Wavelet-Based Solutions to the Digital Image Watermarking Problem

Emina Torlak
emina@mit.edu
# Contents

1 INTRODUCTION........................................................................................................................................... 3

2 THE IMAGE WATERMARKING PROBLEM ........................................................................................................ 4

   2.1 THE GENERAL WATERMARKING PROCESS ......................................................................................... 4

   2.1.1. Watermark Encoding ....................................................................................................................... 5

   2.1.2. Watermark Decoding ....................................................................................................................... 6

   2.2 WATERMARKING APPLICATIONS ......................................................................................................... 7

3 WATERMARKS AND WAVELETS .............................................................................................................. 7

4 THE XIE ALGORITHM .................................................................................................................................. 8

   4.1 WATERMARK GENERATION .................................................................................................................. 9

   4.2 WATERMARK EMBEDDING .................................................................................................................... 9

   4.3 WATERMARK DETECTION ..................................................................................................................... 9

   4.4 SECURITY OF THE XIE WATERMARK

       4.4.1. Insufficient Fragility of the Watermark ...................................................................................... 10

       4.4.2. Underspecification of the Algorithm's Security Requirements .................................................... 11

5 THE DUGAD ALGORITHM ................................................................................................................................ 12

   5.1 WATERMARK GENERATION ................................................................................................................ 14

   5.2 WATERMARK EMBEDDING .................................................................................................................. 14

   5.3 WATERMARK DETECTION .................................................................................................................... 14

   5.4 SECURITY OF THE DUGAD WATERMARK .......................................................................................... 15

6 CONCLUSION .................................................................................................................................................. 17

7 REFERENCES................................................................................................................................................ 17
1 Introduction

The digital form of audio, images, and video has become the commercial standard in the past decade. Digitalized multimedia can be easily created, copied, processed, stored, and distributed using commercially and freely available software. The simplicity and cost efficiency of managing digitalized multimedia have greatly benefited both the content providers and the consumers [13]. Unfortunately, the digitalization of multimedia has also rendered the multimedia products vulnerable to digital piracy: illegal copying, use, and distribution of copyrighted digital data.

In an effort to fight digital piracy, the multimedia industry has developed various copyright protection mechanisms over the years. However, most of these mechanisms were quickly found to be flawed and insecure. For example, the Content Scrambling System (CSS) for DVDs was broken within a few weeks of its release in November 1999, illustrating both the weakness of the employed encryption algorithm and the unsuitability of encryption as a copyright protection mechanism [11]. The industry’s failure to develop a robust method for thwarting digital pirates has prompted active academic research into the copyright protection problem. So far, the research has yielded a promising general solution to the problem, the digital watermarking paradigm, and a myriad of effective algorithms based on the general model.

This paper presents and analyzes two wavelet-based digital image watermarking schemes: the authentication algorithm proposed by Xie et al [15] and the copyright protection algorithm proposed by Dugad et al [4]. Section 2 outlines the image watermarking problem along with some common applications of the watermarking model. Section 3 motivates the use of wavelets in the design of image watermarking schemes. Sections 4 and 5 present the Xie [15] and Dugad [4] algorithms, published analyses of their security against common watermark attacks, and a specification of the privacy requirements necessary for the algorithms’ security. Finally, Section 6 summarizes and concludes the paper.
2 The Image Watermarking Problem

Image watermarking is the process that imperceptibly embeds a watermark $w$ into a host image $x$ to form the watermarked image $x_w$ [12]. The watermark $w$ is generally a vector containing either pseudo-random bits or pseudo-random samples from a probability distribution. Much like a real paper watermark, $w$ is physically bound to $x_w$ and can be detected to make an assertion about $x_w$ [9].

The watermark $w$ should be recoverable from the watermarked image $x_w$ even if $x_w$ is altered by one or more image processing procedures such as filtering, geometric distortions, re-sampling, and lossy compression. The application context of a watermarking scheme determines which image processing transformations of $x_w$ should not hinder watermark detection. For example, watermarking schemes used for image authentication are required to withstand distortions of $x_w$ introduced by file format conversion, compression, and re-sampling. However, such schemes must reject $x_w$ if $x_w$ is perceivably different from $x$ due to filtering, “doctoring,” or geometric distortions, since the purpose of authentication watermarking is to detect significant modifications of the image.

2.1 The General Watermarking Process

Abstractly, a watermarking scheme is the six-tuple $(X, W, K, G, E, D)$ where [13]

1. $X$ represents the set of original or host images.
2. $W$ is the set of all watermarks $w$ such that $\exists x \in X, k \in K, w = G(x, k)$.
3. $K$ is the set of numbers called watermark keys.
4. $G$ is the algorithm that generates $W$ using $K$ and $X$:
   \[ G: X \times K \rightarrow W, \ w = G(x, k) \]
5. $E$ denotes the encoding algorithm that embeds a watermark $w$ in an image $x$ with some strength $\alpha$:
   \[ E: X \times W \times \mathbb{R} \rightarrow X, \ x_w = E(x, w, \alpha) \]
6. $D$ is the decoding algorithm that detects whether a watermark $w$ is present in an image $x$:
   \[ D: X \times W \rightarrow \{0, 1\}, \ D(x, w) = \begin{cases} 
   1 & \text{if } w \text{ exists in } x \\
   0 & \text{otherwise}
\end{cases} \]
A generic watermarking scenario is depicted in Figure 1. The watermark generator takes an original image $x$ and a key $k$ to generate the watermark $w$. The encoder then embeds $w$ into $x$ to produce a watermarked image $x_w$. The watermarked image is distributed and potentially distorted by image processing operations and targeted watermark attacks. In the last stage of a watermarking scenario, the decoder attempts to ascertain the presence of the original watermark in the distorted image.

### 2.1.1. Watermark Encoding

The watermark encoding algorithm, $E$, embeds a watermark either in the spatial or a transform domain of an image. Spatial domain watermarking schemes are generally more computationally efficient than the schemes operating in a transform domain. However, transform domain watermarks are far more resistant to various image processing attacks [1]. As a result, almost all recently proposed watermarking schemes operate in a transform domain.

A generic transform domain encoder $E(x, w, \alpha)$ works as follows:

1. Let $T$ be the transform in whose domain $E$ operates. Compute $T(x)$ to get transform coefficients $v_1, v_2, ..., v_n$.
2. For each $v_i$, $1 \leq i \leq n$, compute $v'_i = f(v_i, w_i, \alpha)$
3. Output $x_w = T^{-1}(v'_1, v'_2, ..., v'_n)$

---

Figure 1. A General Watermarking Scenario
Transform domain watermarking schemes can be classified into additive, multiplicative, and quantization-based [12], depending on how their encoders compute the $v'_i$. Additive encoders [1] use the function $f(v_i) = v_i + \alpha w_i$ to produce the watermarked coefficients $v'_i$. Multiplicative encoders [4] perform the embedding with the formula $f(v_i) = v_i(1 + \alpha w_i)$. Finally, quantization-based encoders [15] generate the watermarked coefficients $v'_i$ using some non-linear function $f(v_i)$, [possibly some coefficients $v_j$ other than $v_i$, $w_i$, $\alpha$].

2.1.2. Watermark Decoding

Watermarking schemes can also be categorized according to their decoding technique as blind, semi-blind, or non-blind [8]. Non-blind and semi-blind decoders need the original image or some information about it to perform watermark detection. Blind decoders, on the other hand, are capable of detecting the presence of a watermark using only the watermarked image.

Non-blind watermarking schemes perform comparably to blind watermarking schemes against common image processing attacks. However, Ratakonda et al have shown in [10] that an additive or multiplicative non-blind watermarking scheme can be easily thwarted with a collusion attack that exploits the scheme’s use of the original image for detection. Consequently, blind schemes have become prevalent in the newer literature on digital watermarking.

There are two common ways in which blind watermark decoders decide whether a given watermark $w$ is present in the image $x_w'$:

1. Extract the watermark $w'$ from the transform coefficients $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m$ of $x_w'$ and test if $c(w, w') > t$, where $c$ is the comparison function and $t$ is the detection threshold. The form of $w$ determines which comparison function $c$ is used. For example, if $w \in \{0, 1\}^s$, $c$ might compute the percent bitwise similarity between $w$ and $w'$: $c(w, w') = 1 - \frac{1}{s} \sum_{i=1}^{s} w_i \oplus w'_i$. The extract-and-compare method of detection is usually employed by quantization-based watermarking schemes.

2. Compute a correlation measure $z$ between $w$ and the transform coefficients $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m$ and test if $z$ is greater than the detection threshold $t$. This form of detection is commonly used with additive or multiplicative schemes with the threshold set to $\frac{\alpha}{sm} \sum_{i=1}^{m}|\hat{v}_i|$, where $\alpha$ is the
watermarking embedding strength used by $E$ and $s$ is either 2 or 3. The correlation measure $z$ is found by applying the Central Limit Theorem: 
\[ z = \frac{1}{m} \sum_{i=1}^{m} \hat{v}_i w_i. \]

### 2.2 Watermarking Applications

Copyright protection is one of the most commonly proposed applications of digital watermarking. A watermarking scheme can provide copyright protection in two ways: as a tool for identification of unauthorized copies and as a mechanism for demonstration of ownership. In the first scenario, the watermarking scheme should have an efficient decoder which can be used by a web-crawler to quickly track down unauthorized copies of watermarked images. In the second case, the scheme should be blind, private, and non-invertible [3].

Image authentication is another popular watermarking application. Authentication watermarks are used to detect image tampering visible to the human eye. As a result, any intentional modification of the image should break an authentication watermark. However, image processing operations that preserve perceptual similarity (e.g. file format conversion) should not interfere with watermark detection.

Digital watermarking techniques can also be used for image fingerprinting (insertion of product identification codes) and steganography (data hiding) [8]. These applications place different requirements on a watermarking scheme’s robustness and embedding capacity. A scheme designed for steganographic purposes should be able to embed the maximum possible amount of information into an image without introducing any perceptible distortions. The robustness of such a scheme is irrelevant. On the other hand, watermarking schemes used for fingerprinting need not have high capacity but must be robust to both intentional and unintentional attacks.

### 3 Watermarks and Wavelets

Most image watermarking schemes operate either in the Discrete Cosine Transform (DCT) or the Discrete Wavelet Transform (DWT) domain. A few watermarking algorithms employ more exotic transforms such as the Fourier-Mellin Transform and the fractal transform [7]. The DWT domain is better suited for image watermarking than the DCT and other transform domains for several reasons [7]:

...
1. The DWT offers excellent **space-frequency localization** of salient image features such as textures and edges. Specifically, the high-frequency content of an image corresponds to large coefficient in the detail subbands. Hence, watermark encoders operating in the wavelet domain can easily locate the high-frequency features of an image and embed most of the watermark energy there. Such embedding will result in implicit visual masking of the watermark since the Human Visual System (HSV) has a limited ability to detect high frequency signals [4],

2. The wavelet transform’s **multi-resolution representation** of images facilitates progressive transmission of image data and hierarchical decoding of nested watermarks.

3. The DWT provides **superior modeling of the HVS**. The dyadic frequency decomposition of the wavelet transform resembles the pyramid decomposition of the hypothetical Cortex Transform [14] which models the human visual system. As a result, the DWT allows the different perceptual bands of the HVS to be excited individually.

4. The wavelet transform is **computationally efficient**. The DWT can be computed in linear time, while the DCT has $O(n \cdot \log n)$ time complexity.

5. The DWT is very **flexible**: there are infinitely many wavelet filters. The multitude of possible filters and filter bank configurations enables highly customized processing of individual images. The flexibility in the choice of wavelet filters can also be exploited to increase the security of the watermarking schemes operating in the wavelet domain [6].

Wavelet-based watermarking schemes were first described in the late 1990s. Two of the earliest wavelet-based schemes were proposed by Dugad et al [4] and Xie et al [15]. The following two sections present the Xie and Dugad algorithms and analyze their security against various attacks.

### 4 The Xie Algorithm

The Xie algorithm, also known as the “robust bit engraving” algorithm, was designed by Liehua Xie and Gonzalo R. Arce at the University of Delaware. The algorithm is a blind quantization-based watermarking scheme originally intended for image authentication. It is described in two papers, [1] and [15]. The quantization formula published in [15] slightly differs from that found
in [1]. This paper uses the formula given in [15]. Sections 4.1, 4.2 and 4.3 present the Xie generator, encoder and decoder. Section 4.4 discusses the scheme’s security.

4.1 Watermark Generation

A Xie watermark is generated by encrypting a soft digest of the original image $x$ with a secret key $k_{\text{secret}}$. An edge map of $x$ is used as the soft digest.

4.2 Watermark Embedding

First, the input image $x$ is decomposed using the wavelet filter $\text{wavelet}$ and the level of decomposition $\text{level}$ to obtain one matrix of approximation coefficients, $cA$, and $3 \times \text{level}$ matrices of detail coefficients. Then, we slide a non-overlapping $3 \times 1$ running window through the rows of $cA$, as shown in Figure 2. At each window position, one bit of the watermark $w$ is embedded by quantizing the median bit as follows:

1. Let $b_{\min(i)}$, $b_{\text{med}(i)}$, and $b_{\max(i)}$ denote the smallest, median, and largest coefficient at window position $i$, respectively.

2. Compute $k = \left\lfloor \frac{b_{\text{med}(i)} - b_{\min(i)}}{S_\alpha} \right\rfloor$ where $S_\alpha = \alpha \frac{|b_{\max(i)}| + |b_{\min(i)}|}{2}$ and $\alpha$ is the embedding strength.

3. Modify $b_{\text{med}(i)}$ to reflect the value of the watermark bit $i$ according to the formula

$$b_{\text{med}(i)}' = \begin{cases} b_{\min(i)} + k \cdot S_\alpha & \text{if } k \mod 2 = w_i \mod 2 \\ \min(b_{\min(i)} + (k + 1) \cdot S_\alpha, b_{\max(i)}) & \text{otherwise} \end{cases}$$

4.3 Watermark Detection

The Xie extraction algorithm mirrors the embedding process described above. We decompose the watermarked image $x'$ with the same filter bank used for embedding. A $3 \times 1$ is shifted through $cA'$. The watermark bit $w_i'$ associated with a window position $i$ is extracted from $b_{\text{med}(i)}'$ as follows: if $\left| b_{\text{med}(i)}' - (b_{\min(i)}' + k \cdot S_\alpha) \right| < \left| b_{\text{med}(i)}' - (b_{\min(i)}' + (k + 1) \cdot S_\alpha) \right|$ then $w_i' = k \mod 2$, else $w_i' = (k + 1) \mod 2$. 
The extracted watermark $w'$ is compared to the original watermark $w$ using the percent bitwise similarity measure from Section 2.1.2. This has the same effect as decrypting $w'$ with $k_{secret}$ and comparing the result to the edge map of $x'$ since $x$ and $x'$ should have the same edge map if $x'$ is perceptually unaltered. Since the expected similarity of two random sequences of bits is 0.5, we choose 0.6 as the detection threshold. In other words, if $c(w, w') \geq 0.6$ we consider the watermark $w$ to be present in $x'$.

![Diagram of Xie's Bit Etching Algorithm]

**Figure 2. Xie's Bit Etching Algorithm**

### 4.4 Security of the Xie Watermark

Meerwald [8] has demonstrated that the Xie algorithm is resistant to various image processing attacks including median filtering, geometric distortions, changes in color-depth, down-scaling, smoothing, gamma correction, cropping, and lossy compression (JPEG, JPEG2000, SPIHT, IW4, EPIC, EPWIC-1, WaveKit, and EZW). We will show that the Xie algorithm is insufficiently fragile for image authentication and that the algorithm’s developers underspecified its security requirements.
4.4.1. Insufficient Fragility of the Watermark

As stated in Section 2.2, the goal of image authenticating watermarks is to detect perceivable alterations of the watermarked image. A good image authentication watermark will break if the watermarked image is perceptually degraded by an attacker. The Xie watermark does not satisfy this fragility requirement: the original watermark $w$ can still be detected in a Xie watermarked image $x'$ even when the brightness of $x'$ is reduced by a large factor (which results in severe image degradation). For example, Figure 3 shows the Xie watermarked image of Lena and the same image with a four-fold reduction in brightness. Although the attacked image is clearly degraded, the similarity between the original and the attacked watermark is 0.9, which is well above the detection threshold of 0.6. Note that this attack works because reducing the brightness of an image does not significantly alter its edge map.

![The Xie Watermarked Lena](image1.png) ![Brightness Reduction Attack on the Xie Watermarked Lena](image2.png)

Figure 3. Brightness Reduction Attack on the Xie Watermark
4.4.2. Underspecification of the Algorithm’s Security Requirements

The developers of the Xie algorithm fail to explicitly state which embedding parameters beside the watermark should be kept secret. We will show that the privacy of wavelet, level, and $\alpha$ is imperative to the security of the Xie scheme.

Suppose that the attacker, $A$, intercepts a Xie watermarked image $x'$. If $A$ knows which wavelet, level of decomposition, and embedding strength were used to watermark $x'$, $A$ can launch the following attack on $x'$:

1. Use the Xie decoder with parameters $x'$, wavelet, level, and $\alpha$ to extract the embedded watermark $w'$.

2. Produce an attacked image $x_A$ such that $x_A$ and $x'$ are perceptually different, but their edge maps are similar enough to induce detection. (This can be done by introducing low-frequency distortions into $x'$)

3. Use the Xie encoder with parameters $x_A$, wavelet, level, $\alpha$, and $w'$ to create the fake watermarked image $x_F$.

From the construction of $x_F$, we know that $w'$ will be detected in $x_F$ and validated. The attack is more powerful than the brightness reduction attack because it provides $A$ with finer control over the nature of distortions added to $x'$. Fig demonstrates the attack on the Xie watermarked Lena image: the fake image on the right hand side has the phrase “Who let the bits out?” added to the top of the image (above Lena’s hat). Note that the edge maps of the two images are the same.

5 The Dugad Algorithm

The Dugad algorithm was developed by Rakesh Dugad, Krishna Ratakonda, and Narendra Ahuja at the University of Illinois. It is a blind multiplicative scheme for copyright protection based on the algorithm by Cox et al [2]. The details of the Dugad encoding and decoding process are published in [4]. Sections 5.1, 5.2 and 5.3 describe the scheme’s generator, encoder and decoder. Section 5.4 discusses the security of the Dugad watermark.
The Xie Watermarked Lena
\(\text{wavelet} = \text{’db4’}, \text{level} = 2, |w| = 100, \alpha = 0.05\)

Fake Watermarked Lena

Figure 4. Known Parameter Attack on the Xie Watermark
5.1 Watermark Generation

The Dugad watermark is a sequence of pseudo-random numbers drawn from the normal distribution with $\mu = 0$ and $\sigma^2 = 1$. The sequence is generated using a standard pseudo-random number generator seeded with the owner secret key $k_{secret}$. The size of the sequence is required to match the size of the image so that the locations manipulated during embedding and detection are fixed. Fixing the manipulated locations in this way avoids the dependence on the ordering of coefficients that is inherent in Cox’s scheme [2].

5.2 Watermark Embedding

We first transform the input image $x$ using the wavelet filter $\text{wavelet}$ and the level of decomposition $\text{level}$ to obtain one matrix of approximation coefficients and $3*\text{level}$ matrices of detail coefficients. Then, we pick all the detail coefficients $V_1, V_2, ..., V_P$ greater than the threshold $T_1$ which is generally set to 40. We replace each selected coefficient $V_i$ with $V_i' = V_i + \alpha |V_i| w_i$. As a final step, the inverse DWT of all the coefficients is computed to obtain the watermarked image $x'$. The embedding process is depicted in Figure 5.

5.3 Watermark Detection

The Dugad decoder works by correlating the watermark $w$ with the DWT coefficients of the watermarked image $x'$. The input image is decomposed with the filter bank used for embedding to obtain the approximation and detail coefficients. We select all the detail coefficients $\hat{V}_1, ..., \hat{V}_M$ greater than the threshold $T_2 > T_1$. ($T_2$ is usually taken to be 50.) Then, we correlate the chosen coefficient with the watermark $w$ using the formula $z = \frac{1}{M} \sum \hat{V}_j w_i$. The obtained value is compared to the detection threshold $s = \frac{\alpha}{2M} \sum \hat{V}_i$. If $z > s$ then the watermark $w$ is present in $x'$. The detection process is also shown in Figure 5.
5.4 Security of the Dugad Watermark

Dugad et al [4] have shown that their watermarking algorithm has better detection rates than Cox’s algorithm which operates in the DCT domain [2]. Meerwald’s more exhaustive tests [8] demonstrate that the algorithm’s performance against common image processing attacks is comparable to that of the Xie watermarking scheme. We will show that the security of Dugad’s watermark depends on constricting the attacker $A$ to operate by the rules of the General Watermarking Framework (GWF) defined in [13].

Suppose the attacker $A$ is allowed to use an arbitrary generation algorithm $G$ to generate his watermark. The attacker also has all well-known watermarking schemes at his disposition. Given these assumptions, the attacker $A$ can launch the following collusion attack on the Dugad watermark:

---

**Figure 5. The Dugad Watermarking Scheme**
1. Define $G$ to be the watermark generation algorithm $G: X \times K \rightarrow W$ which, given a binary key $k$ and an image $x$, outputs the watermark $w = G(x, k) = \text{edge-map}(x) \oplus k$.

2. Define $E$ to be the Xie encoder and $D$ to be the Xie decoder. (The Xie algorithm can be used for copyright protection if the embedding parameters are kept secret.)

3. Run the Xie decoder $D$ on the Dugad watermarked image $x'$ with arbitrarily chosen parameters $\text{wavelet}$, $\text{level}$ and $\alpha$. The decoder will output some binary sequence $b$.

4. Compute $k_{\text{fake}} = \text{edge-map}(x') \oplus b$ and use it to claim that $x'$ is the original image watermarked with $k_{\text{fake}}$ using the Xie encoder $E$.

To see why this attack works, note that $G(x', k_{\text{fake}}) = \text{edge-map}(x') \oplus k_{\text{fake}} = b = D(x', \text{wavelet}, \text{level}, \alpha)$. The key $k_{\text{fake}}$ was constructed to generate a “watermark” that is an intrinsic property of the attacked image $x'$. In fact, the attacker’s choice of $E$ and $D$ ensures that the watermark corresponding to $k_{\text{fake}}$ is also present in the true original image $x$. The watermark “extracted” using the Xie decoder comes from the low pass coefficients of $x'$. However, since the Dugad algorithm operates only on the high pass coefficients of $x$, the approximation coefficients of $x'$ are the same as the low pass coefficients of $x$. Hence, the watermark generated from $k_{\text{fake}}$ is in both $x'$ and $x$. This means that both the true owner of $x$ and the attacker have equal claim to the image.

The attack described above is an instance of the general SWICO (Single Watermarked Image Counterfeit Original) attack described in [3]. It should be noted that all watermarking algorithms are susceptible to some form of this attack if the attacker is allowed to use arbitrary watermark generating functions. The SWICO attack cannot be performed within the limits of the GWF which stipulates that only valid watermarks can be used by the watermark encoder $E$ [13]. A watermark is valid if and only if it is generated by a valid watermark generator $G$. All valid generators are required to be non-invertible functions of $x$ and $k_{\text{secret}}$. In other words, a valid generator $G$ cannot have an efficiently computable inversion function $G^{-1}: W \rightarrow K$ such that $G(x, G^{-1}(w)) = w$ for a given watermark $w$. 
6 Conclusion

Digital image watermarking is a hard problem that has not been fully solved despite a heavy industrial demand. However, the researchers have identified some useful tools that will be invaluable in the construction of industry-grade image watermarking software. One such tool is the Discrete Wavelet Transform.

This paper presented and analyzed two image watermarking schemes that operate in the DWT domain. We examined the security requirements of these schemes and explored their limitations in the context of various threat models. We found that the Xie scheme is insufficiently fragile for image authentication purposes and that its privacy requirements were underspecified. The Dugad scheme was shown to be insecure outside the General Watermarking Framework [13].

7 References


