Efficient Implementation of a Buyer-Seller Watermarking Protocol Using a Composite Signal Representation

Mina Deng, Tiziano Bianchi, Alessandro Piva, and Bart Preneel

COSIC, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
Department of Electronics and Telecommunications, University of Florence, Via S. Marta 3,
50139 Firenze, Italy
{mina.deng,bart.preneel}@esat.kuleuven.be
{tiziano.bianchi,alessandro.piva}@unifi.it

Abstract. Buyer-seller watermarking protocols integrate watermarking techniques with cryptography, for copyright protection, piracy tracing, and privacy protection. In this paper, our main contribution is the development of an efficient buyer-seller watermarking protocol based on homomorphic public-key cryptosystem, and the use of composite signal representation in the encrypted domain 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.

Key words: Buyer-Seller watermarking protocol, Composite signal representation, Homomorphic cryptosystem, Signal processing in the encrypted domain

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 [27,26,4], 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

J. Guajardo and A. Piva (Eds.): SPEED 2009 (Electronic Proceedings), pp. 22–41, September 10th, 2009. Lausanne, Switzerland. These are not formal proceedings. Re-publication is allowed.

a cross-disciplinary application, combining cryptography with watermarking to ensure copyright protection, security and privacy for both the content provider and the customer simultaneously.

In the literature, the first known buyer-seller watermarking protocol was introduced by Memon and Wong [23] using homomorphic cryptosystems to embed a 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 [23] including [22,31,13,20].

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 [15]. As to buyer-seller watermarking protocols, the literature offers few examples of s.p.e.d.oriented approaches. In [21], 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 [29]. However, such techniques require processing each content feature as a separate 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 [3,13], 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 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” [11]. 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 [3] 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 Primitives

2.1 Cryptographic Primitives

Privacy Homomorphism

An encryption scheme is said to be *homomorphic* if for any given encryption scheme, 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 plaintext space $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in the ciphertext space $\mathcal{C}$.

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)*. Because homomorphic cryptosystems cannot have the non-malleability property, the strongest security level a privacy homomorphism can reach is *IND-CPA*, instead of *IND-CCA2*. For instance, the deterministic *RSA* cryptosystem [30] and the *ElGamal* cryptosystem [14] are multiplicative privacy homomorphisms. In contrast to deterministic *RSA*, *ElGamal* is *IND-CPA*. The *Goldwasser-Micali* cryptosystem [18], the *Paillier* cryptosystem [24], and *Paillier*’s generalization the Damgård-Jurik cryptosystem [12] are additive privacy homomorphisms. The state of the art of privacy homomorphic cryptosystems is presented in [16].

Group Signature Group signatures [8,5] enable group members, each with its own private signature key to produce signatures on behalf of the group. Group signature schemes can either be used for static groups, where the identities of group members are fixed in the group setup phase; or for dynamic groups, which allow to update group members. Dynamic schemes have the advantage that instead of assigning a high level of trust to a single group manager, the group manager is separated as an issuer, to issue private signature keys to the group members, and an opener, to open signatures. This provides more security with a lower level of trust [2]. The security properties of static and dynamic group signature schemes are formalized in [1,2] 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. The anonymity property implies that the group member is anonymous in the group.
- **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. The anonymity property implies that the group member is anonymous in the group.
- **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.

2.2 Watermarking and Signal Processing Primitives

Dither Modulation Dither modulation techniques belong to the class of data hiding schemes defined informed embedding algorithms or host-interference rejecting methods [10], where the watermarking problem is viewed as one of communications with side information at the encoder. These systems can achieve host-interference rejection since knowledge of the host signal at the encoder is adequately exploited in system design, in such a way that in absence of attacks the probability of missing detection is equal to zero. Within this class of methods, Quantization Index Modulation (QIM) [9] and Rational Dither Modulation (RDM) [25] 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 [21,29].

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:

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

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

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

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

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

Composite Representation Composite representation of signals [3] 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 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 $a_{C,k}$ of a_n of order R and base β is defined as

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

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

If $\beta > 2Q$ and $\beta^{R} \leq N$, where N is a positive integer, it can be shown [3] 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.

3 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, $\mathcal{A}$ identifies 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 formats. As an illustration, we follow the formal definition of dynamic group signatures of Bellare et al. [2].

3.1 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.

3.2 Watermark Generation and Embedding Protocol

The protocol can be executed multiple times for multi-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.

Certificate authority (CA)		Buyer ($\mathcal{B}$)
1. group key generation	$\xleftrightarrow{SecureChannel}$	2. user key generation
$(gpk, ok, ik) \leftarrow \mathtt{GKg}(1^k)$		$(upk_B, usk_B) \leftarrow \mathtt{UKg}(1^k)$
3. group joining if $\mathtt{Vf}(upk_B, pk_B, sig_B) = 1$	$\xleftarrow{pk_B, sig_B}$	$(pk_B, sk_B) \leftarrow K_s(1^k)$, $sig_B \leftarrow \mathtt{Sig}_{usk_B}(pk_B)$
$cert_B \leftarrow \mathtt{Sig}_{ik}(B, pk_B)$, $reg[B] \leftarrow (pk_B, sig_B)$		
$\mathtt{else}\ cert_B \leftarrow \varepsilon$	$\xrightarrow{cert_B}$	$gsk_B \leftarrow (B, pk_B, sk_B, cert_B)$

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

Seller ($\mathcal{A}$)		Buyer ($\mathcal{B}$)
	$\xleftrightarrow{ARG}$	1. $(pk_B^*, sk_B^*) \leftarrow K_B(1^k)$, $\mu = \mathtt{GSig}(gpk, gsk_B, pk_B^*)$
	$\xleftrightarrow{ZKP_1}$	2. $E_{esc} = E_{pk_{CA}}(sk_B^*)$
	$\xleftrightarrow{ZKP_2}$	3. generate $W_B, ew_{B_i} = E_{pk_B^*}(W_{B_i})$
5. $\mathtt{GVf}(gpk, (pk_B^*, \mu))$, $\mathtt{Vf}(s, pk_B^*)$	$\xleftarrow{m, s}$	4. $m \leftarrow (pk_B^*, \mu, ARG, ew_B, E_{esc})$, $s = \mathtt{Sig}_{sk_B^*}(m)$
generate W_A, ϕ, $W_{AB} \leftarrow (W_A, W_B), W \leftarrow (W_{AB}, \phi)$		
$E_{pk_B^*}(Y_i) = E_{pk_B^*}(X_i \oplus W_i)$, $(i = 1, ..., n)$		
$Table_A \leftarrow [\phi, m, s, W_A]$	$\xrightarrow{E_{pk_B^*}(Y)}$	3. get Y with sk_B^*

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

1. $\mathcal{B}$ first generates a one-time anonymous key pair (pk_B^*, sk_B^*). Then $\mathcal{B}$ applies the *group signing* algorithm $\mathtt{GSig}$ to create a signature μ to pk_B^* with his group signature key gsk_B and the group public key gpk, as $\mu = \mathtt{GSig}(gpk, gsk_B, pk_B^*)$.
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 an honest-verifier 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 an honest-verifier 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 $s = \mathtt{Sig}_{sk_B^*}(m)$ to $\mathcal{A}$.
5. After $\mathcal{A}$ performed the *group signature verification* algorithm $\mathtt{GVf}$ to verify $\mathcal{B}$'s group signature μ with gpk and $\mathcal{B}$'s signature s with pk_B^*, $\mathcal{A}$ 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

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

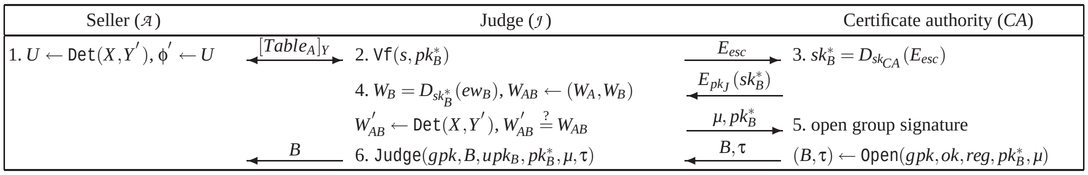
where, for $i = 1, ..., n$

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

Note that $||$ denotes concatenation, and $\oplus$ denotes exclusive OR.
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 $D_{sk_B^*}(E_{pk_B^*}(Y))$.

3.3 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.


Seller ($\mathcal{A}$)
Judge ($\mathcal{J}$)
Certificate authority (CA)
1. $U \leftarrow \mathtt{Det}(X, Y'), \phi' \leftarrow U$
$[Table_A]_Y$
2. $\mathtt{Vf}(s, pk_B^*)$
E_{esc}
3. $sk_B^* = D_{sk_{CA}}(E_{esc})$
$E_{pk_J}(sk_B^*)$
4. $W_B = D_{sk_B^*}(ew_B)$, $W_{AB} \leftarrow (W_A, W_B)$
$W'_{AB} \leftarrow \mathtt{Det}(X, Y')$, $W'_{AB} \overset{?}{=} W_{AB}$
μ, pk_B^*
5. open group signature
B
6. $\mathtt{Judge}(gpk, B, upk_B, pk_B^*, \mu, \tau)$
B, τ
$(B, \tau) \leftarrow \mathtt{Open}(gpk, ok, reg, pk_B^*, \mu)$

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

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

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.

3.4 Zero Knowledge Proofs

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 $\mathcal{CA}$'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. [6], proving that a committed (or revealed) number 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. [7] and the fair encryption of RSA keys by Poupard et al. [28]. Despite the claim of [7] that [28] 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. The encryption scheme of the buyer's private key and the proof of fairness works as follows:
Key Generation: Let N be an RSA modulus $N = P \cdot Q$, where P and Q are primes, $gcd(N, \varphi(n)) = 1$, and G be an integer of order multiple of N modulo N^2. The third party *CA*'s public key is (N, G) and private key is $\lambda(N)$. The buyer's private key is $\lambda(n) = lcm(p-1, q-1)$ with the factoring components p and q such that $n = pq$, which is the modulus of the Paillier cryptosystem's between the buyer and the seller.
Encryption: $\mathcal{P}$ (the buyer) computes $x = n - \varphi(n) = p + q - 1$, randomly chooses $u \in \mathbb{Z}_n^*$ and computes $\Gamma = G^x \cdot u^N \bmod N^2$.
Non-interactive proof: The common input to $\mathcal{P}$ and $\mathcal{V}$ are randomly chosen integers $z_i \in Z_n^*$ for $i = 1..K$.

$\mathcal{P}$ randomly chooses $r_1, r_2 \in [0, A[$ and $v_1, v_2 \in Z_n^*$, computes the commitment $t_1 = \left(G_{r_1} {v_1}^N \bmod N^2, (z_j^{r_1} \bmod n)_{j=1..K}\right), t_2 = \left(G_{r_2} {v_2}^N \bmod N^2, (z_j^{r_2} \bmod n)_{j=1..K}\right)$, and $e_1 = H(t_1, N, G, (z_j)_{j=1..K}, n)$, $e_2 = H(t_2, N, G, (z_j)_{j=1..K}, n)$. $\mathcal{P}$ computes $y_1 = r_1 + e_1(n - \varphi(n))$, $y_2 = r_2 + e_2(n - \varphi(n))$ and $s_1 = u^{e_1} \cdot v_1 \bmod N$, $s_2 = u^{e_2} \cdot v_2 \bmod N$. The non-interactive proof is a 6-tuple $(y_1, s_1, e_1, y_2, s_2, e_2)$.

$\mathcal{V}$ checks $0 \leq y_1 < A$ and $0 \leq y_2 < A$,
computes ${t_1}' = \left((G^{y_1} \cdot y_1^N / \Gamma^{e_1} \bmod N^2, (z_j^{y_1 - e_1 n} \bmod n)_{j=1..K}\right)$
and ${t_2}' = \left((G^{y_2} \cdot y_2^N / \Gamma^{e_2} \bmod N^2, (z_j^{y_2 - e_2 n} \bmod n)_{j=1..K}\right)$,
checks $e_1 = H({t_1}', N, G, (z_j)_{j=1..K}, n)$ and $e_2 = H({t_2}', N, G, (z_j)_{j=1..K}, n)$.
$\mathcal{V}$ accepts if and only if this holds.

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 the prover $\mathcal{P}$) needs to prove to the seller (as the verifier $\mathcal{V}$) that a given ciphertext C is an encryption of a bit, i.e., the corresponding plaintext is one of the two candidate plaintexts $w_1 = 1$ or $w_2 = 0$, 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 honest-verifier zero knowledge proof by Damgård and Jurik [12], and is depicted as follows:

As explained above, Paillier encryption is $E(i) = g^i \cdot r^n \bmod n^2$, 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 $u_1 = cg^{-w_1} \bmod n^2$ and $u_2 = cg^{-w_2} \bmod n^2$. 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 ZKP_3. It is necessary to first outline how ZKP_3 works.

ZKP_C**: Prove a value is n-th residue modulo** n^2
Common input: n, u
Private input for $\mathcal{P}$: v, such that $u = v^n \bmod n^2$

1. $\mathcal{P}$ chooses at random $r \in \mathbb{Z}_n^*$, and sends $a = r^n \bmod n^2$ to $\mathcal{V}$.
2. $\mathcal{V}$ choose a challenge e, a random k-bit number, and sends e to $\mathcal{P}$.
3. $\mathcal{P}$ sends the response $z = rv^e \bmod n^2$ to $\mathcal{V}$.
4. $\mathcal{V}$ checks that $z^n = au^e \bmod n^2$, and accepts if and only if this holds. Otherwise, the protocol halts.

ZKP_D**: Prove a value is 1-out-of-2 n-th residue modulo** n^2
Common input: n, u_1, u_2
Private input for $\mathcal{P}$: v_1, such that $u_1 = {v_1}^n \bmod n^2$
1. $\mathcal{P}$ chooses at random $r_1 \in \mathbb{Z}_n^*$, and then invokes $\mathcal{M}$ on input n, u_2 to get a conversation a_2, e_2, z_2. $\mathcal{P}$ sends $a_1 = {r_1}^n \bmod n^2$, a_2 to $\mathcal{V}$.
2. $\mathcal{V}$ choose a challenge d, a random t-bit number, and sends d to $\mathcal{P}$. Note that if k is

the bit length of n, we can set $t = k/2$ and be assured that a cheating prover can made the verifier accept with probability $\leq 2^{-t}$.
3. $\mathcal{P}$ computes $e_1 = d - e_2 \bmod n^2$ and $z_1 = r_1 v_{1_{e_1}} \bmod n^2$, and sends e_1, z_1, e_2, z_2 to $\mathcal{V}$.
4. $\mathcal{V}$ checks that $d = e_1 + e_2 \bmod 2^t$, $z_1^n = a_1 {u_1}^{e_1} \bmod n^2$ and $z_2^n = a_2 {u_2}^{e_2} \bmod n^2$, and accepts if and only if this holds. Otherwise, the protocol halts.

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 [19].

4 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.

In the proposed protocol, we adopt a secure watermark embedding scheme based on dither modulation techniques and homomorphic cryptosystems; such class of embedding schemes has been proposed in [21,29]. 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

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

distortion compensated QIM (DC-QIM) is obtained as

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

and rational dither modulation (RDM) is obtained as

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

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 [25,29].

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

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

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 underlying cryptosystem.

4.1 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 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

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

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 can be 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 come 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 [21], 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

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

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

5 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.

5.1 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 [24], with a modulus N of 1024 bits.

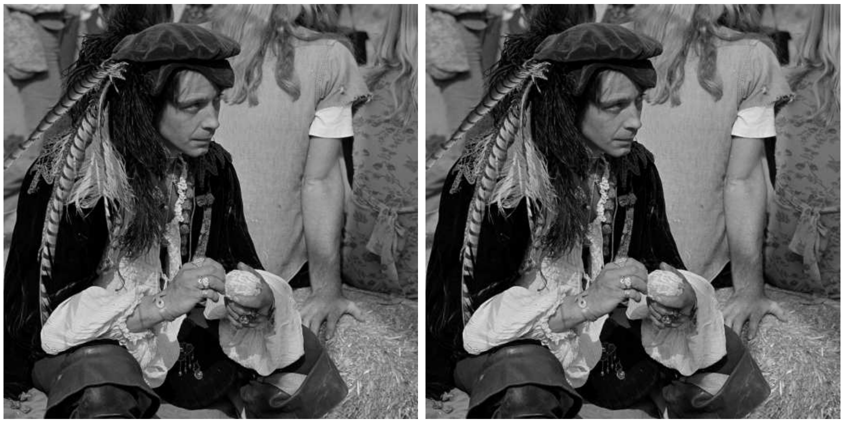
The aforementioned versions have been implemented in C++ using the GNU Multi-Precision (GMP) library [33] and the NTL library [32], and have been run on an Intel(R) Core(TM)2 Quad CPU at 2.40 GHz, used as a single processor. In order to verify the efficiency of the proposed solutions, we measured the 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

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

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 pixelwise 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.



(a) (b)

Fig. 4. Example of a watermarked image: (a) original "Man" image; (b) RDM watermarked image (PSNR = 46.62 dB). The watermarked image has been obtained with efficient composite embedding, using $Q = 2^{11}$.

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 4, 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 $p = 1$ [25].

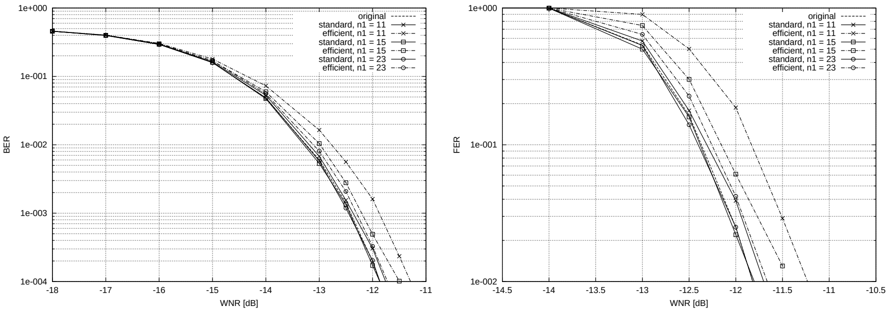
The detection performance has been evaluated in terms of bit error rate (BER) and fingerprint error rate (FER). A fingerprint error is counted every time the detected fingerprint differs from the correct fingerprint by at least one bit. The BER and FER have been measured on 1000 tests, where in each 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

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

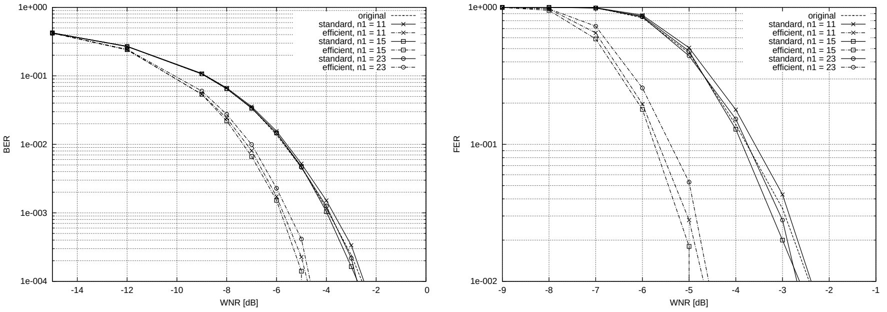
where σ_n^2 is the variance of the noise.

The BER and FER curves versus the WNR are plotted in Fig. 5-6. 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



(a) (b)

Fig. 5. Performance of DC-QIM under AWGN attack: (a) BER; (b) FER. $n1 = \log_2 Q$.



(a) (b)

Fig. 6. Performance of RDM under AWGN attack: (a) BER; (b) FER. $n1 = \log_2 Q$.

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.

5.2 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 [24], 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. [5], 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. [28]. The proof of bit encryptions is modified from the auxiliary protocols of Damgård-Jurik cryptosystem [12], 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

Table 2. Computational complexity and communication complexity estimation.

Protocol	number of exp. or multi-exp. (group size)	size (bit)
Protocol 1 total	2 exp.(on 282 bits), (2 exp.+ 4 multi.) (on 2048 bits)	12,853
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
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

number is the product of two safe primes [6]. For implementation efficiency, we use the non-interactive statistical zero-knowledge proof for quasi-safe prime products by Gennaro et al. [17]. 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 5.2. 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 registration 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 5.2 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 5.2, 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 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.

As for the work in progress, we are currently working on the formal security proof of the proposed buyer-seller watermarking protocol.

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”.

References

1. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Topics in Cryptology - Eurocrypt 2003*, LNCS 2656, pages 614–629. Springer-Verlag, 2003.
2. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: the case of dynamic groups. In *Topics in Cryptology - CT-RSA 2005*, LNCS 3376, pages 136–153. Springer-Verlag, 2005.
3. 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.
4. Jan Camenisch. Efficient anonymous fingerprinting with group signatures. In *ASIACRYPT*, LNCS 1976, pages 415–428. Springer-Verlag, 2000.
5. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *In proceedings of SCN’04, LNCS series*, pages 120–133. Springer, 2005.
6. Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In *Advances in Cryptology - EUROCRYPT 1999*, LNCS 1592, pages 106–121, 1999.
7. Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology - CRYPTO 2003*, LNCS 2729, pages 126–144, 2003.
8. D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology - EUROCRYPT ’91*, LNCS 547, pages 257–265, 1991.
9. 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.
10. 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.
11. Scott Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, May 1998.
12. 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. Springer-Verlag, 2001.
13. Mina Deng, Li Weng, and Bart Preneel. Anonymous buyer-seller watermarking protocol with additive homomorphism. In *Proc. of International Conference on Signal Processing and Multimedia Applications*, page 11, 2008.
14. 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.
15. 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*, 2007, Article ID 78943, 20 pages, 2007.

16. Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for non-specialists. *EURASIP Journal on Information Security*, October 2007. http://www.hindawi.com/RecentlyAcceptedArticlePDF.aspx?journal=IS&number=13801.
17. Rosario Gennaro, Daniele Micciancio, and Tal 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, California, November 1998. ACM, ACM Press.
18. S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker hiding all partial information. In *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, pages 365–377, 1982.
19. Ronald Cramer Ivan, Ivan Damgård, and Philip Mackenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000: 3rd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1751 of *LNCS*, pages 354–372. Springer-Verlag, 2000.
20. Stefan Katzenbeisser, Aweke Lemma, Mehmet Utku Celik, Michiel van der Veen, and Martijn Maas. A buyer–seller watermarking protocol based on secure embedding. *IEEE Transactions on Information Forensics and Security*, 3(4):783–786, Dec. 2008.
21. Minoru Kuribayashi and Hatsukazu Tanaka. Fingerprinting protocol for images based on additive homomorphic property. *IEEE Transactions on Image Processing*, 14(12):2129–2139, December 2005.
22. Chin-Laung Lei, Pei-Ling Yu, Pan-Lung Tsai, and Ming-Hwa Chan. An efficient and anonymous buyer-seller watermarking protocol. *IEEE Transactions on Image Processing*, 13(12):1618–1626, 2004.
23. Nasir D. Memon and Ping Wah Wong. A buyer-seller watermarking protocol. *IEEE Transactions on Image Processing*, 10(4):643–649, 2001.
24. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT'99*, LNCS 1592, pages 223–238. Springer-Verlag, 1999.
25. 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, October 2005.
26. B. Pfitzmann and A.-R. Sadeghi. Anonymous fingerprinting with direct non-repudiation. In *Advances in Cryptology - ASIACRYPT '00*, LNCS 1976, pages 401–414. Springer-Verlag, 2000.
27. B. Pittzmann and M. Schunter. Asymmetric fngerprinting. In *Advances in Cryptology - EUROCRYPT'96*, LNCS 1070, pages 84–95. Springer-Verlag, 1996.
28. Guillaume Poupard and Jacques Stern. Fair encryption of *RSA* keys. In *In Proceedings of Eurocrypt 2000, volume 1807 of LNCS*, pages 173–190. Springer-Verlag, 2000.
29. 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.
30. R. L. Rivest, A. Shamir, and L. M. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. of the ACM*, 21(2):120–126, Feb. 1978.
31. J. Zhang, W. Kou, and K. Fan. Secure buyer-seller watermarking protocol. *IEE Proceedings Information Security*, 153(1):15–18, March 2006.
32. NTL: A library for doing number theory. Available online: http://www.shoup.net/ntl/.
33. GNU Multiple Precision Arithmetic Library. Available online: http://gmplib.org/.