Simple Collusion-Secure Fingerprinting Schemes for Images

Josep Domingo-Ferrer and Jordi Herrera-Joancomartí
Dept. of Computer Engineering and Mathematics
Universitat Rovira i Virgili
ETSE-Autovia de Salou, s/n
E-43006 Tarragona, Catalonia, Spain
e-mail {jdomingo,jherrera}@etse.urv.es

Abstract

This paper describes a robust watermarking algorithm and a collusion-secure fingerprinting scheme based on it. Watermarking robustness is obtained by using the JPEG algorithm to decide mark location and magnitude; the proposed algorithm supports multiple marking. The properties of dual binary Hamming codes are exploited to obtain a fingerprinting scheme secure against collusion of two buyers. The fingerprinting construction proposed here is simpler than previous collusion-secure proposals.

1 Introduction

Electronic copyright protection schemes based on the principle of copy prevention have proven ineffective in the last years (see [7],[8]). The trend for electronic copyright protection is to rely on copy detection. The merchant M selling the piece of information (*e.g.* image) embeds a *mark* in the copy sold. There are two basic kinds of mark: fingerprints and watermarks. One may think of a fingerprint as a serial number (*i.e.* something identifying the buyer of the copy) while a watermark is an embedded copyright message.

There are two important differences between watermarks and fingerprints. First, while in watermarking the hidden message (mark) is the same for all buyers, in fingerprinting the mark depends on the buyer's identity. Second, buyer collusion is not an issue in watermarking (the marked copies being the same for all buyers); however, in fingerprinting the mark is different for every buyer, and it makes sense for a collusion of buyers to collude by comparing their copies and try to locate and delete some mark bits.

In Section 2 a robust watermarking scheme is described and its main properties are given. Section 3 presents a collusion-secure fingerprinting scheme which is simpler than those proposed in [2]. Section 4 is a conclusion also listing topics for future research.

2 A robust watermarking scheme

We present in this Section a watermarking scheme which is a variant eliminating much of the complexity of the proposal [6] but still preserving the robustness properties of the latter scheme (see Subsection 2.2 below).

2.1 General description

The scheme can be described in two stages: mark embedding and mark reconstruction. Like all practical schemes known to date, the following one is symmetric in the sense of [9]: mark embedding is entirely performed by the merchant M, who cannot prove in court that he recovered a mark from a redistributed copy rather than from an unredistributed one (M knows all marked copies he sells from embedding time and could simulate a redistribution).

For mark embedding, we assume that the image allows sub-perceptual perturbation. Assume that q is a JPEG quality level chosen in advance by the merchant M; q will be used as a security parameter. Let $\{s_i\}_{i\geq 1}$ be a random bit sequence generated by a sound stream cipher with secret key k only known to M. Let w and h be the width and the height of the image X to be watermarked (in pixels), *i.e.* $X = \{x_i : 1 \leq i \leq w \times h\}$ where x_i is the gray-scale level of the i-th pixel (without loss of generality, we can assume X to be monochrome to simplify the subsequent discussion).

Algorithm 1 (Mark embedding)

1. *Compress X using the JPEG algorithm with quality q as input parameter. Call the bitmap of the resulting*

compressed image X'. Let $\delta_i := x_i - x'_i$ be the difference between corresponding pixels in X and X'. Only positions i for which $\delta_i \neq 0$ will be usable to embed bits of the mark.

2. *Call ε the mark to be embedded. Encode ε using an error-correcting code (ECC) to obtain the encoded mark E; call $|E|$ the bit-length of E. Replicate the mark E to obtain a sequence E' with as many bits as pixels in X with $\delta_i \neq 0$.*

3. *Let $j := 0$. For $i = 1$ to $w \times h$ do:*

   (a) *If $\delta_i = 0$ then $x''_i := x_i$.*

   (b) *If $\delta_i \neq 0$ then*

      i. *Let $j := j + 1$. Compute $s'_j := e'_j \oplus s_j$, where e'_j is the j-th bit of E'. The actual bit that will be embedded is s'_j.*

      ii. *If $s'_j = 0$ then compute $x''_i := x_i - \delta_i$.*

      iii. *If $s'_j = 1$ then compute $x''_i := x_i + \delta_i$.*

$X'' = \{x''_i : 1 \leq i \leq w \times h\}$ is the marked image. In particular, the embedded marks will survive JPEG compression of X'' down to quality level q (see Subsection 2.2 for a more comprehensive robustness assessment). Quality metrics such as the ones in [4, 5] can be used to measure imperceptibility of the mark. If imperceptibility is not satisfactory, then re-run the mark embedding algorithm with a higher quality level q.

For mark reconstruction, knowledge of the original image and the secret key k is assumed (k is used to regenerate the random sequence $\{s_i\}_{i \geq 1}$). Note that assuming knowledge of the original image in mark reconstruction is quite realistic in the case of symmetric marking algorithms, where mark reconstruction is performed by and for the merchant M.

Algorithm 2 (Mark reconstruction)

1. *Upon detecting a redistributed item $\hat{X}$, M restores it to the bitmap format.*

2. *Compress the corresponding original image X with quality q to obtain X'.*

3. *Let $j := 0$. For $i = 1$ to $w \times h$ do:*

   (a) *Compute $\delta_i := x_i - x'_i$.*

   (b) *If $\delta_i \neq 0$ then*

      i. *Let $j := j + 1$. If $j > |E|$ then $j := 1$.*

      ii. *Compute $\hat{\delta}_i := \hat{x}_i - x_i$.*

      iii. *If $\delta_i \times \hat{\delta}_i > 0$ then $\hat{s}_j := 1$; otherwise $\hat{s}_j := 0$.*

      iv. *Compute $\hat{e'}_j := \hat{s}_j \oplus s_j$.*

      v. *If $\hat{e'}_j = 1$ then $ones_j := ones_j + 1$; otherwise $zeroes_j := zeroes_j + 1$.*

4. *For $j = 1$ to $|E|$ do: if $ones_j > zeroes_j$ then $\hat{e}_j := 1$ otherwise $\hat{e}_j := 0$ ($\hat{e}_j$ is the j-th bit of the recovered mark $\hat{E}$).*

5. *Decode $\hat{E}$ with the same ECC used for embedding. If the decoded $\hat{\varepsilon}$ matches the original ε, then verification succeeds.*

Note that the redistributed $\hat{X}$ may have width $\hat{w}$ and height $\hat{h}$ which differ from w and h due to manipulation by the re-distributor. In the next Subsection, we discuss ways to restore the original width and height.

2.2 Robustness assessment

The scheme described in Subsection 2.1 was implemented using a dual binary Hamming code as ECC. The well-known image Lena was thereafter marked using $q = 60\%$, which kept marks still imperceptible. Then the marked image was manipulated using the base test of the StirMark 3.1 benchmark [7].

The embedded marks survived the following StirMark manipulations:

1. Colour quantisation.

2. All low pass filtering manipulations. More specifically:

   (a) Gaussian filter (blur).

   (b) Median filter (2×2, 3×3 and 4×4).

   (c) Frequency mode Laplacian removal [1].

   (d) Simple sharpening.

3. All compression manipulations. More specifically, JPEG compression for qualities 90% down to 10%.

4. Rotations with scaling of -1.00 up to 0.75 degrees.

5. Shearing up to 5% in the Y direction.

There are additional StirMark attacks which can be easily survived if knowledge of the original image X is used to do some pre-processing on the redistributed $\hat{X}$:

- Removal of rows and columns from the marked image X'' can be automatically detected and approximately undone by M by replacing the removed rows and columns with the corresponding ones of X, which restores the original image size (*i.e.* makes $\hat{w} = w$ and $\hat{h} = h$) and allows correct mark recovery (the probability of correct recovery decreases as the number of removed rows and columns increases).

- Cropping attacks can be dealt with in a similar way. An additional countermeasure against cropping is to use region-based watermarking (*e.g.* as in [3]), which attempts to counter loss of synchronization during mark reconstruction by embedding copies of the mark in several *invariant domains* or distinct meaningful regions of the image (for example, a face, a fruit, etc.).

- Rotation, scaling and shearing attacks can also be detected and undone by M by comparing with the original image.

The really dangerous attacks for the scheme presented here are random geometric distortions and attacks combining several of the aforementioned elementary manipulations.

An additional interesting feature of the presented algorithm is that multiple marking is supported. For example, merchant M_1 can mark an image and sell the marked image to merchant M_2, who re-marks the image with its own mark, and so on. We have been able to embed up to five successive markings without perceptible quality degradation and without loss of robustness.

3 Collusion-secure fingerprinting scheme

In addition to manipulations by a single buyer (like those described in Subsection 2.2), collusion attacks are possible when the above scheme is used for fingerprinting (*marking assumption*, [2]).

Attack 1 *Two or more different buyers can detect the mark bits by comparison of their copies of the same image, because the key k that generates the sequence s_i for a given image is the same for all buyers. After detection, there are two attacking strategies:*

Mark bit deletion *The mark bits that differ between buyers can be deleted by simply adding the pixels that embed them. Clearly, if two mark bits differ, then the corresponding pixels in the marked images are $x_i + \delta_i$ and $x_i - \delta_i$. Since M knows that there must be an embedded bit in that position, the probability of correctly restoring a deleted mark bit is 1/2.*

Mark bit tweaking *The buyer having pixel $x_i+\delta_i$ replaces it with $x_i - \delta_i$ (or viceversa), which tweaks the embedded bit. This attack is worse than deletion, as M has no way to detect that the bit was tweaked.*

If the number of bits tweaked (or incorrectly restored) is greater than the maximum number of errors that the ECC can correct, then mark reconstruction will fail.

In what follows, we restrict ourselves to collusions of two buyers; given the danger inherent to piracy, the size of collusions tends to be small, so concentrating on collusions of size two is not unrealistic and can be done efficiently as shown below.

The effectiveness of Attack 1 depends on the distance between codewords of the ECC used in the mark embedding algorithm. The larger the distance, the more bits will differ between the codewords of colluders, *i.e.* the easier to tweak a codeword bit. On the other hand, the smaller the distance, the less error-correcting capacity will be obtained from the ECC. Thus, it is interesting to use an ECC whose distance between codewords is both lower and upper-bounded. Dual binary Hamming codes are a natural choice, because the distance between any two codewords is fixed: the distance $d(x, y)$ between any two codewords x, y of a dual binary Hamming code of length $N = 2^n - 1$ is 2^{n-1}.

For simplicity, in what follows we will focus on the codeword E that identifies a specific buyer once the ECC has been applied. The subsequent key-dependent transformation of E to obtain the marked image is not relevant for our discussion because the key k is the same for all buyers of the image. The ECC used will be a dual binary Hamming code H with length $2^n - 1$. The distance between any two codewords in H is then 2^{n-1}, so that up to $2^{n-2} - 1$ errors can be corrected. The following definition introduces some useful notation.

Definition 1 *Let $a^1, \cdots, a^l$ be l codewords of length N (*i.e. *$a^j = a_1^j a_2^j \cdots a_N^j$). The i-th position of the set of codewords $a^1, \cdots, a^l$ is called* invariant position *if*

$$a_i^j = a_i^k \quad \forall j, k = 1, \cdots, l$$

We denote by $ivt(a^1, \cdots, a^l)$ the set of invariant positions of $a^1, \cdots, a^l$. Also, we denote by $< a^1, \cdots, a^l >$ the set of words that can be generated by taking as first bit one of $a_1^1, \cdots, a_1^l$, as second bit one of $a_2^1, \cdots, a_2^l$, and so on.

Regarding Attack 1, the following lemma holds.

Lemma 1 *The colluders using Attack 1 cannot tweak the bits at invariant positions.*

Proof: Attack 1 assumes that colluders can tweak a bit only if such a bit differs in their copies. Since bits at invariant positions have all the same value, they cannot be changed. $\diamond$

The following properties of dual binary Hamming codes are stated for later use.

Proposition 1 *Let H be a dual binary Hamming code with length $2^n - 1$. Then any three codewords of H have exactly $2^{n-2} - 1$ invariant positions.*

Proof: Let $x, y \in H$ be any pair of codewords. Define $I = ivt(x, y)$ and $\overline{I}$ the positions not in I. We denote by $x_{|I}$

the bits of the word x in the positions in I. Since $d(x,y) = 2^{n-1}$ it follows that $|ivt(x,y)| = 2^{n-1} - 1$.

By construction, we will prove that given any codeword $z \in H$ and a value $k \leq 2^{n-1} - 1$, if $|ivt(x,y,z)| = k$ then $k = 2^{n-2} - 1$. We have

$$d(x_{|I}, z_{|I}) = d(y_{|I}, z_{|I}) = 2^{n-1} - 1 - k$$

since $x_{|I} = y_{|I}$. Now, x, y and z belong to a dual binary Hamming code with length $2^n - 1$, so $d(x,z) = d(y,z) = 2^{n-1}$; therefore,

$$d(x_{|\overline{I}}, z_{|\overline{I}}) = d(y_{|\overline{I}}, z_{|\overline{I}}) = 2^{n-1} - (2^{n-1} - 1 - k) = k + 1$$

But $d(x_{|\overline{I}}, y_{|\overline{I}}) = 2^{n-1}$, so that the only possibility is

$$k + 1 = \frac{2^{n-1}}{2}$$

Otherwise, we would have

$$d(x_{|\overline{I}}, z_{|\overline{I}}) \neq d(y_{|\overline{I}}, z_{|\overline{I}})$$

since if $x_i \neq y_i$ then $z_i = y_i \Rightarrow z_i \neq x_i$ as we work in a binary domain (x_i denotes the i-th bit of x). Then

$$k = 2^{n-2} - 1$$

$\diamond$

Proposition 2 *Let $x, y \in H$. If $z \in < x, y > \setminus \{x, y\}$, then the correction of z using H cannot yield a codeword different from x and y.*

Proof: Assume that $z \in < x, y >$ exists such that it decodes into a $z' \in H \setminus \{x, y\}$. Since the code H corrects up to $2^{n-2} - 1$ errors, candidates to be z are words with $\leq 2^{n-2} - 1$ errors. But $|ivt(x,y)| = 2^{n-1} - 1$ and $|ivt(x,y,z)| = 2^{n-2} - 1$, so z has at least $(2^{n-1} - 1) - (2^{n-2} - 1) = 2^{n-2}$ errors, which is more than $2^{n-2} - 1$ (the error-correcting capacity of H). $\diamond$

Proposition 3 *Let $x, y \in H$. The probability of obtaining a word $z \in < x, y > \setminus H$ such that z does not uniquely decode into a codeword of H is*

$$p \leq \left(\frac{1}{2}\right)^{2^{n-1}} \cdot 2^n \qquad (1)$$

Proof: The only way that z does not uniquely decode is that it contains exactly 2^{n-2} errors.

By Proposition 1, $|ivt(x,y,z)| = 2^{n-2} - 1$. So, from the $2^{n-1} - 1$ invariant positions of x and y, only $2^{n-2} - 1$ correct values will remain in the new word z and thus $(2^{n-1} - 1) - (2^{n-2} - 1) = 2^{n-2}$ will be errors.

Thus, the total amount of errors in z must be in $I = ivt(x,y)$, so $z_{|\overline{I}}$ has to be exactly equal to the corresponding part of a correct codeword in $< x_{|\overline{I}}, y_{|\overline{I}} >$. The probability of this event for a particular codeword is

$$\left(\frac{1}{2}\right)^{2^{n-1}}$$

because $d(x_{|\overline{I}}, y_{|\overline{I}}) = |\overline{I}| = 2^{n-1}$ so that in every position there is exactly one 1 and one 0. Since the total number of codewords in a dual binary Hamming code is 2^n the probability of non-unique decoding is

$$p \leq \left(\frac{1}{2}\right)^{2^{n-1}} \cdot 2^n$$

$\diamond$

We are now in a position to state the main theorem of this section:

Theorem 1 *The watermarking scheme described in Section 2 can be transformed into a fingerprinting scheme secure against collusions of two buyers by taking as ECC in the mark generation algorithm a dual binary Hamming code H of length $N = 2^n - 1$. An innocent buyer will never be declared guilty and the probability that a participant in a two-buyer collusion can be identified can be made arbitrarily close to 1.*

Proof: It follows from Propositions 2 and 3. Proposition 2 guarantees that an innocent buyer will never be declared guilty. The probability of identifying one of two colluders is $1 - p$, where p is defined in Equation (1); as n increases, $1 - p$ tends to 1. $\diamond$

4 Conclusion and future research

We have presented a new collusion-secure fingerprinting construction based on a robust watermarking algorithm, which allows multiple marking. In the fingerprinting scheme, the problem of two-buyer collusion is solved using dual binary Hamming codes.

The constructions for collusion-secure fingerprinting given here are simpler than those in [2] and are attractive because they are built on top of a robust watermarking algorithm.

Future research will be directed to increasing:

- The range of attacks that can be survived by the watermarking algorithm.
- The size of collusions that can be successfully resisted by the fingerprinting algorithm.

Acknowledgments

Thanks go to Josep Rifà for useful suggestions and to Francesc Sebé for discussions on an earlier version of this paper. This work was partly supported by the Spanish CICYT under grant no. TEL98-0699-C02-02.

References

[1] R. Barnett and D. E. Pearson. Frequency mode L.R. attack operator for digitally watermarked images. *Electronics Letters*, 34(19): 1837-1839, September 1998.

[2] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology-CRYPTO'95*, LNCS 963. Springer-Verlag, Berlin, 1995, pp. 452-465. Journal version in *IEEE Transactions on Information Theory*, IT-44(5): 1897-1905, September 1998.

[3] G. Brisbane, R. Safavi-Naini and P. Ogunbona. Region-based watermarking for images. In *Information Security*, LNCS 1729. Springer-Verlag, Berlin, 1999, pp. 154-166.

[4] J. Fridrich and M. Goljan. Comparing robustness of watermarking techniques. In *Security and Watermarking of Multimedia Contents*, vol. 3567. The Society for Imaging Science and Technology and the International Society for Optical Engineering, San José CA, 1999, pp. 214-225.

[5] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Security and Watermarking of Multimedia Contents*, vol. 3657. The Society for Imaging Science and Technology and the International Society for Optical Engineering, San José CA, 1999, pp. 226-239.

[6] H. J. Lee, J. H. Park and Y. Zheng. Digital watermarking robust against JPEG compression. In *Information Security*, LNCS 1729. Springer-Verlag, Berlin, 1999, pp. 167-177.

[7] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. Attacks on copyright marking systems. In *2nd International Workshop on Information Hiding*, LNCS 1525. Springer-Verlag, Berlin, 1998, pp. 219-239.

[8] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. Information hiding–A survey. *Proceedings of the IEEE*, 87(7): 1062-1078, July 1999.

[9] B. Pfitzmann and M. Schunter. Asymmetric fingerprinting. In *Advances in Cryptology-EUROCRYPT'96*, LNCS 1070. Springer-Verlag, Berlin, 1996, pp. 84-95.