cryptosystem, the above equation can be translated into the encrypted domain as

$$E[z_i] = E[f_Q(x_i, \mathbf{x})] \cdot E[w_i]^{\Delta_Q(x_i, \mathbf{x})}. \tag{8}$$

Note that the seller, being the content owner, knows the plaintext version of $\mathbf{x}$ and can compute both $f_Q(x_i, \mathbf{x})$ and $\Delta_Q(x_i, \mathbf{x})$ in the clear. Hence, equation (8) can be implemented by the seller relying only on the homomorphic properties of the cryptosystem.

5.1 Composite Embedding

One of the main problems of the secure embedding approach presented in equation (8) is that each sample of $\mathbf{x}$ must be encrypted separately. Since the number of features can be very large when marking multimedia contents, the computational cost of encrypting such data may become prohibitive for a practical implementation of the above technique. Also, security of the underlying cryptosystem requires the use of very large algebraic structures. For instance, a secure implementation of Paillier will require at least a 1024 bit modulus, which means that each encrypted word will be represented as a 2048 bit integer. As a consequence, the bandwidth requirements of such an application may soon become very demanding.

In traditional watermarking applications the number of bits required to correctly represent the features is usually quite small, typically ranging from 8 to 16 bits. This suggests that the composite signal representation introduced in Section 3.2.2 may be successfully used to reduce both the number of encryptions and the operations performed on encrypted values.

Let us define the signals $a_i = f_Q(x_i, \mathbf{x})$ and $b_i = w_i \cdot \Delta_Q(x_i, \mathbf{x})$. If we divide the feature vector into blocks of size M, then the composite representations of the above signals can be defined as

$$a_{C,k} = \sum_{j=0}^{R-1} a_{jM+k}\beta^j \qquad b_{C,k} = \sum_{j=0}^{R-1} b_{jM+k}\beta^j. \tag{9}$$

Note that each composite word contains R values that are spaced M positions apart in the original vector. That is, a block of M composite words can be viewed as the superposition of R adjacent blocks of features.

The composite embedding can be defined as the sum of $a_{C,k}$ and $b_{C,k}$. The result is the composite representation of the watermarked features:

$$z_{C,k} = a_{C,k} + b_{C,k} = \sum_{j=0}^{R-1} \left\{ a_{jM+k} + b_{jM+k} \right\}^{\beta^j} = \sum_{j=0}^{R-1} z_{jM+k}\beta^j. \tag{10}$$

As long as $|z_i| < \frac{\beta}{2}$, $\forall i$, the vector of watermarked features can be safely extracted from $z_{C,k}$. Hence, by a suitable choice of β the watermark embedding in (7) can be efficiently performed using (10). The proposed composite embedding can be performed in the encrypted domain by simply using an additively homomorphic cryptosystem. Namely, a secure composite embedding is defined as

$$E[z_{C,k}] = E[a_{C,k}] \cdot E[b_{C,k}]. \tag{11}$$

In the model we consider $E[a_{C,k}]$ is simply obtained as the encryption of $a_{C,k}$, since the seller can compute $a_{C,k}$ in the clear. Conversely, $E[b_{C,k}]$ must be obtained from operations in the encrypted domain applied to the encrypted bits $E[w_i]$. A possible solution is to compute the $E[b_i]$ and then compose them by using the homomorphic property:

$$E[b_{C,k}] = \prod_{j=0}^{R-1} E[b_{jM+k}]^{\beta^j} = \prod_{j=0}^{R-1} \left\{ E[w_{jM+k}]^{\Delta_Q(x_{jM+k}, \mathbf{x})} \right\}^{\beta^j} \tag{12}$$

The above strategy will be referred to as *standard composite embedding*.

A possible drawback of the previous strategy is the necessity of computing the composite representation after the encryption of b_i. Although such encrypted values comes from the product between $E[w_i]$ and $\Delta_Q(x_{jM+k}, \mathbf{x})$, that is, they do not require to actually encrypt anything, nevertheless the amount of intermediate encrypted data and the complexity of the encrypted domain composition may result in an unacceptable computational overhead.

To solve this problem we may resort to an alternative embedding strategy. Usually, the number of bits that compose the watermark is very small with respect to the available features. This suggests that the same bit may be embedded in more than one feature [17], in order to provide a simple repetition code and protect the watermark message from possible detection errors.

In our alternative strategy, we assume that the repetition code is designed so that each feature within the same composite word, say $z_{C,k}$, encodes the same watermark bit, say w_k. The composite component $b_{C,k}$ is then obtained as

$$b_{C,k} = \sum_{j=0}^{R-1} w_k \Delta_Q(x_{jM+k}, \mathbf{x})\beta^j = w_k \sum_{j=0}^{R-1} \Delta_Q(x_{jM+k}, \mathbf{x})\beta^j. \tag{13}$$

Hence, the encrypted component $E[b_{C,k}]$ can be simply obtained as

$$E[b_{C,k}] = E[w_k]^{\Sigma_{j=0}^{R-1} \Delta_Q(x_{jM+k}, \mathbf{x})\beta^j} \tag{14}$$

where the composite representation is computed on plaintext data. This strategy will be referred to as *efficient composite embedding*.

6. IMPLEMENTATION

The efficiency of the proposed solution is verified by means of a practical implementation of the buyer-seller watermarking protocol. Namely, we will implement a prototype of the watermark embedding part, which is deemed the most computational demanding phase of the protocol. As to the setup and watermark generation parts of the protocol, we will refer to a complexity estimate considering well-known practical implementation designs for the cryptographic primitives employed in the protocol.

6.1 Watermark Embedding

In our implementation, we assume that the content is an image and that the features are obtained by applying a block 2D-DCT to the pixel values. Namely, the image is divided into square blocks of 8×8 pixels and an 8×8 DCT is applied to each block. The features are the 14 lowest frequency DCT coefficients of each block, excluding the DC value: they are obtained by reordering the DCT coefficients in the classical zig-zag scan and taking the coefficients from the second to the fifteenth.

The output of the embedder is a vector of encrypted and watermarked DCT coefficients. In order to keep secret the exact set of features, the embedder outputs all the DCT coefficients of the image in encrypted form. The marked coefficients are obtained as in (8). The other coefficients are simply multiplied by Q and rounded before encryption. More formally, the plaintext output values, i.e., after decryption by the buyer, can be expressed as

$$z_i = \begin{cases} f_Q(x_i, \mathbf{x}) + w_i \cdot \Delta_Q(x_i, \mathbf{x}) & x_i \in \mathcal{M} \\ \lceil x_i \cdot Q \rfloor & x_i \notin \mathcal{M} \end{cases} \tag{15}$$

where $\mathcal{M}$ indicates the set of marked features.

After receiving the encrypted and watermarked coefficients, the Buyer will decrypt them, divide them by Q, and reconstruct the watermarked image by applying an inverse DCT. When a composite signal representation is used, the Buyer shall also extract the DCT coefficients from their composite representation. In this case, we assume that the parameters β and R of the composite signal representation are made public by the Seller.

We have implemented three versions of the watermark embedding algorithm. The first version is based on the direct implementation of (8), by encrypting each marked coefficient separately. We will refer to this version as *pixelwise*. The second version uses the composite signal representation according to (12) and will be referred to as *standard composite*. The third one employs the composite signal representation according to (14) and will be referred to as *efficient composite*. All versions are based on the Paillier's cryptosystem [21], with a modulus N of 1024 bits.

The aforementioned versions have been implemented in C++ using the GNU Multi-Precision (GMP) library and the NTL library, and have been run on an Intel(R) Core(TM)2 Quad CPU at 2.40 GHz, used as a single processor. In order to verify the efficiency of the proposed solutions, we measured the execution time of the three versions using three different image sizes: 256×256, 512×512, and 1024×1024. The marked features have been quantized using three different choices for Q: 2^{11}, 2^{15}, and 2^{23}. In each version, a random bit sequence with the same length as the total available features has been embedded using QIM. Both the seller's side computations and the buyer's side computations have been considered. The results are shown in Table 1.

It is evident that the composite signal representation permits to reduce the computational complexity of secure watermark embedding to a great extent. Namely, when $Q = 2^{11}$ the execution time of the efficient composite embedding is 70 times lower than the pixel-

Table 1: Execution time (in seconds) of the different implementations of the secure embedding algorithm: (1) pixelwise; (2) standard composite; (3) efficient composite.

	Q	256×256			512×512			1024×1024		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
embedding	2^{11}	493.2	10.9	7.3	2058.4	44.2	30.5	7528.3	164.1	110.5
	2^{15}	489.3	14.9	9.6	1909.1	58.5	37.8	8704.2	250.7	170.7
	2^{23}	497	22.9	14.6	1953	89.8	57	7926.7	362.1	231.4
extraction	2^{11}	133.8	1.8	1.8	546.2	7	7	2171.4	27.8	27.8
	2^{15}	133.8	2.3	2.3	528.3	9.1	9	2113.7	36	36
	2^{23}	134	3.4	3.4	539.1	13.5	13.5	2122.5	53	53

wise embedding and the corresponding extraction time is about 80 times faster with respect to the pixelwise version. A 1024×1024 image can be processed by the seller in less than two minutes, whereas the buyer can extract the plaintext image in less than 30 seconds. Such timing constrains do not seem prohibitive in view of a practical application of the proposed techniques.

In order to assess the robustness of the watermark in the images processed with the proposed algorithms, we have measured the detection performance after an additive white Gaussian noise (AWGN) attack. We considered the "Man" image with a resolution of 512×512 pixels. The watermark strength is measured by the Document-to-Watermark Ratio (DWR), defined as

$$\mathrm{DWR} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_w^2} \tag{16}$$

where σ_x^2 is the variance of the original image, and σ_w^2 is the variance of the watermark signal, defined as the difference between the original image and the watermarked one.

The image has been watermarked with the DC-QIM, and RDM algorithms described in Section 5, implemented using the standard composite and efficient composite strategies and using different scaling factors Q. The quantization step size has been set in order to obtain a nominal DWR of 33 dB on all images. As to DC-QIM, the parameter α has been set to 0.5, whereas for RDM, the function $\mu(\mathbf{x}, i)$ has been defined as

$$\mu(\mathbf{x}, i) = \left(\frac{1}{2L+1} \sum_{j=i-L}^{i+L} |x_j|^p \right)^{1/p} \tag{17}$$

where $L = 15$ and $p = 1$ [22].

The detection performance has been evaluated in terms of bit error rate (BER) and fingerprint error rate (FER). A fingerprint error is counted every time the detected fingerprint differs from the correct fingerprint by at least one bit. The BER and FER have been measured on 1000 tests, where in each test a 128 bit long fingerprint was embedded into the image. Since the number of available features is much greater than the fingerprint length, the fingerprint has been encoded with a repetition code exploiting the maximum available length.

The detection performance has been measured with different noise levels. The strength of the additive Gaussian noise is expressed through the Watermark-to-Noise Ratio, defined as

$$\mathrm{WNR} = 10 \log_{10} \frac{\sigma_w^2}{\sigma_n^2} \tag{18}$$

where σ_n^2 is the variance of the noise.

The BER and FER curves versus the WNR are plotted in Fig. 4. To facilitate comparison, we also considered the performance of a plaintext embedder using floating point computations, which is referred to as *original* in the figures. As can be seen, for all watermarking algorithms the performance of the standard composite version is very near to the performance of the plaintext version, irrespective of the value of Q. This means that the secure embedding can be safely implemented using the smaller value of Q, which guarantees the higher gain when using the composite signal representation. In the case of the efficient composite version, the results are quite different. As to DC-QIM, the performance decreases slightly when a lower Q is used. As to RDM, quite surprisingly, the efficient version gains about 2 dB with respect to the standard version. We deem that such results can be ascribed to the particular repetition coding pattern of the efficient version, which encode the same bit into DCT coefficients having the same position within the 8×8 blocks. In the case of QIM and DC-QIM, this will slightly correlate the errors on the code bits, since DCT coefficients having the same position will have similar magnitude and will exhibit similar error patterns. Conversely, in the case of RDM, adjacent features are correlated due to the division by $\mu(\mathbf{x}_i)$. Hence, a repetition code avoiding code bits on adjacent features will perform better.

6.2 Efficiency Considerations

In this section, we measure the protocol efficiency in terms of computational and communication complexity for realistic values. As a practical implementation, the following cryptographic primitives are employed in our protocol. The parameters are outlined below or the same as recommended in the original papers. For privacy homomorphism, we choose the Paillier cryptosystem [21], with public key size of 1024 bits, which is the product of two large safe primes of 512 bits each. We employ the group signature scheme by Camenisch et al. [4], with 2048-bit RSA modulus. The key escrow of Paillier private key is based on fair encryption of RSA(-like) keys by Poupard et al. [25]. The proof of bit encryptions is modified from the auxiliary protocols of Damgård-Jurik cryptosystem [11], with the security parameter $s = 1$ for Paillier's cryptosystem. The proof of the correctness of public key is based on proving in zero knowledge that a number is the product of two safe primes [5]. For implementation efficiency, we use the non-interactive statistical zero-knowledge proof for quasi-safe prime products by Gennaro et al. [15]. Because of the foreseen attacks to the hash function SHA-1 and SHA-2 series, we choose to employ SHA-512. Digital signature scheme is RSA-PSS, based on RSA, and hence brings the convenience of generating signature and keys from Paillier's RSA factorizing based keys.

For the computational complexity, the number of exponentiations in each message and the total number of exponentiations required by the protocols, with the group size on which they are performed, are presented in Table 6.2. The communication complexity is evaluated as the sum of the sizes of all messages or rounds, i.e., the number of bits exchanged during the protocols. The reg-

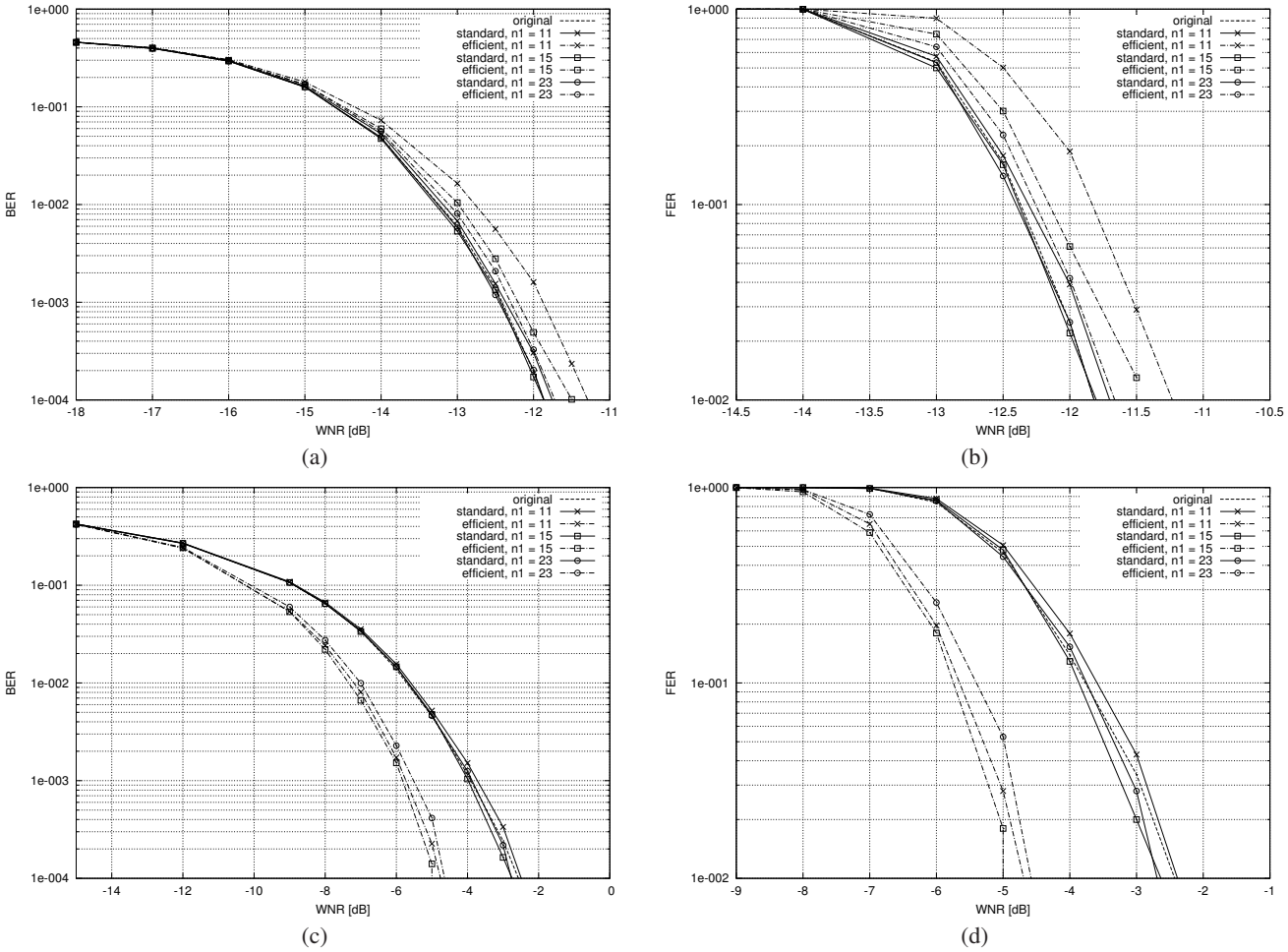


Figure 4: Performance under AWGN attack. DC-QIM: (a) BER; (b) FER. RDM: (c) BER; (d) FER. $n1 = \log_2 Q$**.**

istration protocol contains 2 rounds, namely round 1 as key generation and round 2 as group joining, as detailed in Fig. 1. The messages exchanged in the other protocols are indicated in Fig. 2 and 3. Based on the same group, we distinguish single exponentiations (denoted as exp.) with multi-exponentiations (denoted as multi.), taking into consideration that there are algorithms to compute multi-exponentiations that are faster than first computing each exponentiation separately and then multiplying the results.

In Table 6.2 we consider a 512×512 image, so that the size of the host signal is 262,144 DCT coefficients, with a fingerprint of 128 bits, of which 96 bits for the watermark generated by the buyer and the seller and 32 bits for the index. When using a pixelwise approach, each DCT coefficient is encrypted using Paillier's cryptosystem, requiring 262,144 multi-exponentiations on a 2048-bit group. The size of the encrypted image is $262,144 \times 2048 = 536,870,912$ bits (indicated in message 2.5.3). When using the composite signal representation, we assume that $Q = 2^{11}$, which result in $R = 85$, so that we have roughly 3,760 multi-exponentiations and 6,318,080 transmitted bits. The efficient composite scheme has been assumed.

From Table 6.2, it is evident that the total number of exponentiations are reigned by the number of multi-exponentiations, and that most of the computational effort is required to encrypt the whole image. Most of the computational complexity is located on the Seller's side, since he/she has to encrypt the digital content and perform the embedding in the encrypted domain. However, the composite signal representation can significantly lower this burden. In the pixelwise case, the number of exponentiations required to encrypt the image takes 99.5% of the total number of 2048-bit exponentiations, whereas in the composite case it takes only 73.5%. As to the communication efficiency, the transmission of the encrypted image takes 99.7% of the bandwidth in the pixel wise case and 81.6% of the bandwidth in the composite case. This data also show that the overhead of the protocol is small compared to image encryption: to protect a $512 \times 512 \times 8 = 2$ Mbit image, the data exchanged in the whole protocol (composite version) is about 7.4 Mbit. With an expansion rate of 3.7, small compared to most public key cryptosystems, and with the modern network bandwidth capacity, we can conclude the communication overhead is within an acceptable range.

6.3 Discussions

Due to the scope and space limit of this paper, instead of formal security proofs, we discuss intuitively how the design requirements clarified in Sec. 2 are fulfilled in the proposed protocol. The correctness and completeness of the protocol rely on the security and robustness of the underlying cryptographic and watermarking primitives. In terms of cryptographic requirements, traceability, anonymity, and unlinkability are essentially ensured due to the traceability and anonymity property of the underlying group sig-

Table 2: Computational complexity and communication complexity estimation.

Protocol	number of exp. or multi-exp. (group size)	size (bit)
round 1.1	2 exp. (on 282 bit)	0
round 1.2	(2 exp.+ 4 multi.) (on 2048 bits)	12,853
Protocol 1 total	2 exp.(on 282 bits), (2 exp.+ 4 multi.) (on 2048 bits)	12,853
message 2.0	0	2000
message 2.1	(6 exp. + 2 multi.) (on 2048 bits)	0
message 2.2	1 multi. (on 2048 bits)	0
ZKP_1	4 multi. (on 2048 bits), 25 exp. (on 1024 bits)	15,362
(ZKP_A)	(7 exp. (on 1024 bits))	(8,192)
(ZKP_B)	(4 multi. (on 2048 bits), 18 exp. (on 1024 bits))	(7,170)
message 2.3	96 multi. (on 2048 bits)	0
ZKP_2	1152 exp. (on 2048 bits)	933,888
message 2.4	1 exp. (on 1024 bits)	215,620
message 2.5		
–pixelwise	1 exp. (on 1024 bits), $(262,276)$ multi. (on 2048 bits)	536,870,912
–composite	1 exp. (on 1024 bits), $(3,796)$ multi. (on 2048 bits)	6,318,080
(message 2.5.1)	(4 multi. (on 2048 bits), 1 exp. (on 1024 bits))	(0)
(message 2.5.2)	$((A+32)$ multi. (on 2048 bits))	(0)
(message 2.5.3)		
(–pixelwise)	(262,144 multi.(on 2048 bits))	(536,870,912)
(–composite)	(3,760 multi.(on 2048 bits))	(6,318,080)
message 2.6	1 multi. (on 2048 bits)	0
Protocol 2 total		
–pixelwise	27 exp. (on 1024 bits), (1158 exp.+$(262,332)$ multi.) (on 2048 bits)	538,053,144
–composite	27 exp. (on 1024 bits), (1158 exp.+$(3,948)$ multi.) (on 2048 bits)	7,500,312
message 3.1	0	215,748
message 3.2	1 exp. (on 1024 bits)	2048
message 3.3	1 multi. (on 2048 bits), 1 exp. (on 1024 bit)	1024
message 3.4	1 exp. (on 1024 bits)	13,940
message 3.5	3 exp. (on 2048 bits)	32
Protocol 3 total	3 exp. (on 1024 bits), (3 exp.+1 multi.) (on 2048 bits)	232,664
Protocol in total		
–pixelwise	2 exp. (on 282 bits), 30 exp. (on 1024 bits), (1163 exp.+ $(262,337)$ multi.) (on 2048 bits)	538,298,661
–composite	2 exp. (on 282 bits), 30 exp. (on 1024 bits), (1163 exp.+ $(3,953)$ multi.) (on 2048 bits)	7,745,829

nature scheme as well as the one-time key pair generated by Bob for each transaction. Non-framing (buyer's security) is guaranteed because Alice only knows the encrypted watermarked content, but not Bob's secret watermark and the watermarked content decrypted by Bob. Therefore, Alice cannot frame Bob by distributing replicas herself. Furthermore, Bob generates his watermark without the involvement of any third party, and hence it is infeasible for Alice to recover Bob's watermark via conspiracy attacks, in order to transplant Bob's watermark to another content to fabricate piracy. Non-repudiation (seller's security) is guaranteed because Bob only knows his watermark, but not the joint watermark embedded to the content. Moreover, there is no third party involved in the protocol, so Bob cannot obtain any secret information via conspiracy attacks, which ensures the security of the watermark. In terms of signal processing requirements, the robustness and perceptual quality are guaranteed thanks to the properties of QIM and RDM schemes, whereas the s.p.e.d. requirements are ensured by the use of the composite representation. As to collusion resistance, this is similar to that of the underlying watermarking schemes: in order to achieve a higher collusion resistance, specific anti-collusion codes should be employed.

7. CONCLUSIONS

In this paper, we propose an efficient buyer-seller watermarking protocol based on homomorphic public-key cryptosystem and composite signal representation in the encrypted domain. On one hand, the proposed protocol takes into account all the security concerns related to this kind of applications. Particularly, it avoids double watermark insertion and generalizes to every watermarking algorithm which preserves privacy homomorphism. On the other hand, it employs a recently proposed composite signal representation which allows us to reduce both the computational overhead and the large communication bandwidth which are due to the use of homomorphic public-key encryption schemes.

Our complexity estimates show that the most computational demanding part of the protocol is the encryption of the content and the embedding of the watermark in the encrypted domain. In order to evaluate the feasibility of this part, a practical implementation of an encrypted domain watermark embedding method, based on different watermarking algorithms, has been proposed and tested on images. The results show that the version using composite signal representation can run in less than two minutes on realistic size

images, with a performance in terms of robustness almost indistinguishable from that of the corresponding plaintext embedding algorithms. Considering the computational and network capacity of modern systems, the results suggest that the proposed technique can be successfully used in practical applications.

Acknowledgement

The work reported here has been funded in part by the European Community's Sixth Framework Programme under grant number 034238, SPEED project - Signal Processing in the Encrypted Domain. The work reported reflects only the authors' views; the European Community is not liable for any use that may be made of the information contained herein. This work was also supported in part by the Concerted Research Action (GOA) AMBioRICS 2005/11 of the Flemish Government, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and by the Italian Research Project (PRIN 2007): "Privacy aware processing of encrypted signals for treating sensitive information".

8. REFERENCES

[1] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: the case of dynamic groups. In *Topics in Cryptology - CT-RSA 2005*, LNCS 3376, pages 136–153, 2005.

[2] T. Bianchi, A. Piva, and M. Barni. Efficient pointwise and blockwise encrypted operations. In *Proc. of ACM Multimedia & Security Workshop 2008*, pages 85–90, Oxford, UK, 2008.

[3] J. Camenisch. Efficient anonymous fingerprinting with group signatures. In *ASIACRYPT*, LNCS 1976, pages 415–428, 2000.

[4] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *proceedings of SCN'04*, pages 120–133, 2005.

[5] J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. In *Adv. in Cryptology - EUROCRYPT 1999*, LNCS 1592, pages 106–121, 1999.

[6] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Adv. in Cryptology - CRYPTO 2003*, LNCS 2729, pages 126–144, 2003.

[7] B. Chen and G. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 47(4):1423–1443, May 2001.

[8] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, July 1999.

[9] R. Cramer, I. Damgard, and P. Mackenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *LNCS*, pages 354–372, 2000.

[10] S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, 1998.

[11] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *4th International Workshop on Practice and Theory in Public-Key Cryptography*, LNCS 1992, pages 119–136, 2001.

[12] M. Deng, L. Weng, and B. Preneel. Anonymous buyer-seller watermarking protocol with additive homomorphism. In *Proc. of International Conference on Signal Processing and Multimedia Applications*, page 11, 2008.

[13] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Adv. in Cryptology - CRYPTO 84*, LNCS 196, pages 10–18, 1985.

[14] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni. Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing. *EURASIP Journal on Information Security*, Article ID 78943, 20 pages, 2007.

[15] R. Gennaro, D. Micciancio, and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In *5th ACM Conference on Computer and Communication Security (CCS'98)*, pages 67–72, San Francisco, CA, Nov. 1998.

[16] S. Katzenbeisser, A. Lemma, M. U. Celik, M. van der Veen, and M. Maas. A buyer–seller watermarking protocol based on secure embedding. *IEEE Trans. on Information Forensics and Security*, 3(4):783–786, Dec. 2008.

[17] M. Kuribayashi and H. Tanaka. Fingerprinting protocol for images based on additive homomorphic property. *IEEE Trans. on Image Processing*, 14(12):2129–2139, Dec. 2005.

[18] C.-L. Lei, P.-L. Yu, P.-L. Tsai, and M.-H. Chan. An efficient and anonymous buyer-seller watermarking protocol. *IEEE Trans. on Image Processing*, 13(12):1618–1626, 2004.

[19] N. D. Memon and P. W. Wong. A buyer-seller watermarking protocol. *IEEE Trans. on Image Proc.*, 10(4):643–649, 2001.

[20] P. Moulin and R. Koetter. Data-hiding codes. *Proceedings of the IEEE*, 93(12):2083–2126, Dec. 2005.

[21] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Adv. in Cryptology - EUROCRYPT'99*, LNCS 1592, pages 223–238, 1999.

[22] F. Perez-Gonzalez, C. Mosquera, M. Barni, and A. Abrardo. Rational dither modulation: a high-rate data-hiding method invariant to gain attacks. *IEEE Trans. on Signal Processing*, 53(10):3960–3975, Oct. 2005.

[23] B. Pfitzmann and A.-R. Sadeghi. Anonymous fingerprinting with direct non-repudiation. In *Adv. in Cryptology - ASIACRYPT '00*, LNCS 1976, pages 401–414, 2000.

[24] B. Pittzmann and M. Schunter. Asymmetric fingerprinting. In *Adv. in Cryptology - EUROCRYPT'96*, LNCS 1070, pages 84–95, 1996.

[25] G. Poupard and J. Stern. Fair encryption of RSA keys. In *Adv. in Cryptology - EUROCRYPT'00*, LNCS 1807, pages 172–190, 2000.

[26] J. P. Prins, Z. Erkin, and R. L. Lagendijk. Anonymous fingerprinting with robust QIM watermarking techniques. *EURASIP Journal on Information Security*, 2007, Article ID 31340, 13 pages, 2007.