Tiziano Bianchi and Alessandro Piva

Secure Watermarking for Multimedia Content Protection

A review of its benefits and open issues

©ISTOCKPHOTO.COM/ALEX

SIGNAL PROCESSING IN THE ENCRYPTED DOMAIN

Distribution channels such as digital music downloads, video-on-demand, and multimedia social networks pose new challenges to the design of content protection measures aimed at preventing copyright violations. Digital watermarking has been proposed as a possible brick of such protection systems, providing a means to embed a unique code as a "fingerprint" into each copy of the distributed content. However, application of watermarking for multimedia content protection in realistic scenarios poses several security issues.

Secure signal processing, by which name we indicate a set of techniques able to process sensitive signals that have been obfuscated either by encryption or by other privacy-preserving primitives, may offer valuable solutions to the aforementioned issues. More specifically, the adoption of efficient methods for watermark embedding or detection on data that have been secured in some way, which we name in short secure watermarking, provides an elegant way to solve the security concerns of fingerprinting applications.

The aim of this contribution is to illustrate recent results regarding secure watermarking to the signal processing community, highlighting both benefits and still open issues. Some of the most interesting challenges in this area, as well as new research directions, will also be discussed.

INTRODUCTION

A digital watermark provides a communication channel multiplexed into the original content through which it is possible to transmit some application-dependent information; in forensic tracing, a watermark can be used to embed a unique code, as a fingerprint, into each copy of the content to be distributed, linking the copy either to a particular user or to a specific device. When unauthorized published content is found, the fingerprint allows to trace the user who has redistributed the content [4], [24].

The adoption of digital watermarking techniques for multimedia content protection in realistic scenarios raises a set of important issues. One is represented by a possible malicious incrimination of an honest buyer (known in the literature as customer's rights problem): when the watermark is embedded at the distribution server, a customer whose watermark has

Digital Object Identifier 10.1109/MSP.2012.2228342
Date of publication: 12 February 2013

been found on unauthorized copies can claim that he/she has been framed by a malicious seller who inserted his/her identity as watermark in an arbitrary object. The mere existence of this problem may discredit the reliability of the forensic tracing architecture. A possible solution to this problem is to construct fingerprinting asymmetric schemes, where only the buyer has access to the fingerprinted content; however, if the merchant later finds a copy of the content, he/she can still identify the buyer and prove to third parties that this buyer bought this copy. While cryptographically secure asymmetric fingerprint protocols have been proposed several years ago [29], the actual implementation of such protocols for realistic multimedia contents have been investigated only recently [34]. Another problem is the system scalability: in classical distribution models, the watermark embedding process is carried out by a trusted server before releasing the content to the user. However, in large-scale systems, servers may become overloaded, since the computational burden due to watermark embedding grows linearly with the number of users. In addition, since the distribution of individually watermarked copies requires to resort to point-to-point communication channels, bandwidth requests can become prohibitive. A third question is the existence of untrusted verifiers: in the watermark detection process, a content owner can be asked to prove to another party that a watermark is present in his/her copy. This process usually requires to reveal secret information related to watermark embedding, such that a cheating party could then exploit the knowledge of the secrets to remove the watermark from the content.

THE ADOPTION OF EFFICIENT METHODS FOR WATERMARK EMBEDDING OR DETECTION ON DATA THAT HAVE BEEN SECURED IN SOME WAY PROVIDES AN ELEGANT WAY TO SOLVE THE SECURITY CONCERNS OF FINGERPRINTING APPLICATIONS.

Within this challenging research area, previous questions have been answered by resorting to secure signal processing techniques [21]. The first two problems have been addressed introducing secure watermark embedding, that is, mechanisms where the watermark embedding is carried out in a way that the content owner does not have access to the final watermarked version, while not disclosing the original content. Solutions exist to securely and efficiently embed a watermark both at the server's side and at the client's side. Secure server-side embedding can be used as a building block in asymmetric fingerprint protocols, providing a cryptographically secure solution to the customer's rights problem, while secure client-side embedding offers a very efficient solution to the system scalability problem. The presence of untrusted verifiers can be solved by resorting to secure watermark detection, i.e., to an interactive proof scheme where the content owner convinces another interested party that his/her content contains a given watermark without disclosing sensitive information that could facilitate the watermark removal, like the secret key of the watermarking algorithm or the actual watermark.

In this article, we will illustrate the aforementioned techniques, trying to provide the reader with a clear understanding of their merits and their present limitations. The article will end with a discussion about possible new research directions, focusing on research challenges that may be particularly interesting for the signal processing community.

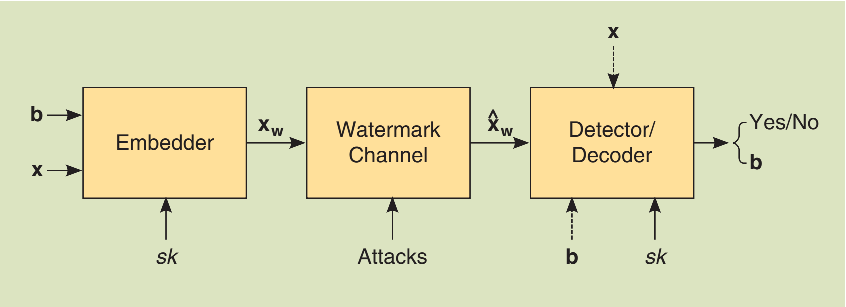
A FEW PRELIMINARIES

In this section, we briefly review some basic concepts useful to understand the described solutions: digital watermarking model [4], homomorphic encryption [27], commitment schemes [12], zero-knowledge protocols [2], and asymmetric fingerprinting protocols [29].

WATERMARKING MODEL

A common model [4] for a digital watermarking system is shown in Figure 1. The inputs of the system are a vector $\mathbf{x} = [x_1, x_2, \ldots, x_M]$, representing either the original host signal samples or, more generally, a set of features of the host signal computed by a suitable transform (common examples are the discrete Fourier tranform (DFT) and the discrete cosine transform (DCT)), and some application dependent to-be-hidden information, here represented as a binary vector $\mathbf{b} = [b_1, b_2, \ldots, b_L]$, with b_i taking values in $\{0, 1\}$. The embedder inserts the watermark code $\mathbf{b}$ into the host signal to produce a watermarked signal $\mathbf{x}_w$, usually making use of a secret key sk to control some parameters of the embedding process and allow the watermark recovery only to authorized users. The general form of the embedding function can thus be written as

$$\mathbf{x}_w = \mathcal{E}(\mathbf{x}, \mathbf{b}, sk). \qquad (1)$$



[FIG1] A digital watermarking system.

It is often useful to describe the embedding function by introducing a watermarking signal w, so that the watermarked signal can be expressed as $\mathrm{x}_w = \mathrm{x} + \mathrm{w}$. When the watermarking signal w depends only on b and sk, the scheme is usually referred to as blind embedding. More advanced watermarking techniques, however, take into account also the host signal

$\mathbf{x}$ according to the principle of digital communications with side information at the encoder, which permits the achievement of higher embedding capacities [11]. Such schemes are referred to as informed embedding.

All manipulations (both intentional and nonintentional) the watermarked content may undergo during distribution and use are modeled by the watermark channel, which modifies $\mathbf{x}_w$ into the received version $\hat{\mathbf{x}}_w$. Based on $\hat{\mathbf{x}}_w$, the hidden information can be retrieved either by a watermark detector, which verifies the presence, or the absence of a specific message given to it as input, that is,

$$\mathcal{D}(\hat{\mathbf{x}}_w, \mathbf{b}, sk) = \text{yes/no}, \tag{2}$$

or by a watermark decoder, which reads the binary information conveyed by the watermarked signal, that is,

$$\mathcal{D}(\hat{\mathbf{x}}_w, sk) = \mathbf{b}. \tag{3}$$

When detectors and decoders do not depend from the original content $\mathbf{x}$, as in the examples above, they are referred to as blind or oblivious detector/decoder. In some cases, however, detectors and decoders may also use the original content $\mathbf{x}$ to retrieve the hidden information, in which case they are referred to as nonblind detector/decoder.

[FOR USE IN SIGNAL PROCESSING APPLICATIONS, COMMITMENT SCHEMES THAT ARE ADDITIVELY HOMOMORPHIC ARE OF SPECIFIC IMPORTANCE.]

HOMOMORPHIC CRYPTOSYSTEMS

A cryptosystem is said to be homomorphic with respect to an operation $\star$ if there exists an operator $\varphi(\cdot,\cdot)$ such that for any two plain messages m_1 and m_2, we have

$$D[\varphi(E[m_1], E[m_2])] = m_1 \star m_2, \tag{4}$$

where $E[\cdot]$ ($D[\cdot]$) denotes the encryption (decryption) operator. It is evident that homomorphic encryption provides an elegant way of performing a set of operations by working on encrypted data. In particular, an additively homomorphic cryptosystem maps an addition in the plain text domain to an operation in the ciphertext domain, (usually a multiplication). Given two plain texts m_1 and m_2, the following equalities are then satisfied:

$$D[E[m_1] \cdot E[m_2]] = m_1 + m_2, \tag{5}$$

and, as a consequence,

$$D[E[m]^a] = am, \tag{6}$$

where a is a public integer. Additively homomorphic cryptosystems allow then to perform in the encrypted domain additions, subtractions and multiplications with a known (nonencrypted) value (but not division, since it could lead to noninteger values), thus providing a way of applying any linear operator in the encrypted domain.

Another desirable property of a homomorphic cryptosystem is that given two encrypted values it should not be computationally feasible to decide whether they conceal the same value or not. The above property guarantees the confidentiality of the cryptosystem when encrypting data with a restricted set of possible values (for example bits), or when a set of data exhibiting a peculiar correlation structure (for example, consecutive signal samples) is encrypted as separate encryptions. A scheme that satisfies the above property is referred to as semantically secure and is commonly implemented by letting the encryption function E depend on a random parameter r. A well-known additively homomorphic and semantically secure scheme is the one presented by Paillier in [27].

COMMITMENT SCHEMES

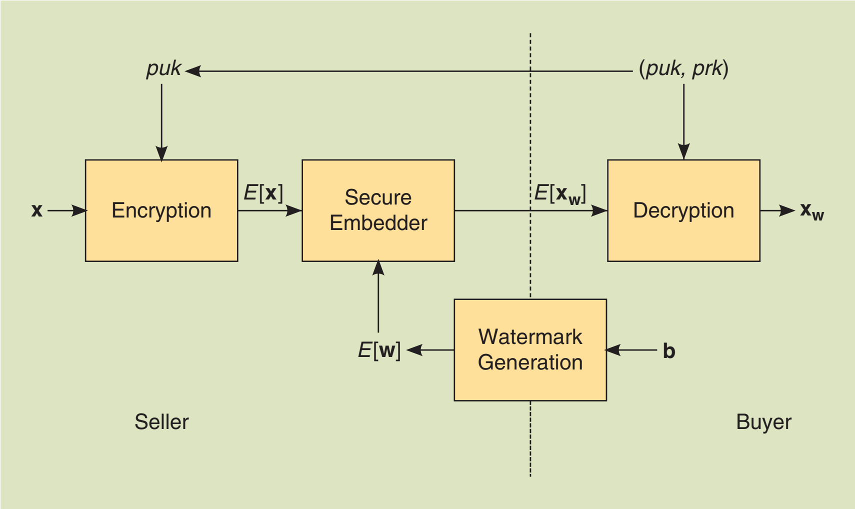
A commitment scheme is a method that allows a party, let us say Alice, to commit to a value while keeping it hidden from another party, let us say Bob, and while also preserving Alice's ability to reveal the committed value later to Bob. A useful way to visualize a commitment scheme is to think of Alice as putting the value in a locked box, and giving the box to Bob. The value in the box is hidden from Bob, who cannot open the lock without the help of Alice (hiding property), but since Bob has the box, the value inside cannot be changed by Alice; hence, Alice is "committed" to this value (binding property). At a later stage, Alice can "open" the box and reveal its content to Bob.

For use in signal processing applications, commitment schemes that are additively homomorphic are of specific importance: here, knowledge of two commitments allows one to compute—without opening—a commitment of the sum of the two committed values, i.e., $C[m_1] \cdot C[m_2] = C[m_1 + m_2]$.

Again, the homomorphic property only supports additions. However, there are situations where it is not possible to prove a relation by additive homomorphism as in proving that a committed value is the square of the value of another commitment. In such cases, zero-knowledge proofs can be used.

ZERO-KNOWLEDGE PROOF PROTOCOLS

A zero-knowledge protocol allows a party, called prover of the statement, to prove a certain statement or condition to another party, called verifier of the statement, without revealing any knowledge to the verifier except the fact that the assertion is valid [15]. As a simple example, consider the case where the prover claims to have a way of factorizing large numbers. The verifier will send the prover a large number and he/she will send back the factors. Successful factorization of several large integers will decrease verifier's doubt in the truth of the prover's claim. At the same time, the verifier will learn nothing about the actual factorization method.



[FIG2] Server-side secure watermark embedding.

Although simple, the example shows an important property of zero-knowledge protocol proofs, namely that they are interactive in nature. The interaction should be such that with increasing number of "rounds," the probability of an adversary to successfully prove an invalid claim decreases significantly.

ASYMMETRIC FINGERPRINTING

In the most common case, distribution tracing is made possible by letting the entity selling the content (seller) insert a distinct watermark (fingerprint) identifying the person purchasing the content (buyer), within any copy of data that is distributed. Unfortunately, this scheme does not protect the buyer's rights, since the watermark is inserted solely by the seller. A buyer whose watermark is found in an unauthorized copy can claim that the unauthorized copy was created and distributed by the seller, or by a reselling agent. A possible solution consists in resorting to a trusted third party (TTP) who takes care of watermark embedding and decoding. However, TTPs are difficult to implement in real-life scenarios and may easily become the bottleneck of the whole system.

A FUNDAMENTAL BUILDING BLOCK OF ASYMMETRIC FINGERPRINTING IS A FUNCTIONALITY THAT ALLOWS THE SELLER AND BUYER TO JOINTLY PERFORM WATERMARK EMBEDDING IN SUCH A WAY THAT THE ORIGINAL CONTENT IS A PRIVATE INPUT OF THE SELLER, WHEREAS THE FINGERPRINT DATA IS A PRIVATE INPUT OF THE BUYER.

An elegant solution to the aforementioned problems is asymmetric fingerprinting [29], where the buyer first commits to a secret that only he/she knows (registration phase), then both buyer and seller follow a protocol (named buyer-seller watermarking protocol) after which only the buyer receives a copy of the watermarked work. However, if the copy is illegally distributed, the seller can identify the buyer from whom the copy originated, and prove it to a judge by using a proper dispute resolution protocol. A fundamental building block of asymmetric fingerprinting is a functionality that allows the seller and buyer to jointly perform watermark embedding in such a way that the original content **x** (and the secret key *sk*) is a private input of the seller, whereas the fingerprint data **b** is a private input of the buyer.

SECURE WATERMARK EMBEDDING

As discussed above, secure watermark embedding techniques can provide an elegant solution to both the customer's rights problem in copyright protection and the system scalability problem in distribution models. In the first case, secure embedding usually occurs at the server's side, as a building block in an asymmetric fingerprinting protocol. In the second case, secure embedding is performed by each client of the distribution system, after the original content has been encrypted and distributed using a broadcast channel.

SERVER-SIDE WATERMARK EMBEDDING

Secure watermark embedding at the server's side is a classical example of a problem that can be efficiently solved by resorting to secure signal processing techniques based on homomorphic encryption. Let us assume that the buyer holds the public/private key pair (*puk*,*prk*) of an additively homomorphic cryptosystem. By recalling the watermarking model in the section "Watermarking Model," it is evident that if the seller and buyer can compute an encryption of the watermark signal **w**, then watermark embedding can be performed by the seller in the encrypted domain as follows:

$$E[x_{w,i}] = E[x_i] \cdot E[w_i]. \qquad (7)$$

In the above equation, the seller, knowing the plain text values of x_i, can compute the encryptions $E[x_i]$ by relying on the buyer's public key *puk*. However, the computed value $E[x_{w,i}]$ is meaningless for the seller, since he/she does not know the private key for decrypting. Hence, only the buyer can have access to the watermarked content, as illustrated in Figure 2.

Even if the above simple protocol satisfies the requirements of a secure embedding mechanism to be used in asymmetric fingerprinting, to be applied with realistic watermarking algorithms we have to ensure that also the embedding of the watermark signal can be computed in the encrypted domain. In the early solution proposed by Pfitzmann and Schunter [29], both the original content and the watermark signal were modeled as bit strings, with the latter being equal to the buyer's code. Hence, a simple privacy homomorphism allowing the server to compute the exclusive or (XOR) of encrypted bits was sufficient. However, the above technique is not adequate in the case of realistic multimedia contents, since it guarantees neither the robustness nor the perceptual fidelity required in this kind of scenario. Let us describe how an embedding in the encrypted domain can be implemented for the two main classes of watermarking techniques.

SECURE WATERMARK EMBEDDING SCHEMES BASED ON DM TECHNIQUES CAN BE EFFICIENTLY IMPLEMENTED BY RELYING ON HOMOMORPHIC CRYPTOSYSTEMS.

1) *Secure spread-spectrum:* One of the most known watermarking algorithms for multimedia content is the spread-spectrum technique [4]. The watermark embedding rule is defined as follows:

$$x_{w,i} = x_i + \gamma(2b-1)s_i, \qquad (8)$$

where $b \in \{0,1\}$ is the embedded bit, s_i is the ith component of a spreading sequence, and γ is a scaling factor controlling the watermark strength. The above scheme can be used to encode a single bit. Multiple bits can be encoded by partitioning the host features into several chunks and by using the above technique to embed a bit in every chunk.

A secure version of spread-spectrum watermarking can be obtained in a very simple way by relying on an additively homomorphic and semantically secure cryptosystem. If we assume that the seller receives the encrypted bit $E[b]$ as the output of the registration phase of the fingeprinting protocol, then the encrypted watermarked signal can be computed as

$$E[x_{w,i}] = E[x_i] \cdot E[b]^{2\gamma s_i} \cdot E[\gamma s_i]^{-1}. \qquad (9)$$

The seller can easily compute the above expression, since he/she knows the plain text values of both x_i and s_i. Similar schemes based on Cox's spread spectrum watermarking technique [4] are also possible based either on multiplicatively [26] or additively [19] homomorphic cryptosystems.

2) Secure dither modulation: Dither modulation (DM) techniques, belonging to the class of data hiding schemes defined informed embedding, hide signal-dependent watermarks using as embedding rule the quantization of some content features. The simplest example of such techniques is a binary DM with uniform scalar quantizers: in this realization, we assume that each bit of $\mathbf{b}$, say b_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 $b = 0$ and $b = 1$ are built as

$$\begin{aligned} \mathcal{U}^{\Delta}_{\delta,0} &= \{u_{0,k}\} = \{k\Delta + \delta, k \in \mathbb{Z}\}, \\ \mathcal{U}^{\Delta}_{\delta,1} &= \{u_{1,k}\} = \{k\Delta + \Delta/2 + \delta, k \in \mathbb{Z}\}, \end{aligned} \qquad (10)$$

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

A 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 $b = \{0,1\}$

$$Q^{\Delta}_{\delta,b}(x) = \arg \min_{u_{b,k} \in \mathcal{U}^{\Delta}_{\delta,b}} |u_{b,k} - x|, \qquad (11)$$

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

Secure watermark embedding schemes based on DM techniques can be efficiently implemented by relying on homomorphic cryptosystems [20], [33]. 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 DM can be expressed as

$$x_{w,i} = f(\mathbf{x}, i) + b_i \cdot \Delta(\mathbf{x}, i), \qquad (12)$$

where $f(\mathbf{x}, i)$ and $\Delta(\mathbf{x}, i)$, denoting respectively a suitable function of the original features and a signal dependent quantization step, depend on the chosen embedding technique. For example, quantization index modulation (QIM) [9] can be obtained by choosing

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

whereas rational DM (RDM) [28] can be obtained as

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

where $\operatorname{sgn}(x) = x/|x|$ and $\mu(\mathbf{x}, i)$ is a suitable function of the features around x_i [28], [33], [13].

By assuming an additively homomorphic cryptosystem, (12) can be translated into the encrypted domain as

$$E[x_{w,i}] = E[f(\mathbf{x}, i)] \cdot E[b_i]^{\Delta(\mathbf{x}, i)}. \qquad (13)$$

Note that the seller, being the content owner, knows the plain text version of $\mathbf{x}$ and can compute both $f(\mathbf{x}, i)$ and $\Delta(\mathbf{x}, i)$ in the clear. Hence, (13) can be implemented by the seller relying only on the homomorphic properties of the underlying cryptosystem.

3) Composite embedding: One of the main problems of the secure embedding approach presented in (9) and (13) is that each sample of $\mathbf{x}$ must be encrypted separately.

In traditional watermarking applications, the number of bits required to correctly represent each feature is usually quite small, typically ranging from 8 to 16 b. On the contrary, security of the underlying cryptosystem requires the use of very large algebraic structures (e.g., a secure implementation of Paillier requires that each encrypted word is represented at least as a 2,048-b integer):

the combination of these conditions results in a high-data expansion from the plain text to the encrypted representation of signals, so that the bandwidth requirements of such an application may soon become very demanding. In addition, 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.

A PARTICULARLY INTERESTING APPROACH IS REPRESENTED BY METHODS USING A STREAM-CIPHER THAT ALLOWS THE USE OF MULTIPLE DECRYPTION KEYS, WHICH DECRYPT THE SAME CIPHERTEXT TO SLIGHTLY DIFFERENT PLAIN TEXTS.

As a solution to the above problems, composite representation of signals [5] has been proposed. This representation permits to group several signal samples into a single word and to perform basic linear operations on them, then allowing to speed up linear operations on encrypted signals via parallel processing and to reduce the size of the whole encrypted signal. Let us consider a signal a_n. Given a pair of positive integers β, R, the composite representation of a_n of order R and base β is defined as

$$a_{C,k} = \sum_{i=0}^{R-1} a_{i,k}\beta^i, \quad k = 0, 1, \ldots, M-1, \tag{14}$$

where $a_{i,k}$, $i = 0, 1, \ldots, R-1$ indicate R disjoint subsequences of the signal a_n. Under suitable hypotheses [5], the composite representation $a_{C,k}$ can be processed through modular arithmetic without losing information, and several kinds of linear processing can be directly applied to the composite representation of the signal, allowing for a parallel processing of the original signal samples. As an example, a much more efficient secure embedding algorithm can be obtained [13]. Let us define the signals $\tilde{x}_i = f(\mathrm{x}, i)$ and $\tilde{w}_i = b_i \cdot \Delta(\mathrm{x}, i)$. By dividing the feature vector into blocks of M samples, the composite representations of the above signals can be expressed as $\tilde{x}_{C,k} = \sum_{j=0}^{R-1} \tilde{x}_{jM+k}\beta^j$ and $\tilde{w}_{C,k} = \sum_{j=0}^{R-1} \tilde{w}_{jM+k}\beta^j$, and composite embedding can be defined as

$$\begin{aligned} x_{w,C,k} &= \tilde{x}_{C,k} + \tilde{w}_{C,k} \\ &= \sum_{j=0}^{R-1} \{\tilde{x}_{jM+k} + \tilde{w}_{jM+k}\}\beta^j = \sum_{j=0}^{R-1} x_{w,jM+k}\beta^j, \end{aligned} \tag{15}$$

where the result is the composite representation of the watermarked features $x_{w,i}$.

4) Signal representation: The watermarked features in the previous example are not suitable for direct processing through a homomorphic cryptosystem, since they are represented as real values. An integer valued watermarked feature is usually obtained as

$$\begin{aligned} z_i &= \lfloor f(\mathrm{x}, i) \cdot Q \rceil + b_i \cdot \lfloor \Delta(\mathrm{x}, i) \cdot Q \rceil \\ &= f_Q(\mathrm{x}, i) + b_i \cdot \Delta_Q(\mathrm{x}, i), \end{aligned} \tag{16}$$

where $\lfloor \cdot \rceil$ is the rounding function and Q is a scale factor that can be adjusted according to the required precision. It is worth noting that computing with the above representation is somewhat different than traditional fixed point arithmetic. Since secure division of encrypted values requires an interactive protocol, in computations relying only on the privacy homomorphism the scale factor Q accumulates after each multiplication, and in general particular care must be taken in choosing the right number of bits of the scale factor. Nevertheless, in the case of the schemes presented in the section "Secure Dither Modulation," experimental results show that in most cases 11–15 b are sufficient to obtain the same watermarking performance as a floating point implementation [13].

An alternative approach is to use integer valued features that can be obtained by using integer transforms [37]. In this case, however, the watermarking algorithm has to be modified such that it satisfies the integer constraint.

CLIENT-SIDE WATERMARK EMBEDDING

Client-side watermark embedding systems transmit the same encrypted version of the original content to all the clients, but a client-specific decryption key allows to decrypt the content and at the same time implicitly embed a watermark. When the client uses his/her key to decrypt the content, he/she obtains a uniquely watermarked version of the content. The security properties of the embedding scheme usually guarantee that obtaining either the watermark or the original content in the clear is of comparable hardness as removing the watermark from the personalized copy.

In the literature, several approaches for secure client-side embedding can be found. A particularly interesting approach is represented by methods using a stream-cipher that allows the use of multiple decryption keys, which decrypt the same ciphertext to slightly different plain texts. The difference between the original and the decrypted content represents the embedded watermark. The first scheme following this approach was proposed by Anderson et al. [3] who designed a special stream cipher, called Chameleon, which allows to decrypt Chameleon-encrypted content in slightly different ways. During encryption, a secure number generator, driven by a secret key, produces a sequence of indices, used to select four entries from a look-up table (LUT). These entries are XORed with the plain text to form a word of the ciphertext. The decryption process is identical to encryption except for the use of a decryption LUT, obtained by properly inserting bit errors in some entries of the encryption LUT. Decryption superimposes these errors onto the content, thus leaving a unique watermark.

Celik et al. [7] have proposed a generalization of Chameleon that operates on LUTs composed of real numbers and replace the XOR operation by an addition. The scheme is suitable for embedding robust spread spectrum watermarks. In addition, perceptual requirements can be taken into account, as demonstrated by a recent extension to audio watermarking

[16], and the scheme can be modified to handle joint decryption and watermarking on vector quantized data [23].

The secure LUT-based embedding solution works as follows. The distribution server generates a long-term master encryption LUT E of size L, whose entries are properly generated random samples; E will be used to encrypt the content to be distributed to all the clients. Next, for the kth client, the server generates a personalized watermark LUT W_k according to a desired probability distribution, and builds a personalized decryption LUT D_k by combining the master LUT and the watermark LUT

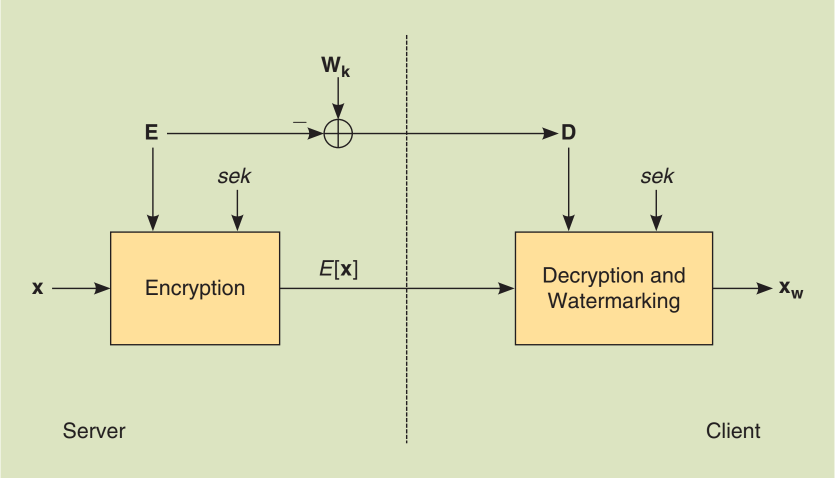
$$\mathrm{D}_k[i] = -\mathrm{E}[i] + \mathrm{W}_k[i]. \tag{17}$$

The personalized decryption LUTs are then transmitted once to each client over a secure channel. Let us note that the generation of the LUTs is carried out just once at the setup of the application. A content x is encrypted by adding to it a pseudorandom sequence obtained by selecting some entries of the LUT with a secure pseudorandom sequence generator driven by a session key *sek*. Each client receives the encrypted content $E[\mathrm{x}]$ along with the session key *sek* and decrypts it using some entries of his/her personalized decryption LUT D_k (again chosen according to *sek*), with the final effect that a spread-spectrum watermark sequence is embedded into the decrypted content. This process is summarized in Figure 3.

A FLEXIBLE SOLUTION FOR ZKWD IS TO COMPUTE THE WATERMARK DETECTION STATISTIC IN THE ENCRYPTED DOMAIN AND THEN USE ZERO-KNOWLEDGE PROOFS TO CONVINCE THE VERIFIER THAT THE DETECTION STATISTIC EXCEEDS A FIXED THRESHOLD.

In detail, driven by the session key *sek*, a set of indices t_{ij} is generated, where $0 \le i \le M-1$, $0 \le j \le S-1$, $0 \le t_{ij} \le L-1$. Each feature of the content x_i is encrypted by adding S entries of the encryption LUT, obtaining the encrypted feature $E[x_i]$ as follows:

$$E[x_i] = x_i + \sum_{j=0}^{S-1} \mathrm{E}[t_{ij}]. \tag{18}$$



[FIG3] Client-side secure watermark embedding.

Joint decryption and watermarking is accomplished by reconstructing with the session key *sek* the same set of indices t_{ij} and by adding S entries of the decryption LUT to each encrypted feature $E[x_i]$

$$x_{w,i} = E[x_i] + \sum_{j=0}^{S-1} \mathrm{D}[t_{ij}] = x_i + \sum_{j=0}^{S-1} \mathrm{W}[t_{ij}] = x_i + w_i. \tag{19}$$

1) Client-side DM: LUT-based client-side embedding can be also extended to DM techniques. The key observation is that these algorithms can be also viewed as watermarking schemes using syndrome coding, i.e., in which the information is coded in the residual error after quantization. Specifically, the marked feature of a QIM watermarking system can be modeled as

$$x_w = Q_0(x) + \theta_b, \tag{20}$$

where $\theta_b = b\Delta/2$ is a shift encoding the information bit b and can be considered as the error made after quantizing x_w with the quantizer Q_0, i.e., the information is encoded in the syndrome obtained after decoding x_w as an element of the codebook $\mathcal{U}_0$.

The syndrome coding approach is useful for client-side embedding, since it permits to separate the watermarked feature into a server-dependent part $Q_0(x)$, which depends on the cover content, and a client-dependent part θ_b, which depends on the information to be embedded.

Syndrome coding can be also used to generalize conventional DM by allowing arbitrary syndrome codewords θ_b, so as to provide an additional degree of freedom in the generation of the client-dependent part. For example, if a generic syndrome codeword can be expressed as the sum of S entries from a LUT, a simple client-side QIM embedding can be obtained by encrypting the original content as

$$E[x_i] = Q_0(x_i) + \sum_{j=0}^{S-1} \mathrm{E}[t_{ij}], \tag{21}$$

and the client can compute the watermarked features as

$$\begin{aligned} x_{w,i} &= E[x_i] + \sum_{j=0}^{S-1} \mathrm{D}[t_{ij}] \\ &= Q_0(x_i) + \sum_{j=0}^{S-1} \mathrm{W}[t_{ij}] = Q_0(x_i) + \theta_{b,i}. \end{aligned} \tag{22}$$

More sophisticated client-side embedding algorithms can be devised by applying the above principle along random projections of the original content, so as to increase the security, like the spread-transform DM [30].

SECURE WATERMARK DETECTION

To tackle the problem of watermark detection in the presence of an untrusted verifier (to whom watermark secrets cannot be disclosed), a possible solution offered by secure signal processing is represented by zero-knowledge watermark detection (ZKWD) that uses a cryptographic protocol to wrap a standard watermark detection process.

In general, a ZKWD algorithm is an interactive proof system where a prover tries to convince a verifier that a digital content $\mathbf{x}$ is watermarked with a given watermark $\mathbf{b}$ without disclosing $\mathbf{b}$. In contrast to the standard watermark detector, in ZKWD the verifier is given only properly encoded (or encrypted) versions of security-critical watermark parameters. Depending on the particular protocol, the watermark code, the watermarked object, a watermark key, or even the original unmarked object is available in an encrypted form to the verifier. The prover runs the zero-knowledge watermark detector to demonstrate to the verifier that the encoded watermark is present in the object in question, without removing the encoding. A protocol run will not leak any information except for the unencoded inputs and the watermark presence detection result.

[BUILDING A TTP-FREE ASYMMETRIC FINGERPRINTING PROTOCOL BASED ON CLIENT-SIDE EMBEDDING IS CURRENTLY AN OPEN ISSUE THAT DESERVES MORE RESEARCH.]

A flexible solution for ZKWD is to compute the watermark detection statistic in the encrypted domain (e.g., by using additive homomorphic public-key encryption schemes or commitments) and then use zero-knowledge proofs to convince the verifier that the detection statistic exceeds a fixed threshold. This approach was first proposed by Adelsbach and Sadeghi [2], who use a homomorphic commitment scheme to compute the detection statistic; their approach, later refined in [1], adopts a zero-knowledge protocol based on Cox's spread spectrum watermarking scheme.

In Cox's scheme [4], a set $\mathbf{x}$ of DCT coefficients is selected from the original image and a multiplicative watermark embedding rule is defined as follows:

$$x_{w,i} = x_i + \gamma w_i x_i = x_i(1 + \gamma w_i). \tag{23}$$

To determine if a given content contains the watermark $\mathbf{w}$, in case of blind detection, the decoder extracts the set $\mathbf{x}_w$ of DCT coefficients and then computes the correlation ρ between the features $\mathbf{x}_w$ and the watermark $\mathbf{w}$ for which we are looking. If the correlation is larger than a threshold T, then the watermark is considered present in the content. In [2], it is assumed that the watermark and DCT-coefficients are integers and not real numbers (this can be achieved by appropriate quantization). Moreover, for efficiency reasons the correlation-based detection criterion is expressed as

$$\begin{aligned}\rho &= (\mathbf{w}^T \mathbf{x}_w)^2 - \mathbf{x}_w{}^T \mathbf{x}_w \cdot T^2 \\ &= (A)^2 - B \geq 0;\end{aligned} \tag{24}$$

the above detection criterion is equivalent to a classical correlation detector with threshold T, provided that the factor A is positive.

The following zero-knowledge detection protocol has been designed to allow the prover to prove to a verifier that the watermark committed to in $C[\mathbf{w}]$ is present in the watermarked content $\mathbf{x}_w$, without revealing any information about $\mathbf{w}$. In the protocol, an additively homomorphic commitment scheme (such as the one proposed by Damgård and Fujisaki [12]) is used. Let $\mathbf{x}_w, C[\mathbf{w}], T$ be the common inputs of prover and verifier and let us assume that the commitment can be opened by the prover. First, both the prover and verifier select the watermarked features $\mathbf{x}_w$ and compute the value B of (24); the prover sends a commitment $C[B]$ to the verifier and opens it immediately, allowing him to verify that the opened commitment contains the same value B he computed himself. Now both compute the commitment

$$C[A] = \prod_{i=1}^{M} C[w_i]^{x_{w,i}}, \tag{25}$$

by taking advantage of the homomorphic property of the commitment scheme. Subsequently the prover proves in zero-knowledge that $A \geq 0$. Next, the prover computes the value A^2, sends a commitment $C[A^2]$ to the verifier, and gives him a zero-knowledge proof that it really contains the square of the value contained in $C[A]$. Being convinced that $C[A^2]$ really contains the correctly computed value A^2, the two parties compute the commitment $C[\rho] = C[A^2] \cdot C[B]^{-1}$ on the value ρ. Finally the prover proves to the verifier, with a proper zero-knowledge protocol, that $C[\rho] \geq 0$. If this proof is accepted then the detection algorithm ends with true, otherwise with false.

While early protocols addressed only correlation-based watermark detectors, the approach has been also extended to Gaussian maximum likelihood detectors [36] and DM watermarks [31], [25].

CHALLENGES AND NEW DIRECTIONS

In the previous sections, we have illustrated some well-established approaches, combining secure signal processing and watermarking, which can be used for multimedia content protection. Although every approach provides a valuable tool for solving a specific problem, there are still some issues to be dealt with to make the above solutions appealing in real-life applications. In the following, we will present the main challenges in this area and we will discuss possible solutions and new research directions.

Server-side embedding: Current solutions based on homomorphic encryption offer provable security at the price of a very high complexity [34]. Here, the bottleneck is the secure embedding module, since all watermarked features have to be encrypted using a costly homomorphic cryptosystem. As an example, in [13] it is reported that on a $1,024 \times 1,024$ image secure embedding takes about two minutes using a standard personal computer.

Apart from the foreseeable evolution of the hardware equipment or advancements in homomorphic encryption, an appealing solution from a signal processing point of view could be combining these schemes with partial encryption techniques, which are often employed in video encryption [35]. In closely related fields, partial encryption has been employed in secure client-side watermarking [22] and as a means for implementing commutative watermarking and encryption [6]. The rationale behind such an approach is that signals are fuzzy entities, which do not require complete protection, so that we can trade off security for a better efficiency.

Client-side asymmetric fingerprinting: Although client-side embedding provides an elegant solution to the system scalability problem, the incorporation of the aforementioned technique in an asymmetric fingerprinting protocol does not appear an easy task. Here, the main problem is that the watermarking LUT should not be revealed to the server. At the same time, the client should not have access to the watermarking LUT, since the knowledge of both decryption and watermarking LUTs will immediately disclose the encryption LUT as $\mathrm{E}[i] = \mathrm{W}_k[i] - \mathrm{D}_k[i]$.

Current solutions propose using a TTP to manage both encryption and watermarking LUTs [17], [32], however such a TTP can become quickly overloaded in a realistic system. Building a TTP-free asymmetric fingerprinting protocol based on client-side embedding is currently an open issue that deserves more research. From a signal processing point of view, an interesting question is whether the LUT framework can be exploited to reliably convey a binary vector $\mathbf{b}$, since this will enable the use of existing asymmetric fingerprint protocols.

Collusion resistance: A common problem of fingerprinting is that several clients may collude and try to remove the fingerprint by comparing the respective watermarked copies. Collusion resistance can be achieved by using specific anticollusion codes in the design of the fingerprint. However, merging collusion resistant techniques and secure embedding is in general a difficult task.

As to client-side embedding, a natural solution is to design the watermarking LUTs so that they produce a specific anticollusion code for each client [18]. Nevertheless, the above strategy still suffers from the fact that watermarking LUTs should be managed by a TTP.

In secure server-side embedding, the fingerprint depends on private inputs from the buyer, so that it is not easy to enforce the use of specific anticollusion codes. A recently proposed solution consists in letting the buyer pick up fingerprint elements from a list controlled by the seller, in such a way that the seller does not know the chosen elements [8]. In our opinion, this is a promising research direction, where the interaction between cryptographic tools and specific anticollusion code designs appears fundamental.

ZKWD: One of the main concerns is that zero-knowledge proofs do not provide protection against blind sensitivity attacks [10], but they can only slow the efficiency of this kind of attacks. Moreover, current solutions are still very complex for real life deployment and often made scarcely appealing by the fact that detectors can be implemented within secure environments. It is also worth noting that research on this particular subject has stalled in the last five years, probably due to the difficulty of bringing together the required expertise in both signal processing and cryptography. Because of the above limits, this is the research area on secure watermarking where further advancements appear more difficult.

DIGITAL WATERMARKING HAS BEEN PROPOSED AS A POSSIBLE BRICK OF MULTIMEDIA CONTENT PROTECTION SYSTEMS, BUT ITS APPLICATION IN REALISTIC SCENARIOS HAS RAISED SEVERAL ISSUES.

Apart from the specific challenges cited above, secure watermarking techniques may also benefit from specific cryptographic advancements, like the design of cryptographic tools offering both additive and multiplicative homomorphism [14]: with such schemes from the encrypted content $E[x]$ it would be possible to compute the ciphertext of any polynomial function of x, solving any secure computation problem. Fully homomorphic encryption schemes are still computationally very expensive, but their use in practical applications could become feasible in the future.

CONCLUSIONS

Digital watermarking has been proposed as a possible brick of multimedia content protection systems, but its application in realistic scenarios has raised several issues. Secure watermarking has then been designed to solve some of these issues. In this article, we reviewed three classes of secure watermarking that have received particular attention in the literature: secure server-side embedding, introduced to solve the customer's rights problem, secure client-side embedding, proposed to work out the system scalability problem, and secure watermark detection, suggested to take into account the presence of untrusted watermark verifiers.

Even if this is a field that has reached a certain degree of maturity, so that some areas like secure watermark detection witness little activity in recent years, we showed that there are still some interesting challenges for which to be dealt. An important observation is that currently available tools do not provide yet a complete solution, simultaneously solving all of the above problems. Since secure signal processing is a highly interdisciplinary area of research needing contributions from both the signal processing and the cryptographic communities, it is hard to imagine that a workable solution might be provided by the signal processing community alone. Nevertheless, we believe that some challenges may be very interesting for the signal processing community, like merging selective encryption and secure watermarking and developing collusion resistant solutions for secure watermarking. As a final comment, we deem that there is still an open space for research in this interdisciplinary and interesting research area.

AUTHORS

Tiziano Bianchi (tiziano.bianchi@unifi.it) received the M.Sc. degree (Laurea) in electronic engineering and the Ph.D. degree in information and telecommunication engineering from the University of Florence, Italy, in 2001 and 2005, respectively. Since 2005, he has been a research assistant with the Department of Electronics and Telecommunications, University of Florence. His research interests have involved the processing of SAR images, signal processing in communications, multicarrier modulation techniques, and ultrawideband systems. His current research topics include multimedia security technologies and signal processing in the encrypted domain. He has published more than 80 papers on international journals and conference proceedings.

Alessandro Piva (alessandro.piva@unifi.it) received the Ph.D. degree in computer science and telecommunications engineering from the University of Florence, Italy, in 1999. From 2002 to 2004, he was a research scientist at the National Interuniversity Consortium for Telecommunications. He is currently an assistant professor at the University of Florence. His research interests lie in the areas of information forensics and security and image and video processing. He is the coauthor of more than 100 papers published on international journals and conference proceedings. He holds three Italian patents and an international patent regarding watermarking. He is a member of the IEEE Information Forensics and Security Technical Committee. He is an associate editor of *IEEE Transactions on Information Forensics and Security, IEEE Transactions on Circuits and Systems for Video Technology, EURASIP Journal on Information Security*, and *LNCS Transactions on Data Hiding and Multimedia Security*. He is a Senior Member of the IEEE.

REFERENCES

[1] A. Adelsbach, M. Rohe, and A.-R. Sadeghi, “Non-interactive watermark detection for a correlation-based watermarking scheme,” in *Communications and Multimedia Security* (Lecture Notes in Computer Science, vol. 3677). Berlin/Heidelberg: Springer, 2005, pp. 129–139.

[2] A. Adelsbach and A.-R. Sadeghi, “Zero-knowledge watermark detection and proof of ownership,” in *Proc. 4th Int. Workshop Information Hiding LNCS* 2137, pp. 273–288.

[3] R. J. Anderson and C. Manifavas, “Chameleon: a new kind of stream cipher,” in *Proc. 4th Int. Workshop Fast Software Encryption* (*FSE’97*), London, 1997, pp. 107–113.

[4] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. New York: Marcel Dekker, 2004.

[5] T. Bianchi, A. Piva, and M. Barni, “Composite signal representation for fast and storage-efficient processing of encrypted signals,” *IEEE Trans. Inform. Forensics Sec.*, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[6] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. D. Natale, and A. Neri, “A commutative digital image watermarking and encryption method in the tree structured Haar transform domain,” *Signal Process. Image Commun.*, vol. 26, no. 1, pp. 1–12, 2011.

[7] M. Celik, A. Lemma, S. Katzenbeisser, and M. van der Veen, “Look-up table based secure client-side embedding for spread-spectrum watermarks,” *IEEE Trans. Inform. Forensics Sec.*, vol. 3, no. 3, pp. 475–487, 2008.

[8] A. Charpentier, C. Fontaine, T. Furon, and I. Cox, “An asymmetric fingerprinting scheme based on Tardos codes,” in *Proc. 13th Int. Conf. Information Hiding, IH’11*, Berlin, Heidelberg, 2011, pp. 43–58.

[9] B. Chen and G. Wornell, “Quantization index modulation: A class of provably good methods for digital watermarking and information embedding,” *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[10] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, “Blind Newton sensitivity attack,” *IEE Proc. Inform. Sec.*, vol. 153, no. 3, pp. 115, 2006.

[11] I. J. Cox, M. L. Miller, and A. L. McKellips, “Watermarking as communications with side information,” *Proc. IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[12] I. Damgård and E. Fujisaki, “A statistically-hiding integer commitment scheme based on groups with hidden order,” in *Proc. Advances in Cryptology—ASIACRYPT’02, LNCS* 2501, 2002, pp. 125–142.

[13] M. Deng, T. Bianchi, A. Piva, and B. Preneel, “An efficient buyer-seller watermarking protocol based on composite signal representation,” in *Proc. 11th ACM Workshop Multimedia and Security*, Princeton, NJ, 2009, pp. 9–18.

[14] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proc. 41st Annu. ACM Symp. Theory Computing* (*STOC ‘09*), New York, 2009, pp. 169–178.

[15] O. Goldreich, *Foundations of Cryptography I*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[16] J.-J. Jiang and C.-M. Pun, “Secure client-side digital watermarking using optimal key selection,” in *Communication and Networking* (Communications in Computer and Information Science, vol. 266), T.-H. Kim, H. Adeli, W.-C. Fang, T. Vasilakos, A. Stoica, C. Z. Patrikakis, G. Zhao, J. G. Villalba, and Y. Xiao, Eds. Berlin/Heidelberg, Germany: Springer, 2011, pp. 162–168.

[17] 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. Inform. Forensics Sec.*, vol. 3, no. 4, pp. 783–786, Dec. 2008.

[18] S. Katzenbeisser, B. Škori, M. Celik, and A.-R. Sadeghi, “Combining Tardos fingerprinting codes and fingercasting,” in *Information Hiding* (Lecture Notes in Computer Science, vol. 4567), T. Furon, F. Cayre, G. Doërr, and P. Bas, Eds. Berlin/Heidelberg, Germany: Springer, 2007, pp. 294–310.

[19] M. Kuribayashi, “On the implementation of spread spectrum fingerprinting in asymmetric cryptographic protocol, *EURASIP J. Inform. Sec.*, vol. 2010, pp. 1:1–1:11, Jan. 2010.

[20] M. Kuribayashi and H. Tanaka, “Fingerprinting protocol for images based on additive homomorphic property,” *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2129–2139, Dec. 2005.

[21] R. L. Lagendijk, Z. Erkin, and M. Barni, “Encrypted signal processing for privacy protection,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 82–105, 2013.

[22] A. Lemma, S. Katzenbeisser, M. Celik, and M. van der Veen, “Secure watermark embedding through partial encryption,” in *Proc. 5th Int. Conf. Digital Watermarking, IWDW’06*, Berlin/Heidelberg, Germany, 2006, pp. 433–445.

[23] C.-Y. Lin, P. Prangjarote, L.-W. Kang, W.-L. Huang, and T.-H. Chen, “Joint fingerprinting and decryption with noise-resistant for vector quantization images,” *Signal Process.*, vol. 92, no. 9, pp. 2159–2171, 2012.

[24] W. Lin, H. Zhao, and K. Liu, “Game-theoretic strategies and equilibriums in multimedia fingerprinting social networks,” *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 191–205, Apr. 2011.

[25] M. Malkin and T. Kalker, “A cryptographic method for secure watermark detection,” in *Proc. 8th Int. Workshop on Information Hiding, IH’06*, Old Town Alexandria, VA, 10–12 July 2006.

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

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

[28] 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. Signal Process.* vol. 53, no. 10, pp. 3960–3975, Oct. 2005.

[29] B. Pfitzmann and M. Schunter, “Asymmetric fingerprinting,” in *Advances in Cryptology—EUROCRYPT’96* (Lecture Notes in Computer Science, no. 1070). Berlin/Heidelberg: Springer, 1996, pp. 84–95.

[30] A. Piva, T. Bianchi, and A. De Rosa, “Secure client-side ST-DM watermark embedding,” *IEEE Trans. Inform. Forensics Sec.*, vol. 5, no. 1, pp. 13–26, Mar. 2010.

[31] A. Piva, V. Cappellini, D. Corazzi, A. D. Rosa, C. Orlandi, and M. Barni, “Zero-knowledge ST-DM watermarking,” in *Proc. Security, Steganography, and Watermarking of Multimedia Contents VIII*, 2006.

[32] G. Poh and K. Martin, “An efficient buyer–seller watermarking protocol based on chameleon encryption,” in *Digital Watermarking* (Lecture Notes in Computer Science, vol. 5450), H.-J. Kim, S. Katzenbeisser, and A. Ho, Eds., pp. 433–447. Berlin/Heidelberg, Germany: Springer, 2009.

[33] J. P. Prins, Z. Erkin, and R. L. Lagendijk, “Anonymous fingerprinting with robust QIM watermarking techniques,” *EURASIP J. Inform. Sec.*, vol. 2007, pp. 13, 2007.

[34] A. Rial, M. Deng, T. Bianchi, A. Piva, and B. Preneel, “A provably secure anonymous buyer–seller watermarking protocol,” *IEEE Trans. Inform. Forensics Sec.*, vol. 5, no. 4, pp. 920–931, Dec. 2010.

[35] T. Stutz and A. Uhl, “A survey of H.264 AVC/SVC encryption,” *IEEE Trans. Circuits Syst Video Technol*, vol. 22, no. 3, pp. 325–339, Mar. 2012.

[36] J. R. Troncoso-Pastoriza and F. Pérez-Gonzalez, “Zero-knowledge watermark detector robust to sensitivity attacks,” in *Proc. ACM Multimedia Security Workshop*, 2006, pp. 97–107.

[37] P. Zheng and J. Huang, “Walsh-Hadamard transform in the homomorphic encrypted domain and its application in image watermarking,” in *Proc. 14th Information Hiding Conf.*, 2012.