Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Signal Processing 83 (2003) 2133–2170

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Secure hybrid robust watermarking resistant against tampering and copy attack

F. Deguillaume*, S. Voloshynovskiy, T. Pun

Department of Computer Science, University of Geneva, CUI, 24 rue du Général Dufour, CH-1211 Geneva 4, Switzerland

Received 10 November 2002; received in revised form 14 May 2003

Abstract

Digital watermarking appears today as an efficient mean of securing multimedia documents. Several application scenarios in the security of digital watermarking have been pointed out, each of them with different requirements. The three main identified scenarios are: copyright protection, i.e. protecting ownership and usage rights; tamper proofing, aiming at detecting malicious modifications; and authentication, the purpose of which is to check the authenticity of the originator of a document. While robust watermarks, which survive to any change or alteration of the protected documents, are typically used for copyright protection, tamper proofing and authentication generally require fragile or semi-fragile watermarks in order to detect modified or faked documents. Further, most of robust watermarking schemes are vulnerable to the so-called copy attack, where a watermark can be copied from one document to another by any unauthorized person, making these schemes inefficient in all authentication applications. In this paper, we propose a hybrid watermarking method joining a robust and a fragile or semi-fragile watermark, and thus combining copyright protection and tamper proofing. As a result this approach is at the same time resistant against copy attack. In addition, the fragile information is inserted in a way which preserves robustness and reliability of the robust part. The numerous tests and the results obtained according to the Stirmark benchmark demonstrate the superior performance of the proposed approach.

Keywords: Hybrid watermarking; Tamper proofing; Authentication; Copyright protection; Copy attack; Substitution attacks

1. Introduction

These last years, the rapidly growing multimedia market and use of digital technologies in general has revealed an urgent need for securing documents. Numerous threats have been identified yet, but one of the first to be pointed out was the incredible ease with which *exact* copies could be done without any authorization. Classical protection such as cryptography appeared soon not to be a solution, since once a document has been decrypted, even by an authorized customer, this latter can always distribute it in plain form without any restriction. Therefore, more sophisticated document security methods have been proposed, first aiming at solving the *copyright protection* problem based on watermarking technologies.

1.1. Copyright protection

The main requirements for copyright-protection watermarking algorithms are *robustness* (denoting

* Corresponding author. Tel. +41-22-705-7671; fax: +41-22-705-7780.
E-mail addresses: frederic.deguillaume@cui.unige.ch (F. Deguillaume), svolos@cui.unige.ch (S. Voloshynovskiy), thierry.pun@cui.unige.ch (T. Pun).
URL: http://watermarking.unige.ch/

0165-1684/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0165-1684(03)00172-5

how the watermark can survive any kind of malicious or unintentional transformations), *visibility* (does the watermark introduce perceptible artifacts), and *capacity* (the amount of information which can be reliably hidden and extracted from the document). For copyright applications, robustness should be as high as possible, and visibility as low as possible in order to preserve the value of the marked document. Note, however, that capacity can be low since copyright information generally requires a rather small amount of information, which can be an index inside a database holding copyright information. Other requirements can be outlined, which are: *security* (from the cryptographic point of view), and that the scheme should be *oblivious* (the original or *cover* image is not needed for the extraction process).

Many robust watermarking schemes have been proposed, consisting in either spatial domain or transform domain watermarks. The main issue addressed for these schemes these last years is the robustness of watermarks against various intentional or unintentional alterations, degradations, geometrical distortions or media-conversion which can be applied to the watermarked (*stego*) image. The four main issues of the state-of-the-art watermark robustness are described in more details in Voloshynovskiy et al. work [54,55] and can be summarized as follows:

(1) *Host interference cancellation*: In order to ensure low visibility, the energy of the watermark is usually drastically lower than the energy of the data to which it is embedded (the *host*), leading to strong interference. Host interference cancellation can then be performed: either at the encoder side by precoding or structured codes such as the Quantization Index Modulation (QIM) [8], the Scalar Costa Scheme (SCS) [11], or the Dither Modulation (DM) [7] in the case of printer-generated documents; or at the decoder side by the *Estimation-Subtraction* (ES) approach which consists of the robust estimation of the embedded watermark from the stego image. The ES approach is used by our robust watermarking scheme [58], and is given in more details in Section 2 of this paper.
(2) *Intersymbol interference cancellation*: The various attacks applied to the stego data introduce filtering which results into the interference between the symbols used for the encoding of the watermark. The cancellation of this intersymbol interference then consists of the inversion of the filtering or the equalization of the attacking channel. Some methods are the Tomlinson–Harashima Precoder [21,51] (THP) at the encoder side, or the Decision-Feedback Equalization (DFE) or the Turbo Equalization [52] at the decoder side.
(3) *Channel state estimation*: Due to various attacks the watermark can also suffer fading, which is not necessarily uniform. Techniques such as the embedding of a reference signal and diversity techniques (Section 2) can be employed, in order to estimate the channel variations resulting from the attacks and to invert them at the decoder side.
(4) *Geometrical synchronization*: Geometrical distortions applied to a stego image resynchronize the embedded signal and make it unreadable. Methods are then needed to compensate these distortions and to resynchronize the signal at the decoder.

Solutions against geometrical transform can use either a transform invariant domain watermark like the Fourier–Mellin transform [45], or an additional template for resynchronization [46], or a *self-reference* watermark based on the Auto Correlation Function (ACF) of a repetitive watermark [30]. Self-reference watermarks have been shown to have as main advantage over other methods the fact that they exploit the redundancy of the regular structure of the watermark in order to robustly estimate the undergone geometrical distortions. We previously proposed a method based on this concept, which is robust to general affine transforms [10,58] as well as to non-linear distortions and to the Random Bending Attack (RBA) [59]; our approach uses the ACF or magnitude spectrum of a periodical watermark, at the global level to recover from affine transforms, and at the local level to recover from the RBA.

1.2. Tamper proofing and authentication

Other important threats have been recently identified with respect to multimedia document, the most important of them being the ease offered by today technologies for tampering or counterfeiting. Digital cameras are constantly growing in quality while

becoming widely available, and software such as Jasc Software Paint Shop Pro™ or Adobe Photoshop™ make it very easy to perform complex modifications without visible artifact. Although this is useful for artistic applications, this is a serious problem for legal applications such as evidences in trials, for healthcare insurances in medical imaging, for counterfeiting, etc. Classical analysis techniques used for authenticating analog photographs are ineffective. Another important issue is the ability to authenticate the originator of a visual document.

Of course global cryptographic signatures can detect tampering and authenticate documents, but are unable either to highlight which areas have been modified, or to assess the severity of the alteration; moreover, format conversion kills this meta-data. Such a global authentication has been proposed by Friedman in his trusted digital camera [19]. Therefore, one proposed solution to both tamper proofing and authentication is again watermarking, which is used here to attach check-codes of local areas inside the image itself, in order to achieve the ability to localize altered regions. Such watermarks do not need the same level of robustness as for copyright protection, since in case of removal or cancellation the image can just be considered as non authentic. Two cases can be distinguished: the watermark can be either *fragile*, meaning that any modification, even a limited change of a small set of pixels, is detected, or *semi-fragile*, offering a level of tolerance to some "acceptable" alterations such as low-level lossy compression or slight contrast adjustment.

Fragile or semi-fragile tamper proofing/authentication watermarking schemes divide the image into local areas, compute a key-dependent function from each of them, and embed the results into the image itself. Usually the function results are stored into their corresponding areas, and these areas can be as small as single pixels. Yeung and Mintzer [68] proposed a pixel-wise method, known as the Yeung–Mintzer scheme, which works as follows: a key-dependent function is used to map the gray-scale value of each pixel from 0 to 255 to a binary value, either 0 or 1. For color images, three such functions are used, one for each color channel, and the outputs of the three function are combined together by exclusive-or (XOR). The gray-scale (or color) values of the image are altered in order to get a specific binary logo, when the image is used as input of this function. Both the key and the logo should be kept secret. The verification takes place by recomputing the binary logo from the possibly altered image, and the comparison between the recomputed logo and the original one gives a map of modified areas. The main advantage of a pixel-wise approach is its degree of locality, which is the highest possible. Moreover, the Yeung–Mintzer scheme is fast and simple, making it well suited for hardware implementation.

However pixel-wise approaches present security weaknesses, mostly resulting from the limited number of discrete values that a pixel can take, making block-wise approaches a better solution. In a block-wise scheme the image is first divided into blocks, a key-dependent hash function is applied to each of them, and the obtained hash-codes are embedded into their corresponding blocks—generally by replacing the least significant bits (LSB) of pixels for fragile schemes. The key is kept secret, and unauthorized changes are then detected where the recomputed codes do not match the stored codes. Wong [64] proposed such a block-wise approach, which was then improved to enhance its security against specifics attacks by Coppersmith et al. [9]. Other state-of-the-art fragile watermarking schemes are Celik et al. scheme [6], an interesting hierarchical approach which can recursively authenticate blocks and then sub-blocks, and the earlier Walton scheme [62], based on encrypted checksums using a secret key.

At the opposite, semi-fragile watermarks are more tolerant, and can be even used to measure the severity of the alteration; robust watermarking has been proposed for tamper proofing or authentication, however this approach is insecure since robust watermarks are usually additive, making them vulnerable to the so-called *copy attack* described by Kutter et al.: the signal can be easily estimated using denoising techniques and copied to another image [31]. One solution to resist the copy attack is to make the watermark somehow dependent on the image content or more generally unpredictable. One possibility is to compute *robust* hashes which are tolerant to slight modifications, and to embed them robustly (such concept of embedding robust image-dependent information into the same image is called *fingerprinting*). Existing semi-fragile algorithms are Kundur and Hatzinakos telltale tamper proofing and authentication [29],

embedding a watermark in the wavelet domain which characterizes the altered frequencies, Lin et al. method [34] based on the embedding of pseudo-random spread-spectrum signals into the Discrete Cosine Transform (DCT) blocks of JPEG encoded images, and Lin an Chang scheme [33] based on the encoding of invariant features from DCT coefficient. Another fragile scheme is Wolfgang and Delp Variable Watermark Two-Dimensional (VW2D) algorithm [63].

We can also mention self-embedding watermarks where a lower resolution version of the visual content is embedded into the image itself; Wu and Liu [66] propose such a scheme which embeds the visual content using the look-up table (LUT) of the frequency domain coefficients, and Fridrich [15] proposes to embed the visual content in the bit representation of chosen DCT coefficients. Self-embedding watermarks not only can detect tampered areas by locally analyzing mismatches between the stego image and the actually extracted visual information, but can even reconstruct these areas.

1.3. A hybrid solution

While robust watermarks are typically required for copyright protection, fragile or semi-fragile watermarks have been proposed to solve tamper proofing and authentication. Watermarking methods above are either robust schemes, or fragile/semi-fragile schemes; however approaches combining both for copyright and tamper proofing/authentication applications are rarely proposed. Fridrich [12] proposed such a hybrid method, but uses a watermark with relatively low robustness. Joining robust and fragile/semi-fragile watermarks have two main potential advantages.

First we mentioned that most of robust watermarking schemes are vulnerable to the copy attack [31]. While most of studied attacks against robust watermarking aim at removing the watermark or making it unreadable, one could believe that copying a wrong watermark into another document is useless to the attacker. This is not true, since the copy attack puts in question the link between the cover data and the embedded information. By creating this ambiguity about the validity of the watermark, the copy attack falls in the class of *protocol attacks*. Consequently, such a robust watermark cannot be used for the authentication of a document (for example a passport), since the attacker is able to produce a fake document, and then copy the watermark information from an authentic one. Even in the case of copyright protection or data monitoring applications, the user cannot be sure that the extracted information really belongs to the document. This makes the watermark useless in many applications.

Secondly, due to their localization functionality based on independent areas, most of tamper proofing/authentication watermarking schemes fail to detect *substitution attacks*. These attacks consist of pasting parts or blocks which have been copied either from the same image or from other images already watermarked with the same key, under certain synchronization conditions. The fabrication of completely arbitrary images are even possible without being detected when a large number of watermarked images are available, based on vector quantization techniques or cryptographic implementation weaknesses. Moreover, even if the scheme has been designed to resist the attacks above, images compositions are still possible where rather large areas (from a small number of source images) are copied: we call such a composition *collage attack*. In this case only boundaries between the zones from different origins are detected as tampered, leading to the ambiguity about the authenticity of the zones in the context of the composed image: do these zones come from different sources, or did local tampering just take place? Consequently due to this ambiguity the collage attack fall again in the protocol attacks.

Therefore, joining robust and fragile/semi-fragile algorithms can help in the design of a watermarking algorithm which resists against the two types of protocol attacks above. To this extent no real-working scheme for secure hybrid robust watermarking, tamper proofing and authentication has been proposed yet. Consequently we propose: firstly, to join a highly robust watermark described in Section 2, with a fragile block-wise watermark for combined copyright protection, tamper proofing and authentication given in Section 3; secondly, a smart embedding of the fragile part which fully preserves the robust watermark, as described in the same section; thirdly, an extension of this scheme to a joint robust and semi-fragile watermark in Section 5. Section 4 outlines the cryptography and security aspects of image hash-coding and signatures with localization capabilities, and

summarizes possible solutions to defeat substitution attacks. Section 6 briefly discusses some practical scenarios of hybrid watermarking, shows the ability for tamper detection of our approach and demonstrates its resistance against the mentioned protocol attacks—the copy and the collage attacks.

2. Robust watermarking

The robust watermarking scheme we developed is a content adaptive multiresolution algorithm with channel state estimation, exploiting a self-reference watermark in order to resist against geometrical transformations. The principles of this technology are explained in more details in our previous publications [10,58–60], and is implemented in the prototype known as *Berkut 1.0* [38].

2.1. Watermark encoding and embedding

The robust algorithm we propose consists in the novel use of three components: a self-reference watermark for the recovering and the compensation of global affine geometrical distortions, a specially designed method for the recovering from local or non-linear geometrical transforms, and a signal in addition to the watermark carrying the message as side information for the extraction process.

The robust watermark $\mathbf{w}$ to embed thus consists of two components: first, a part which carries the useful information, usually the copyright message $\mathbf{b}$; secondly, a key-dependent reference component (depending on a secret key $\mathbf{k}$) carrying additional information which is used as a pilot during the extraction and decoding stage to estimate the characteristic of the embedding channel [57], as well as for synchronization purpose. We will use the term of *informative watermark* to refer to the part $\mathbf{w}_{\text{inf}}$ carrying $\mathbf{b}$, and of *reference watermark* for the reference part $\mathbf{w}_{\text{ref}}$ (therefore $\mathbf{w} = \mathbf{w}_{\text{inf}} + \mathbf{w}_{\text{ref}}$). The informative and reference parts however are orthogonal with respect to each other, i.e. they are embedded into non-overlapping positions in the host image.

2.1.1. Message encoding

The message, a bit string $\mathbf{b} = (b_1, \ldots, b_L)^{\mathrm{T}}$ with $b_i \in \{0, 1\}$, $i = 1, \ldots, L$, is first encoded in a (usually longer) bit string using some error correction codes (ECC). The bit string is then encrypted using a key-dependent primitive (based on the secret key $\mathbf{k}$) and mapped from $\{0, 1\}$ to $\{-1, 1\}$, resulting into a codeword $\mathbf{c} = (c_1, \ldots, c_K)^{\mathrm{T}}$, with $c_i \in \{-1, 1\}$, $i = 1, \ldots, K$, followed by a spreading over a rectangular or square block of size $t_1 \times t_2$ with some density D_{inf}. The remaining positions are reserved for the reference watermark, with a density of $D_{\text{ref}} \leqslant 1 - D_{\text{inf}}$, as a pseudo-random bit string depending on $\mathbf{k}$ and also mapped to $\{-1, 1\}$; if $D_{\text{ref}} < 1 - D_{\text{inf}}$, then free positions still remain which contain no information at all and are represented in this encoding as 0's. The embedding positions are still allocated based on the key $\mathbf{k}$.

The message is encoded using ECC with soft decoding [23] such as Turbo codes [3] or the Low Density Parity Check Codes (LDPC) [20], that belongs to the iterative coding codes, and which we tested at rates $r_{\text{ecc}} = \frac{1}{2}$ or $r_{\text{ecc}} = \frac{1}{3}$; the lower is the rate, the more robust to distortions is the codeword. Such ECC encoding achieves significantly superior performance than binary modulation or any other codes based on hard decoding, like the Bose–Chaudhuri–Hocquenghem (BCH) cyclic codes which are often used in the watermarking community. The selection of binary encoding is explained by the low watermark-to-noise ratio (WNR) typical for the robust watermarking operations regime. In this case, binary constellations practically approach the channel capacity. In the following we will use only the Turbo codes.

In our implementation, for the informative part we encode a 64 bit message plus additional bits used for assessing the reliability of the decoding (check code). The used rate is $r_{\text{ecc}} = \frac{1}{3}$, leading to a codeword of length $K \simeq 250$. The reference part comprises $N \simeq 60$ bits. The resulting $K + N \simeq 310$ bits are then allocated in a block of size $t_1 \times t_2 = 19 \times 19$ pixels.

Afterwards the $t_1 \times t_2$ watermark block is upsampled by a factor of 2 to receive a low-pass watermark and then flipped and copied once in each direction, producing a symmetrical block of final size $4t_1 \times 4t_2$. Finally, the latter $4t_1 \times 4t_2$ block is repeated to cover the whole image size, resulting into a *symmetrical and periodical* watermark with periods $T_1 = 4t_1$ and $T_2 = 4t_2$. In our implementation we have $T_1 = T_2 = 76$. This symmetrical block, which we will call $\mathbf{w}_p$, can be seen as a watermark with not all null values

belonging to $\{-1,0,1\}$ (according to the message encoding) inside the coordinates domain $[0,T_1-1]\times[0,T_2-1]$, and zero elsewhere. The resulting watermark $\mathbf{w}$ can then be written as

$$\mathbf{w}(n_1,n_2)=\sum_{m=0}^{(M_1/T_1)-1}\sum_{n=0}^{(M_2/T_2)-1}\mathbf{w}_p((n_1,n_2)^{\mathrm{T}}-(m\cdot T_1,n\cdot T_2)^{\mathrm{T}})^{\mathrm{T}}, \quad (1)$$

where $\mathbf{w}_p$ are the $T_1\times T_2$ individual upsampled and flipped blocks, M_1,M_2 the image size in pixels (width and height), and n_1,n_2 the individual pixels coordinates. If T_1 or T_2 are not exact multiples of M_1 or M_2, respectively, then the upper integer bounds of M_1/T_1 and M_2/T_2 can be taken in Eq. (1) in order to cover the whole image, and the final stego image just cropped to its original size.

Since the reference watermark is inserted close to the positions of the informative watermark, it can give additional information: for the determination of the watermark presence or absence for the given key; as a pilot for the estimation of a channel state for the optimal design of a matched filter in the decoder for the informative watermark; for the evaluation of the reliability of the local and global geometrical transforms recovering; for the estimation of the reliability of the decoding of the informative watermark. The informative watermark itself, organized in a special spatial structure, can be used for the last purpose as well. Note that the watermark is not restricted to be of square shape, but can also be of any regular or irregular shape that is then replicated in a special manner (not necessary strictly periodical) over the image and can be predistorted to avoid removal attacks based on the exploitation of the periodical watermark structure.

2.1.2. Watermark embedding

To embed the resulting robust watermark $\mathbf{w}$ in a cover image $\mathbf{x}$ a linear additive scheme is used in the wavelet domain. Although this scheme is known to suffer from the host interference, we will exploit further the channel state estimation at the decoder to compensate for the influence of the host signal. Moreover, in the low WNR regime, this influence is not crucial to reach the channel capacity [61]. Both the cover image and the watermark are first decomposed into a multiresolution pyramid using the Discrete (critically sampled) Wavelet Transform (DWT), resulting into $\tilde{\mathbf{x}}$ for the image, and $\tilde{\mathbf{w}}$ for the watermark. $N_{\mathrm{wt}}=5$ resolution levels (indexed by k) are used for the DWT based on the Daubechies 8-tap filter. Biorthogonal and over-complete expansions can be used for this purpose as well and even with better success for many reasons that are out of the scope of this paper. Moreover, each resolution sub-band has three components corresponding to distinct *orientations* (indexed by l) for the vertical, the horizontal, and the diagonal directions respectively, except for the lowest resolution $k=N_{\mathrm{wt}}$ which consists of only one low-pass component. The watermarking process is applied and adapted to each k,l sub-band component separately. Finally, the stego image is reconstructed by computing the inverse DWT of the marked image pyramid.

Perceptual masking is modeled based on a Noise Visibility Function (NVF), computed for each pixel (n_1,n_2) of each sub-band component $\tilde{\mathbf{x}}_{k,l}$ of the image's DWT, which is expressed as

$$\mathrm{NVF}_{k,l}(n_1,n_2)=\frac{\tilde{\mathbf{x}}_{k,l}(n_1,n_2)}{\tilde{\mathbf{x}}_{k,l}(n_1,n_2)+\sigma^2_{\tilde{\mathbf{x}}_{k,l}}}. \quad (2)$$

The NVF is based on a Stationary Generalized Gaussian (SGG) model proposed by Voloshynovskiy et al. [60]. $\sigma^2_{\tilde{\mathbf{x}}_{k,l}}$ is the global variance of the wavelet image coefficients from the sub-band (k,l), and $\tilde{\mathbf{x}}_{k,l}(n_1,n_2)$ can be written as

$$\tilde{\mathbf{x}}_{k,l}(n_1,n_2)=\gamma_{k,l}\cdot[\eta(\gamma_{k,l})]^{\gamma_{k,l}}\frac{1}{\|\mathbf{r}_{k,l}(n_1,n_2)\|^{2-\gamma_{k,l}}} \quad (3)$$

with $\eta(\gamma)=\sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}}$ where Γ is the Gamma function, and $\mathbf{r}_{k,l}(n_1,n_2)=\tilde{\mathbf{x}}_{k,l}(n_1,n_2)/\sigma_{\tilde{\mathbf{x}}_{k,l}}$ where $\tilde{\mathbf{x}}_{k,l}(n_1,n_2)$ are the wavelet cover image coefficients. Fig. 1 shows the NVF of the cover image and watermark pyramids computed from image "Lena".

Finally, the weighted watermark is added to the cover image as follows:

$$\tilde{\mathbf{y}}_{k,l}(n_1,n_2)=\tilde{\mathbf{x}}_{k,l}(n_1,n_2)+(1-\mathrm{NVF}_{k,l}(n_1,n_2))\cdot\tilde{\mathbf{w}}_{k,l}(n_1,n_2)\cdot S^{\mathrm{e}}_{k,l}+\mathrm{NVF}_{k,l}(n_1,n_2)\cdot\tilde{\mathbf{w}}_{k,l}(n_1,n_2)\cdot S^{\mathrm{f}}_{k,l}, \quad (4)$$

where $\tilde{\mathbf{x}}_{k,l}$ are the cover image sub-bands, $\tilde{\mathbf{y}}_{k,l}$ the obtained stego wavelet sub-bands and $\tilde{\mathbf{w}}_{k,l}$ the watermark wavelet sub-bands. $S^{\mathrm{e}}_{k,l}$ is an embedding strength for

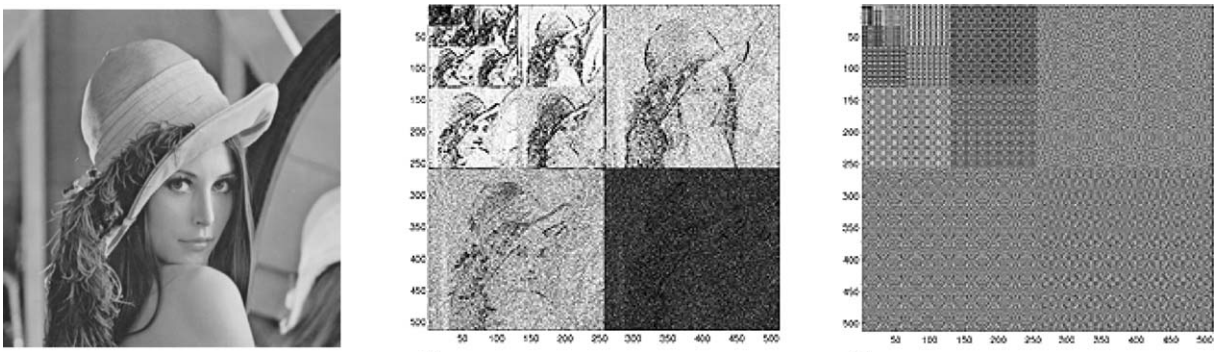

Fig. 1. Watermark embedding in the wavelet domain. (a) The cover image "Lena" to be watermarked, (b) NVFs computed for each sub-band of the cover image DWT, and (c) the watermark DWT pyramid.

the edges and textures, and $S^{\mathrm{f}}_{k,l}$ is a strength for the flat regions. Visual masking is then ensured first by choosing $S^{\mathrm{e}}_{k,l}$ greater than $S^{\mathrm{f}}_{k,l}$ for edges and textures hiding, and secondly by using adapted strengths for each resolution and for each orientation, according to the Modulation Transfer Function (MTF) [56] of the Human Visual System (HVS). Finally, the inverse DWT is applied on all the $\tilde{\mathbf{y}}_{k,l}$ to get the stego image $\mathbf{y}$.

2.2. Robust watermark extraction

From the stego and possibly distorted or attacked image $\mathbf{y}'$, we have to get an estimate $\hat{\mathbf{w}}$ of the watermark. For this purpose a Maximum a Posteriori (MAP) probability estimate is used:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}\in\mathbb{R}^N}\{p_{\mathbf{y}\,|\,\mathbf{w}}(\mathbf{y}'\,|\,\mathbf{w})\cdot p_{\mathbf{w}}(\mathbf{w})\}, \tag{5}$$

where $p_{\mathbf{y}|\mathbf{w}}(.)$ and $p_{\mathbf{w}}(.)$ are the p.d.f.'s of the cover image and watermark, respectively, and $N = M_1M_2$.

Let us consider the cover image $\mathbf{x}$ and the watermark $\mathbf{w}$ to be matrices with the elements $\mathbf{x}(n_1,n_2)$ and $\mathbf{w}(n_1,n_2)$, with $0 \leqslant n_1,n_2 < M_1,M_2$. Assuming that the image $\mathbf{x}$ and watermark $\mathbf{w}$ are conditionally i.i.d. locally Gaussian, i.e. $\mathbf{x}(n_1,n_2) \sim \mathcal{N}(\bar{\mathbf{x}}(n_1,n_2),\sigma^2_{\mathbf{x}}(n_1,n_2))$ and $\mathbf{w} \sim \mathcal{N}(0,\sigma^2_{\mathbf{w}}(n_1,n_2))$, an estimate of the watermark is given by

$$\hat{\mathbf{w}}(n_1,n_2) = \frac{\sigma^2_{\mathbf{w}}(n_1,n_2)}{\sigma^2_{\mathbf{w}}(n_1,n_2)+\hat{\sigma}^2_{\mathbf{x}}(n_1,n_2)}(\mathbf{y}'(n_1,n_2) - \bar{\mathbf{y}}'(n_1,n_2)), \tag{6}$$

where $\mathbf{y}'$ is the attacked stego image (including the effect of perceptual watermark modulation), and where it is assumed $\bar{\mathbf{y}}'(n_1,n_2) \approx \bar{\mathbf{x}}(n_1,n_2)$ to be the local mean of $\mathbf{y}(n_1,n_2)$, and $\hat{\sigma}^2_{\mathbf{x}}(n_1,n_2) = \max\{0,\sigma^2_{\mathbf{y}'}(n_1,n_2) - \sigma^2_{\mathbf{w}}(n_1,n_2)\}$.

An important issue is the estimation of the watermark variance $\sigma^2_{\mathbf{w}}$ in the above estimate. This can be done based on the available copy of the stego image. However, the severe distortions due to lossy JPEG compression could destroy the information about the texture masking that was used for the watermark embedding, and a histogram modification attack could damage the relevant information about contrast sensitivity masking. Since no reliable information about perceptual mask is available after these attacks (we assume low-WNR regime as the main operational scenario), we propose to use a *global* estimate of the watermark strength based on the available copy of the attacked image. This practically means that we assume spatial stationarity of the watermark. To estimate a global watermark variance we use the following formula:

$$\hat{\sigma}^2_{\mathbf{w}} = \frac{1}{M_1M_2}\sum_{n_1=0}^{M_1-1}\sum_{n_2=0}^{M_2-1}\hat{\sigma}^2_{y}(n_1,n_2), \tag{7}$$

where $\hat{\sigma}^2_{y}(n_1,n_2)$ is the local variance of the stego image in a small neighborhood at coordinates (n_1,n_2), for an image of size $M_1 \times M_2$. Estimate (7) is a global mean value of the watermark variance. Obviously, other robust versions of (7) such as a robust median estimate of the variance could be applied here.

2.3. Compensation of geometrical distortions

The main idea for this feature is to consider the geometrical transforms at two hierarchical levels: first at the global level (for the whole image), assuming global affine transform; secondly at the local level, in order to approximate any global non-linear or random local distortions as a juxtaposition of local affine transforms. This observation is especially true for the RBA. In the case of global affine transforms, the parameters of local affine transforms will be the same as the global one, and this allows to utilize the same unified approach for modeling both global affine transforms and local or non-linear distortions.

2.3.1. Global affine transforms

Geometrical affine distortions which may have occurred are retrieved and compensated at the global level, by analyzing the magnitude of either the ACF, or of the Discrete Fourier Transform (DFT) of the estimated watermark $\hat{\mathbf{w}}$. An affine transform consists in a linear component, which can be represented by the four coefficients a, b, c and d forming the matrix A, plus a translation component represented by the two coefficients v_1 and v_2 for the vector $\mathbf{v}$.

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}. \tag{8}$$

Therefore, an affine transform maps each point of Cartesian coordinates (n_1, n_2) to (n'_1, n'_2) according to:

$$\begin{pmatrix} n'_1 \\ n'_2 \end{pmatrix} = A \cdot \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} + \mathbf{v}. \tag{9}$$

The translation component $\mathbf{v}$ can be estimated separately, for example based on a cross-correlation between the extracted watermark $\hat{\mathbf{w}}$ and the known reference watermark $\mathbf{w}_{\text{ref}}$, or by a zero-phase search of the Fourier transform when symmetrical watermark blocks are used. This recovers also cropping, which conceptually corresponds to a translation. A similar approach to the compensation of translation and cropping based on cross-correlation has been proposed by Kalker et al. [27]. This step will then be ignored in the following.

With respect to the watermark blocks $\mathbf{w}_p$ mentioned in Eq. (1), the resulting distorted watermark $\mathbf{w}'$ after a global affine transform can be written as

$$\mathbf{w}'(n_1, n_2) = \sum_{m=0}^{(M_1/T_1)-1} \sum_{n=0}^{(M_2/T_2)-1} \mathbf{w}_p(A^{-1}(n_1, n_2)^{\mathrm{T}} - (m \cdot T_1, n \cdot T_2)^{\mathrm{T}})^{\mathrm{T}}, \tag{10}$$

where A is equally applied to all repeated blocks of the image.

Due to this periodicity, the ACF or the magnitude spectrum exhibits a regular grid of periodically placed local maxima, or *peaks*. The ACF approach consists in calculating $\hat{\mathbf{w}} * \hat{\mathbf{w}} = F^{-1}(|F(\hat{\mathbf{w}})|^2)$ from the estimated watermark $\hat{\mathbf{w}}$, where F is the DFT and F^{-1} the inverse DFT; for the magnitude spectrum we just compute $|F(\hat{\mathbf{w}})|^2$. Assuming that the watermark is white noise within blocks, its spectrum is additionally uniform. Therefore, both $\hat{\mathbf{w}} * \hat{\mathbf{w}}$ and $|F(\hat{\mathbf{w}})|^2$ show aligned and regularly spaced peaks. For the ACF, however, peaks are spaced with periods equal to the block size T_1, T_2, while for the magnitude spectrum they are placed with periods $M_1^F \cdot 1/T_1, M_2^{\mathrm{F}} \cdot 1/T_2$ if a 2D DFT domain of size $M_1^F \times M_2^F$ was used. If an affine distortion was applied to the stego image, the peaks layout will be rescaled, rotated and/or sheared, but alignments are preserved. Therefore, it is easy to estimate any affine geometrical distortion from these peaks by fitting alignments and estimating periods. Fig. 2 shows peaks extracted from the magnitude spectrum of the watermark $|F(\hat{\mathbf{w}})|^2$. In (a) the embedded and perceptually masked watermark $\mathbf{w}$ is considered by using the knowledge of the cover image in a non-oblivious approach, while in (b) the watermark $\hat{\mathbf{w}}$ predicted from the stego image is considered. Therefore, these peaks can be extracted from the stego data with high quality from the estimated watermark without knowledge of the cover image. Fig. 3 shows the peaks extracted from the estimated watermark, after JPEG lossy compression of the stego image with a quality factor (QF) of 50%, without and with geometrical distortions. The applied affine transform in then estimated from the regular structure of these peaks, using a Hough transform [25] based approach in order to detect the principle axes of the peaks alignments, as well as a robust period estimation along these main axes; the complete method is described in more details in Deguillaume et al. [10]. In contrast with other state-of-the-art self-reference approaches [27,30], the high redundancy of repetition of the watermark blocks, resulting in a large number

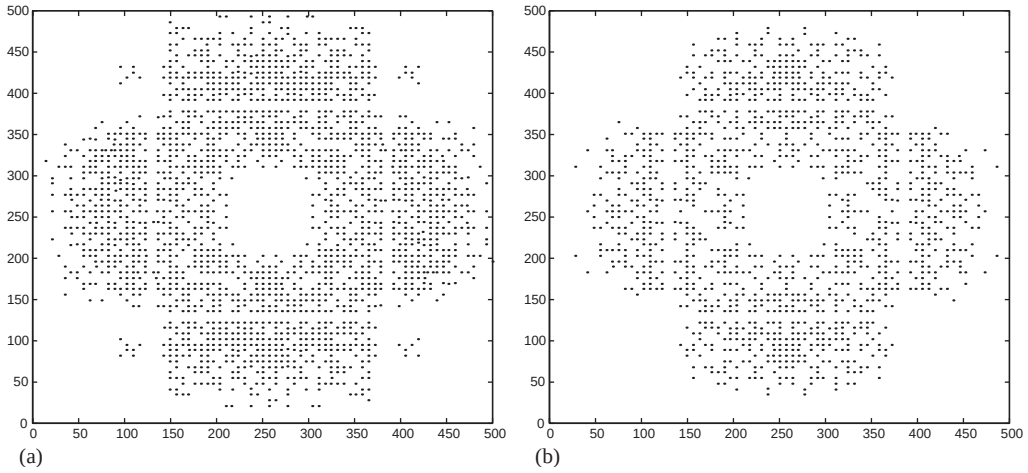

Fig. 2. Extracted peaks from the magnitude spectrum. (a) Peaks obtained from the embedded and perceptually masked watermark, and (b) peaks obtained from the stochastic estimate of the watermark.

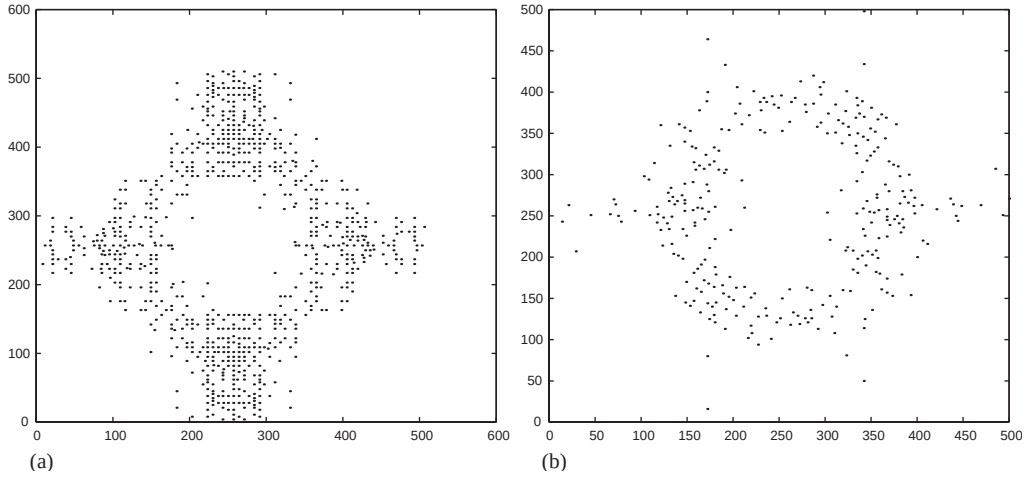

Fig. 3. Extracted peaks from the magnitude spectrum of the estimated watermark after JPEG compression with QF = 50%: (a) without geometrical distortion, and (b) after a rotation of 37° and auto cropping.

of peaks, makes our approach robust against attacks based on template analysis. In our experiments peaks could be properly extracted from JPEG compressed images with a QF as low as 50%; no known watermarking method is able to resist to global affine transforms combined with such a compression.

2.3.2. Local and non-linear transforms

The method for recovering from local geometrical transforms is based on the assumption that all linear or non-linear geometrical transforms as well as RBA can be considered as a set of local affine transforms. This approximation is possible due to the restricted amount of invisible distortions that can be introduced by the random bending to keep the quality of image within acceptable ranges, especially in commercial applications. Then Eq. (10) should be replaced by the following:

$$\mathbf{w}'(n_1,n_2) \approx \sum_{m=0}^{(M_1/T_1)-1} \sum_{n=0}^{(M_2/T_2)-1} \mathbf{w}_p(A_{m,n}^{-1}(n_1,n_2)^{\mathrm{T}} - (m\cdot T_1, n\cdot T_2)^{\mathrm{T}})^{\mathrm{T}}, \tag{11}$$

where $A_{m,n}^{-1}$ are the independent local affine transforms estimated at the block level. These transforms can be estimated either by local self-reference by computing *locally* the ACFs or the magnitude spectrums, or by local resynchronization based on the reference watermark. Although for simplicity Eq. (11) assumes $T_1 \times T_2$ blocks at the local level, the local self-reference option requires the computation of the ACF or magnitude spectrum of at least $2 \times 2 = 4$ blocks in order to exploit periodicity locally (that corresponds to approximately 160×160 pixels for our 76×76 repeated blocks $\mathbf{w}_p$). If local resynchronization with the reference watermark is used however, only one block $\mathbf{w}_p$ is required (thus corresponding to 76×76 pixels), or even less since these blocks are both upsampled and flipped.

One possible strategy consists in estimating the applied transform at the global level first, and then to correct this estimate at the local level. Another one could be to directly estimate locally the distortions. Note that in both cases, the local compensation of the distortions allows us to decode the watermark message locally, even if the watermark has been destroyed from most parts of the image. This could be a great advantage, for example, to extract the copyright information from parts of protected images which have been pasted into another image. The complete approach and application of this approach is detailed in Voloshynovskiy et al. [59].

2.4. Robust message decoding

Assuming that attack, prediction and extraction errors could be modeled as additive Gaussian, the detector is designed using the Maximum Likelihood (ML) formulation for the detection of a known signal (projection sets are known due to the key) in Gaussian noise, that results in a correlator detector:

$$\mathbf{r} = \langle \hat{\mathbf{w}}, \mathbf{p} \rangle. \tag{12}$$

In more general cases, the detector should be designed for stationary non-Gaussian noise or for the non-stationary Gaussian case, detailed in [57]. Finally, given an observation vector $\mathbf{r}$, the optimum decoder that minimizes the conditional probability of error assuming that all codewords $\mathbf{b}$ are equi-probable is given by the ML decoder:

$$\hat{\mathbf{b}} = \arg\max_{\tilde{\mathbf{b}}} p(\mathbf{r} \mid \tilde{\mathbf{b}}, \mathbf{x}), \tag{13}$$

based on the central limit theorem (CLT), since most researchers assume that the observed vector $\mathbf{r}$ can be accurately approximated as the output of an additive Gaussian channel noise [23,30] for a large sample space. We use the Bahl–Cocke–Jelinek–Raviv (BCJR) decoder [3] for the Turbo codes.

2.5. Assessing the message reliability

A remaining problem is to obtain a diagnostic assessing the reliability of the message, or the matching of the decoded message with respect to the original one. Since the message is ECC encoded, we consider that *all* errors in the extracted codeword resulting from the attacks should be corrected to claim a successful decoding. That means that the estimated message $\hat{\mathbf{b}}$ is successfully decoded if and only if all its bits are equal to those of the original message $\mathbf{b}$, that is if $\hat{\mathbf{b}} = \mathbf{b}$. Unfortunately in oblivious multibit watermarking schemes the original message itself is generally not known by the decoder, therefore other criterions for assessing the reliability of the watermark extraction and decoding are needed. A first approach could consist in the use of the additional information carried by the structure of the watermark $\mathbf{w}$ such as its spatial allocation. The peaks from the ACF or magnitude spectrum can give us a first estimate of the probability of presence of the watermark. As a second approach the distortion occurred to the extracted reference watermark $\hat{\mathbf{w}}_{\text{ref}}$ informs us about the probability of presence of the said watermark, since in contrary to $\mathbf{b}$ the original $\mathbf{w}_{\text{ref}}$ is known. Further, if the statistics of the pseudo-random process used for the encoding of $\mathbf{w}$ is precisely known, it is possible to design an estimator of the probability that the predicted watermark $\hat{\mathbf{w}}$ contains a message given a key $\mathbf{k}$; such an approach was proposed for Spread Spectrum watermarking by Óruanaidh and Csurka [43] based on a Bayesian estimator.

However, the methods cited above give an estimate of the *presence* of a watermark, rather than of the *reliability* of the message decoding itself. Then to assess the accuracy of the watermark decoding, we propose to attach a binary encoded check code $\mathbf{h}$ to the message $\mathbf{b}$ to form the string $\mathbf{b}' = (\mathbf{b}, \mathbf{h})$, with $\mathbf{h}$ derived

from $\mathbf{b}$ using some function as: $\mathbf{h}=H(\mathbf{b})$. This function H should produce different codes for different inputs with high probability, and could be a cryptographic hash function (cryptographic hash-codes and signatures are summarized in Section 3). Upon decoding, we get $\hat{\mathbf{b}}' = (\hat{\mathbf{b}}, \hat{\mathbf{h}})$. The check code is computed again with the same function as for the embedding from the estimated message as $\tilde{\mathbf{h}} = H(\hat{\mathbf{b}})$, and then one can tell that the message was correctly decoded if $\tilde{\mathbf{h}} = \hat{\mathbf{h}}$, with high probability if the length of $\mathbf{h}$ was sufficient. These considerations lead to the important concept or error probability, or to the probability of *false alarm*, that is the probability that the decoder claims a successfully extracted watermark and decoded message, when actually there was none.

Practically, we use jointly the reference watermark and the message-dependent check code above. A successful decoding is claimed when the ratio of correct bits in the reference bit string is greater or equal to a threshold τ_{ref}, *and* if the extracted and recomputed check codes strictly match. Let us consider a binary bit string representing the extracted reference bit string $\hat{\mathbf{b}}_{\text{ref}}$, in comparison with the expected string $\mathbf{b}_{\text{ref}}$, and N their length. When no watermark is present, then $\hat{\mathbf{b}}_{\text{ref}}$ is random and uncorrelated with $\mathbf{b}_{\text{ref}}$ and on average 50% of bits are in common between the two bit strings. With bits 0 and 1 occurring with a probability $p = \frac{1}{2}$ each, one can compute that the probability that n or more bits are in common by chance ($0 \leqslant n \leqslant N$) is given by the binomial distribution:

$$P_{1/2}^{\text{ref}}(k \geqslant n \,|\, N) = \sum_{k=n}^{N} \binom{N}{k} 2^{-N}. \tag{14}$$

For example, for a reference bit string of length $N = 64$ bits, the probability of getting at least 70% of correct bits ($\tau_{\text{ref}} = 45$ bits) is $P_{1/2}^{\text{ref}}(k \geqslant 45 \,|\, 64) \simeq 7.8 \times 10^{-4}$. Of course τ_{ref} should be sufficiently low in order to limit the *misdetection* probability (rejection of watermark which was present), but high enough regarding the average ratio of errors that the ECC can correct. For a check code $\mathbf{h}$ of N' bits, we can compute the probability that $H(\hat{\mathbf{b}}) = \hat{\mathbf{h}}$ (for *all* bits) as

$$P_{1/2}^{\text{chk}}(N') = 2^{-N'}. \tag{15}$$

With a 18 bit check code we get $P_{1/2}^{\text{chk}}(18) \simeq 3.8 \times 10^{-6}$. Then the product of expressions (14) and (15): $P_{1/2}^{\text{ref}}(k \geqslant n \,|\, N) \cdot P_{1/2}^{\text{chk}}(N')$ estimates the false alarm probability P_{false}, and with the lengths and threshold above we get finally $P_{\text{false}} \simeq 3.0 \times 10^{-9}$. A reference watermark threshold of 75% ($\tau_{\text{ref}} = 48$) and a longer check code with $N' = 24$ lead to $P_{\text{false}} \simeq 2.3 \times 10^{-12}$. These false alarm rates could be considered as realistic values for most robust watermarking applications.

3. Hybrid watermarking

We propose to join the highly robust watermarking scheme described in Section 2 with a block-wise fragile algorithm based on cryptographically secure hash-codes similar to Wong [64] or Coppersmith et al. approaches [9], including several improvements for security reasons which will be discussed in Section 4. This technology is implemented in our prototype *Berkut 2.0* [38].

3.1. Hybrid watermark embedding

The block diagram of Fig. 4 shows the hybrid embedding process at the image level. This is the symmetrical version of tamper proofing/authentication, that means that both the signature embedding and verification require the same user key $\mathbf{k}$, which should be kept secret. As shown in Section 2, a robust watermark block consists in the two non-overlapping (thus orthogonal) components: the informative watermark $\mathbf{w}_{\text{inf}}$ and the reference watermark $\mathbf{w}_{\text{ref}}$. While $\mathbf{w}_{\text{inf}}$ contains the message $\mathbf{b}$, encoded and encrypted to a codeword $\mathbf{c}$ with a secret key $\mathbf{k}$, $\mathbf{w}_{\text{ref}}$ depends only on the key $\mathbf{k}$. $\mathbf{w}$ is then embedded as previously described to the cover image $\mathbf{x}$, taking into account the perceptual model $\mathbf{M}_{\text{HVS}}$ computed from $\mathbf{x}$ to ensure low visual distortions. $\mathbf{w}_{\text{empty}}$ corresponds to positions where no watermark information is embedded when the watermark density is lesser than 1.

Obviously, the fragile component has to be applied *after* the robust one, in order to hash the robust watermark with the image. The fragile watermark $\mathbf{w}_{\text{frag}}$ is then based on a key-dependent block-wise cryptographically secure hash function, of which input key is derived from $\mathbf{k}$. The resulting codes (i.e. a set of computed hash-codes $\mathbf{s}$ consisting of one code for each

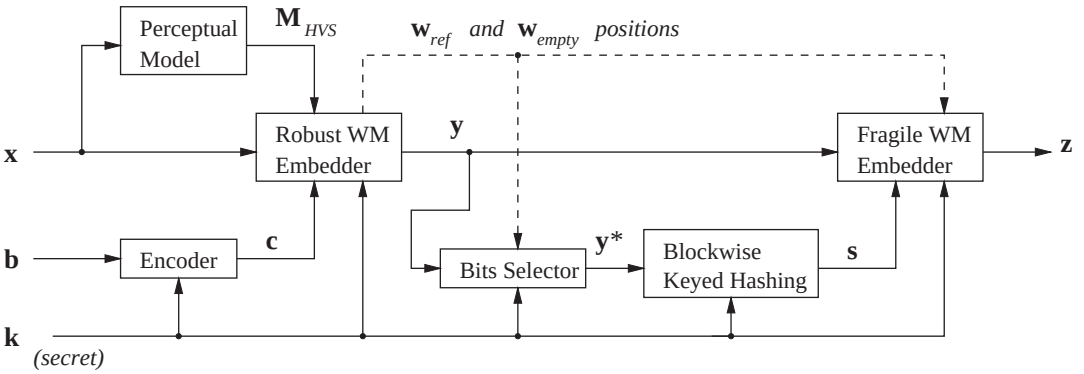

Fig. 4. Hybrid embedding. First the robust watermark is embedded, then the fragile watermark. The key **k**, used for both watermarks embedding and hashing, should be kept secret in order to prevent anyone to produce valid hash-codes from the forged images.

block) are then embedded as local *signatures*[1] in a fragile way within each block: a set of least significant bits (LSB) of **y** is pseudo-randomly selected based on **k** to embed the bits of the code. In order to keep hash-codes valid, the hash function takes as input $\mathbf{y}^{\star}$, a version of **y** where all LSBs selected for the embedding of $\mathbf{w}_{\text{frag}}$ have been cleared (set to 0) by the "Bits Selector" block. The "Keyed Hashing" block could be any keyed hashing algorithm, or an unkeyed one encrypted afterwards. The hash function requirements could be summarized as

$$\begin{aligned} &\mathbf{I} = \mathbf{I}' \Rightarrow H_{\mathbf{k}}(\mathbf{I}) = H_{\mathbf{k}}(\mathbf{I}'), \\ &\mathbf{I} \neq \mathbf{I}' \Rightarrow H_{\mathbf{k}}(\mathbf{I}) \neq H_{\mathbf{k}}(\mathbf{I}'), \end{aligned} \tag{16}$$

where **I** and $\mathbf{I}'$ are any input (not necessary visual data), and H is a hash function optionally depending on a random key **k**. Moreover, when $\mathbf{I} \neq \mathbf{I}'$ even for a single bit, $H_{\mathbf{k}}(\mathbf{I})$ and $H_{\mathbf{k}}(\mathbf{I}')$ are completely uncorrelated.

Any cryptographically secure functions can be used for hash-coding or signature. Common unkeyed hash primitives which can be used are Rivest's Message Digest version 5 (MD5) [48] or NISTs Secure Hash Algorithm version 1 (SHA-1) [39]. The output hash-codes could be encrypted by a symmetric block cipher such as the Data Encryption Standard (DES) or its triple version (Triple-DES) [40], Lai and Massey IDEA [32], or the recently accepted NIST's Advanced Encryption Standard (AES/Rijndael) [42]. Keyed hash functions could be a Hash Message Authentication Code (HMAC) function based either on MD5 or on SHA-1.[2] Finally we obtain **z**, the stego image containing both the robust and the fragile watermark.

The robustly marked image **y** (already containing **w**) is divided by the fragile algorithm into contiguous and non-overlapping blocks of indexes i, j, and results into a final stego image **z** which contains both **w** and $\mathbf{w}_{\text{frag}}$. Therefore **y** and **z**, as well as the set of signatures **s**, can be written in term of these blocks as follows:

$$\begin{aligned} &\mathbf{y} = \{\mathbf{y}_{i,j}\}, \quad \mathbf{z} = \{\mathbf{z}_{i,j}\}, \quad \mathbf{w}_{\text{frag}} = \{\mathbf{w}_{\text{frag } i,j}\}, \\ &\mathbf{s} = \{s_{i,j}\} \\ &\qquad \text{with } i = 1,\ldots,\frac{M_1}{t_1'} \text{ and } j = 1,\ldots,\frac{M_2}{t_2'}, \end{aligned} \tag{17}$$

where M_1, M_2 is the image size and t_1', t_2' the block size in number of pixels. The block size should be selected as small as possible in order to ensure good localization of the detection of image alterations, but large enough to contain sufficient signatures bits for an "acceptable" level of security. Practically we performed experiments with block sizes $t_1' \times t_2'$ from 19×19 to 38×38 pixels. Note that if the image size is not an

[1] From the cryptographic point of view: in the case of a symmetrical (i.e. with a secret key) scheme, we should talk about *Message Digest Code* (MDC); in the case of an asymmetrical (i.e. public key based) scheme as discussed later, we should talk about *signature*. Actually we will use the term *signature* in both cases.

[2] For security Triple-DES is preferable to DES which uses too short keys. Moreover, SHA (including all variants) is currently the only FIPS-approved method for secure hashing, and should be preferred to MD5 as added security measures.

1. for $j = 1$ to M_2/t'_2
2. for $i = 1$ to M_1/t'_1
3. $\mathbf{y}^{\star}_{i,j}, \mathbf{y}^{\star}_{\eta(i,j)} = BitsSelector\left(\mathbf{y}_{i,j}, \mathbf{y}_{\eta(i,j)}\right)$;
4. $s_{i,j} = KeyedHash_{\mathbf{k}}\left(\mathbf{y}^{\star}_{i,j}, \mathbf{y}^{\star}_{\eta(i,j)}, \ldots\right)$;
5. $\mathbf{z}_{i,j} = FragileEmbed_{\mathbf{k}}\left(\mathbf{y}_{i,j}, s_{i,j}\right)$;
6. end for
7. end for

Fig. 5. Fragile watermark embedding pseudo-code at the block level. The robustly watermarked image $\mathbf{y}$, of size $M_1 \times M_2$ pixels, is divided into $t'_1 \times t'_2$ adjacent blocks; $\mathbf{y}_{i,j}$ denotes the ith, jth block, $\mathbf{y}^{\star}_{i,j}$ the same block with the fragile watermark embedding bits set to 0, and $\eta(i,j)$ the indexes of neighboring blocks (e.g. the 8 neighbors). The "..." at line 4 denotes additional information which could be included in the input of the hash function. Then the signature $s_{i,j}$ is computed using the key $\mathbf{k}$, and is finally embedded (as $\mathbf{w}_{\text{frag}}$) to form the authenticated block $\mathbf{z}_{i,j}$.

exact multiple of the block size, one can take for example the lower integer bounds of M_1/t'_1 and M_2/t'_2. The embedding of the fragile part $\mathbf{w}_{\text{frag}}$ is detailed for the block level in pseudo-code of Fig. 5, and illustrated in Fig. 7.

In contrast to Wong's approach where blocks are independently hashed, our hash function takes as input the current i,j-block itself as well as some neighboring blocks, the resulting code being then embedded into the i,j-block only. Our approach is similar to Coppersmith et al. method [9], which proposed to hash overlapping blocks, and to embed the result only into smaller non-overlapping blocks. This ensures that the signature hold by each block depends not only on that block itself, but also from a neighboring area in some extent. However, we propose here to include *complete* neighboring blocks in the hash function input, while Coppersmith et al. proposed only slightly larger overlapping blocks (of 32×32 pixels) with respect to the non-overlapping ones (24×24 pixels). Moreover, the neighborhood function can be parameterized as shown below depending on the targeted application.

Such hashing of the current block and neighboring blocks together is a first step to introduce local contextual dependencies, and could be called *Hash-code Block Chaining* (HBC) as proposed by Barreto et al. [2]. In pseudo-code of Fig. 5, for each block of indexes i,j, the neighboring indexes are denoted by $\eta(i,j)$, with the possible configurations examples:

$$\eta(i,j) = \{(i-1,j-1),(i-1,j),(i-1,j+1),\\ (i,j-1),(i,j+1),(i+1,j-1),\\ (i+1,j),(i+1,j+1)\} \text{ (the 8 neighbors)},$$

$$\eta(i,j) = \{(i-1,j),(i,j-1),(i,j+1),\\ (i+1,j)\} \text{ (4 neighbors)},$$

$$\eta(i,j) = \{(i-1,j),(i+1,j)\} \text{ (2 neighbors)},$$

$$\text{or} \quad \eta(i,j) = \{(i-1,j)\} \text{ (only one neighbor)}.$$

For those blocks which are beside the image borders ($i = 1$ or M_1/t'_1, $j = 1$ or M_2/t'_2), for which the neighboring blocks fall outside of the image, one can just consider that the image is infinitely padded with the value 0. These configurations are illustrated in Fig. 6.

Note also that each signature detects a modification applied to it's block or to any of it's neighboring blocks. Although this could lead to a decrease of the localization property of the method, it is also possible to consider that each signature preserved by attacks validates it's block *plus* all it's neighbors. Such consideration can be used to compensate the loss of localization, and in our case even results into the same localization capability as if no HBC was used (except along the image borders). This is due to the fact that the neighborhood includes complete blocks: one block which is not validated by some signatures can be completely validated by one of its neighbors. In contrast, the scheme of Coppersmith et al. cannot fully compensate the loss in localization for alterations occurred close to block corners or boundaries, because of its smaller neighboring zone with respect to the block size.

In addition to HBC, other local or global contextual information can be included in the input of hash functions, such as current block indexes (i,j), the image size (M_1, M_2), owner-related data like in the case of robust watermarking, date and time, place, a unique image identification name or number, etc. Such hashed additional information is denoted by the "..." in pseudo-code of Fig. 5 (line 4). Linking individual block hashing with both local and global contextual information is important from the security point of view, in order to defeat the substitution

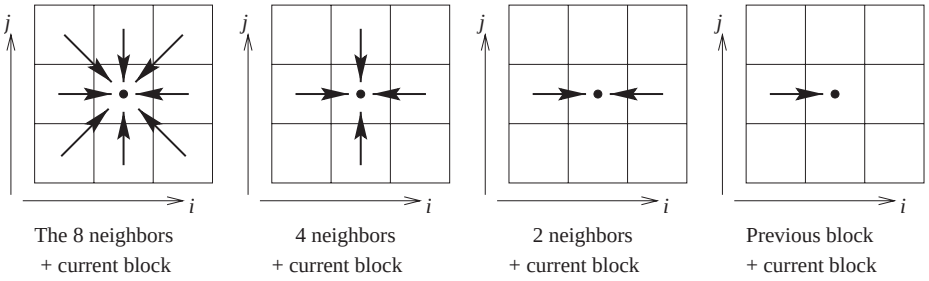

Fig. 6. Hash-code Block Chaining (HBC) examples. The block-wise hash function can take as input: the 8 neighbor blocks, 4 neighbors, 2 neighbors, or only 1 neighbor, in addition to the current block. During verification, if the extracted signature matches the computed signature then the current block and all its considered neighbors are validated.

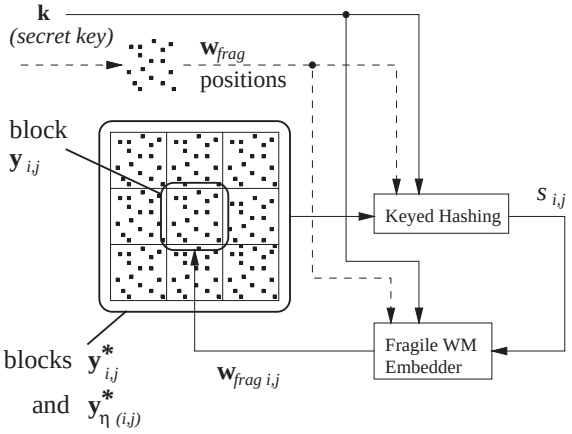

Fig. 7. Fragile watermark embedding for one block $\mathbf{y}_{i,j}$, in the case of 8 neighbors HBC. Square points correspond to the positions reserved for the fragile watermark, and which should be excluded from the input of the hash function (giving $\mathbf{y}^{\star}_{i,j}$ and $\mathbf{y}^{\star}_{\eta(i,j)}$). The resulting hash-code $s_{i,j}$ is then embedded to $\mathbf{y}_{i,j}$.

attacks dedicated to fragile and semi-fragile watermarking schemes. These attacks, as well other security aspects, will be explained in more details, and countermeasures proposed, in Section 4.

Note that $\mathbf{w}_{\text{frag}}$ fragile blocks may or may not coincide with $\mathbf{w}$ robust blocks, actually fragile blocks may be sub-blocks from robust blocks for better locality in the tamper detection. However an important issue is to preserve the original robustness of the robust watermark: first, embedding the fragile part by LSB modulation of selected pixels ensures very limited modification, which is unlikely to destroy the robust watermark which has a larger amplitude; secondly, we propose to embed the fragile watermark in selected positions not belonging to the robust watermark copyright information component $\mathbf{w}_{\text{inf}}$, i.e. we embed $\mathbf{w}_{\text{frag}}$ in positions of the reference watermark $\mathbf{w}_{\text{ref}}$ and in positions containing no watermark at all $\mathbf{w}_{\text{empty}}$), thus fully preserving $\mathbf{w}_{\text{inf}}$. This characteristic is shown by the dashed arrows transmitting the $\mathbf{w}_{\text{ref}}$ and $\mathbf{w}_{\text{empty}}$ positions in Fig. 4, and by the block squared points inside the image blocks in Fig. 7. Thus $\mathbf{w}_{\text{inf}}$ is untouched, and on average at most 50% of positions in $\mathbf{w}_{\text{ref}}$ are altered by $+1$ or -1 due to the LSB modulation. Since also $\mathbf{w}_{\text{ref}}$ usually covers not more than 20% of the area of $\mathbf{w}$, this makes $\mathbf{w}_{\text{frag}}$ and $\mathbf{w}$ almost orthogonal. At the same time the visual impact of the fragile part is much lower than the visual distortions of the robust part.

3.2. Hybrid watermark extraction and verification

At the extraction stage, the robust extractor first estimates the robust watermark $\hat{\mathbf{w}}$ from the possibly attacked and tampered stego image $\mathbf{z}'$, and decodes an estimation of the copyright message $\hat{\mathbf{b}}$; the possibly applied global (affine) and local geometrical distortions (RBA) are compensated in this part.

The block diagram of Fig. 8 shows the extraction and authentication part. The authentication part takes $\mathbf{z}'$ as input; recomputes signatures $\tilde{\mathbf{s}}$ from $\mathbf{z}'^{\star}$ (a version of $\mathbf{z}'$ where the LSBs used for $\mathbf{w}_{\text{frag}}$ have been set to 0); extract $\hat{\mathbf{w}}_{\text{frag}}$ from $\mathbf{z}'$ and get the estimated embedded signatures $\hat{\mathbf{s}}$; and finally outputs a tampered blocks map estimate $\hat{\mathbf{T}}$ (consisting of one "tampering index" per block) obtained by comparing signatures $\tilde{\mathbf{s}}$

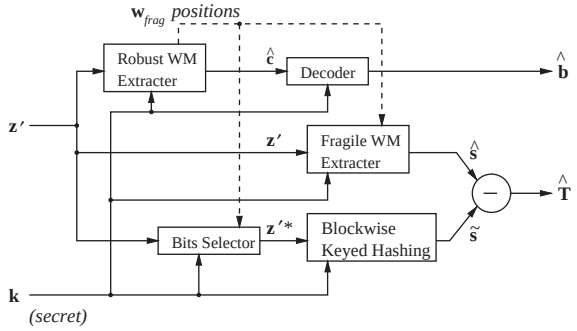

Fig. 8. Hybrid extraction. The robust and the fragile watermark are extracted separately, using the secret key $\mathbf{k}$ for hashing. Afterwards a decision have to be taken based on the hash-codes differences.

1. for $j = 1$ to M_2/t_2'
2. for $i = 1$ to M_1/t_1'
3. $\mathbf{z}'^{\bigstar}_{i,j}, \mathbf{z}'^{\bigstar}_{\eta(i,j)} = BitsSelector\left(\mathbf{z}'_{i,j}, \mathbf{z}'_{\eta(i,j)}\right)$;
4. $\tilde{s}_{i,j} = KeyedHash_{\mathbf{k}}\left(\mathbf{z}'^{\bigstar}_{i,j}, \mathbf{z}'^{\bigstar}_{\eta(i,j)}, \ldots\right)$;
5. $\hat{s}_{i,j} = FragileExtract_{\mathbf{k}}\left(\mathbf{z}'_{i,j}\right)$;
6. $\hat{T}_{i,j} = \hat{s}_{i,j} \ominus \tilde{s}_{i,j}$;
7. end for
8. end for

Fig. 9. Fragile watermark extraction pseudo-code at the block level. $\mathbf{z}'_{i,j}$ is the current block of the possibly attacked stego image $\mathbf{z}'$. $\ominus$ is the comparison operator between signatures (line 6), and each block i,j for which $\hat{T}_{i,j} = 0$ authenticates this block i,j plus its neighbors $\eta(i,j)$.

and $\hat{\mathbf{s}}$. For the authentication the input image $\mathbf{z}'$ is divided into blocks of the same size and same positions as for the embedding process. Thus $\mathbf{z}'$ and $\hat{\mathbf{T}}$ can be expressed as

$$\mathbf{z}' = \{\mathbf{z}'_{i,j}\}, \quad \hat{\mathbf{T}} = \{\hat{T}_{i,j}\}, \quad \text{with}$$
$$i = 1, \ldots, M_1/t_1' \quad \text{and} \quad j = 1, \ldots, M_2/t_2'. \tag{18}$$

The pseudo-code of Fig. 9 and block diagram of Fig. 10 show the extraction and verification of the fragile signature at the block level.

We can then define the estimated block tampering index $\hat{T}_{i,j} \in \{0,1\}$ for each block index i,j as 1 if the block $\mathbf{z}'_{i,j}$ and/or its neighbors $\mathbf{z}'_{\eta(i,j)}$ are modified and 0 otherwise, as given by the comparison operator $\ominus$ in the pseudo-code of Fig. 9, line 6. It could

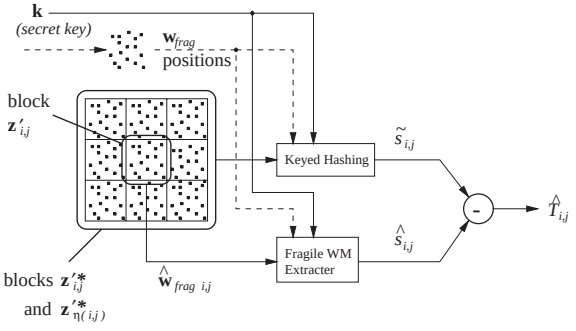

Fig. 10. Fragile watermark extraction for one block $\mathbf{z}'_{i,j}$, in the case of 8 neighbors HBC. The recomputed code $\tilde{s}_{i,j}$ is compared with the extracted one $\hat{s}_{i,j}$: the block and its neighbors are authenticated if $\hat{T}_{i,j} = 0$.

be written as

$$\hat{T}_{i,j} = 1 - \delta(\tilde{s}_{i,j} - \hat{s}_{i,j}), \tag{19}$$

where $\delta(.)$ is the Kroneker symbol ($\delta(x) = 1$ if $x = 0$, and 0 otherwise), considering $\hat{s}_{i,j}$ and $\tilde{s}_{i,j}$ as binary encoded scalars.

At the end a global normalized authenticity measure A_T indicates the ratio of authentic blocks over the total number of blocks for the whole image, and could be defined as

$$A_T = \frac{1}{\frac{M_1}{t_1'}\frac{M_2}{t_2'}} \sum_{i=1}^{M_1/t_1'} \sum_{j=1}^{M_2/t_2} 1 - \hat{T}_{i,j} \tag{20}$$

with the following interpretation:

$A_T = 1 \Rightarrow$ authentic image,

$0 < A_T < 1 \Rightarrow$ partially tampered image,

$A_T = 0 \Rightarrow$ non−authentic image. (21)

At the end the generic following decision can then be made, based on the diagnostics of both robust and fragile watermarks:

(1) $\hat{\mathbf{b}}$ *is correctly decoded and* $A_T = 1$: The image is fully authenticated and has not been tampered.
(2) $\hat{\mathbf{b}}$ *is correctly decoded but* $A_T < 1$: If $A_T > 0$ then only malicious local modification probably occurred: we partially authenticate the image and we point out modified regions where $\hat{T}_{i,j} = 1$; if $A_T = 0$, we reject the image as globally

non authentic, but since $\hat{\mathbf{b}}$ is valid at the same time, we can claim that a copy attack may have occurred.

(3) $\hat{\mathbf{b}}$ *failed or multiple* $\hat{\mathbf{b}}_k$, $k = 1, 2, \ldots$ *are decoded, and* $A_T > 0$: If we get $A_T = 1$, then we can immediately claim that an advanced substitution attack may have been applied using different images watermarked with the same key; if $A_T < 1$, we can suspect a collage attack for example if some of the $\hat{T}_{i,j}$ were 0 (matching signatures) simultaneously for at least two regions containing valid distinct robust messages $\hat{\mathbf{b}}_k$ and $\hat{\mathbf{b}}_l$.

(4) $\hat{\mathbf{b}}$ *was not decoded and* $A_T = 0$: We reject the image as globally non-authentic, and at the same time we cannot claim any copyright.

Simple attacks are easily detected in items 1, 2, and 4. If the marked image has been simply replaced by another one, the input will obviously be rejected; any local modification in an valid image will destroy signatures in the altered blocks, and *simple tampering* (local or global) is detected by item 2. A copy attack further corresponds to the second item when $A_T = 0$: the copy of a robust watermark $\mathbf{w}$ from another image would make the robust message $\hat{\mathbf{b}}$ still correctly decoded, but all signatures will mismatch ($\hat{T}_{i,j} = 1$ for all i, j); therefore by rejecting this case, our hybrid approach is resistant to the copy attack.

Item 3 is a particular case: if the robust watermark is altered, then we could expect $A_T < 1$ due to signatures mismatches since the robust watermark is included in the input of hash functions. However, this situation can occur if different robust watermarks are present, all embedded with the same key; note that our robust watermarking algorithm, which works at the local level to achieve resistance to the RBA [59], can successfully decode *different messages* $\hat{\mathbf{b}}_i$ at the local level. This situation can occur if a collage attack was applied. This consists of the composition of an image, with parts coming from other images watermarked with the same key. By keeping the blocks synchronization of the fragile scheme, such compositions can be made where the areas coming from different sources are wrongly validated—only the boundaries between areas being detected as non-authentic. This scenario and other substitution attacks are discussed in more details in Section 4, as well as countermeasures against them.

In general the analysis of the $\hat{T}_{i,j}$ locally, with respect to blocks from which one $\hat{\mathbf{b}}$ or several $\hat{\mathbf{b}}_k$ were correctly decoded, can be useful for both items 2 and 3 in order to get more detailed diagnostics about what probably happened to the image.

3.3. Asymmetrical hybrid watermark

The version presented above is a symmetrical scheme, which is also the case of most of today watermarking technologies. This means that the same key $\mathbf{k}$ is used for the embedding and the extraction/authentication, and should be obviously kept secret. However, at the opposite from what stands for robust watermarking, in the case of tamper proofing and authentication an asymmetrical scheme is straightforward. We propose first to use a symmetrical key $\mathbf{k}_0$ for the generation of both $\mathbf{w}$ and $\mathbf{w}_{\text{frag}}$ positions, and possibly encrypt the robust watermark message $\mathbf{b}$. This key is needed for both the embedding and the extraction stage, consequently it should be made public. Then the robust part in particular is still a symmetrical part. Secondly a private key $\mathbf{k}_{\text{priv}}$ is used for the fragile part to encrypt the resulting set of hash-codes $\mathbf{h}$ obtained from a block-wise *unkeyed* hash function ($\mathbf{h} = \{h_{i,j}\}$), producing the signatures $\mathbf{s}$ which are embedded within blocks. Any asymmetrical encryption or signature algorithm can be used for this purpose, such as the Rivest–Shamir–Adleman (RSA) algorithm [49], which is based on the difficulty to factorise large numbers into prime components, or the NISTs Digital Signature Algorithm (DSA) [41], based on the intractability of a certain particular case of the problem of computing a discrete logarithm. The asymmetrical hybrid watermark embedding is shown in Fig. 11 at the image level; the corresponding fragile part algorithm is given in pseudo-code of Fig. 12 and shown in Fig. 13 at the block level.

The extraction/verification stage requires the symmetrical key $\mathbf{k}_0$ used for the embedding, and a public key $\mathbf{k}_{\text{pub}}$ to decrypt the embedded signatures back to hash-codes. In this approach, everyone can then verify the document using $\mathbf{k}_{\text{pub}}$, but only the holder of the private key $\mathbf{k}_{\text{priv}}$ can generate valid signatures.

Concerning the robust watermark, one can argue that it would be easy to remove the robust watermark since $\mathbf{k}_0$ is publicly known in this application;

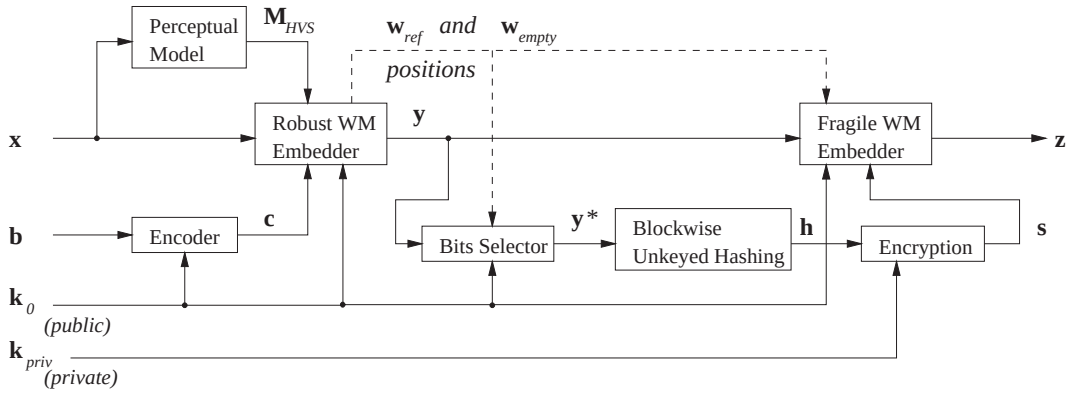

Fig. 11. Hybrid embedding, asymmetrical version. The signature part requires a private key $\mathbf{k}_{\text{priv}}$ for signature generation; only the holder of the private key can generate valid signatures. However, a symmetrical key $\mathbf{k}_0$ is still needed for encoding $\mathbf{b}$ to the robust watermark $\mathbf{w}$ and the position allocation of both $\mathbf{w}$ and $\mathbf{w}_{\text{frag}}$; $\mathbf{k}_0$ should be public since it is needed for both the embedding and the extraction stages.

1. for $j = 1$ to M_2/t_2'
2. for $i = 1$ to M_1/t_1'
3. $\mathbf{y}^{\star}_{i,j}, \mathbf{y}^{\star}_{\eta(i,j)} = BitsSelector\left(\mathbf{y}_{i,j}, \mathbf{y}_{\eta(i,j)}\right)$;
4. $h_{i,j} = UnkeyedHash\left(\mathbf{y}^{\star}_{i,j}, \mathbf{y}^{\star}_{\eta(i,j)}, \ldots\right)$;
5. $s_{i,j} = Encrypt_{\mathbf{k}_{priv}}\left(h_{i,j}\right)$;
6. $\mathbf{z}_{i,j} = FragileEmbed_{\mathbf{k}_0}\left(\mathbf{y}_{i,j}, s_{i,j}\right)$;
7. end for
8. end for

Fig. 12. Fragile watermark embedding pseudo-code at the block level, asymmetrical version. An unkeyed hash function is used, and the hash-codes $h_{i,j}$ is encrypted using an asymmetrical encryption algorithm with private key $\mathbf{k}_{\text{priv}}$. The resulting signature $s_{i,j}$ is then embedded to form block $\mathbf{z}_{i,j}$; note that the auxiliary public key k_0 may be needed for allocating embedding positions.

however, in this case the signatures would fail and this kind of attack would be detected. The asymmetrical hybrid extraction is shown in Fig. 14, the fragile part algorithm detailed in pseudo-code of Fig. 15, and shown for one block in Fig. 16.

The only changes at the block level with respect to the symmetrical version are: first, use an unkeyed hash function to generate hash-codes $h_{i,j}$; secondly, encrypt these hash-codes to $s_{i,j}$ with $\mathbf{k}_{\text{priv}}$ before embedding, and decrypt the extracted signatures $\hat{s}_{i,j}$ to hash-codes $\hat{h}_{i,j}$ with $\mathbf{k}_{\text{pub}}$ *before* comparison. Therefore the recomputed and extracted decrypted hash-codes themselves should be compared, and Eq. (19) can be rewritten as

$$\hat{T}_{i,j} = 1 - \delta(\tilde{h}_{i,j} - \hat{h}_{i,j}). \tag{22}$$

4. Security of hybrid watermarking

Many attacks or malicious changes can by mounted against hybridly watermarked documents, either targeting the robust watermark and the fragile watermark separately, or addressing the interactions or relationship between both parts. Since attacks on robust watermarking have been already widely discussed, here we will mainly focus on intentional attacks specific to the fragile part. Unlike those dedicated to robust watermarks, the usual goal of attacks on fragile watermark is not to remove the information (otherwise the host data would be invalidated), but rather to perform tampering or manipulations which will be undetected at the verification stage. While well-known attacks are based on weaknesses inherent to the embedding and verification algorithms, other rather exploit security flaws of specific scenarios or verification protocols. More details about the classification of attacks against authentication watermarks as well as a resistant block-based scheme are addressed by Fridrich [14]. Analysis of tampering attacks and countermeasures recommendations for block-wise cryptographically based algorithms are studied by Barreto et al. [2], and will be the

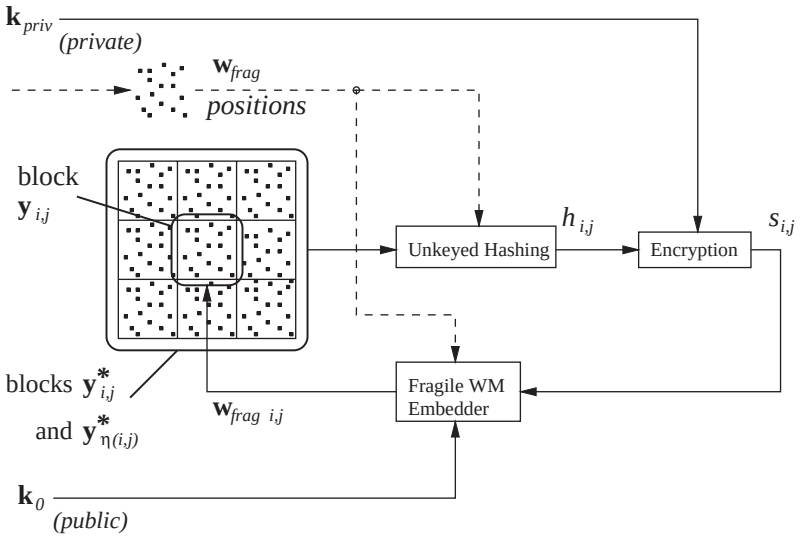

Fig. 13. Fragile watermark embedding for one block $\mathbf{y}_{i,j}$, asymmetrical version. The principal difference with symmetrical embedding is that an unkeyed hash is used to get the hash-code $h_{i,j}$, which is encrypted with the user's private key $\mathbf{k}_{\text{priv}}$.

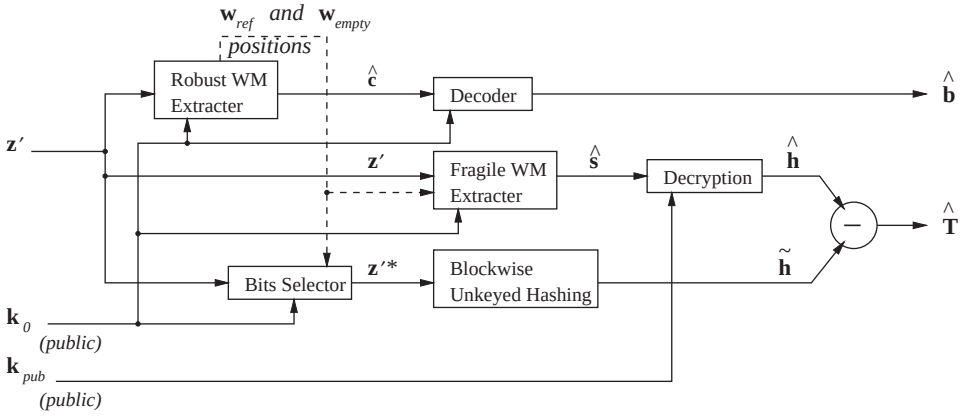

Fig. 14. Hybrid extraction, asymmetrical version. The public keys $\mathbf{k}_0$ and $\mathbf{k}_{\text{pub}}$ are used for the watermarks extraction, and for decrypting the extracted hash-codes, respectively.

main source of inspiration for the following of this section. Finally, we propose to used jointly the diagnostic and information given by the robust and the fragile watermarks in a hybrid approach, in order to increase the security of the system against attacks individually targeting either the robust part, or the fragile part—and in particular protocol attacks such as the copy attack and the collage attack.

4.1. Pixel based vs. block based schemes

In a fragile approach, any change is in theory detected, since the change of one pixel would result into the mismatch of embedded and recomputed corresponding function results or hash-codes. However, the retained method for the generation of signatures and their embedding should be carefully designed in order to achieve resistance to various tampering attack.

1. for $j = 1$ to M_2/t'_2
2. for $i = 1$ to M_1/t'_1
3. $\mathbf{z}'^{\star}_{i,j}, \mathbf{z}'^{\star}_{\eta(i,j)} = BitsSelector\left(\mathbf{z}'_{i,j}, \mathbf{z}'_{\eta(i,j)}\right)$;
4. $\widetilde{h}_{i,j} = UnkeyedHash\left(\mathbf{z}'^{\star}_{i,j}, \mathbf{z}'^{\star}_{\eta(i,j)}, \ldots\right)$;
5. $\widehat{s}_{i,j} = FragileExtract_{\mathbf{k}_0}\left(\mathbf{z}'_{i,j}\right)$;
5. $\widehat{h}_{i,j} = Decrypt_{\mathbf{k}_{pub}}\left(\widehat{s}_{i,j}\right)$;
6. $\widehat{T}_{i,j} = \widehat{h}_{i,j} \ominus \widetilde{h}_{i,j}$;
7. end for
8. end for

Fig. 15. Fragile watermark extraction pseudo-code at the block level, asymmetrical version. The extracted signatures $\hat{s}_{i,j}$ are decrypted with public key $\mathbf{k}_{\text{pub}}$, and decrypted hash-codes compared ($\ominus$ operator). Each block i,j and neighbors $\eta(i,j)$ are validated where $\hat{T}_{i,j} = 0$.

One of the most attacked state-of-the-art approach is probably the pixel-wise Yeung–Mintzer scheme. First this algorithm is vulnerable to the *Multiple Stego-Image Attack*, when the same logo and the same key have been used for several images [17,18]. This attacks uses images marked with the same key and the same logo in order to build $M_1 \times M_2$ equations for 256 unknowns (M_1, M_2 being the image size), and on average only 2 images are needed to recover 90% of the logo bits. Fridrich et al. improved Yeung–Mintzer scheme by introducing a key-dependency of each current logo bit with a block of previously processed pixels, at the prices of a very high computational complexity and a loss of localization [16].

However, both original and improved versions are still vulnerable to the *Verification Device Attack*, when a verification device or center is available to the attacker (for example on-line), which accepts an image for verification and returns an image where tampered or untampered pixels are indicated [14]. To mount this attack, the attacker first produces any arbitrary image, and submit it to the verification center. He needs to modify the first pixel of the image, and to resubmit the modified image until that pixel get a non-tampered status. Then the attacker repeats this process pixel by pixel, in a row-by-row manner, until the whole image is claimed as untampered by the verification center. This attack takes on average only $2M_1M_2$ tries to succeed. Since all single-pixel authentication watermarking schemes are subject to this attack, block-wise algorithms based on secure cryptographic functions are preferable.

Regarding block-wise algorithms, it has been noticed very soon that schemes based on the hashing of non-overlapping and independent blocks like in

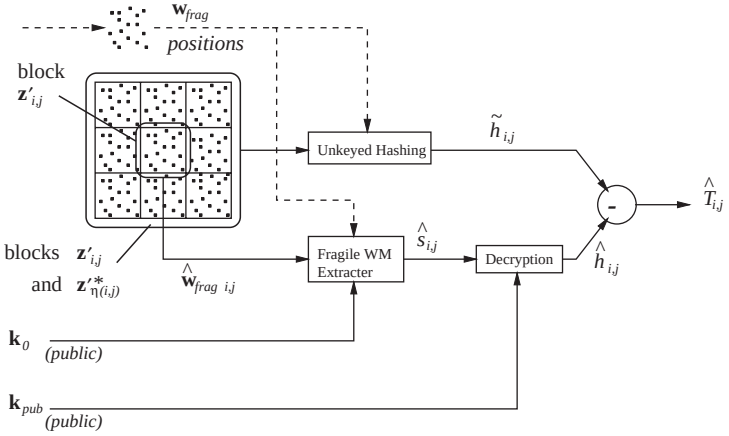

Fig. 16. Fragile watermark extraction for one block $\mathbf{z}'_{i,j}$, asymmetrical version. The extracted signature $\hat{s}_{i,j}$ is first decrypted with the user's public key $\mathbf{k}_{\text{pub}}$ to get the hash-code $\hat{h}_{i,j}$, which is compared with the recomputed hash-code $\tilde{h}_{i,j}$. The current block i,j and its neighbors are authenticated if $\hat{T}_{i,j} = 0$.

Wong's approach were also vulnerable to various tampering attacks, and especially to substitutions attacks described by Holliman and Memon [24] and Barreto et al. [2]. Other weaknesses could result from the design of the used cryptographic primitives and the way they are implemented, the signatures lengths, etc. Many of these attacks have been pointed out and advanced solutions proposed [2]. Below we describe the most significant attacks against block based schemes, and propose countermeasures against them.

4.2. Substitution attacks

The most simple of these attacks could consist in exchanging color planes in color images, in the case where each plane is hashed separately. Therefore, an obvious solution is to hash the three color planes together. Generally, the hashing and marking of independent blocks, without any other contextual information, is vulnerable to simple copy, cutting and pasting inside the same watermarked image: a few valid blocks copied from a suitable area can be pasted in another place in order to hide or to replace objects in the scene, generally without visible artifact; the only restriction for this attack to succeed is to respect the synchronization within the block division, which is usually not difficult when the block size is publicly known. The knowledge of the key is not required, since each block is independently authenticated by itself. If the copied areas come from other images, two cases can be distinguished: either the other images are not watermarked or are watermarked with keys which are different from the one of the attacked image, and the copied zones will be easily detected as tampered—we fall in the class of *simple tampering*; or the other images are all watermarked with the *same* key as the attacked image, and then the copied areas can be seen as authentic.

Therefore for most authentication schemes with localization, a security problem arises when the attacker can access a large number of images all watermarked with the same key. We call all attacks which aim at replacing parts of or the entire image, either within the image itself or using other sources protected by the same security parameters, *substitution attacks*. The different variants of substitution attacks, which will be described later, could be named: the *Cut-and-Paste Attack* when the properties of the authentication algorithm allows to construct images which are completely validated (usually at the block level), the *Birthday Attack* based on the cryptography theory and which results into the same effect, and the previously mentioned collage attack when the pasted zones are still validated but the boundaries between them detected.

In this framework, Wong's scheme in its original version as well as all independent block based schemes are vulnerable to an advanced substitution attack which can be mounted using vector-quantization (VQ) techniques [24], and which is known as the *Vector Quantization Attack* (VQ Attack), or the *Holliman-Memon Attack*. This is an enhancement of the Cut-and-Paste Attack which is able to construct an completely arbitrary image using the smallest possible areas—the blocks themselves. For this purpose the attacker first needs to gather L watermarked images, all marked with the same key, as given by

$$S_{\text{images}} = \{\mathbf{I}^{(1)}_{M_1,M_2}, \mathbf{I}^{(2)}_{M_1,M_2}, \ldots, \mathbf{I}^{(L)}_{M_1,M_2}\}. \tag{23}$$

For some algorithms, images from S_{images} should also contain the same embedded information (usually a binary bitmap logo), and also have the same size M_1, M_2 when the algorithm also hashes the image size. It is assumed that the attacker does not know the key, but knows the correct block size and synchronization. The L images from S_{images} are all divided into blocks as used by the fragile algorithm (of size t'_1, t'_2), resulting into the set of blocks:

$$S_{\text{blocks}} = \left\{ \mathbf{v}^{(1)}_r, \mathbf{v}^{(2)}_r, \ldots, \mathbf{v}^{(L)}_r, r := 1, 2, \ldots, \frac{M_1}{t'_1} \times \frac{M_2}{t'_2} \right\}. \tag{24}$$

These blocks are sorted in order to regroup together blocks corresponding to the same embedded logo and synchronization; this already is the case for all blocks having the same index r, if the division is made in the same order for all images. Let $\mathbf{I}'_{M_1,M_2}$ be the faked image that the attacker wants to make valid: the idea is to build a visual approximation $\mathbf{I}''_{M_1,M_2}$ of $\mathbf{I}'_{M_1,M_2}$ which will be fully authenticated. $\mathbf{I}'_{M_1,M_2}$ is then also divided into blocks $\{\mathbf{u}_s\}$, and for each block $\mathbf{u}_s$ a subset of blocks $\mathbf{v}^{(i)}_r$ from S_{blocks} is selected which correspond to the same bitmap logo and block synchronization; then the attacker replaces $\mathbf{u}_s$ by the block $\mathbf{v}^{(i)}_r$ which is

visually the closest to $\mathbf{u}_s$. This operations is repeated for all blocks $\mathbf{u}_s, s = 1, 2, \ldots$ to construct the approximated faked image $\mathbf{I}''_{M_1,M_2}$. This approach is merely the same as vector quantization, where we can think of a codebook as the collection of all blocks that would be correctly decoded. The gathering of a sufficient number of set of images marked with the same key is quite realistic, for example from a database; actually a small number of images (i.e. less than 10) is often sufficient to apply this attack, with very little visual artifact.

4.3. Cryptographic attacks

The underlying cryptographic primitives are obviously important too. Secure and well-studied cryptographic algorithms should be used, using keys of sufficient lengths. However, since the fragile watermarking is based on hash-codes and signatures, one important point to mention is the lengths of such hash-codes. Wong's scheme uses 64 bit hash-codes. It could be believed 64 bits are secure enough, since an exhaustive search would take $2^{64} \simeq 1.84 \times 10^{19}$ tries to find an input resulting into a given hash-code.

However, the possible weakness here rather consists in the possibility to find hash-code collisions, i.e. two blocks from different images (watermarked with the same key) which result into the same hash-code—which would help for generating a faked image. Here the problem is not to find an input which results into one fixed hash-code, but to find *two arbitrary* hash-codes which collide. Collision search can be performed on a set of images assuming they are all watermarked with the same key, without knowing the actual key by comparing the bits used for the embedding (the LSBs selected positions in our case). This problem is subject to the *Anniversary* or *Birthday Paradox* [35], which states that for hash-codes of N bits, the probability to obtain a collision is already equal to about 50% when only $\sqrt{N}$ random blocks are gathered. With hash-codes of 64 bits, only $2^{32} \simeq 4.29 \times 10^9$ block samples are needed to have already a probability of 0.5 to get a collision.

This property of hash-codes can help an attacker to mount the so-called *Birthday Attack* [2]. In a concrete example, an image of 1200×1800 pixels (2.16 Megapixels) can be divided into about 8400 complete blocks of size 16×16; therefore 511′306 images would contain the average 2^{32} complete blocks needed to mount a Birthday Attack if 64 bit hash-codes are used. The possible availability of large databases of images all protected with the same key makes this attack realistic. Wong's scheme and any algorithm using independent blocks and "short" signatures are potentially vulnerable to this attack. Of course the situation is even worse with smaller blocks, and one solution could be to increase the block size, at the price of a loss of locality. But even with the block size of 19×19 that we previously mentioned, the average number of needed images is 725′257, which is not very different from the previous number. Then to achieve a higher security level, it is recommended to use hash-codes of at least 128 bits: in this case the Birthday Attack would actually require 2^{64} block samples, which is the number that we expected at first, at the beginning of this sub-section.

4.4. Countermeasures against attacks

In the following, we discuss the countermeasures needed to defeat all known attacks regarding fragile/semi-fragile block-wise watermarking, as well as in the context of our hybrid concept. First the Birthday Attack could be simply avoided by using signatures of sufficient length. Secondly, our algorithm introduces inter-block dependencies to make substitutions attacks more difficult. Thirdly, we propose to hash additional global and local contextual information with each block, including the image size, the current block indexes, as well as other unique information for each image, and which could be embedded with the signatures. Since some of these security measures are redundant, we can select the most convenient ones depending on the targeted application. Finally, we take advantage of the hybrid approach in order to defeat the protocol attacks.

4.4.1. Hash-code Block Chaining (HBC)

Substitutions attacks are actually made possible due to the independence of blocks. The solution is therefore to introduce local dependencies as well as other local contextual information. First hashing the three planes together in color image prevents from color swapping. Secondly, hashing each block with some of its neighbors as proposed in Section 3 with HBC (Figs. 6 and 7), makes substitution attacks more difficult to mount (as mentioned HBC is equivalent to the

overlapping blocks of Coppersmith et al. [9]). Note that another way to introduce inter-regions dependencies was given by Celik et al.'s approach [6], based on a multi-level hierarchy of blocks and the calculation of block signatures in this hierarchy: while the lowest level of the hierarchy ensures better tamper localization, higher level block signatures increase the resistance against substitution attacks. However, we did not retain this idea in our practical implementation, since the case when low level hash-codes are verified while higher-level ones are not could be difficult to interpret in some scenarios.

4.4.2. Undeterministic Hash-code Block Chaining (HBC2)

Barreto et al. [2] further show that even with HBC, a fragile watermarking algorithm is not secure against a more sophisticated Cut-and-Paste Attack which considers the groups of blocks linked together instead of individual blocks. They call this attack the *Transplantation Attack*. The same problem stands for a more powerful version of the Birthday Attack, called by the same authors *Improved Birthday Attack*, and which also takes into account groups of dependent blocks. Increasing the number of chained blocks could make these attacks more difficult to perform but not impossible, since the attacker could simply consider larger groups of blocks. Therefore, Barreto et al. proposed to enhance HBC by chaining previous hash-codes in addition to neighboring blocks, combined with *undeterministic* signatures, calling this variant *Hash-code Block Chaining version 2* (HBC2). The modified version do the following: first, the hash function takes as input not only the neighboring blocks, but also neighboring (and already computed) signatures; secondly, "undeterministic signature" means that two strictly identical input hashed using the same key produce two randomly different signatures: consequently the assumption that images are all watermarked with the same key does not help anymore, since signatures always look random to an attacker. In the case of a public key cryptosystem, undeterministic signatures can easily be achieved by using DSA [41] in place of RSA; however, note that any deterministic hash function may be turned into an undeterministic one by using a *random salt*, taken as input and appended to the signature. The salt consists in a random string r which is appended to the hash-code h or the signature s; at the embedding stage r is included in the input of the hash function as

$$h = H(r,\ldots) \quad \text{or} \quad s = S(r,\ldots). \tag{25}$$

and *both* r and h (or s) are embedded as (r,h) (or (r,s)), since this salt r is needed at the verification stage.

4.4.3. Global and local contextual information

Unfortunately, the previously given solutions are still not enough to ensure full resistance against the collage attack previously described, when areas large enough are copied and pasted: only the boundaries between areas coming from different images are detected as tampered, but nothing can tell us that these different areas come from different sources. We could then think of hashing the binary representation of blocks indexes (i,j), or the image size (M_1,M_2) as well. However, a successful collage attack is still possible by preserving the blocks original positions and by using images of the same size. This situation will be illustrated by Fig. 24 in Section 6.

A second solution we can think of and that was proposed by Wong and Memon [65] is to hash some global additional information, chosen unique for each image, such as an image identification number (ID). The consequence of this method is that given an image ID, only the corresponding areas will be authenticated, but the pasted areas (coming with a different ID) will be rejected. Any additional global and local information hashed is then represented by the "..." in pseudo-codes of Figs. 5, 9, 12, and 15, line 4 (Section 3).

4.4.4. Embedded unique stamps

Since any hashed additional information is also needed at the verification stage, it should be stored with its corresponding key, which could make the images ID method above inconvenient for many applications. A solution to this problem first proposed by Fridrich [14,16] consists of storing this additional ID within the signatures themselves as *stamps* or *time-stamps* in encrypted form. Such a stamp can also be used to carry useful additional information, such as the original block position, helping us to determine the original locations of areas which has been cropped in the source image. The stamp does

not need to be stored separately from the image anymore, and can even be random, just acting as an additional salt (Eq. (25)) for the signature process. It can be the same for all blocks of the same image, but it should be at least different from one image to another in order to resist against the previously described collage attack (which use different source images). However, including locally dependent information (such as a block index) in the stamp could be another mean to resist against substitution attacks and more precisely the collage attack. We actually propose to use a time-stamp indicating the date and time of the watermark embedding, plus other optional information if necessary depending on the targeted application.

The time-stamp is included in the input of the hash functions, and at the verification stage it should be decrypted before recomputing the signatures. In this approach, signatures will be authenticated again in every copied area again, but the extraction of different time-stamps can alert us that a collage attack probably occurred, as illustrated by Fig. 24 of Section 6. It becomes then possible to count the number of copied areas and to localize them.

4.4.5. Jointly exploiting the robust watermark

Finally, the proposed hybrid watermarking concept gives us an opportunity for the interpretation of the decoded information, which robust systems or fragile/semi-fragile systems do not have separately. Therefore, we propose to use the results of watermark extraction from both the robust and the fragile parts, in addition to the previously detailed security recommendations. Consequently, a more precise verification diagnostic can be given with respect to protocol attacks.

First, the detection of the collage attack can be enhanced, since the robust algorithm could either fail, or decode different independent messages correctly when the RBA-resistant version of our robust method is used [59] (coming from the fact that the RBA robust version extracts the watermark at the local level). This feature corresponds to the item 3 of the decision enumeration given at the end of Section 3. When used jointly with the stamp/time-stamp feature, we have then another criteria to detect such attacks; further, if the same robust message was embedded in all parts (resulting into *only one* decoded robust message), the extracted stamps can still distinguish the different parts.

Secondly, as concluded in Section 3, joint robust and fragile watermarking is resistant to the copy attack: as we pointed it out, it is generally easy to estimate the robust watermark and to copy it into another marked or unmarked image. The wrong robust watermark will then be decoded from the target image, but the fragile watermark will fail. Even if the fragile part is also copied to the destination image (e.g. by copying the LSB planes), the signatures would not match since the input of the hash functions are different. Therefore, the copy attack can be detected by the decision item 2 at the end of Section 3.

4.4.6. Summary of security measures

Consequently, to conclude this section, we can summarize the main security measures that could be implemented by the items below:

1. *Use hash-codes of sufficient lengths*: Hash-codes of at least 128 bits should be used, and we propose SHA-1 or HMAC based on SHA-1 (160 bits) in order to defeat the Birthday Attack. For hash-codes of more than 128 bits it is also possible to keep only the 128 leftmost bits to save space.
2. *Chain blocks in hash-coding*: For each block compute the hash-code of this blocks plus neighboring blocks (Fig. 6), in order to make the Cut-and-Paste and collage attacks more difficult. This is HBC.
3. *Chain signatures in hash-coding*: In addition to HBC of item 2, make hash-codes also dependent from at least one previously computed signature.
4. *Use undeterministic hash-coding*: Undeterministic hash-codes or signatures, jointly used with item 3 above, defeats the more advanced Transplantation Attack and Improved Birthday Attack. Items 3 and 4 together are HBC2.
5. *Hash extra global and local information*: Hashing the indexes i, j of the current block makes block synchronization necessary for an attack to succeed; hashing the image's size M_1, M_2 restrict attacks to images of the same size; hashing a unique ID for each image makes the collage attack merely infeasible, but may be not practicable in many applications.
6. *Hash and embed a unique stamp*: Hash a unique stamp for each image (e.g. a random ID or date

and time), which is embedded beside the signatures, to defeat the collage attack, and to allow to distinguish and localize pasted areas; can also carry other useful information. This method can replace the non-embedded ID approach of item 5.

7. *Joint information from robust and fragile parts*: Analyzing the decoding of both parts gives us a more powerful diagnostic to defeat protocol attacks: first to confirm the detection of a collage attack and to separate the different areas, and secondly to detect the copy attack regarding the robust part.

Therefore, by first using these suggested countermeasures for the fragile part, and secondly by taking advantage of the hybrid approach by exploiting the additional information coming from the robust part, we can expect a highly robust and secure approach for both copyright protection, authentication and tamper proofing.

4.5. Payload for asymmetrical signatures

We want to mention that watermarking for copyright, authentication or tamper proofing applications generally requires that a limited payload is embedded: first in order to preserve the robustness for the robust part $\mathbf{w}$, and secondly in order to limit the block size for the fragile part $\mathbf{w}_{\text{frag}}$ to preserve acceptable localization capabilities in tamper detection. This is not a problem for symmetrical authentication/tamper proofing schemes with the actually proposed hash-code lengths of 128 or 160 bits above. But in the case of public key signatures, asymmetrical algorithms may require a much larger payload to achieve acceptable security of embedded signatures; this is especially the case for RSA, which requires encryption/decryption keys of at least 1024 bits for a sufficient level of security, resulting into signatures of the same length.

However, more appropriate algorithms could be used instead like DSA: for DSA/DSS key lengths from 512 to 1024 bits are suggested for use with the same security as in RSA, and produced signatures are only 320 bits long. Further, we can even use Elliptic Curve (EC) cryptography, proposed independently by Miller and Koblitz [28,36]; EC cryptography can be adapted to many asymmetrical algorithms, and especially to the DSA as ECDSA [26,41]. Today it is believed that a 256 bit EC cryptosystem could achieve the same level of security as a 1024 bit RSA, representing a reasonable payload in watermarking applications.

5. Hybrid semi-fragile extension robust to media conversion

The hybrid watermarking scheme described above is based an a strictly *fragile* embedding of authentication codes. Consequently any modification is detected by the fragile part and leads to the rejection of the corresponding block as non authentic; no lossy compression nor image enhancement is allowed. Further, we have no way to measure the level of the applied alteration. An innocent modification cannot be distinguished from severe tampering, nor even from a copy attack when the robust part is still correctly extracted. Therefore, the hybrid approach based on fragile watermarking is suitable only for the protection of *digitally* stored or transmitted documents, excluding any modification. [3]

At the opposite, the robust part used in our hybrid approach is resistant to a large range of distortions, including signal degradation, geometrical distortions, and printing/rescanning. Such difference of robustness between both types of watermark is not ideal when joint in a hybrid approach. Even if authentication watermarks do not need to achieve the same level of resistance as robust watermarks, a large variety of applications require some robustness to an certain amount of "acceptable" modification of the visual data (such as good-quality lossy compression or limited contrast enhancement). The protection of analog media such as hard copy documents further requires that the authentication part resists printing and rescanning too.

Therefore, we propose here an extension of our hybrid approach to a more tolerant one based on *semi-fragile* watermarking (a concept introduced in Section 1). The semi-fragile watermark should

[3] An obvious application of a strictly fragile watermark could be the protection of medical images, for which no modification is permitted; however, it should be possible (knowing the key) to remove all watermarking information in order to revert to the original image: such a scheme is known as *invertible watermarking*.

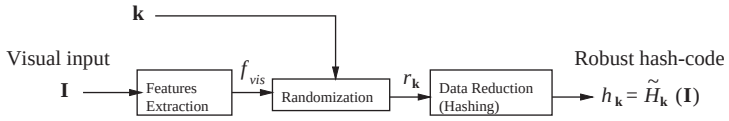

Fig. 17. Robust visual hashing $\tilde{H}$ steps. Visual features f_{vis} are extracted which are robust to some level of permitted distortions from the visual input data $\mathbf{I}$; they are then randomized based on a key $\mathbf{k}$ giving $r_{\mathbf{k}}$; finally they are reduced in size to form the hash-code $h_{\mathbf{k}} = \tilde{H}_{\mathbf{k}}(\mathbf{I})$.

fulfill two conditions: first, visually slightly modified inputs should produce the same or almost the same hash-codes, which means that *robust visual hashing* should be used instead of classical cryptographic hash functions. Secondly, the highly sensitive LSB modulation approach has to be replaced by a more robust embedding.

5.1. Robust visual hashing

The idea of robust visual hashing is to generate a key-dependent secure digest which changes continuously with the input, differing at most by a small number of bits for two distinct but perceptually equivalent inputs. Robust hashing can be seen as a three-steps operation: first, *features extraction* which resists the transformations that we define as acceptable; secondly, a (generally key-dependent) *randomization* process on these features in order to achieve security; thirdly, a data reduction step which maps the randomized information to a shorter bit string representing the input data. Fig. 17 summarize these steps.

5.1.1. Features extraction

For the features f_{vis} extraction step from the visual input $\boldsymbol{I}$, we have to define what is an "acceptable" alteration, and which inputs can be considered as "perceptually equivalent". This aspect concerns both the type and the level of distortion we want to allow, and is obviously dependent on the targeted application. Permitted distortions could include signal processing changes such as slight lossy compression, signal fading, noise addition, gray-scale conversion, etc. as well as some classes of geometrical distortions. The selected features should be robust and invariant to the allowed distortions. Early tolerant visual hashing for images have been proposed by Schneider and Chang [50] which uses features like edges, color/gray-scale histograms or DCT, and by Brandt and Lin [5] which are also robust to translation, rotation, and scaling. Xie and Arce [67] extract edges information using the DWT. Bhattacharjee and Kutter [4] extract perceptually interesting feature points that are not embedded within the image but are stored separately. Later Hel-Or et al. [22] proposed geometric hashing based on salient points and voting algorithm, and Fridrich [13] proposes a function using the low-pass of DCT coefficients, which can be made invariant to translation, scaling and rotation using the Fourier-Mellin transform [44].

5.1.2. Randomization and data reduction

The randomization step (mapping f_{vis} to $r_{\mathbf{k}}$, generally based on a key $\mathbf{k}$) is essential, since the generated code should keep the same properties as classical cryptographic hash function beyond their continuous character: codes should be unpredictable for random inputs, and two completely different inputs should result into uncorrelated codes. In the case of keyed hashing, two different keys (differing even by a single bit) should also produce totally different codes. Fridrich [13] uses key-dependent random matrices to randomize low-pass DCT, and Venkatesan et al. [53] propose a random tiling of the wavelet transform (DWT) of the input prior to features extraction.

Finally the data reduction step (mapping $r_{\mathbf{k}}$ to $h_{\mathbf{k}} = \tilde{H}_{\mathbf{k}}(I)$) is the irreversible data compression which reduces the length of the encoded features to a compact digest code. Both the randomization and the data reduction steps should preserve the continuous property of the input features, and for this purpose these two last steps could be done together rather than separately.

The tolerance to acceptable visual distortion of the input as well as the sensitivity to the key should be fulfilled by the three described steps, and the requirements of Eq. (16) should be replaced by the

following [13]:

$$\begin{aligned}&\mathbf{I} \sim \mathbf{I}' \Rightarrow \tilde{H}_{\mathbf{k}}(\mathbf{I}) \approx \tilde{H}_{\mathbf{k}}(\mathbf{I}'),\\ &\mathbf{I} \neq\!\!\neq \mathbf{I}' \Rightarrow \tilde{H}_{\mathbf{k}}(\mathbf{I}) \neq\!\!\neq \tilde{H}_{\mathbf{k}}(\mathbf{I}'),\\ &\mathbf{k} \neq \mathbf{k}' \Rightarrow \tilde{H}_{\mathbf{k}}(\mathbf{I}) \neq\!\!\neq \tilde{H}_{\mathbf{k}'}(\mathbf{I}'),\end{aligned} \tag{26}$$

where $\mathbf{I}$ and $\mathbf{I}'$ are two different visual inputs, $\tilde{H}$ is a key-dependent visual hash function, and $\mathbf{k}$ a random key. The symbol "$\sim$" means that the inputs are visually similar to each other, and "$\approx$" that codes differ at most by a small percentage of their bits. "$\neq\!\!\neq$" indicates visually completely different data, as well as completely mismatching codes (differing by about 50% of their bits). The verification is then done by counting the percentage of mismatching bits with a threshold τ_{vis} representing the amount of allowed distortion.

In another variant, visually equivalent inputs generate exactly *the same* hash-code. This could be done using an error correction code (ECC) to *decode* the encoded features bit strings as the data reduction step [53]: first, ECC decoding ensures irreversible data compression; secondly, such reduction is robust in the sense that bit strings resulting from similar inputs can be mapped to the same code, assuming that they are inside the same ECC decoding sphere with high probability.[4] This variant can be formalized by replacing the first line in Eq. (26) by

$$\mathbf{I} \sim \mathbf{I}' \Rightarrow \tilde{H}_{\mathbf{k}}(\mathbf{I}) = \tilde{H}_{\mathbf{k}}(\mathbf{I}'). \tag{27}$$

In this case, the verification can be again processed by strict hash-codes comparison as for fragile watermarking, then the allowable level of distortion depends only on the features extraction step.

5.2. Semi-fragile watermark embedding

The generated robust signatures $s_{i,j}$ has then to be embedded as a semi-fragile watermark $\mathbf{w}_{\text{sfrag}}$ within each i,j-block $\mathbf{y}_{i,j}$ already watermarked with the robust watermark $\mathbf{w}$. Obviously, the robustness of $\mathbf{w}_{\text{sfrag}}$ should at least correspond to the level of tolerance of the robust visual hashing $\tilde{H}$ used; the global robustness of the semi-fragile part then corresponds to the less robust component from $\tilde{H}$ and $\mathbf{w}_{\text{sfrag}}$. In particular, if $\tilde{H}$ is invariant to some geometrical distortions such as rotations and scaling, then $\mathbf{w}_{\text{sfrag}}$ should resist the same.

Therefore $\mathbf{w}_{\text{sfrag}}$ is block-wise embedded, for which two approaches of embedding could be used:

1. *Include* $\mathbf{w}_{\text{sfrag}}$ *in* $\mathbf{w}$: In a hybrid approach it is possible to take advantage of the presence of the robust part to include signatures $s_{i,j}$ in its payload, beside the robust message $\mathbf{b}$. The main advantage is that no additional algorithm is needed, and the $s_{i,j}$ benefit from the resistance of $\mathbf{w}$, at least at the robust block level. Further, less robustness can be achieved for the $s_{i,j}$ than for $\mathbf{b}$ by using less encoding bits. The main drawback of this method is that robust blocks may have size increased in order to achieve higher payload, reducing the localization capability of the tamper proofing part as well as the robustness of the robust watermark.
2. *Embed* $\mathbf{w}_{\text{sfrag}}$ *using a specific approach*: The $s_{i,j}$ can be embedded using a robust approach different from the $\mathbf{w}$ embedder, in a way which interferes as less as possible with $\mathbf{w}$. For example, for the authentication of hard copy documents, resistance to printing and rescanning is desirable: in this case for host interference cancellation one can use the Quantization Index Modulation (QIM) [8],[5] or the Dither Modulation (DM) [7] if only printed documents are targeted. Currently, our research is in progress to design joint source/channel coding approach for hierarchical data embedding.

In both cases, as for the strictly fragile version $\mathbf{w}_{\text{sfrag}}$ could be embedded in the free and reference watermark ($\mathbf{w}_{\text{empty}}$ and $\mathbf{w}_{\text{ref}}$) positions: first in order to limit the need for additional payload and resulting block size increase, and secondly in order to achieve almost orthogonal robust and semi-fragile parts.

We further assume the tolerance of $\tilde{H}$ used to the presence $\mathbf{w}_{\text{sfrag}}$ itself (which cannot be including in the hashing input during the embedding stage). Then embedding positions do not need to be excluded from the hashing input like in the strictly fragile approach. If needed, some prefiltering or noise removal could be

[4] Venkatesan et al. propose to use Reed–Muller code of the first order, however many others ECCs could be used; further, 2 or more inverse ECCs could be sequentially applied and/or decoding rate chosen to achieve the wanted length reduction.

[5] Note that the LSB modulation approach used in our strictly fragile approach is a particular case of QIM.

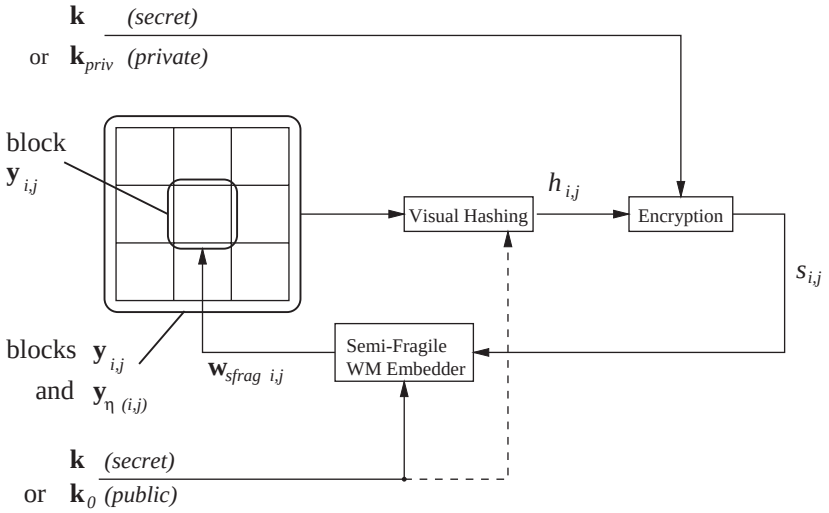

Fig. 18. Semi-fragile embedding at the block level. A robust visual hash-code is computed from the current block and its neighbors, and encrypted to a signature $s_{i,j}$ which is embedded in a *semi-fragile* manner. The visual hashing may depend on a key for randomization. The symmetrical case uses the secret key $\mathbf{k}$, while the asymmetrical case uses the private key $\mathbf{k}_{\text{priv}}$ and the auxiliary shared key $\mathbf{k}_0$.

included in the features extraction of $\tilde{H}$ to ensure this tolerance.

5.3. Hybrid semi-fragile approach

The hybrid semi-fragile embedding and extraction processes are mainly the same as for the strictly fragile approaches. However, we propose to use feature extraction which is robust to signal processing alterations, such as compression (with high quality), limited blurring or sharpening, as well as noise addition.

We suggest first the preprocessing of input blocks such as gray-scale conversion, histogram equalization, and sharpening. Then the Fourier transform (DFT) or wavelet transform (DWT) can be used for the extraction of frequency-dependent or resolution-dependent features. We propose not to ensure any invariance to geometrical transformations: one possibility is to let the robust part perform the estimation of the geometrical distortion at both the global level (affine transforms) and the local level (RBA), and to compensate them *before* the tamper proofing verification part.

Otherwise the general approach at the image level is analog to the strictly fragile version illustrated in Figs. 4 and 11 (for the embedding), and 8 and 14 (for the extraction/verification, except that the compensated image is passed to the verification box). The semi-fragile watermark embedding and extraction are shown at the block level by Figs. 18 and 19.

Then Eq. (19) should be replaced by the calculation of a continuous authenticity factor $\hat{T}'_{i,j} \in [0,1]$ for each block $\mathbf{z}'_{i,j}$ and neighbors $\mathbf{z}'_{\eta(i,j)}$:

$$\hat{T'}_{i,j} = \text{dist}(\tilde{h}_{i,j}, \hat{h}_{i,j})_{\text{normalized}}, \tag{28}$$

where $\text{dist}(h,h')_{\text{normalized}}$ is a normalized distance metrics varying from 1 when h and h' are completely uncorrelated, to 0 when we have strictly $h = h'$. For binary continuous hash-codes h and h' of equal number of bits N, such a distance can be defined for example as:

$$\text{dist}(h,h')_{\text{normalized}} = 2\min\left(\frac{1}{N}\sum_{k=1}^{N} h(k) \oplus h'(k), \frac{1}{2}\right), \tag{29}$$

where $\oplus$ is the bitwise exclusive-or (XOR) operator, and $h(k)$, $h'(k)$ the individual bits of h and h', respectively. Finally, a binary decision $T_{i,j}$ about the local tampering in i,j can be defined equivalently to

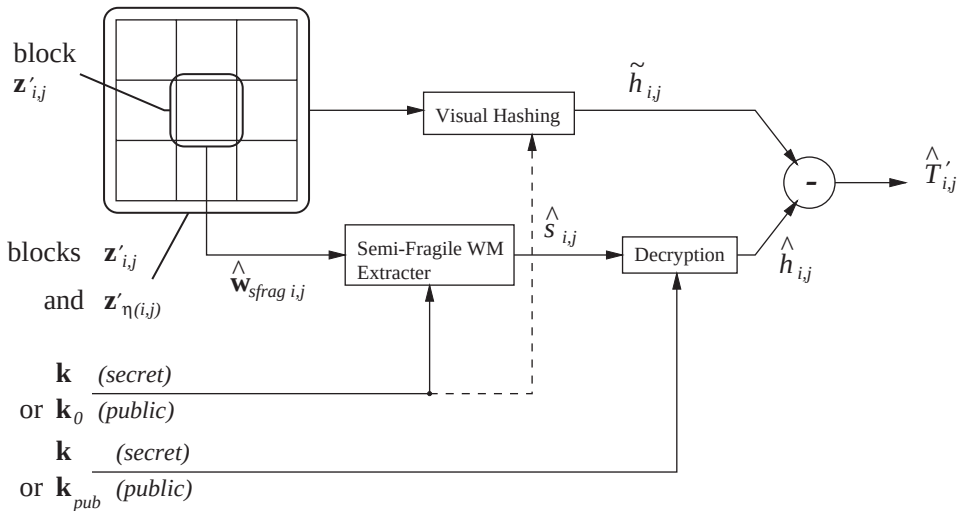

Fig. 19. Semi-fragile extraction at the block level. The robust visual hash-code is computed again giving $\tilde{h}_{i,j}$; the embedded signature $\hat{s}_{i,j}$ is extracted and decrypted back to $\hat{h}_{i,j}$, and the comparison done in a continuous manner resulting into $\hat{T}'_{i,j}$—a normalized distance between $\tilde{h}_{i,j}$ and $\hat{h}_{i,j}$. Again $\mathbf{k}$ is used in the symmetrical case, while $\mathbf{k}_{\text{pub}}$ (the public key) and $\mathbf{k}_0$ (the auxiliary shared key) are used in the asymmetrical case.

Eq. (19) using the fixed threshold τ_{vis}, $0 < \tau_{\text{vis}} \leqslant 1$ as

$$\hat{T}_{i,j} = \begin{cases} 0 & \text{if } \hat{T}'_{i,j} \leqslant \tau_{\text{vis}}, \\ 1 & \text{otherwise.} \end{cases} \tag{30}$$

Therefore, we can still use Eq. (20) to compute the global authenticity factor A_T for the image, and apply the same decision rules as for the fragile version; then τ_{vis} sets the maximum amount of acceptable visual distortions. Further, we cannot only authenticate (partially or totally) an image in a tolerant manner, but also characterize the level of locally applied distortion based on the $\hat{T}'_{i,j}$. In a hybrid approach, this comes in addition to the reference watermark $\mathbf{w}_{\text{ref}}$ contained in the robust part $\mathbf{w}$ and used to estimate its fading.

5.4. Fragile vs. semi-fragile approaches

Fragile watermarking can be made secure and allows good localization, but is suitable for digital documents only and does not allow any distortions. Semi-fragile watermarking can be tolerant to limited or innocent distortions such image compression, enhancement, or digital-analog conversion, but the embedding blocks may have to be increased in size, reducing the localization properties of our approach; moreover, the security level of robust hashing with continuous behavior is probably not completely proven today and is still an open problem.

Consequently, we propose two variants depending on the targeted application: first, a *strict* hybrid watermarking as described in the previous sections–based on cryptographic hash functions and fragile LSB modulation; secondly, a *tolerant* hybrid watermarking, using robust or visual hash-codes and robust embedding within the joint robust watermarking. While the strict variant allows to authenticate digital documents only, the tolerant variant would be mostly suitable in the case of digital/analog conversion and hard-copy documents such as banknotes, identity cards, passports, or value papers.

6. Applications and results

To demonstrate the main functionalities of the proposed approach, we first briefly outline the main applications and goals of robust, fragile and semi-fragile watermarks, and the corresponding attacks. Secondly we experimentally studied the robust version of the algorithm, demonstrating its performance and its

resistance against a set of standard attacks. The size of the blocks of the robust part was 19×19 pixels (upsampled and flipped to 76×76 sized blocks—Section 2). All images used for the testing were watermarked with a Peak Signal-to-Noise Ratio (PSNR) of about 38 dB with respect to their originals. Then we experimented the fragile part of our method in concrete scenarios. The fragile watermarking block size was 19×19 pixels. We first tested the addition or removal of objects in an image. Then we targeted the ability of the hybrid approach to defeat the copy attack, as well as the collage attack under various conditions. The authentication/tamper proofing part was tested in its fragile and symmetrical version only.

6.1. Goals and scenarios

Copyright protection is clearly the main class of applications targeted by researches on digital watermarking in the year 1990. For this purpose highly robust watermarks have been developed. The embedded message gives some semantic information about the host image, such as its owner, its creator, intellectual property, its rights holder, etc., and is used to identify the protected document. The robust watermark message is either an identification number, a short text, or an index pointing into a copyright database. It can also be used for copy control mechanisms, allowing devices (for example a CD or a DVD reader) to automatically accept or refuse the copy or the playback of protected material [37].

However robust watermarking has many other applications which are not related to copyright protection. A typical example is the "Media Bridge" [1] of Digimarc corporation, where the embedded message redirects on a Web site when a printed version of the stego image is scanned or presented in front of a camera. The message can act as an embedded labelling for various purposes such as advertisement, the price or reference of the represented commercial product, or any other descriptive information. This leads to the concept of *smart images*, of which one typical example consists of the embedding of a pointer to the original image into a database. It becomes then possible from a severely damaged or cropped hard-copy image to retrieve its original version, when it is very difficult to figure out from the damaged copy what the original was. Robust watermark can even be used to carry specific commands for *intelligent devices*, for example parameters settings for the playback of a broadcasted work. Another application is *document tracking*, which can be used for example to control when and how many times a work has been diffused, such as advertisements or movies on television networks.

Fragile and semi-fragile watermarking was then rapidly proposed for authentication and tamper proofing. Authentication aims at checking the authenticity of a document and especially of its source, while tamper proofing is used to detect unauthorized modifications. There is clearly a huge need for authentication and tamper proofing not only for digital media, but also for a large variety of physical objects, hard-copy documents and valued papers. This includes identity cards, passports, official documents, and banknotes, for which embedded information or signatures can be a low-cost alternative to more classical solutions based on special physical features. Semi-fragile watermarking based technologies could then require a simple scanner instead of more sophisticated devices. Regarding malicious tampering, three practical examples could be: alteration of pictures from digital cameras to forge faked evidences to be used in trials; modification of digitized medical images by patients for fraudulent declaration to healthcare insurances; and fabrication of faked identity cards, for example by replacing the photograph by another one. Then fragile and semi-fragile watermarks aim at making falsifications and unauthorized modifications easy to detect and characterize.

6.1.1. Scenarios of protocol attacks

Since generally simple local alterations are easily detected, we pointed out in this article the weakness of many fragile schemes regarding substitution attacks using images protected with the same key. These attacks, by making faked material wrongly authenticated, make the main goal of tamper proofing missed, but they can be defeated by the described security measures. However, a successful collage attack makes it difficult to figure out the actual authenticity of the copied regions with respect to each other in the context of the composed image, and cannot be distinguished with certainty from simple tampering.

Regarding robust watermarking, the copy attack not only causes the ambiguity in copyright protection

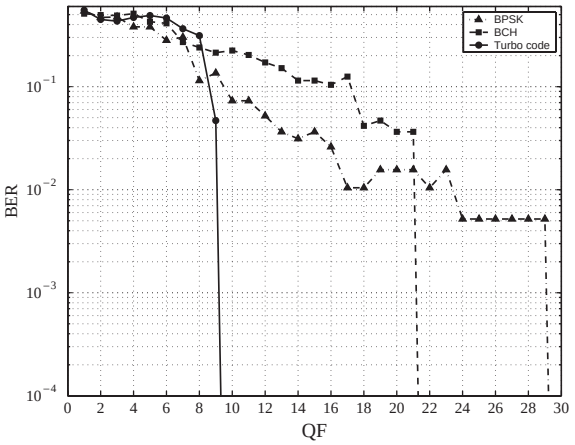

Fig. 20. BER of the decoded watermark message under JPEG lossy compression, for BPSK, BCH, and Turbo codes encoding. Turbo codes achieves the highest performance up to a compression QF of 10, while BPSK already gives errors for QF less than 30.

applications. Schemes vulnerable to this attack are ineffective for document tracking or smart images scenarios: how to be sure that a movie or an advertisement was actually broadcasted? How to guaranty that the pointed description or original image really corresponds to the copy which was scanned? Intelligent devices could receive malicious commands, a threat known under the name of *stego viruses*. Finally passports or identity cards cannot be efficiently secured if a watermark can be copied from a valid document and re-embedded into the faked one without being detected. Therefore, an hybrid approach which is able to defeat these protocol attacks is of great advantage.

6.2. The robust watermarking part

The main issue of robust watermarking is the accuracy with which the embedded message can be decoded after a certain level of attacks. To illustrate this aspect, Fig. 20 shows the bit-error-rate (BER) that we receive at the decoder from a lossy JPEG compressed stego material with respect to QF varying from 100 (the highest quality) down to 1. This testing shows how the use of an efficient ECC can drastically increase the performance of a robust watermarking scheme. The plot shows the average decoding BER from 6 test images of 512×512 pixels marked with a PSNR of about 38 dB. We tested the Binary Pulse Antipodal Signaling (BPSK), BCH encoding, and Turbo codes. BPSK achieved the worse performance, with errors already occurring with JPEG compression at a QF of about 30. At the opposite the Turbo codes (which is actually used in our scheme) allowed us to approximate the best the watermarking channel capacity even with a QF of 10.

Table 1
Averaged results of system performance according to Stirmark 3.1.

Stirmark attack	Averaged score
Signal enhancement	1.00
Compression (JPEG/GIF)	0.99
Scaling	1.00
Cropping	0.99
Shearing	1.00
Rotation (auto-crop, auto-scale)	0.99
Column and line removal	1.00
Flip	1.00
Random Bending Attack (RBA)	1.00

The success of the decoding can be assessed by the decoder based on the check codes and on the reference watermark accuracy as discussed in Section 2. We did not perform testing regarding the false alarm rate in assessing a successful decoding, the number of test images being too small. According to Eqs. (14) and (15) this probability can be around 10^{-9} to 10^{-12}, depending on the lengths of the reference bits and of the message check code.

We have tested our method according to the set of experimental attacks defined by the *Stirmark 3.1* benchmark [47], using the 6 standard images proposed by the Stirmark team (of sizes from 512×512 to 600×800) and the recommended embedding PSNR of 38 dB. The results are presented in Table 1. As argued in Section 2, since ECC encoding is used for each experiment a decoding is considered as successful if and only if *all* bits of the message were correctly decoded for a given test. For each group of tests, marks are ratios of correct decoding of the robust message over the number of testing of that group, thus varying from 0 (no watermark was decoded) to 1 (all decoding successful). The watermark was also successfully decoded from randomly distorted images (resulting from RBA), significantly rising the total score (averaged over all experiments) up to the ratio of 0.993 over 1. Moreover, the performances were the same

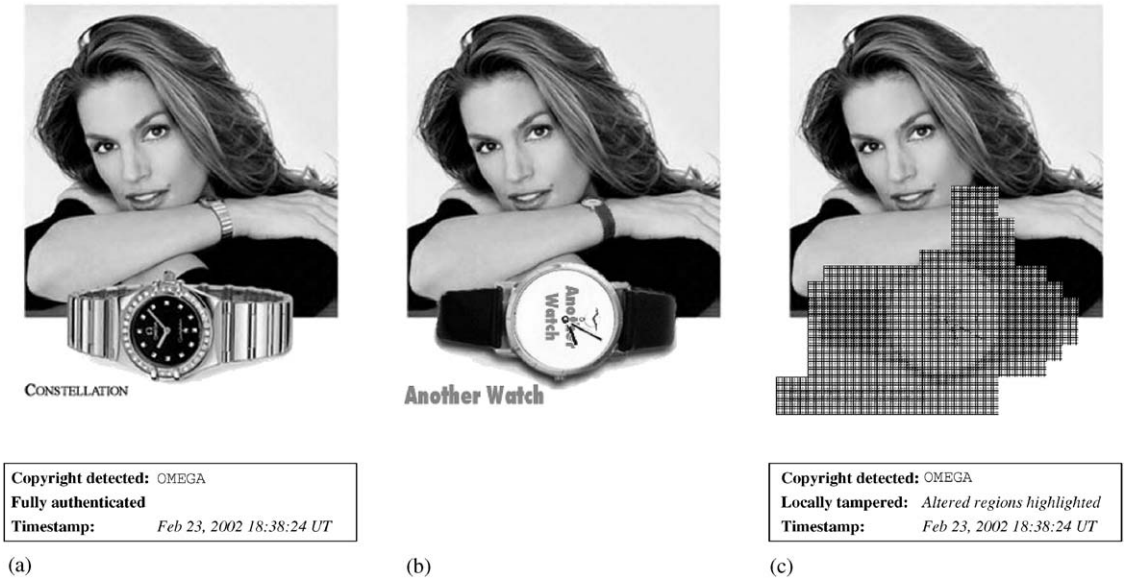

Fig. 21. Local tampering experiment. (a) Watermarked "Cindy" with PSNR = 38 dB, an advertisement for the OMEGA^{TM} Constellation watch; (b) tampered watermarked "Cindy" where the watch is replaced by another one; (c) local tampering is detected and the modified zones highlighted.

with or without the presence of the fragile part of the watermark. More detailed results of our robust watermarking approach can be found in Voloshynovskiy et al. work [55]. This is the highest score obtained for a known multibit watermarking algorithm in year 2002.

6.3. Local tampering

The first experiment illustrated in Fig. 21 shows the scenario where an advertisement poster is tampered in favor of a competitor product. First, we embedded the hybrid watermark into the 512×633 pixels sized image "Cindy", showing Cindy Crawford advertising an OMEGA^{TM} watch of the Constellation collection.[6] Then we saved this image without any compression in a non-lossy mode. The robust watermark was successfully decoded showing the trademark of the advertised watch "OMEGA", and the image fully authenticated (a). The embedded stamp carries the time-stamp indicating the time of embedding. Then as malicious persons we locally modified the image by replacing the watch by another one of arbitrary mark "Another Watch",[7] and we updated also the text on the advertisement to reflect this change (b). Except these changes, no other distortion was introduced, and the tampered image was saved again in a non-lossy mode. The faked poster now advertises "Another Watch". However the hybrid scheme detected and highlighted the changed regions by a dashed grid (c), and the extracted copyright still indicated the trademark "OMEGA": then we cannot only detect the modification, but we can also get an idea of the motivation of the tampering and retrieve the original trademark.

The result was the same after simple transformations such as horizontal mirroring, vertical flipping, and 90° rotations—which are fully conservative operations (i.e. free of interpolation errors): this capability was implemented by just applying the verification process for any of these 8 possible orientations. Since we tested here a strictly fragile version of our algorithm, it is clear that lossy compression as well as geometrical transforms would make the signatures fail, and consequently the watermarked image claimed as fully non-authentic. For such gray-scale or color images

[6] With the written permission of OMEGA Ltd., http://www.omegawatches.com.

[7] "Another Watch" is a fictive trademark.

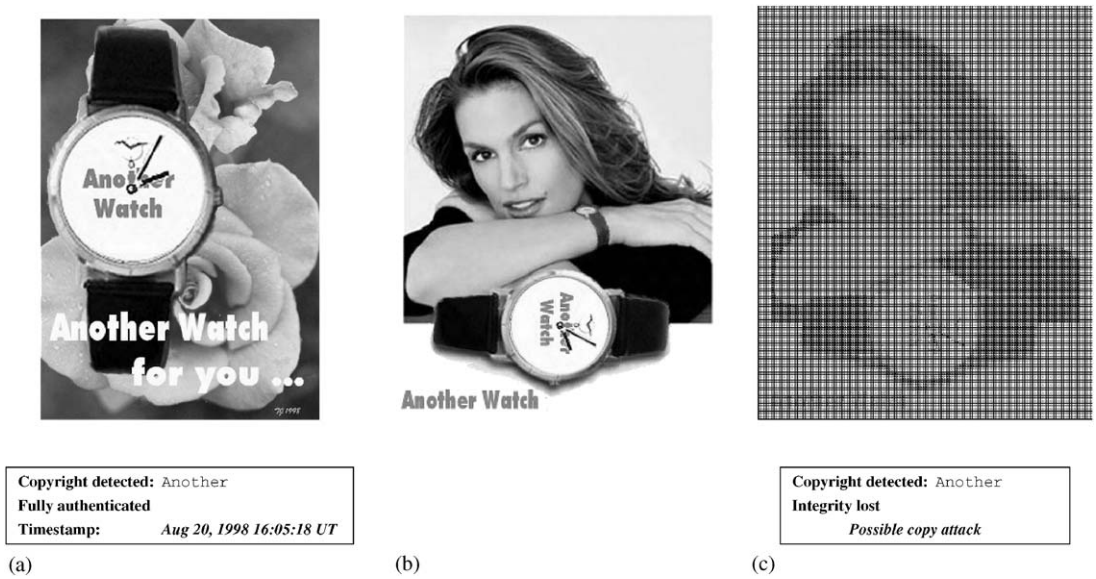

Fig. 22. Copy attack experiment. (a) The authentic and watermarked poster advertising "Another Watch"; (b) the watermark is copied from "Another Watch" to the faked advertisement forged in Fig. 21; (c) the robust watermark now claims "Another", but the document integrity is completely lost: a copy attack can then be suspected.

of about 500 × 600 pixels, the execution time of the signatures verification stage is in average 0.5 s on a Pentium III computer with 512 Mb RAM/600 MHz processor frequency, and not more than 2 s in the worst case when all signatures fail.

6.4. Protocol attacks

The following experiments illustrate how the copy and the collage attacks can lead to ambiguities making the scheme useless in some applications. It is then demonstrated how the hybrid approach can solve these ambiguities.

6.4.1. Copy attack

The second experiment that we performed outlines a possible consequence of the ambiguity resulting from a copy attack and how it is solved. This testing is shown in Fig. 22. We suppose that the original advertisement for the other watch of previous experiment is also available with its own watermark (a). [8]

Since the faked image forged in Fig. 21b still indicates the original trademark, we wanted to change the copyright to reflect the replacement watch instead. Then we copied the watermark from the latter advertisement and re-embedded it into the faked image (Fig. 22b). The watermark detector extracted the copyright "Another" as expected, but the authentication part of the hybrid approach fully rejected the image as non authentic, claiming a possible copy attack. Consequently, we can still know that this image was faked. Here the time-stamp was not available, since it was embedded within the authentication part. Copying the embedded signatures within the robust part (e.g. copying the LSB plane also) does not help for the attack, since anyway the recomputed signatures completely mismatch.

6.4.2. Collage attack

The third experiment targeted the collage attack, shown in Figs. 23 and 24. Two images of paintings of famous persons, "Mona Lisa" [9] and "Napoleon" [10],

[8] Composition with "Brandy rose", copyright photo courtesy of Toni Lankerd, 18347 Woodland Ridge Dr. Apt #7, Spring Lake, MI 49456, U.S.A.

[9] "Mona Lisa" or "La Gioconde", painted by Leonardo da Vinci.

[10] "Napoleon" is a cropped version of the "Master of Europe", painted by Appiani the Elder.

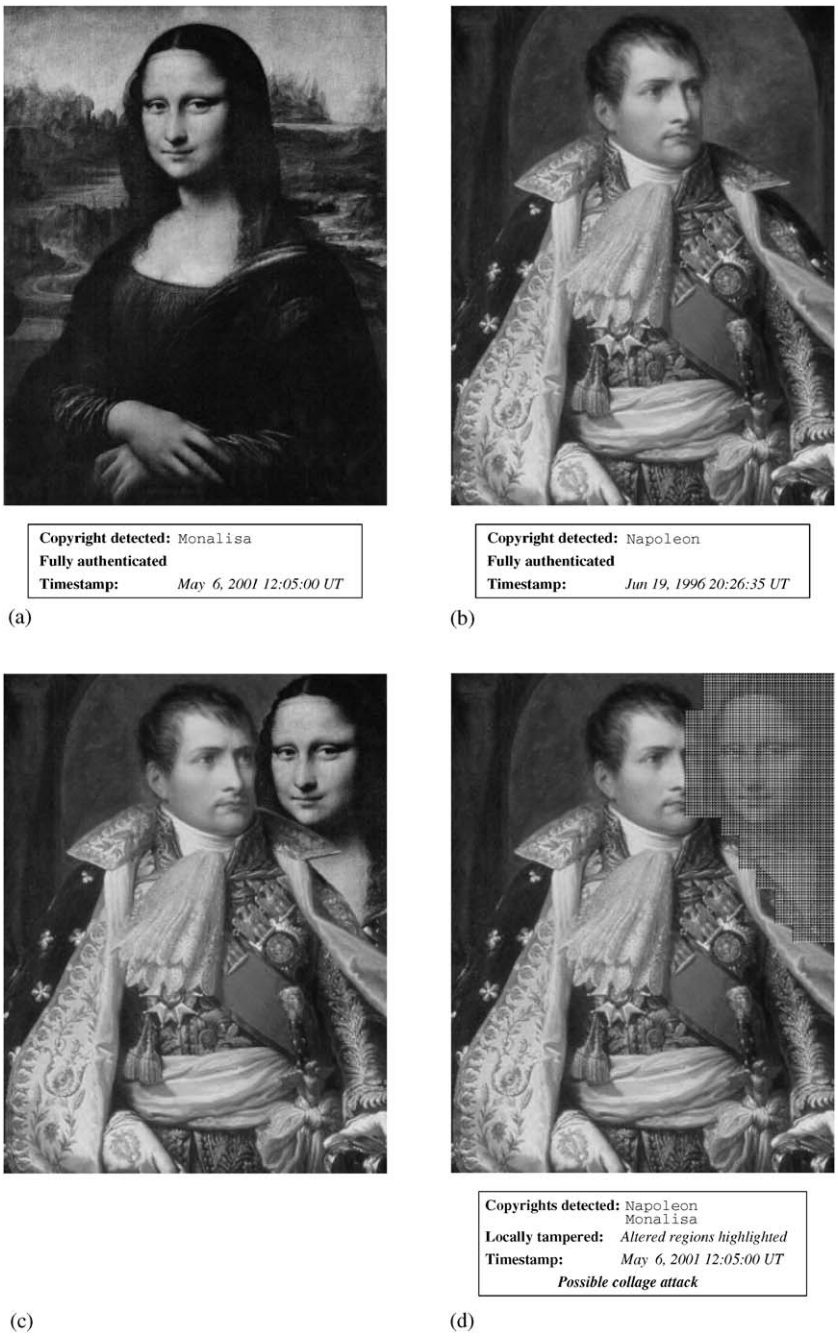

Fig. 23. First collage attack experiment. Both images: (a) "Mona Lisa"; (b) "Napoleon" are marked using the *same* key; (c) Mona Lisa's face is then copied and pasted into "Napoleon" at an arbitrary position; (d) since the pasted area is not synchronized within the host image, it is detected as an invalid area.

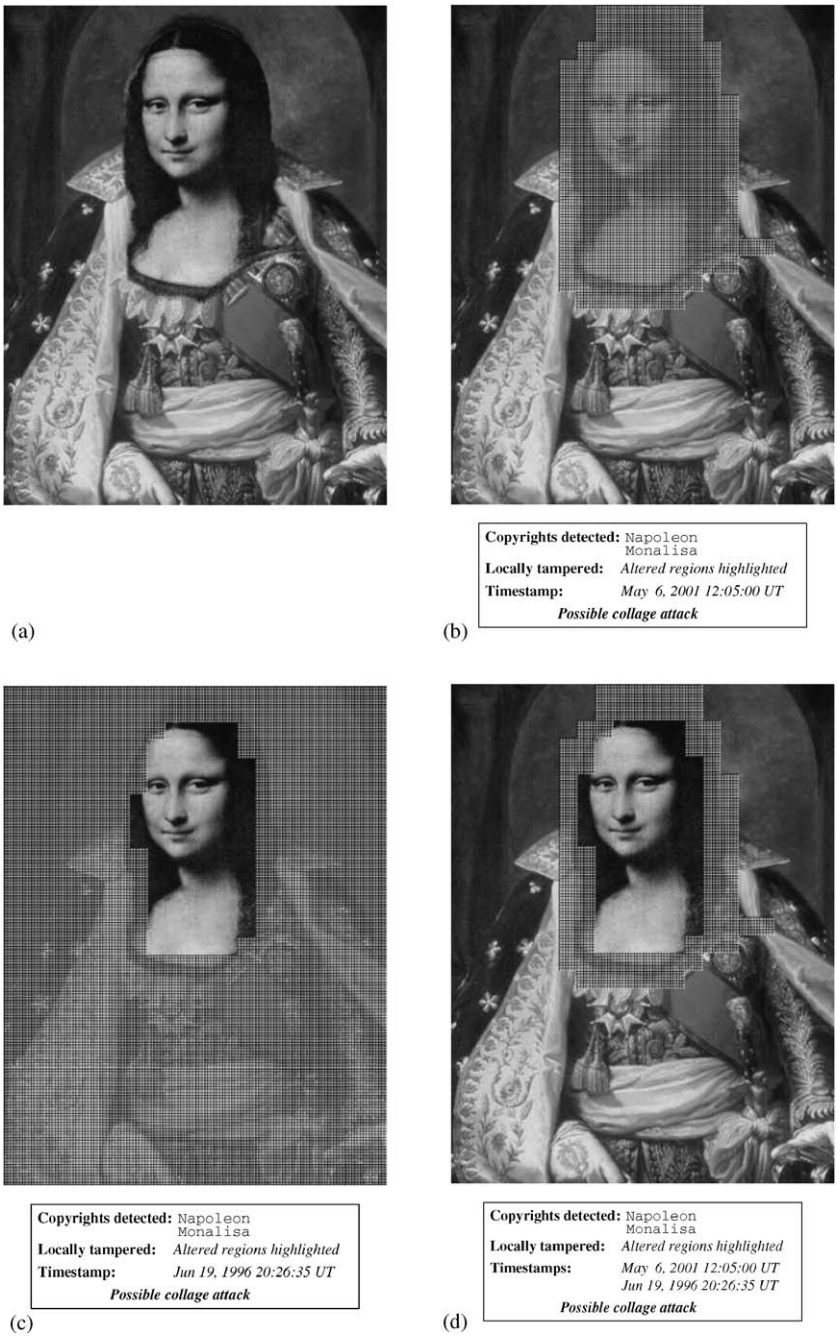

Fig. 24. Second collage attack experiment preserving the synchronization: (a) Mona Lisa's face is placed in "Napoleon" at the same offset as in "Mona Lisa"; (b) "Mona Lisa" part is rejected when a unique ID for "Napoleon" is used, (c) "Napoleon" part is rejected when "Mona Lisa" ID is entered; (d) when no ID used, only the boundaries between the two zones are invalidated, but the different time-stamps and copyrights separate these areas, and a composition is suspected.

both 512 × 669 pixels sized, were watermarked with the same key but with different robust messages. These images have been used as the source of our collage attack. Collages and compositions could be performed by malicious persons to build faked pictures, in this example art or historical paintings which have never been produced.

A first collage experiment consisted of copying a part from one image and to insert it into the other one, without care concerning the synchronization of the watermarks, and is shown in Fig. 23. Of course both images were separately authenticated (a,b). We copied Mona Lisa's face from "Mona Lisa" and pasted it at an arbitrary location into "Napoleon" (c). Afterwards we tried the watermark extraction from the resulting "Napoleon" (d). The robust part extracted the two copyright messages (coming from the two images), but the tamper proofing part detected the pasted area as an invalid region.

Secondly, we performed again this attack, but this time keeping the original synchronization of the pasted area in the target image relatively to the upper-left image corner, thus keeping all corresponding blocks from both images at the same positions (Fig. 24a). In this case two situations occurred: if one unique image ID was included in the hash inputs (that could be the original images sizes—in the case where their are of different sizes), then one area was rejected. At the verification stage, when the ID used for signing "Napoleon" was entered, Mona Lisa's face was rejected (b), and when the ID associated to "Mona Lisa" was used, everything but Mona Lisa's face was rejected (c). In the case where no additional external information was hashed the zones from both source images were authenticated, except the boundary between them which was detected as tampered (d). However, in this latter case the decoding of 2 robust messages, as well as the extraction of 2 distinct time-stamps, are the indication of a possible collage attack.

7. Conclusions

In this paper, we presented a hybrid robust watermarking scheme for visual data, which combines copyright protection, authentication, and detection of tampering. For this purpose we jointly used the highly robust watermarking scheme we previously developed, and a fragile watermark based on local signatures. Note that little work has been done today on such hybrid robust and fragile/semi-fragile watermarking.

The robust part exhibits high robustness to signal processing attacks, geometrical transforms as shown by the Stirmark results, as well as robustness to printing and rescanning. The algorithm is resistant against random local geometrical distortions too as well as to projective and non-linear transforms, and can also defeat collage attack by extracting and decoding the copyright information locally. The fragile part does not decrease the robustness of the robust part, due to its nearly orthogonal embedding with respect to the robust information. Extended security analysis of the scheme has been performed, especially from the cryptographic point of view. Exploiting the diagnostics from both the robust and the fragile parts, the algorithm is resistant against different kinds of attacks, including the copy attack and the collage attack.

A semi-fragile extension has further been proposed for applications which require limited distortions to be tolerated, such as image enhancement or lossy compression with acceptable quality, as well as for the protection of physical media or value papers. Both hybrid fragile and semi-fragile based algorithms are suitable for joint copyright protection or tracking, as well as tamper proofing/authentication purpose. Any visual media could benefits from this approach, especially video, where a semi-fragile version could be used to authenticate a movie under lossy compression such as the widely used MPEG2 format.

Acknowledgements

This work has been partially supported by SNF project Number 2100-064837.01 and SNCCR project IM2. We appreciate many fruitful discussions with S. Pereira (former University of Geneva), A. Herrigel and Y. Rytsar (DCT), M. Kutter and F. Jordan (AlpVision), F. Perez-Gonzalez and F. Balado (University of Vigo) and Igor Kozintsev (Microprocessor Research Labs., Intel) that lead to the appearance of BERKUT algorithm. We are also thankful to J. Eggers (Siemens

former University of Erlangen-Nürnberg) for interesting discussions about the SCS and corresponding results of its performance.

References

[1] A. Alattar, Smart images using Digimarc's watermarking technology, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Contents II. Vol. 3971, San Jose, CA, USA, January 23–28, 2000.

[2] P.S.L.M. Barreto, H.Y. Kim, V. Rijmen, Toward a secure public-key blockwise fragile authentication watermarking, in: IEEE Proceedings of the International Conference on Image Processing ICIP 2001, Thessaloniki, Greece, October 2001, pp. 494–497.

[3] C. Berrou, A. Glavieux, Near optimum error correcting coding and decoding: Turbo-codes. IEEE Trans. Comm. (October 1996) 1261–1271.

[4] S. Bhattacharjee, M. Kutter, Compression tolerant image authentication, in: IEEE International Conference on Image Processing '98 Proceedings, Chicago, IL, USA, October 1998, Focus Interactive Technology Inc.

[5] R.D. Brandt, F. Lin, Representations that uniquely characterize images modulo translation, rotation, and scaling, Pattern Recognition Lett. 17 (1996) 1001–1015.

[6] M.U. Celik, G. Sharma, E. Saber, A.M. Tekalp, A hierarchical image authentication watermark with improved localization and security, in: IEEE Proceedings of the International Conference on Image Processing ICIP 2001, Thessaloniki, Greece, October 2001, pp. 502–505.

[7] B. Chen, G.W. Wornell, Digital watermarking and information embedding using dither modulation, in: IEEE Second Workshop on Multimedia Signal Processing (MMSP-98), Redondo Beach, USA, December 1998, pp. 273–278.

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

[9] D. Coppersmith, F.C. Mintzer, C.P. Tresser, C.W. Wu, M.M. Yeung, Fragile imperceptible digital watermark with privacy control, in: IS& T/SPIE's 11th Annual Symposium, Electronic Imaging '99: Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, CA, USA, January 1999, pp. 79–84.

[10] F. Deguillaume, S. Voloshynovskiy, T. Pun, Method for the estimation and recovering of general affine transforms in digital watermarking applications, in: IS& T/SPIE's 14th Annual Symposium, Electronic Imaging 2002: Security and Watermarking of Multimedia Contents IV, Vol. 4675, San Jose, CA, USA, January 20–25 2002, pp. 313–322.

[11] J. Eggers, J. Su, B. Girod, A blind watermarking scheme based on structured codebooks, in: Secure images and image authentication, IEE Colloquium, London, UK, April 2000, pp. 4/1–4/6.

[12] J. Fridrich, A hybrid watermark for tamper detection in digital images, in: ISSPA'99 Conference, Brisbane, Australia, August 22–25 1999, pp. 301–304.

[13] J. Fridrich, Visual hash for oblivious watermarking, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Contents II, Vol. 3971, San Jose, CA, USA, January 24–26, 2000, pp. 286–294.

[14] J. Fridrich, Security of fragile authentication watermarks with localization, in: IS& T/SPIE's 14th Annual Symposium, Electronic Imaging 2002: Security and Watermarking of Multimedia Contents IV, Vol. 4675, San Jose, CA, USA, January 2002, pp. 691–700.

[15] J. Fridrich, M. Goljan, Protection of digital images using self embedding, in: Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, USA, May 1999.

[16] J. Fridrich, M. Goljan, A.C. Baldoza, New fragile authentication watermark for images, in: IEEE International Conference on Image Processing ICIP2000, Vancouver, Canada, September 2000.

[17] J. Fridrich, M. Goljan, N. Memon, Further attacks on Yeung–Mintzer fragile watermarking scheme, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Contents II, San Jose, CA, USA, January 2000, pp. 428–437.

[18] J. Fridrich, M. Goljan, N. Memon, Cryptanalysis of the Yeung–Mintzer fragile watermarking technique, J. Electron. Imaging 11 (2) (April 2002) 262–274.

[19] G.L. Friedman, The trustworthy digital camera: restoring credibility to the photographic image, IEEE Trans. Consumer Electr. 39 (November 1993) 905–910.

[20] R.G. Gallager, Low-Density Parity-Check Codes, MIT Press, Cambridge, 1963.

[21] H. Harashima, H. Miyakawa, Matched-transmission technique for channels with intersymbol interference, IEEE Trans. Comm. 20 (4) (August 1972) 774–780.

[22] H. Hel-Or, Y. Yitzhaki, Y. Hel-Or, Geometric hashing techniques for watermarking, in: ICIP 2001, Thessaloniki, Greece, January 2001.

[23] J.R. Hernández, F. Pérez-González, J.M. Rodríguez, G. Nieto, The impact of channel coding on the performance of spatial watermarking for copyright protection, in: Proceedings of the ICASSP'98, Vol. 5, May 1998, pp. 2973–2976.

[24] M. Holliman, N. Memon, Couterfeting attacks on oblivious block-wise independant invisible watermarking schemes, IEEE Trans. Image Process. 9 (March 2000) 432–441.

[25] P. Hough, Method and means for recognizing complex patterns, 1962, U.S. Patent 3069654.

[26] D. Johnson, A. Menezes, The Elliptic Curve Digital Signature Algorithm (ECDSA), Corr 99-34, Department of C&O, University of Waterloo, August 1999.

[27] A.C.C. Kalker, G. Depovere, J. Haitsma, M.J. Maes, A video watermarking system for broadcast monitoring, in:

IS&T/SPIE's 11th Annual Symposium, Electronic Imaging'99: Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, CA, USA, January 1999, pp. 103–112.
[28] N. Koblitz, Elliptic curve cryptosystems, Math. Comput. 48 (1987) 203–209.
[29] D. Kundur, D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, Proc. IEEE 87 (7) (July 1999) 1167–1180.
[30] M. Kutter, Digital image watermarking: hiding information in images, Ph.D. Thesis, EPFL, Lausanne, Switzerland, August 1999.
[31] M. Kutter, S. Voloshynovskiy, A. Herrigel, Watermark copy attack, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Contents II, Vol. 3971, San Jose, CA, USA, January 23–28 2000.
[32] X. Lai, J.L. Massey, A proposal for a new block encryption standard, in: I.B. Damgard (Ed.), Lecture Notes in Computer Science (EUROCRYPT'90), Advances in Cryptology, Springer, Vol. 473, 1991, pp. 389–404.
[33] C.-Y. Lin, S.-F. Chang, Semi-fragile watermarking for authenticating JPEG visual content, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, Vol. 3971, San Jose, CA, USA, January 23–28 2000, pp. 140–151.
[34] E.T. Lin, C.I. Podilchuk, E.J. Delp, Detection of image alterations using semi-fragile watermarks, in: IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, Vol. 3971, San Jose, CA, USA, January 23–28 2000.
[35] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, October 1996.
[36] V.S. Miller, Uses of elliptic curves in cryptography, in: Advances in Cryptology CRYPTO'85, Vol. 218, 1985, pp. 417–426.
[37] M.L. Miller, I.J. Cox, J.A. Bloom, Watermarking in the real world: an application to DVD, in: P. Horster, J. Dittmann, P. Wohlmacher, R. Steinmetz (Eds.), Proceedings of Multimedia and Security Workshop at ACM Multimedia 98 (GMD Report), Vol. 41, Bristol, UK, September 1998, pp. 71–76.
[38] University of Geneva Watermarking Group, Watermarking Group Technology, 2002, http://watermark.unige.ch/wmg_technology.html.
[39] National Institute of Standards and Technology (NIST), Secure Hash Standard, May 1993, Federal Information Processing Standards Publications (FIPS) PUB 180-1.
[40] National Institute of Standards and Technology (NIST), Data Encryption Standard (DES), October 1999, Federal Information Processing Standards Publications (FIPS) PUB 46-3.
[41] National Institute of Standards and Technology (NIST), Digital Signature Standard (DSS), February 2000, Federal Information Processing Standards Publications (FIPS) PUB 186-2.
[42] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), November 2001, Federal Information Processing Standards Publications (FIPS) PUB 197.
[43] J.J.K. Óruanaidh, G. Csurka, A Bayesian approach to spread spectrum watermark detection and secure copyright protection for digital image libraries, in: IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA, June 1999.
[44] J.J.K. Óruanaidh, T. Pun, Rotation, scale and translation invariant digital image watermarking, in: IEEE International Conference on Image Processing ICIP1997, Santa Barbara, CA, USA, October 1997, pp. 536–539.
[45] J.J.K. Óruanaidh, T. Pun, Rotation, scale and translation invariant spread spectrum digital image watermarking, Signal Process. 66 (3) (1998) 303–317.
[46] S. Pereira, J.J.K. Óruanaidh, F. Deguillaume, G. Csurka, T. Pun, Template based recovery of Fourier-based watermarks using Log-polar and Log-log maps, in: International Conference on Multimedia Computing and Systems, Special Session on Multimedia Data Security and Watermarking, Vol. 1, Florence, Italy, June 1999, pp. 870–874.
[47] F.A.P. Petitcolas, Stirmark benchmark 4.0. 2002, http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/.
[48] R. Rivest, The MD5 message-digest algorithm, April 1992, Request for Comments (RFC) 1321, MIT LCS and RSA Data Security, Inc.
[49] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Comm. ACM 21 (February 1978) 120–126.
[50] M. Schneider, S. Chang, A robust content based digital signature for image authentication, in: Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland, September 1996, pp. 227–230.
[51] T. Tomlinson, New automatic equalizer employing modulo arithmetic, Electron. Lett. 7 (March 1971) 138–139.
[52] M. Tüchler, R. Kötter, A. Singer, Turbo equalization: principles and new results, IEEE Trans. Commun. 50 (5) (May 2002) 754–767.
[53] R. Venkatesanan, S. Koon, M. Jacubowski, P. Moulin, Robust image hashing, in: ICIP 2000, Vancouver, BC, Canada, September 2000.
[54] S. Voloshynovskiy, F. Deguillaume, O. Koval, T. Pun, Robust watermarking with channel state estimation, Part I: theoretical analysis, Signal Processing: Security of Data Hiding Technologies, (Special Issue) 2003–2004, to appear.
[55] S. Voloshynovskiy, F. Deguillaume, O. Koval, T. Pun, Robust watermarking with channel state estimation, Part II: applied robust watermarking, Signal Processing: Security of Data Hiding Technologies, (Special Issue) 2003–2004, to appear.
[56] S. Voloshynovskiy, F. Deguillaume, S. Pereira, A. Herrigel, T. Pun, Method for adaptive digital watermarking robust against geometric transforms, European Patent Application PCT/IB00/01089, August 3 2000.
[57] S. Voloshynovskiy, F. Deguillaume, S. Pereira, T. Pun, Optimal diversity watermarking with channel state estimation, in: IS& T/SPIE's 13th Annual Symposium, Electronic Imaging 2001: Security and Watermarking of Multimedia

Contents III, Vol. 4134, San Jose, CA, USA, January 21–26 2001, pp. 23–27.
[58] S. Voloshynovskiy, F. Deguillaume, T. Pun, Content adaptive watermarking based on a stochastic multiresolution image modeling, in: Tenth European Signal Processing Conference EUSIPCO2000, Tampere, Finland, September 2000.
[59] S. Voloshynovskiy, F. Deguillaume, T. Pun, Multibit digital watermarking robust against local nonlinear geometrical distortions, in: IEEE International Conference on Image Processing ICIP2001, Thessaloniki, Greece, October 2001, pp. 999–1002.
[60] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, T. Pun, A stochastic approach to content adaptive digital image watermarking, in: Third International Workshop on Information Hiding, Vol. 1768, Dresden, Germany, September 29–October 1st 1999, pp. 212–236.
[61] S. Voloshynovskiy, T. Pun, Capacity-security analysis of data hiding technologies, in: IEEE International Conference on Multimedia and Expo ICME2002, Lausanne, Switzerland, August 26–29, 2002.
[62] S. Walton, Information authentication for a slippery new age, Dr. Dobbs J. 20 (4) (April 1995) 18–26.
[63] R.B. Wolfgang, E.J. Delp, Fragile watermarking using the VW2D watermark, in: IS& T/SPIE's 11th Annual Symposium, Electronic Imaging '99: Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, CA, USA, January 25–27 1999, pp. 204–213.
[64] P.W. Wong, A public key watermark for image verification and authentication, in: IEEE International Conference on Image Processing '98 (ICIP'98) Proceedings, Vol. 1, 1998, MA11.07.
[65] P.W. Wong, N. Memon, Secret and public key authentication watermarking schemes that resist vector quantization attack, in: IS& T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Contents II, Vol. 3971, San Jose, CA, USA, January 2002, pp. 417–427.
[66] M. Wu, B. Liu, Watermarking for image authentication, in: IEEE International Conference on Image Processing '98 (ICIP'98) Proceedings, Chicago, IL, USA, October 1998, Focus Interactive Technology Inc., TA10.11.
[67] L. Xie, G.R. Arce, Joint wavelet compression and authentication watermarking, in: IEEE International Conference on Image Processing '98 Proceedings, Chicago, IL, USA, October 1998, Focus Interactive Technology Inc.
[68] M.M. Yeung, F.C. Mintzer, An invisible watermarking technique for image verification, in: 1997 International Conference on Image Processing (ICIP '97), Vol. 2, Washington, DC, USA, October 26–29 1997, pp. 680–683.