INTRUSIVE AND NON-INTRUSIVE WATERMARKING

Hari Krishna V.J, K.R Ramakrishnan

Dept. **of** Electrical Engineering, Indian Institute of Science, Bangalore-560012
email:{vj,krr}@ee.iisc.ernet.in

ABSTRACT

"Can we watermark without perturbing an image?"
We present the salient results of the investigation carried out to find an answer to this question.

1. INTRODUCTION

The digital revolution and the growing popularity of the internet has brought with it many problems. Ownership right protection of digital images is one of them. The principles of information hiding have been used in the form of watermarking to tackle this problem.

Image Watermarking involves inserting a watermark in the image to be protected. This is done by perturbing an image in either it's spatial domain or in **a** domain obtained by transforming the image. There are a number of techniques which watermark an image in this manner [3].

Is it necessary that we perturb an image to watermark it ? Suppose one claims that a watermarking scheme which embeds without perturbing an image exists, based on what can one prove or disprove such a claim ? This leads **us** to another fundamental question. How do we judge a given watermarking scheme ?

1.1. Governing factors

For an ***invisible*** watermarking scheme transparency is a governing factor. An invisible watermarking scheme can either be fragile or robust. Fragile watermarking is useful for image authentication. For ownership right protection which is what we are aiming to achieve, the watermarking scheme needs to be robust. *Non-invertibility* [2] works independent of robustness and hence it is a governing factor. Capacity is of fundamental importance since it governs the number of bits one can insert and detect with a low probability of error. Thus, we judge any candidate watermarking scheme based on ***transparency, robustness, capacity*** and ***non-invertibility*** together referred to as the governing factors.

Let us refer to schemes that embed by perturbing an image as ***intrusive*** schemes. Initially, we present a watermarking scheme which claims to embed without perturbing an image. We call this a ***non-intrusive*** scheme. We judge the scheme on the basis of the governing factors and find out, that, transparency-wise, robustness-wise and capacity-wise the scheme is promising but is easily invertible. We discover that, it is only to ensure non-invertibility that an image needs to be perturbed in order to be watermarked. We propose a ***watermarking framework*** designed such that, any watermarking scheme which embeds by adhering to the rules of the watermarking framework will carry with it the positive aspects of non-intrusive embedding and at the same time be non-invertible.

Non-invertibility and non-intrusive embedding seem to be at loggerheads with each other. The attacker seems to have more chances of inverting the scheme in our framework. We discuss ways of using the standard, key based one-way functions along with the" allowable" watermark strategy to nullify the advantage that an attacker seems to have. For a given one-way function we analyse the maximum amount of non-intrusive embedding that can be done without giving the attacker any more chances of inverting the scheme.

We also prove that, given a transparency constraint, for a class of watermarking schemes our framework is at least as robust as the scheme chosen for intrusive embedding. Infact, robustness increases with the amount of non-intrusive embedding. This ensures that, the flexibility in the choice of the scheme to embed, which our framework allows, is truly an advantage.

2. NON-INTRUSIVE EMBEDDING

Let **us** analyse a watermarking scheme(Fig 1) which claims to embed a watermark W with P components[1], in a given N **x** M image **I** without perturbing **I.**

$$\boldsymbol{I} = \{x_{mn}, m \in U_I, n \in V_I\}$$
$$W = \{s_n, n \in V_W\}$$
$$x_{mn}, s_n \in \{0, 1 \ldots L - 1\}$$

Where, $U_I = \{0, 1, \ldots, M-1\}, V_I = \{0, 1, \ldots, N-1\}$ and $V_W = \{0, 1, \ldots, \boldsymbol{P} - 1\}$, L is the number of pixel intencity levels.

[1] A component is a collection of bits

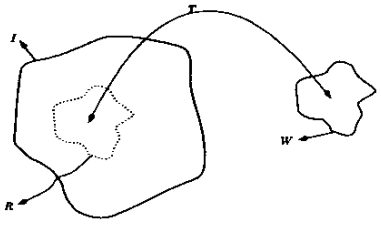

Fig. 1. Embedding without perturbing the image

Watermark insertion:

- Alice identifies a set R with P elements, $R = \{r_n, n \in V_W\}$. R can be a subset of I (spatial domain insertion) or a subset of a transform of I (transform domain insertion).
- Alice generates a *key* set $K = W - R = \{k_n = s_n - r_n, n \in V_W\}$.
- This completes watermark insertion. The set K is kept secret by Alice.

Watermark detection:

- To test for the presence of the watermark, Alice has to set up the set R from the given image and $K + R = W - R + R = W$ gives the watermark.

The key K links R and W. It is not necessaty that Alice obtain K as shown in the scheme discussed above. Alice can use any mapping $T : R \rightarrow W$ and keep a K associated with this mapping secret. Obviously, the secret K depends on W the watermark.

Transparency:

The image I to be watermarked is not perturbed in any way to insert a given watermark W. The scheme is transparent. In fact, transparency is guaranteed independent of the distance of view.

Robustness:

Robustness of the scheme depends on the choice of elements that constitute set R.

For example, one can choose perceptually relevant components of the image I and form set R. This will make the scheme robust to some of the commonly used compression attacks. The most natural equivalent of this scheme would be the *Cox et al* [4] kind of scheme , where embedding is done in the perceptually relevant components. There is one fundamental difference between these two seemingly similar approaches. The trade-off between the watermark signal amplitude and the transparency criterion which exists in *Cox et al* kind of schemes does not exist here since the elements of set R are not perturbed.

Capacity:

Capacity analysis with attack modelling is pretty complicated. To test if there is any serious shortcoming in the capacity of the scheme being discussed, we assume that the watermarked image is not subjected to any innocent or malicious processing. So, how many watermark bits can we insert and retrieve using this scheme? Given a set R with P components and a watermark of arbitrary length we can partition W into sets with P components each and associate with eachpartition a K as described in the scheme. To detect the watermark each P length watermark partition is extracted individually and assembled together to get the entire watermark. In this manner watermark of any arbitrary length can be inserted and detected. In fact, since we are not considering any kind of processing on the watermarked image it is not necessary that R contain only the " safe spots" in the image I. R can be the entire image I.

Even though the whole discussion above was with the "no attack" assumption, the fact that this watermarking scheme has infinite capacity is not trivial. Consider the example of a scheme which embeds by perturbing the elements of some set R. If we try to embed repeatedly in the same R, the watermark will not continue to be transparent. In which ever way R is chosen there will be a limit on the number of watermark bits that can be inserted in a transparent manner since the watermark bits rely on perturbation for their insertion.

Obviously, if attacks are considered the capacity of the scheme will come down. But, to begin with, the scheme seems to have an advantage over schemes that rely on perturbation for insertion when it comes to capacity. We can safely say that if we are abandoning the scheme for lack of capacity, that is not justifiable.

Nnn-Invertibility:

The scheme is invertible. In this watermarking scheme, since the process of embedding does not perturb the original image Alice's original image and her watermarked image are the same. The original image is public and Bob who has access to the original image can start off with the original image and watermark it afresh to claim ownership. Even a rudimentary attack will work since Bob can easily cook up his own R_B, K_B and W_B such that $R_B + K_B = W_B$.

Even if we impose the constraint that the watermark be allowable, Bob will not face any problems since he can start off with the original image(publicly available) and $f(I) = W_B$ (f is a one-way function) will give him his "allowable watermark"

Transparency-wise and capacity-wise the scheme looks promising, robustness-wise there are no obvious drawbacks but the scheme is easily invertible.

Any watermarking scheme that relies on allowable watermarks to ensure non-invertibility has to perturb the original image to create a watermarked image so that the wa-

termarked image is similar to but not same as the original image.

Suppose we allow a part of the watermark to perturb the image and embed the rest without perturbing as described in the scheme discussed so far, in theory, we are giving ourselves a chance to ensure non-invertihility and at the same time retain some of the favourable traits of watermark insertion without perturbation. We hypothesize a ***watermarking framework***.

3. A WATERMARKING FRAMEWORK

Given a watermark W with P components to he embedded in a given image I, we split the watermark into two main parts, an ***intrusive*** part and a ***non-intrusive*** part. The components forming the intrusive part are referred to as the ***intrusive components*** and the components forming the non-intrusive part are referred to as the ***non-intrusive components.*** Any **bit** in the binary representation of the non-intrusive part of the watermark **is** a ***non-intrusive bit*** and any bit in the binary representation of the intrusive part of the watermark **is** a ***intrusive bit.***

Any watermarking scheme that belongs to this framework treats the intrusive and the non-intrusive parts differently. Insertion of a intrusive component is by perturbation and insertion of a non-intrusive component is by generation of a key set K and thus without perturbation. Watermark detection treats the two parts accordingly.

Given a watermark W with P components, as we alter the percentage of intrusive and non-intrusive components we traverse from one extreme to another. At one extreme the entire watermark **is** non-intrusive. Transparency-wise and capacity-wise the best possible situation but, any scheme which embeds such a watermark is bound to be invertible. Another extreme is when the entire watermark is intrusive. Relatively speaking, any scheme which embeds such a watermark will perturb the image by the maximum extent, giving us the best possible chance to make the scheme non-invertible, but, transparency-wise and capacity-wise the scheme suffers most. **All** existing watermarking schemes belong to this category. Between the extremes, intermediate percentages of intrusive and non-intrusive parts expose all other shades.

We further hypothesize, that if it were possible to parameterize the governing factors to get governing parameters(values), it should be possible to choose the correct proportion of intrusive and non-intrusive parts to get a watermark. The watermarking scheme which inserts this watermark would match the governing parameter requirements. For a given application, if the governing parameters can be specified, the watermarking framework can be used to get a tailor made watermarking scheme.

The watermarking framework **is** just a concept or an idea, implementation of which requires one to identify the proportion of intrusive and non-intrusive components in the watermark and choose an embedding algorithm to embed the intrusive components.

The hypothesis seems to have only an intuitive basis. Many claims need to be justified.

The mere presence of intrusive components does not bring about non-invertibility. We have to find a way to use these componets to guarantee non-invertibility. How will the non-intrusive components behave in such a scenario? Will Bob have better chances of inverting the scheme?

The frameworkneither specifies an embedding algorithm for the intrusive components nor does it specify where the elements of R are going to come from to embed the non-intrusive components. This gives us the freedom to plug in any algorithm to embed the intrusive components. This freedom will truly translate to **an** advantage if we can say that the framework is at least as robust as the embedding scheme chosen for the intrusive components.

How do we handle non-intrusive and intrusive components to obtain bounds on capacity? Does the initial capacity. wise promise shown by the non-intrusive components translate to any tangible gains?

In the sections to come we briefly state the results of "the governing factors based analysis" for our watermarking framework.

3.1. Non-invertibility

If Bob sticks to IBM type of attack **[2]** to invert a watermarking scheme belonging to our framework, he needs to guess only the intrusive part of his watermark(W_{B_I}) to get his counterfeit original I_B. To get his watermark(W_B) allowed, Bob has to find a key S_B such that $f_{S_B}(I_B) = W_B$. Let m_B and n_B be the number of bits in the binary representation of W_{B_I} and W_B respectively. Bob has $2^{n_B - m_B}$ watermarks to work with [I].

Let, set L contain all these $2^{n_B - m_B}$ watermarks. All elements in L have the m_B hits in the hinary representaion of W_{B_I} in their most significant positions. Bob's simplified problem is to find a key S_B such that $f_{S_B}(I_B) \in L$. Not all W $\in L$ are useful to Bob. Bob would like to work only with valid watermarks. **A** $W \in L$ **is** a valid watermark if, there exists some key S such that $f_S(I_B) = W$. Otherwise, W is a invalid watermark.

If f_S is one-way, it will continue to act one-way, when the cardinality of L is 1(entire watermark is intrusive), or when the number of valid watermarks in L for any choice of W_{B_I} is **1.** On the other hand, the easiest cases for Bob to solve occur when the cardinality of L is 2^{n_B}(entire watermark is non-intrusive) or when all possible valid watermarks corresponding to I_B are in L.

3.1.1. *Bound on number of non-intrusive components*

Let a watermark with n bits be obtained using a one-way function f_S. Let m_{bound} be the least number of intrusive bits, such that, in each of the $2^{m_{bound}}$ possible L sets there exists at most one valid watermark. $n - m_{bound}$ is the maximum number of non-intrusive bits such that f_S continues to be one-way for Bob's simplified problem.

3.1.2. *Symmetric key one-way functions*

In case of functions like DES it is not" easy" to arrive at m_{bound} by making use of the definition given above. We need to start with an acceptably low probability of a hit for Bob, say $1/2^{32}$ and find the number of non-intrusive bits we can push in. Working with 64 bit DES and 32 non-intrusive bits, The probability that Bob finds the correct key is the following:
(probability that the chosen watermark W_B is valid) $\times$(probability that Bob chooses the right key)$\times$(cardinality of L set) = $(2^{56}/2^{64})$ x $(1/2^{56})$ x $2^{32} = 1/2^{32}$. In fact, we discover that the probability of a hit for Bob does not depend on the key length [1]. Thus, it is useful to work with one-way functions like **128 bit Twofish** which allow variable length keys.

3.1.3. *Goppa code based one-way function*

The McElice cryptosystem **[5]** which is based on the ***NP*** hard problem of decoding an arbitrary linear code suits our requirements.
Let ***k*** be the number of bits in a message m. Let n be the number of bits in the ciphertext ***c***. t is a system parameter.
G_G is a ***k*** x n generator matrix. for a binary ***(n,k)*** linear code, which can correct upto t errors.
G_S is any ***k*** x ***k*** non-singular matrix.
G_P is any n x n permutation matrix.
$\hat{G} = G_S G_G G_P$.
If used for encryption ***(8,)*** would be the public key and (G_S, G_G, G_P) would be the private key. $c = \hat{G}(m)$.
Since G_G is a generator matrix the minimum distance(d_{min}) will be at least $2t + 1$. If we choose the number of non-intrusive bits to he less than d_{min} and the make the traditional message space our key space, with the condition that $\hat{G}$ is obtained with I or some hash of I as the seed, then Bob's–simplified problem which is to find a key S_B such that, $G(S_B) \in L$ continues to be NP hard.

3.2. Robustness

Given a watermark to be embedded with a transparency criterion, let us assume that all the components are intrusive. The transparency criterion, if distributed over the intrusive components gives us the the average value by which each intrusive component can perturb the image. Stronger the watermark presence better the robustness of the scheme. Thus, greater the average perturbation greater will be the robustness. Given a fixed number of components in the watermark, if a percentage of them can be inserted without perturbation as in our framework, the average value by which each intrusive component can perturb the image for its insertion will be greater and thus the intrusive components are more immune to attacks.

It can be shown that the robustness of the intrusive components in our framework increases with the number of non-intrusive components as $\sqrt{\beta/\gamma}$, where β is the number of components in a given watermark, γ is the number of intrusive components. The non-intrusive components are at least as robust as the intrusive components **[1]**. Infact, while embedding the non-intrusive components, by choosing a large embedding constant their immunity to noise can be further increased.

4. CONCLUSION

The watermarking framework discussed above is our answer to the question "Can we watermark without perturbing an image?". The framework specifies a one-way function to ensure non-invertibility,the minimum number of intrusive components (maximum number of non-intrusive components) and a way to identify the key set K. One can plug in any scheme to embed the intrusive components. The robustness of the framework increases with the number of non-intrusive components. The capacity-wise gain in our framework, which is intutively obvious, has not been formally captured.

5. REFERENCES

[I] Hari Krishna **V.J. *Intrusive and Non-intrusive Watermarking,*** MSc thesis, Indian Institute of Science, January **2002.**

[2] Scott Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M.Yeung. ***On The Invertibility Of Invisible Watermarking Techniques,*** ICIP **1998,** pp. 540-543.

[3] Gerhard C.Langelaar, Iwan Setyawan and Reginald L and Lagendjik. ***Watermarking Digital Image and Video Data. A State-of-the-Art Overview,*** IEEE Signal Processing Magazine, pp. 20-46 September 2000.

[4] Ingemar J. Cox, Killian Tom, Leighton and Talal Shamoon. ***A Secure, Robust Watermark for Multimedia,*** ICIP **1996,** pp. 243-246.

[5] Rudolf Niederreiter. ***Intmduction to finite fields and their applications,*** Cambridge university press, **1994.**