RESEARCH

Digital Image Watermarking: A Formal Model, Fundamental Properties, and Possible Attacks

Hussain Nyeem[1*], Wageeh Boles[1] and Colin Boyd[1,2]

Abstract

While formal definitions and security proofs are well established in some fields like cryptography and steganography, they are not as evident in digital watermarking research. A systematic development of watermarking schemes is desirable, but at present their development is usually informal, ad hoc, and omits the complete realization of application scenarios. This practice not only hinders the choice and use of a suitable scheme for a watermarking application, but also leads to debate about the state-of-the-art for different watermarking applications.

With a view to the systematic development of watermarking schemes, we present a formal generic model for digital image watermarking. Considering possible inputs, outputs, and component functions, the initial construction of a basic watermarking model is developed further to incorporate the use of keys. On the basis of our proposed model, fundamental watermarking properties are defined and their importance exemplified for different image applications. We also define a set of possible attacks using our model showing different winning scenarios depending on the adversary capabilities. It is envisaged that with a proper consideration of watermarking properties and adversary actions in different image applications, use of the proposed model would allow a unified treatment of all practically meaningful variants of watermarking schemes.

Keywords: digital watermarking; data protection; image watermarking; watermarking model

1 Introduction

Digital watermarking—a data hiding technology—has already justified its suitability for different multimedia applications. Watermarking generally operates on different digital media or cover objects (*e.g.*, image, audio, video) and is considered to have three major components [1, 2]: watermark *generation*, *embedding*, and *detection*. Watermark generation yields the desired watermark, which can optionally depend on some keys. The generated watermark is embedded into the cover object by the watermark embedding, sometimes based on an embedding key. During detection, the embedded watermark in a cover object is extracted and verified. The basic realization of watermarking may be valid for other multimedia applications; however, we restrict our attention in this paper only to the digital image applications.

An image watermarking application may have different objectives, which determine the necessary watermarking properties for that application. Those objectives can be classified into two types: (i) security objectives (*i.e.*, to achieve certain security properties such as integrity of the watermarked image) and (ii) non-security objectives (*e.g.*, annotation for an efficient image-database management). Achieving these objectives requires determining and considering the necessary properties of the individual watermarking components. The watermark generation and embedding properties generally include *visibility*, *blindness*, *embedding capacity*, and *perceptual similarity*. Similarly, *blindness*, *robustness*, *error probability*, *etc.* are studied for watermark detection. (We formally define these properties later in Sec. 4. Until then, inverted commas are used to refer to them for their abstract meaning.) A general consideration of these properties, however, is more than difficult for the diverse requirements of the applications. Consequently, without a proper consideration of the properties and the application scenarios, various watermarking schemes are being developed and evaluated.

Proper consideration of watermarking properties and application scenarios, on the other hand, is highly critical for the development and use of a watermarking scheme. A loose consideration of the properties may af-

[*] Correspondence: hussain.nyeem@student.qut.edu.au
[1] School of Electrical Eng. & Computer Science, Queensland University of Technology (QUT), Brisbane, Queensland, Australia
Full list of author information is available at the end of the article

fect the overall watermarking performance. Similarly, an improper realization of an application scenario may leave security vulnerabilities. For example, if the development (*i.e.*, design and evaluation) of a scheme is motivated by the high embedding capacity and high perceptual similarity requirements (and thus ignores the other properties), the scheme may eventually require high embedding time. On the other hand, in an image content authentication application, if the scenario is not considered properly (*e.g.*, a watermark is generated without considering the required properties such as "collision resistance" property), the scheme can have security flaws and may not be reliable in practice [3]. Therefore, a *systematic development* of watermarking schemes is essential.

A systematic development means to have mathematical formalism and operation determination for watermarking schemes. Here, operation determination helps identify the objectives and properties of a watermarking scheme with their explicit consideration for an application scenario, and mathematical formalism is used to specify them. An informal study of watermarking is easier to grasp first, but its formal study is desirable since formalism has several benefits: (*i*) the potential to provide rigorous analysis of the required watermarking properties, (*ii*) the completeness for resolving ambiguities and misconceptions, and (*iii*) the readiness for supporting a computer aided fashion of analysis.

However, the present development of watermarking schemes is rather informal, ad hoc, and usually omits the realization of the application scenarios as mentioned above. This practice not only hinders watermarking applications from choosing a suitable scheme, but also leads to debate about the state-of-the-art for different watermarking applications. Addressing this problem requires a complete generic model with well defined properties of digital watermarking as a basis for its formal study. Since watermarking may also need to achieve various security properties (along with any non-security objectives), the expected adversary capabilities must also be considered.

In support of a systematic development (*i.e.*, design and evaluation) of the watermarking schemes, in this paper, we aim at developing a formal generic model of digital image watermarking. A generic and formally defined watermarking model gives the big picture of watermarking and helps identify all of its possible variants for different (image, video, *etc.*) applications. In other words, by determining the required (watermarking) inputs, outputs, and properties for different objectives, this model helps characterize a watermarking scheme. Using the proposed model, we seek to define a set of watermarking properties based on the application requirements. The proposed model also helps thorough analysis of watermarking schemes. An incomplete model here may lead to an inadequate computational analysis of a scheme resulting in various technical flaws and protocol weaknesses, which can be exploited later by an adversary. To this end, we also study a set of possible attacks to show the winning conditions for an adversary in different scenarios.

This paper is organized as follows. Section 2 reviews the relevant literature addressing the need for a formal generic watermarking model. Section 3 presents the construction of a formal generic watermarking model. In Section 4, the systematic definition of necessary properties are given with examples to demonstrate their technical use in digital image applications. Section 5 explains different security aspects of the model providing with the common attack models. The conclusions are given in Section 6.

2 Related Work

The construction of an appropriate general model is a fundamental need for watermarking as discussed in previous section. However, only a few relevant research covers the adjoining fields of steganography and data-hiding [1, 4–15]. In this section, we briefly review different models proposed for watermarking (or its adjoining fields) and thoroughly consider a set of selected criteria to study them. Considering objectives, inputs-outputs, component functions, and underlying theory, we briefly overview those models below. We also summarize our findings in Tables 1 and 2.

Jian and Koch [5] presented a model for the abstraction of digital watermarking schemes. From the steganography and spread spectrum communication concepts, that model provides a common basis for performance evaluation of some earlier schemes. However, the inputs and outputs are incomplete for a general watermarking scenario. For example, a watermark is not clearly defined and considered as an identification code using *bit-noise*—the bit-stream of noise-like signals. Therefore, analysing various security issues (*e.g.*, vector quantization attacks [16] arising from an input image independent watermark generation), and abstraction of new schemes (which are not spread-spectrum communication based) may require a further development of that model.

Petitcolas *et al.* [4] illustrated a digital watermark embedding and recovery model from an information hiding viewpoint. To give an overview of the technique, a simplified data-hiding scenario is considered and thus any formal definition of the inputs, outputs, and component functions are omitted. The model, therefore, remains limited to describe a watermarking scheme in a more complete sense. For example, how the watermarking key and/or the *mark* (which represents either

Table 1 Summary of the models used in relevant studies.

Models in Use	Objectives	Inputs & Outputs	Component Functions	Underlying Theory	Limitations
Jian and Koch *et al*. [5]	To describe digital watermarking schemes	Original data Watermarked data Degraded data (as a copy of watermarked data) Identification code (as watermark)	Embedding (bit-carrier selector, bit-noise generator, bit-carrier modifier) Extraction (bit-carrier selector, bit-pattern matching)	Steganography Spread-spectrum communications Signal processing	Limited consideration of the inputs, outputs, component-functions, and watermarking properties for image applications Limited to spread-spectrum communication based watermarking schemes
O'Sullivan *et al*. [8]	To determine the optimal hiding strategy, where watermarking is considered as a game between an attacker and information hider	Input and output data (*e.g*., images, audio, *etc*. as a vector) Message (as watermark)	Encoder Decoder	Information theory Steganography	Limited consideration of the image application scenarios, inputs, outputs, component-functions, and watermarking properties
Cox *et al*. [11]	To examine the similarities between watermarking and traditional communication models	Cover-data (as a vector) Watermark message Watermarked cover-data	Perceptual distance function Encoding function Extraction function Mixing function	Spread-spectrum communications	Limited consideration of image application scenarios (*e.g*., that use only spread spectrum based schemes), inputs, outputs, component functions, and watermarking properties
Petitcolas *et al*. [4]	To illustrate a simplified case of watermarking concept	Mark (as fingerpring or watermark) Stego-image Marked-image	Embedding Recovery	Information hiding	Limited consideration of inputs, outputs, and components May not be useful to study image watermarking schemes rigorously
Cohen and Lapidoth *et al*. [9]	To compute the coding capacity of the watermarking game for a Gaussian cover text and squared mean error distortions	Cover-text Message Stego-text Secret key	Encoder Decoder	Game theory Information theory	Limited consideration of inputs, outputs, and components Watermarking is considered as a game in a copyright application scenario
Adelsbach *et al*. [12]	To analyse security of watermarking schemes against protocol attacks (*e.g*., copy, ambiguity attacks)	Unwatermarked object Watermarked object Watermark Key	Key generation Embedding Detection	Cryptography	Limited consideration of inputs, outputs, and components Application scenarios are limited to dispute resolving protocols
Barni *et al*. [13]	To provide a general security framework for robust watermark	Original content Watermark Watermarked content Key (for embedding and detection)	Embedding (feature extraction and mixing, watermark generation) Decoding	Information theory Cryptography Signal processing	The concept of fair and unfair attacks may not be realistic Limited consideration of inputs, outputs, and components (*e.g*., what original content includes)
Li *et al*. [1]	To illustrate the formulation of the security definitions and the attacker models	Original and watermarked work (as a vector) Watermark (as bit sequence)	Watermark generation Watermark embedding Watermark detector Perceptual distance function	Data-hiding Cryptography Signal processing	Limited consideration of inputs and outputs The model is represents only a simplified case of watermarking

Continued on next page

Table 2 Summary of the models used in relevant studies. (Continued from previous page.)

Models in Use	Objectives	Inputs & Outputs	Component Functions	Underlying Theory	Limitations
Moulin *et al*. [15]	To evaluate hiding capacity in an optimal attack context (as a data-hiding game)	Host-data (image, audio, video, *etc*.) Message Side information Composite data (contains hidden message)	Encoder Decoder	Information theory Data-hiding Game theory	Limited consideration of inputs, outputs, and component functions (*e.g*., inputs and outputs are not conventional for watermarking)
Mittelholzer [6]	To characterize embedding process and attacked stego-image (for analysing secrecy and robustness in terms of mutual information)	Cover-data Key Secret message	Stego-encoder Stego-channel Stego-decoder	Information theory Steganography	Limited consideration of inputs, outputs, and component functions More related to steganography schemes
Cachin [10]	To quantify steganographic security	Cover-text Stego-text Secret key	Key generation Embedding Extraction	Information theory Steganography	Limited consideration of inputs, outputs, and component functions More related to steganography schemes Limited to the passive attack scenarios
Adelsbach *et al*. [7]	To formalize robustness considered as a core security property, of watermarking	Cover-data Stego-data Watermark Key (for embedding and detection) Secret parameter (used as key-generation input)	Key generation Embedding Detection	Cryptography	Limited consideration of inputs, outputs, and component functions Limited to robust watermarking schemes

a fingerprint—hidden serial number, or a watermark—hidden copyright message) is chosen/generated needs to be explicitly defined.

In order to analyse watermarking as a classical communication system for digital multimedia data, Cox *et al*. [11] presented a generic communication model of watermarking. In that model, individual vectors generalize cover-data and distortion. Distortion is assumed to be additive, and a real valued function is considered to measure perceptual distance between content vectors. That model is suitable to describe an optimal embedding scheme that embeds a watermark with its largest possible size (in bits) to offer the highest possible detection ability. There may be some variants of such an embedding scheme (depending upon different watermarking properties like "blindness", "robustness" *etc*.). that can also be described using that model (by defining the functions in different ways). However, that model may not help to define and analyse an image watermarking scheme completely, because of its limited consideration of the inputs, outputs, and/or use of keys, in some application scenarios (*e.g*., authentication, tampering detection and recovery, *etc*.).

Mittelholzer [6] demonstrated a theoretical model to define a case of the embedding process and malicious modification, of a stego-message. The embedding process considers hiding a secret stego-message (as watermark), and thus mainly aims at achieving confidentiality and robustness properties in terms of mutual information. That model provides a theoretical basis for designing some watermarking schemes, for example, where the cover images have statistically Gaussian components. The model, however, may not be able to address many other watermarking properties due to limited considerations of the inputs, outputs and component functions. For example, the "blindness" property that helps determine the requirements of other inputs (different from the input image and watermark), which are not considered in the model.

Following a thorough security analysis, Li *et al*. [1] referred to a general watermarking model. Unlike many other models, that model considers the basic component functions more completely using the signal processing paradigm. It also allows a more structured approach to define various threat models. However, the model still has limited specifications of the inputs and outputs of its components. For example, a watermarking scheme may have other inputs (in addition to the input image and other multimedia signal referred to as *work*) to generate the watermark, which are not present in the model. As a result, it represents only a simplified case of watermarking and may not help realize the overall scenarios completely for the security or other watermarking requirements.

Barni *et al.* [13] presented a watermarking model to generally tackle the security analysis using an attack classification inspired by cryptographic models. Their model includes two main functions: watermark embedding and decoding. The embedding function has three steps: feature extraction from the original content; watermark generation from the message using a key; and feature mixing with the watermark. The decoding function decodes the hidden message from watermarked version using a decoding key. This realization indeed presents a basic watermarking application scenario. However, a more complete set of inputs and outputs, and the separation of functions (for example, separating watermark generation from embedding, and message decoding from watermark detection) may help describe a watermarking scheme with more insights for a broader application scenario. Besides, although modelling the watermark as a game is compelling for the security analysis, the concept of fair and unfair attacks may not be realistic.

Watermarking has also been studied [8, 9, 14, 15] using the formal concepts of game theory and information theory. O'Sullivan *et al.* [8] suggested watermarking can be defined as a game played between an information hider and an adversary. The attacker and information hider scenarios are further studied for watermarking [9, 14]. Later, Moulin and O'Sullivan [15] formalized a *distortion function*, *watermarking code*, and *attack channel*. The main limitation of the models used to demonstrate the game scenarios in those studies is that they only represent a set of cases of watermarking. Such an approach of defining a model can help address particular problems for an application, but may not be able to represent the overall watermarking scenario (which is required to develop a unified watermarking theory). In other studies [7, 12], watermarking models are used as an abstraction of security proofs.

The different models, discussed so far, are mainly established for different digital media and to individually describe and analyse different watermarking schemes. In other words, those models are not general in the sense that neither of them would be sufficient to study most of the digital image watermarking schemes available in the literature. Some of them are influenced by the underlying concept of steganography [5, 6, 8, 10], cryptography [1, 7, 12, 13], information theory [6, 8–10, 15], or spread spectrum communication [11]. In many cases [4–7, 12, 13], a key is used but their respective properties are not clearly defined, especially in achieving a specific security property. Watermark generation and its general inputs-outputs are not considered in most of them [4–7, 12]. A few researchers [5–7, 11] define necessary properties for their model, while others do not. All the above mentioned models are mainly motivated by the "robust" watermarking scenarios (*e.g.*, copyright protection), where unauthorized removal is of core interest. Moreover, the models studied so far are mostly incomplete to be a generic model in terms of: (*i*) considering the inputs, outputs, and basic components, (*ii*) defining necessary properties, and/or (*iii*) realizing the application scenarios. We therefore conclude that despite having a basic need for it, a formal generic image watermarking model is still lacking.

In our earlier work [2], we introduced a formal generic watermarking model for image applications addressing a gap in watermarking literature. We explored the need for the watermarking model and showed some uses of the model to define a few watermarking properties and attacks. In another follow-up work [3], we have also presented the use of the model in describing and analysing security of specific watermarking schemes, where we have shown how these schemes are violating the systematic definition of security. This paper, however, aims at incorporating further clarification and improvements on the constructions and definitions of the model and its uses. We consider here a relatively complete set of fundamental properties and wide range of application scenarios for digital images. With the aid of some practical examples, we also show the uses of the properties addressing a few hidden assumptions in current practice. Further, the set of expected adversaries are reconsidered to show how they can win with a particular attack. In the following sections, the main contributions are presented in three parts: (*i*) a formal watermarking model (Sec. 3), (*ii*) definitions and uses of fundamental properties (Sec. 4), and (*iii*) possible attacks on the watermarking security (Sec. 5).

3 A Formal Generic Watermarking Model

A formal generic watermarking model is of great importance. It is one of the most fundamental requirements for conceptualizing, systematic development and evaluation of the watermarking schemes, as discussed in Sec. 1. It helps avoid any confusion and misconceptions by defining the necessary inputs, outputs, and component functions of a watermarking scheme. The watermarking schemes described using a formal model offer the readiness for implementation and computer aided fashion of analysis. The required properties and design criteria of a watermarking application can also be defined by the model, which helps characterize a watermarking scheme for the application. The model also provides a means for defining attack models and thus for carrying out a rigorous analysis of a watermarking scheme. Moreover, a formal watermarking model creates a common platform for all possible watermarking schemes. Such a platform is expected not only to give a designer sufficient flexibility to describe any watermarking scheme, but also to help others understand the scheme in a systematic way.

In this section, we present a construction of a formal generic watermarking model in two stages, namely the *basic model* and the *key-based model*. The challenge here is to consider a "complete" set of watermarking inputs, outputs, and component functions in general from their specific information domains and function families. However, the problem can be reduced to a watermarking application(s), where a set of "possible" inputs, outputs and component functions can be defined in general to capture the fundamental properties of prominent schemes proposed today for the application(s). We therefore narrow down our scope to only the watermarking applications in digital images, and start constructing a basic model with considering the possible watermarking inputs, outputs, component functions used in the applications. Later, a key-based model is developed by incorporating keys to the basic model for completeness. This would allow a designer to achieve any required security properties (*e.g.*, authentication, confidentiality) and to employ any suitable cryptographic technique as a building block in a watermarking scheme.

3.1 Construction of a Basic Model

A basic model, as it implies, is expected to represent a basic scenario for the image watermarking applications. We firstly identify the fundamental components and their possible inputs and outputs of a watermarking scheme. Irrespective of the system and security requirements, a watermarking scheme can have three fundamental components as mentioned in Sec. 1 and shown in Fig. 1. In order for their systematic definition, we consider three functions: watermark generation, $G(\cdot)$, embedding, $E(\cdot)$, and detection, $D(\cdot)$, and define their possible inputs and outputs as shown in Table 3. The primary roles of these functions in an image watermarking application are described below. To denote different data (*e.g.*, inputs and outputs) within this context, in what follows, plain-letters indicate the original versions, and respective single-bar letters and tilde-letters indicate their watermarked and estimated versions accordingly.

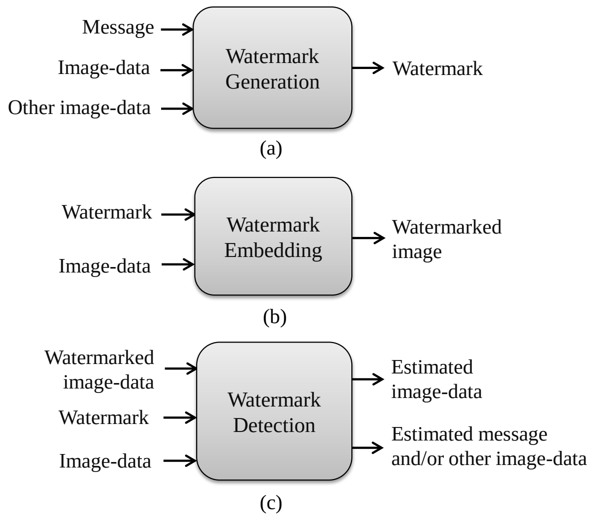

Figure 1 Fundamental components of (digital) image watermarking: (a) watermark generation, (b) watermark embedding, and (c) watermark detection.

Watermark generation, $G(\cdot)$. This function generates a suitable watermark according to the watermarking objectives in an application. In a simple data-hiding application, a watermark can be the embedding-data (*e.g.*, message, m, other image-data, j) itself (along with any side information). In an advanced application, a watermark may require to have certain properties (depending upon the watermarking objectives). For example, in a copyright protection application, a watermark may need to be "robust" against certain processing techniques and/or attacks. (We will discuss the "robustness" property in detail in Section 4.5.) Failure to consider those properties may result in technical flaws and security vulnerabilities. Although watermark generation is mainly constrained by the required properties, it starts with necessary inputs and their properties. For an image application, the generation function, $G(\cdot)$, can take image-data, i, and message, m and/or other image data, j as input, and outputs a watermark, w.

Watermark embedding, $E(\cdot)$. As the data-hiding component, watermark embedding function considers where and how to embed the watermark satisfying various requirements of the cover objects (here, digital images). For example, a "perceptual similarity" requirements (that control which pixels can be modified to what extent) of medical images may limit the embedding region [17]. (We will discuss the "perceptual similarity" property in detail in Sec. 4.1.) There are different domains (*e.g.*, spatial, transform) for embedding,

Table 3 Components of a basic watermarking model.

Components	**Inputs**	**Outputs**
Watermark generation, $G(\cdot)$	image-data, i message, m other image-data, $(j : j \neq i)$	watermark, w
Watermark embedding, $E(\cdot)$	image-data, i watermark, w	watermarked image data, $\bar{i}$
Watermark detection, $D(\cdot)$	watermarked image-data, $\bar{i}$ image-data, i watermark, w	estimated image-data, $\tilde{i}$ estimated message, $\tilde{m}$ estimated other image-data, $\tilde{j}$ or, failure, $\perp$

which are computed directly from an input image. Embedding types may also be different (*e.g.*, invisible, invertible or reversible, blind, *etc.*—will be discussed in Sec. 4). Irrespective of the embedding region, domain and type, however, an embedding function $E(\cdot)$ can take a watermark, w and the original image-data, i as input to output the watermarked image-data, $\bar{i}$.

Watermark detection $D(\cdot)$. This function helps make an objective decision (*e.g.*, to declare whether the content is authentic) and/or initiate further actions (*e.g.*, to extract the embedded data, to engage and retain users of the watermarked objects). In different application scenarios, the additional tasks may vary and depend on the binary decision (*i.e.*, *pass* or *fail*). The basic idea is that $D(\cdot)$ extracts the embedded watermark and regenerates another version of the watermark, from the inputs. If the regenerated version matches the extracted version, a *pass* signal is returned. (The *pass* signal is considered to pass the parameters such as the valid watermark, the estimated image-data, *etc.* to its dependent module that performs the additional tasks, which will be shown later in Fig. 3.) Otherwise a failure is output. The main constraints for this function thus can be the minimum error probabilities (*e.g.*, false negative/positive rates) and computation time. Like the functions, $G(\cdot)$ and $E(\cdot)$, the internal design of $D(\cdot)$ can also vary, but it generally takes watermarked image-data, $\bar{i}$, original image-data, i and a watermark, w to yield either an estimated image-data, $\tilde{i}$, message $\tilde{m}$ and other image-data, $\tilde{j}$, or a failure, $\perp$.

Thus, a basic watermarking scheme for digital images can be defined as a 6-*tuple* $(\mathbb{I}, \mathbb{M}, \mathbb{W}, G, E, D)$ such that:

(i) $\mathbb{I}$, the image-data space, is a set of tuples with value in the positive integers $\mathbb{Z}^+ = \{|a| \geq 0 : a \in \mathbb{Z}\}$. Each tuple is a set of coordinates, (x, y) for $2D$-space, or (x, y, z) for $3D$-space with $x, y, z \in \mathbb{Z}^+$. An element of image data space is called an image of $a \times b$ size for $2D$-space, and of $a \times b \times c$ for $3D$-space, where $a, b, c \in Z^+$ and $x = \{1, 2, 3 \cdots a\}$, $y = \{1, 2, 3 \cdots b\}$, and $z = \{1, 2, 3 \cdots c\}$. I,J,$\bar{I}$, and $\tilde{I}$ are the subsets of $\mathbb{I}$, where:
   - I is the set of original unwatermarked image-data;
   - J is the set of other image-data used for watermark generation and $J \cap I = \phi$;
   - $\bar{I}$ is the set of watermarked image-data;
   - $\tilde{I}$ is the set of estimated original image-data;
   - $\tilde{J}$ is the set of estimated other image-data.

(ii) $\mathbb{M}$ is the plaintext space, and $\mathbb{W} = \{0, 1\}^+$ is the watermark space. A *message* is a string of plaintext symbols. $M \subset \mathbb{M}$ is the set of original messages, and $W \subset \mathbb{W}$ is the set of original watermarks. $\tilde{M} \subset \mathbb{M}$ and $\tilde{W} \subset \mathbb{W}$ are the sets of respective estimates.

(iii) G is a function $G : I \times M \times J \to W$ that is used for watermark generation.

(iv) E is a function $E : I \times W \to \bar{I}$ that is used for watermark embedding.

(v) D is a function $D : \bar{I} \times I \times W \to \tilde{I} \times \tilde{M} \times \tilde{J} \cup \{\perp\}$ that is used for watermark detection, where $\perp$ indicates a failure.

(vi) a watermark w is *valid* if and only if it is obtained from valid inputs, (i, m, j) using the valid watermark generation function, $G(\cdot)$ such that, $G(i, m, j) = w$. Similarly, a watermarked image, $\bar{i} \in \bar{I}$ is valid if and only if $E(i, w) = \bar{i}$ for valid inputs, $(i, w) \in I \times W$. More formally, we can define a digital image watermarking scheme to be complete, if the following is true: for all $(i, m, j) \in I \times M \times J$ there exists $(\tilde{i}, \tilde{m}, \tilde{j}) \in \tilde{I} \times \tilde{M} \times \tilde{J}$, where $\tilde{i} \approx i, \tilde{j} \approx j$, such that $D(E(i, G(i, m, j)), i, G(i, m, j)) = (\tilde{i}, \tilde{m}, \tilde{j})$. Here, the symbol '$\approx$' denotes the *perceptual similarity* between two images. For example, $\tilde{i} \approx i$ implies that the perceptual content of i and $\tilde{i}$ are "sufficiently" similar to each other. (For more complete definition of *perceptual similarity* property, see Def. 4.1.)

It is worth noting here that we consider the original (unwatermarked) version of an image as the input image for the watermarking functions. In most cases, original images are used for watermarking. However, there may be cases where a (valid) watermarked version of an image can be used as an input image. For example, to update/re-embed a watermark in an existing watermarked image, one may need to use the present (or any earlier) watermarked version, rather than using the original image. It depends upon the application scenario which version of images are to be used (and how any restrictions on using them should

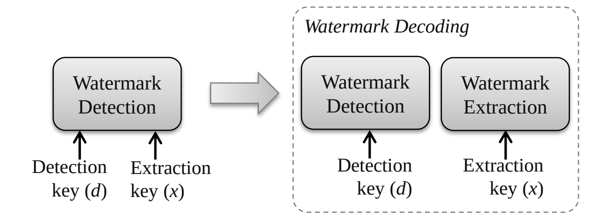

Figure 2 Deploying keys in the detection function of our basic model.

Table 4 Components of a key-based watermarking model.

Components		**Inputs**	**Outputs**
Key generation, $Key(\cdot)$		image-data, i message, m other image-data, $(j : j \neq i)$	generation key, g embedding key, e detection key, d extraction key, x
Watermark encoding	Generation, $G(\cdot)$	generation key, g image-data, i message, m other image-data, $(j : j \neq i)$	watermark, w
	Embedding, $E(\cdot)$	embedding-key, e image-data, i watermark, w	watermarked image-data, $\bar{i}$
Watermark decoding	Detection, $D(\cdot)$	detection-key, d watermarked image-data, $\bar{i}$ image-data, i watermark, w	$\left\{\begin{array}{l}\text{estimated image-data, } \tilde{i}\\ \text{estimated watermark, } \tilde{w}\end{array}\right.$ or, failure, $\perp$
	Extraction, $X(\cdot)$	extraction key, x watermarked image-data, $\bar{w}$ image-data, i estimated watermark, $\tilde{w}$	$\left\{\begin{array}{l}\text{estimated message, } \tilde{m}\\ \text{estimated other image-data, } \tilde{j}\end{array}\right.$ or, failure, $\perp$

be dealt with). However, this variation (in input image versions) can be studied as a special case of the proposed model, where the model may accept either an original image or its existing watermarked versions as an input. Therefore, we consider the fundamental scenario for the proposed model, where an (original) image is watermarked for the first time.

The construction of the above basic model is suitable for realizing a basic watermarking scenario, but it may not be sufficient to capture the recent watermarking advances. Although study of a complete watermarking model is still lacking, many advances are evident [18–22] in the present watermarking context. For example, the concepts of using keys and deploying cryptographic techniques are prominent in addressing different levels of security in various application scenarios such as content/owner authentication and copy-control. Such developments help obtain the combined benefits from the fusion of data-hiding and cryptographic techniques.

3.2 Towards a Complete Watermarking Model

To adopt and generalize the use of keys, we extend the basic scenario to a key-based scenario. We assume two individual keys, generation key, g and embedding key, e for $G(\cdot)$ and $E(\cdot)$, respectively. Although in our basic construction, for simplicity, $D(\cdot)$ is considered to perform the *detection* and *extraction* tasks inherently, this should naturally be split into separate functions for security reasons. We, therefore, separate the computation of extraction from $D(\cdot)$ using an additional function $X(\cdot)$, which we call the extraction function. Thus, an individual detection key, d and extraction key, x can be used as shown in Fig. 2. These two functions, $D(\cdot)$ and $X(\cdot)$ can be further defined as sub-functions of watermark *decoding* (to resemble our earlier construction) as shown in Table 4. The other two functions, $G(\cdot)$ and $E(\cdot)$ can similarly be the sub-functions of watermark *encoding*. Fig. 3 illustrates the watermark encoding and decoding processes.

We note here that the outputs $\left(\tilde{i}, \tilde{w}\right)$ of $D(\cdot)$ and $\left(\tilde{m}, \tilde{j}\right)$ of $X(\cdot)$ can be an exact estimate of their original versions respectively for a *non-blind* decoder (see Def. 4.3 for "blindness" property). Here, exact estimates of (m, j) are obtainable at $X(\cdot)$ from an exact estimate of w as $D(\cdot)$ outputs. For a *blind* decoder, to get an exact estimate of the input image, original information (that is compromised for embedding) is required by $D(\cdot)$. This requirement leads to the construction of $E(\cdot)$ as an *invertible* (or *reversible*) function, a major recent watermarking trend. (We discuss the "invertibility" or "reversibility" property later in Sec. 4.4.) Further, how exactly $\tilde{i}$, $\tilde{w}$, $\tilde{m}$ and $\tilde{j}$ can be produced depends on how much error is allowed in their estimation — an error in estimating $\tilde{w}$ at $D(\cdot)$ propagates through to yielding $\tilde{m}$ and $\tilde{j}$ at $X(\cdot)$. However, $\tilde{w}$ and $\tilde{m}$ are defined as bit strings, and for any decoder (*blind* or *non-blind*), they should be an exact estimate except for a few bit errors that can be handled by error correction codes.

Further, as shown in Fig. 3(b), the detection function in the watermark decoding invokes the extraction function, once the detection is completed. We note here that the detection function is executed independently, and may only output a pass or fail signal depending upon the existence of a valid watermark. This also means that, the extraction is not always required (depending upon the applications such as image content authentication). However, the extraction function can be performed after the detection, when required for the applications like image annotation, since extraction of the information carried by the watermark will

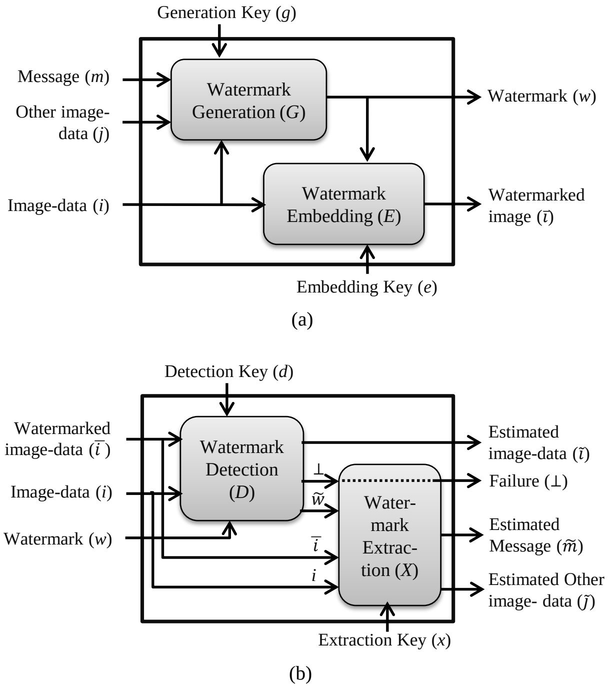

Figure 3 Key-based digital watermarking model: (a) watermark encoding, and (b) watermark decoding.

make sense, only if the image is passed by the detection (*e.g.*, ensuring the authenticity or integrity of the watermarked image).

We, therefore, develop the construction of a basic watermarking model (for digital images) further to incorporate the use of keys. We define here a key-based watermarking scheme as a 8-*tuple* $(\mathbb{I}, \mathbb{M}, \mathbb{W}, K, G, E, D, X)$ such that:

(i) I,J,$\bar{I}$,$\tilde{I}$, and $\tilde{J}$ are subsets of $\mathbb{I}$. Definition for the image-data space, $\mathbb{I}$, the plain text space, $\mathbb{M}$, the watermark space, $\mathbb{W}$, and their respective subsets are the same as defined in the basic model of Sec.3.1.

(ii) K is the set of all keys and a key is a sequence of m binary bits, where $m \in \mathbb{Z}^+$. Sets of watermark generation keys, K_g, embedding keys, K_e, extraction keys, K_x, and decoding keys, K_d are subsets of K (*i.e.*, $K_g \subset K$, $K_e \subset K$, $K_x \subset K$, and $K_d \subset K$).

(iii) $G = \{G_g | g \in K_g\}$ is a family of functions $G_g : I \times M \times J \to W$ that is used for *watermark generation.*

(iv) $E = \{E_e | e \in K_e\}$ is a family of functions $E_e : I \times W \to \bar{I}$ that is used for *watermark embedding.*

(v) $D = \{D_d | d \in K_d\}$ is a family of functions $D_d : \bar{I} \times I \times W \to \tilde{I} \times \tilde{W} \cup \{\perp\}$ that is used for *watermark detection.*

(vi) $X = \{X_x | x \in K_x\}$ is a family of functions $X_x : \bar{I} \times I \times \tilde{W} \to \tilde{M} \times \tilde{J} \cup \{\perp\}$ that is used for *watermark extraction.*

(vii) For each key, $g \in K_g$ and $e \in K_e$ there exists $d \in K_d$ and $x \in K_x$ respectively *i.e.*, for all $(i, m, j) \in I \times M \times J$, there exists $\left(\tilde{i}, \tilde{w}\right) \in \tilde{I} \times \tilde{W} | \tilde{i} \approx i$ such that $D_d\left(E_e\left(i, G_g\left(i, m, j\right)\right), i, G_g\left(i, m, j\right)\right) = \left(\tilde{i}, \tilde{w}\right)$, and for all $\tilde{w} \in \tilde{W}$, there exists $\left(\tilde{m}, \tilde{j}\right) \in \tilde{M} \times \tilde{J} | \tilde{J} \approx j$ such that $X_x\left(E_e\left(i, G_g\left(i, m, j\right)\right), i, \tilde{w}\right) = \left(\tilde{m}, \tilde{j}\right)$.

At this point, we stress the properties of the keys that can differentiate between *private* and *public* watermarking schemes. We define a watermarking scheme as a *private key* (or simply *private* or *symmetric*) scheme if $d = e$, and $x = g$ (*i.e.*, if d and x can at least be easily computed from e and g, respectively). Otherwise, we call it a *public key* (or simply *public* or *asymmetric*) scheme if $d \neq e$ and $x \neq g$, and if computing d and x from e and g is "computationally infeasible" in practice respectively. The phrase "computationally infeasible" follows the standard definition in cryptography. Here, d and x are the private keys and e and g are the public keys. Similar to the watermarking keys, watermarking itself has many properties that may lead to its many practically meaningful variants for different applications. Before discussing these properties and defining them in Sec. 4, we present below a comparative study in support of our above model.

3.3 A Comparative Study

In comparison with the summary of existing models (Table 1 and Table 2), we summarize the features of our proposed model in Table 5. As discussed in Sec. 2, a common limitation is the narrow focus on a particular type of data-hiding, steganography or watermarking scenario with different objectives, in developing a watermarking model. This leads to considering a simplified set of inputs, outputs, and component functions. Although such a simplified and generalized model helps realize the application scenarios of some relevant schemes, in the formal watermarking context, they are incomplete and thus need to be re-defined to be used as a general model for image applications.

Our model addresses the major limitations of relevant models for studying image watermarking schemes. We believe that the proposed model is a first step towards a formalized conception of image watermarking, and allows a unified treatment of all its practically meaningful variants. Considering this, we also define a set of fundamental properties in following sections using our model to further strengthen the watermarking theory in the image application context.

Table 5 Summary of the proposed model.

Model in Use	Objectives	Inputs & outputs	Component functions	Underlying theory	Limitations
Proposed	To provide a means for the systematic development, and thus to develop a unified and more realistic theory, of digital image watermarking	Image-data (with different properties, *e.g.*, original, watermarked, *etc.*, see Fig. 3) Watermark Message Key (for each function)	Key generation Watermark generation Watermark embedding Watermark detection Watermark extraction	Digital image and signal processing Cryptography	May not be suitable for studying steganography schemes

4 Fundamental Watermarking Properties

Defining the properties of watermarking plays an important role in the systematic development of various schemes. For example, in developing a new scheme, the watermarking objectives determine a set of criteria (as discussed in Sec. 1). Each criterion can be expressed in terms of the minimum requirements for a relevant watermarking property. In the design phase, those requirements help characterize the scheme (*e.g.*, by setting constraints for the construction of watermarking functions). In the evaluation phase, measuring (with a suite of tests) how those requirements are fulfilled gives merit to the scheme. The relative importance of each property, thus, can be determined based on the application requirements. This also means that the interpretation and significance of watermarking properties can vary with the application. These properties, in practice, can be interpreted in terms of the inputs and outputs of watermarking components, use of keys, *etc*. They can also be mutually dependent, which requires a trade-off among the improvements in the properties [23] for an application.

In the image watermarking context, a number of defining properties (considering their relative importance) are studied below: *perceptual similarity*, *visibility*, *blindness*, *invertibility*, *robustness*, *embedding capacity*, *error probabilities*, and *security*. In the following sections, we formally define these properties using the developed watermarking model (Sec. 3) and show how they can be interpreted and used in a real application scenario. To simplify reading, from now on, the notations are used without explicitly giving their domains. For example, 'for all $a, b, c, \cdots$' will be used to mean 'for all $(a, b, c, \cdots) \in A \times B \times C \cdots$'.

4.1 Perceptual Similarity

The *perceptual similarity* (or *imperceptibility*) is one of the most important properties for the image applications. Since embedding distortion is inevitable, $E(\cdot)$ exploits the (relatively) redundant information of an image intelligently for a minimum of visual artefacts. In almost any image application, therefore, keeping a watermarked image perceptually similar to the original image becomes an important criterion. Perceptual similarity means the perceptual contents of the two images are "sufficiently" similar to each other, (and thus it is mainly studied for the *invisible* watermarking schemes; the "visibility" property is discussed below). The requirements for this property may vary with the application scenario. In order to ease the problem of dealing with these varying requirements, we now define the perceptual similarity property using a quantitative approach.

Definition 4.1 (Perceptual Similarity)**.** *Any two images, i_1 and i_2, are said to be (d, t)* perceptually similar, *if $d_j(i_1, i_2) \leq t_j$ for all similarity measures $d_j \in d \equiv \{d_1, d_2, \cdots, d_n\}$ and thresholds $t_j \in t \equiv \{t_1, t_2, \cdots, t_n\}$.*

Various measures are used to quantify the requirements for the perceptual similarity. For example, correlation quality (CQ), signal to noise ratio (SNR), peak or weighted SNR (PSNR or WPSNR), mean square error (MSE), structural similarity index (SSIM), mean or weighted SSIM (MSSIM or WSSIM), normalized cross-correlation (NCC), *etc*. However, no globally agreed and effective measures for visual quality currently exist [24]. In addition, not all the measures give the similar estimation. Therefore, we define perceptual similarity by defining a *similarity* measure, which is a set of n-suitable measures that help quantify the perceptual distance between two images. Now, we define two images to be perceptually similar (or imperceptible) for an acceptable value returned by all suitable measures defined for *similarity*.

As an example to use the above definition, we may consider two measures (*i.e.*, $n = 2$): PSNR and MSSIM, for the similarity measure, d such that $d_1 =$ PSNR and $d_2 =$ MSSIM. The given thresholds are: $t_1 = 60$ (dB) and $t_2 = 0.995$. Two images i_1 and i_2 are said to be perceptually similar if both $d_1(i_1, i_2) \geq 60$ and $d_2(i_1, i_2) \geq 0.995$ are satisfied.

4.2 Visibility

A *visible* watermarking scheme deliberately inserts a watermark such that it appears noticeably on the watermarked image to show some necessary information such as company logo, icon, or courtesy. However, in order that the watermark does not become so strongly pronounced that it takes over the main image, the level of visibility can be controlled, for example, by a parameter α. Visible watermarks are important in recognition and support of possessing a digital image. In contrast, an *invisible* watermark is embedded by keeping the perceptual content of the watermarked images similar to that of the original images to address security problems in different application scenarios. Therefore, there are schemes which are either *visible* or *invisible* based on the appearance of watermark on the watermarked images.

Definition 4.2 (Visibility)**.** *A watermarking scheme is called* visible *or* perceptible*, if* $E(\cdot)$ *embeds a given watermark,* w*, into an image,* i*, such that the* w *appears at least noticeably in* $\bar{i}$*. That is,* $|E_e(i,w) - i| = \alpha w$ *for all* i, w*. Here,* α *is weight factor that controls the degree of visibility.*

A watermarking scheme is called invisible *or* imperceptible*, if* $E(\cdot)$ *embeds* w *into* i *such that the* $\bar{i}$ *is perceptually similar to the original image,* i*. That is* $E_e(i,w) \approx i$ *for all* i, w*.*

Although the visibility and perceptual similarity properties share some perceptual aspects of a watermarked image, they need not to be confused with each other. As stated in Def. 4.1, the perceptual similarity property determines if an original image and its watermarked version remain "perceptually" the same. On the other hand, Def. 4.2 states that a visible watermark appears on a watermarked image with a predefined degree of visibility, α, and thus strictly speaking for the visible watermarking, the watermarked image is not perceptually similar to the original image. Perceptual similarity property is thus studied for the invisible watermarking schemes.

An invisible watermarking scheme usually differs from a visible watermarking scheme, not only in the visibility factor, but also in their embedding processes. Invisible embedding of a watermark aims at keeping the perceptual difference (resulting from the embedding distortion) at a "minimum" level such that the watermarked and original images remain perceptually the same. Their perceptual similarity is verified by quantifying the perceptual difference using similarity measures. The commonly used similarity measures do not indicate any subjective quality degradation, rather they quantify the overall perceptual difference either by their local (*e.g.*, block-wise or kernel-based) or global (*e.g.*, whole image based) operations. As a result, the defined perceptual similarity does not directly indicate whether a watermarking scheme is visible or invisible. However, for an invisible watermarking scheme, the quantified perceptual difference between an original image and its watermarked version would naturally be much lower than that for a visible watermarking scheme.

In short, an invisible scheme may be considered a variant of visible watermarking with a "negligible" (*i.e.*, approaching zero) α, and having an additional (and even more strict) perceptual similarity requirement. Visible watermarking is present in a few applications such as video broadcasting. However, recent research is mainly focussed on invisible watermarking with a high perceptual similarity in various image applications [25–41].

4.3 Blindness

Another important watermarking property is *blindness* that helps characterize a scheme to be *blind*, *non-blind*, or *semi-blind.* The term *blindness* (or *oblivious*) is generally used in cryptography to define a detection process independent of any side information. More specifically, blindness is used to define a computational property of information retrieval (*e.g.*, to define the computational independence on the original information or its derivatives to retrieve the required information). Similarly, blindness defines the detection and extraction process in digital watermarking, although there is no complete definition for a watermarking scheme to be *blind* or *non-blind.*

As a requirement for blindness, some schemes consider that no original input image and the information derived from the input image should be required, whereas other schemes consider only avoiding the original input requirement during the detection. Although schemes in both categories are often considered as blind, with a more strict blindness requirement, the schemes in the latter category may eventually fail to achieve the overall design requirements in an image application (*e.g.*, image authentication). Additionally, confusion arises when a scheme is defined as *semi-blind.* Sometimes, it is considered that if the detection and extraction processes can operate objectively without the original image and its derived information, but still require the original watermark, then the scheme can be *semi-blind.*

Cox *et al.* [42] informally defined a blind or oblivious watermark detector in such a way that the detector does not require access to the original (*i.e.*, unwatermarked) image, or some information derived from the original image. Otherwise, the detector is called non-blind or informed. However, their definition is not sufficient to realize three different cases associated with the blindness property. We define here (Def. 4.3) watermarking blindness to distinguish the dependency of $D(\cdot)$ and $X(\cdot)$ on any of the original input data that is used in $G(\cdot)$ and $E(\cdot)$, and thereby distinguish three different cases of this watermarking property.

Definition 4.3 (Blindness)**.** *A watermarking scheme is called* blind (*or* oblivious) *if both $D(\cdot)$ and $X(\cdot)$ are independent of the original image, i and watermark, w. Formally, for all images i_1, i_2 and watermarks w_1, w_2, hold both*

$$\begin{aligned} D_d(\bar{i}, i_1, w_1) &= D_d(\bar{i}, i_2, w_2) \\ \text{and} \quad X_x(\bar{i}, i_1, \tilde{w}) &= X_x(\bar{i}, i_2, \tilde{w}). \end{aligned}$$

A watermarking scheme is called semi-blind *if either one of $D(\cdot)$ and $X(\cdot)$ is independent of i and/or w. Thus, for* semi-blind *watermarking, for all images i_1, i_2 and watermarks w_1, w_2 either*

$$\begin{aligned} D_d(\bar{i}, i_1, w_1) &= D_d(\bar{i}, i_2, w_2) \\ \text{and} \quad X_x(\bar{i}, i, \tilde{w}) &\neq X_x(\bar{i}, i_1, \tilde{w}) \end{aligned}$$

or

$$\begin{aligned} D_d(\bar{i}, i, w) &\neq D_d(\bar{i}, i_1, w_1) \\ \text{and} \quad X_x(\bar{i}, i_1, \tilde{w}) &= X_x(\bar{i}, i_2, \tilde{w}). \end{aligned}$$

Otherwise a watermarking scheme is called non-blind (*or* non-oblivious *or* informed) *if both of $D(\cdot)$ and $X(\cdot)$ are dependent on i and/or w. Thus, for all images i, i_1 and watermarks w, w_1, hold both*

$$\begin{aligned} D_d(\bar{i}, i, w) &\neq D_d(\bar{i}, i_1, w_1) \\ \text{and} \quad X_x(\bar{i}, i, \tilde{w}) &\neq X_x(\bar{i}, i_1, \tilde{w}). \end{aligned}$$

We note here that strictly speaking the detection function, $D(\cdot)$ and the extraction function $X(\cdot)$ must have all three inputs: $\bar{i}$, i, and w. However, for instances of *blind* and *semi-blind* watermarking, some inputs (*e.g.*, i and w) are not used in $D(\cdot)$ and $X(\cdot)$, and thus they can be optionally omitted.

It can also be noted that the blindness property, as defined in Definition 4.3 in terms of the watermark detection and extraction functions, can also be considered for the watermark generation function. A non-blind (*i.e.*, an original image dependent) $G(\cdot)$ can be helpful in resisting *copy attacks* (that aims at counterfeiting the $D(\cdot)$ for any invalid modifications, or invalid watermarked images; see Sec. 5.1.6 for the definition of *copy attack*). The blindness for $D(\cdot)$ is also important, where availability of the original image, watermark or other side information at $D(\cdot)$ can thwart watermarking objectives. *Blind* and *non-blind* watermarking schemes are sometimes confused with *private* and *public* watermarking respectively. However, we insist on defining a watermarking scheme to be *private* and *public* in terms of their keys (as defined in Sec. 3.2) to avoid any confusion.

4.4 Invertibility

Invertibility (or *reversibility* or *losslessness*) is a computational property of watermarking. The meaning of this property is quite intuitive; however, we expect that defining invertibility in the current context would help realize its mutual relation with other properties. In an image application, invertibility is expected to restore any watermarked images to their original versions, where no embedding distortion is allowed in the original image. Such a watermarking criterion motivates construction of an invertible $E(\cdot)$ that helps $D(\cdot)$ to reproduce an original image from the watermarked image [30, 32, 34, 38, 39, 43–60]. Here, we define an *invertible* watermarking scheme such that it allows inverse computation of $E(\cdot)$ during detection.

Definition 4.4 (Invertibility)**.** *A watermarking scheme is* invertible (*or* reversible *or* lossless) *if the inverse of $E(\cdot)$ is computationally feasible to compute and is used in $D(\cdot)$ to estimate an exact original image, i, from the respective watermarked image, $\bar{i}$. Otherwise, the scheme is called* non-invertible *watermarking scheme.*

From the above definition, if $E_e(i, w) = \bar{i}$, then for an *invertible* watermarking scheme, E_e^{-1} the detection must exist and satisfy $E_e^{-1}(\bar{i}) = (i, w)$. Therefore, such watermarking schemes can be either *blind* or a *semi-blind* (according to Definition 4.3). Since, in image applications, an *invertible* watermarking scheme is mainly designed to reverse the effect of embedding on the original image, the embedding function is only considered to define invertibility of the scheme. However, the concept of an invertible function can also be extended for $X(\cdot)$, if an invertible $G(\cdot)$ is computationally feasible.

4.5 Robustness

Robustness in watermarking is often confused with its meaning from cryptography [61]. A main reason is probably that watermarking has to consider some spatial or perceptual properties (*e.g.*, perceptual similarity, visibility). Several attempts have been made to informally define the robustness property of watermarking. For example, Piper and Safavi-Naini [62] considered a watermarking scheme as robust if it can successfully detect the watermark in the "processed" images. The strength of this definition depends on how the "processed" image is defined. In contrast, Cox *et al.* [42] referred to robustness as the ability to detect the watermark after common signal processing techniques. More specifically, robustness can be defined as the degree of resistance of a watermarking scheme to modifications of the host signal due to either common signal processing techniques or operations devised specifically in order to render the watermark undetectable [63]. In summary, watermarking robustness has to deal with: (i) defining a set of processing techniques, and (ii) the detection ability for the "processed" images.

We now formalize the concept of watermarking robustness in terms of the processed images and the detection ability. Firstly, a set of processing techniques ($i.e.$, various operations/transforms) is defined below to define a "processed" image for an application. Here, the same set of processing techniques may not be valid for different watermarking applications, and thus a general consideration of the techniques may not be always useful. Secondly, a detection condition is defined that determines the detection ability, for the set of "processed" images.

Definition 4.5 (Processed Image)**.** *A* processed *image is an image that is not essentially perceptually similar to its original, but a certain amount of distortion, δ is incurred by a processing technique, $p \in P$. That is, if any image, $l \in \mathbb{I}$ is processed by p then, for the processed image, $p\,(l)$ the following is true: $p\,(l) = l + \delta$. Here, P is the set of applicable processing techniques for an application such that $P \subset \mathbb{P}$, where $\mathbb{P}$ is the space of processing techniques.*

It is worth noting that, in our earlier work [2, 61], we aimed at avoiding any confusion between the *robustness* and *security* properties, and considered that a processed image is not perceptually similar to its unprocessed version. That consideration was based on the assumption that only an adversary may want to process a valid watermarked image to achieve the perceptual similarity requirements. However, that assumption is not always valid in practice. For example, a watermarked image can be processed such as by lossless compression and file-format conversion, with the required perceptual similarity property (not only maliciously, but also intentionally as a system requirement). We, therefore, revise our earlier consideration for Def. 4.5 such that a processed image is not necessarily perceptually similar to its unprocessed version. We believe that this revision does not conflict with our earlier intention to avoid the confusion between robustness and security properties.

With the Def. 4.5, now we may wish to define the detection condition for the robustness property. Suppose a processing technique, $p \in P$, causes distortion to a watermarked image, $\bar{i}$. As defined in our proposed model, $D_d\,(\cdot)$ accepts with the property: $D_d\,(p\,(\bar{i})\,, i, w) = \left(\tilde{i}, \tilde{w}\right) \cup \perp$ for all $p\,(\bar{i})\,, i, w | p\,(\bar{i}) \in \bar{I}$. Here, the *pass* that returns with $\left(\tilde{i}, \tilde{w}\right)$ and the failure, $\perp$ can be used to define two potential variants, *robust* and *fragile* respectively, of watermarking schemes for different P. Another variant, *semi-fragile* watermarking scheme can also be defined considering a suitable subset of P. Thus, we define the robustness property in Def. 4.6 considering detection ability at three different levels.

Definition 4.6 (Robustness)**.** *A watermarking scheme is defined for the following levels of* robustness*:*

Robust. A watermarking scheme is called robust *if $D_d\,(p\,(\bar{i})\,, i, w) = \left(\tilde{i}, \tilde{w}\right)$ for all $p \in P$.*

Fragile. A watermarking scheme is called fragile *if $D_d\,(p\,(\bar{i})\,, i, w) = \perp$ for all $p \in P$.*

Semi-fragile. A watermarking scheme is called semi-fragile *if $D_d\,(p\,(\bar{i})\,, i, w) = \left(\tilde{i}, \tilde{w}\right)$ for all $p \in P_1$ and $D_d\,(p\,(\bar{i})\,, i, w) = \perp$ for all $p \in (P \backslash P_1)$, where $P_1 \subset P$.*

As stated in Def. 4.6, a successful detection ($i.e.$, $D_d\,(\cdot) \neq \perp$) is the basic criterion for a watermarking scheme to be *robust* to $p \in P$. However, there is no absolute robustness for watermarking, since taking all known/available processing techniques into consideration (for robustness) is not realistic. It is therefore reasonable to identify only the set of applicable processing techniques for the robustness requirements in an application (like knowing the set of potential adversaries for the security requirements in an application, see Sec. 4.8 below). As Def. 4.6 suggests, we also stress that one must have an explicit consideration on P for design and evaluation of a watermarking scheme in a particular application scenario.

When we consider P (the set of applicable processing techniques), we may notice that different processing techniques ($e.g.$, compression, de-noising) have different parameters ($e.g.$, compression ratio, down sampling rate, type and rank of filter). These parameter settings give different strengths to a processing technique. Therefore, it is worth noting that considering a technique, p, means that p is defined with its all required parameter settings. The technique with other settings thus remains outside of P.

4.6 Embedding Capacity

Embedding capacity (or simply *capacity*) is an important, and maybe the most-studied, property for watermarking schemes. A lot of studies have reported recently on improving this property maintaining the required perceptual similarity in different ways [30, 32, 38, 39, 50–59]. A number of ways to estimate the steganographic/watermarking embedding capacity by using information theoretic and perceptual model based methods, and detection theory are also present in the literature [64–70]. Capacity estimation is a fundamental problem of steganography [69], where the question is how much data can safely be hidden without being detected? However, in watermarking, the primary constraint for the capacity is its mutual dependence on a few others properties (*e.g.*, perceptual similarity, robustness) rather than the detection problem as in steganography. Therefore, we define watermarking capacity on the basis of perceptual similarity of $(i, \bar{i})$, for which the scheme works objectively (*e.g.*, without a failure).

Definition 4.7 (Embedding Capacity)**.** *Watermarking* embedding capacity *for an image, i is the maximum size of any watermark, $w = G_g(i, m, j)$ for all m and j, to be embedded in i, such that $E_e(i, w) \approx i$, $D_d(E_e(i, w), i, w) = \left(\tilde{i}, \tilde{w}\right)$, and there exists $\tilde{m}, \tilde{j} | \tilde{j} \approx j$ such that $X_x(E_e(i, w), i, \tilde{w}) = \left(\tilde{m}, \tilde{j}\right)$.*

Def. 4.7 suggests that to know the capacity of a watermarking scheme for an image, one needs to know how many bits can be embedded in the image with achieving the perceptual similarity and error probability (*e.g.*, successful detection) requirements. This capacity estimation method may vary with the type of watermarking schemes. Although several attempts have already been made [64–70] to know the capacity bound as mentioned above, developing a general method for capacity estimation of each type of watermarking schemes could still be interesting. This may also help solve other capacity related problems like the capacity control [50].

In image applications, embedding capacity is usually expressed as a ratio, bit-per-pixel (bpp). According to Def. 4.7, if the watermarking embedding capacity is n-bit, and the size of watermark is m-bit (*i.e.*, $w = \{1, 0\}^m$), then the necessary condition for an invisible watermarking scheme is: $m < n$. This condition suggests that there can be a hidden assumption of recursive embedding in developing an invisible scheme—if the required capacity is not achievable in first run of $E(\cdot)$, the remaining bits can be re-embedded recursively. That assumption may severely affect the performance of a watermarking scheme in practice, and thus needs to be explicitly stated, if applicable.

4.7 Error Probability

Error probability is an important property that helps determine the reliability of a watermarking scheme in practice. Some of the important and commonly used measures of error probability are: bit error rate (BER), false positive rate (FPR), false negative rate (FNR). However, this property is often disregarded in developing a watermarking scheme, assuming a reliable (operating) environment where communication errors are “negligible” and can be managed, for example, by using a suitable *error correction code*. This assumption is useful to simplify the application scenarios, but for some applications (*e.g.*, proof of ownership), this property needs to be studied explicitly. For example BER can be considered to evaluate the performance of the functions $D(\cdot)$ and $X(\cdot)$ in obtaining $\left(\tilde{i}, \tilde{w}\right)$ and $\left(\tilde{m}, \tilde{j}\right)$ respectively. (Here, BER follows its standard definition in communication system.) In our proposed model, we defined $D(\cdot)$ in such a way that the absence of a valid watermark, w in a watermarked image, $\bar{i}$ outputs a detection failure. Otherwise, $D(\cdot)$ returns $\left(\tilde{i}, \tilde{w}\right)$, which indicates that the input image is watermarked. Following this, we define the false positive and false negative for our model below.

Definition 4.8 (False Positive and False Negative)**.** *A watermarking detection in a normal condition is said to be a* false positive *if $D_d(i, w) \neq \perp$ for some i. Conversely, a watermarking detection is a* false negative *if $D_d(\bar{i}, i, w) = \perp$ for some $\bar{i}$. Here, the normal condition allows the scheme to run with all of its valid inputs, outputs, and functions.*

Irrespective of application scenarios, ideally, a zero FNR and FPR represents a reliable detection. Particularly, a watermarking scheme can be of no use if a scheme is unable to detect a valid watermark in normal condition of operation. Achieving a zero FNR and FPR in practice, however, may not be realistic for many reasons like communication errors. So, it is reasonable here to define a highly accurate detection for an application scenario in terms of a very low probability (*e.g.*, in the order of 10^{-6}) of detection failure.

However, error probability may be confused with other watermarking properties. Other properties (*e.g.*, security, robustness, perceptual similarity) may also deal with errors, which can be of different types; for example, bit-errors (often termed as *distortion*) in a valid watermarked/unwatermarked image, which can be incurred maliciously, unintentionally, or as a system requirement, may also cause a detection failure. Further, we note that the function $E(\cdot)$ itself utilize the error signal, *e.g.*, exploiting the redundant bitplanes of an image, for embedding. This embedding error can be considered as a system requirement and thus can be addressed in terms of perceptual similarity requirement. Specifically, while error probability measures can be used to determine the system error rate for the reliability of a watermarking scheme, the other perceptual errors (*i.e.*, distortion) can be studied in terms of the security, robustness and perceptual similarity properties.

4.8 Security

Security property of watermarking schemes as a whole may be far from easy to conceptualize (and may not be always necessary in practice) [71–73]. Two main possible reasons are: (i) application dependent properties, and (ii) the confusion between security and robustness requirements. In practice, different image applications may require different levels of security. Some applications do not need to be secure at all since there is no ultimate benefit in circumvention of watermarking objectives. For example, where a watermark is used only to add value in which they are embedded rather than to restrict uses for some device control applications [42]. Therefore, these types of watermarks do not need to be secure against any hostile attacks, although they still need to be robust against common processing techniques used in those applications. (This is how we defined the robustness property in Def. 4.6.)

Although the requirements for robustness and security properties of a watermarking scheme may overlap [61], they need to be considered separately. For security properties, in contrast to robustness, all possible attacks that an adversary may attempt within a particular scenario are to be studied. This may include different attacks; namely, elimination attack, collusion attack, masking attack, distortion attack, forgery attack, copy attack, ambiguity attack, scrambling attack, and/or other form of active and passive attacks in an application scenario. We will discuss and formally define these attacks in Sec. 5; however, to define a secure watermarking scheme formally, we denote an attack (*i.e.*, a set of adversary actions) in an application scenario by $\mathscr{A}$. Therefore, as discussed above, the possible choices for the attack $\mathscr{A}$ may vary with an application scenario.

Definition 4.9 (Security)**.** *A watermarking scheme is called* $\mathscr{A}$–secure *if the scheme retains the security against the attack* $\mathscr{A}$ (i.e.*, if it is "hard" to succeed with the set of adversary actions mounted by the attack* $\mathscr{A}$*).*

An application-specific analytical approach is often considered to study watermarking security [3, 16, 74–80]. In a broad sense, this practice suggests that the security property can be studied for two main types of watermarking schemes: *robust* and *fragile*. However, instead of focusing on a specific type of watermarking schemes, in this paper (Sec. 5), we are more interested in studying the general scenarios of a set of possible attacks in an abstract level for image applications. The main idea is to demonstrate how an adversary of different capabilities may win with different conditions. We call this a *win condition*. Knowing the inputs, outputs, and the win-conditions would eventually help visualize the possible attacks in an application. (With that visualization, conducting an application-specific security analysis can be easier and more efficient). Here, we consider that identifying the set of attacks in a specific application and defining them in the model are the first steps to defining the watermarking security.

5 Attacks on the Watermarking Security

In the watermarking context, an attack can be roughly defined as any malicious attempt to perform unauthorized embedding, removal, or detection of a (valid or invalid) watermark. An adversary that makes such attempts can be of different capabilities (*e.g.*, can have different inputs, and access to the watermarking functions). In practice, it is quite reasonable to assume the capabilities of expected adversaries in modelling attacks. For example, an adversary knowing nothing may assume an image is watermarked and may want to remove the watermark by applying a *distortion* attack (see Def. 5.4). Having access to the embedding function, an adversary can also find and exploit the weakness of the detection function in applying different active attacks including *elimination* and *masking* attacks, (see Def. 5.1 and Def. 5.3, respectively). Further, more difficult security problems arise if the adversary has both embedding and detection functions and knows how they work.

Attacks on the watermarking security can be mainly divided in two categories [42]: (i) *active* (*i.e.*, unauthorized embedding and unauthorized removal) and (ii) *passive* (*i.e.*, unauthorized detection). An *active* attack attempts to alter the watermarking resources or to affect their operation, whereas a *passive* attack, without doing that, attempts to know or exploit watermarking information. Some active attacks that circumvent the scheme directly are often referred to as *system* or *protocol* attacks. We define different attacks below using our model. Depending on which inputs are available to the adversary, however, there may be different flavours of the definitions. In what follows, the original (valid) watermark is defined as $w_0 \in W$ to distinguish it from other modified versions in an attack. Any other new notations will be defined accordingly.

5.1 Active Attacks

5.1.1 Elimination Attack

In an elimination attack, an adversary tries to output an image, which is perceptually similar to the watermarked image and not be detected as containing the watermark. Thus, the attacked watermarked image cannot be considered to contain a watermark at all. It is important to consider that eliminating the watermark does not necessarily mean reconstructing (or inverting) the watermarked image [42]. Rather, the adversary may output a new image that is perceptually similar to the watermarked image.

Definition 5.1 (Elimination Attack)**.**

Input. *Watermarked image,* $i = E_e\left(i, w_0\right)$*, where* $w_0 \in W$

Output. *Attacked image,* $i_a \in \tilde{I}$ *such that* $i_a \approx \bar{i}$

Win Condition. $D_d\left(i_a, i, w\right) = \perp$ *for all* w

Here, for a stronger adversary, the input can also include w_0 and the adversary can have access to $E_e\left(\cdot\right)$.

5.1.2 Collusion Attack

In a collusion attack, an adversary obtains several watermarked versions of an original image, each with a different or same watermark to obtain a close approximation of the watermarked image and thereby, produces a copy with no watermark.

Definition 5.2 (Collusion Attack)**.**

Input. n *copies* $($*where* $n \geq 2)$ *of watermarked image,* $\bar{i_j} = E_e\left(i, w_j\right)$*, where* $j = \{1, \cdots, n\}$

Output. $i_a \in \tilde{I}$ *such that* $i_a \approx \bar{i_j}$

Win Condition. $D_d\left(i_a, i, w\right) = \perp$ *for all* w

As in Def. 5.2, for example, an adversary has n copies (where $n \geq 2$) of watermarked image, $\bar{i_j} = E_e\left(i, w_j\right)$, where $j = \{1, \cdots, n\}$. In the form of an elimination attack, the adversary outputs $i_a \in \tilde{I}$ such that $i_a \approx \bar{i_j}$, and wins if for all w, $D_d\left(i_a, i, w\right) = \perp$.

5.1.3 Masking Attack

Masking of a watermark means that the attacked watermarked image can still have the watermark, which is, however, undetectable by existing detectors. More sophisticated detectors might be able to detect it.

Let an adversary have a watermarked image, $\bar{i} = E_e\left(i, w_0\right)$, where $w_0 \in W$. Here, the adversary aims to output $i_a \in \bar{I}$ such that $i_a \approx \bar{i}$. The adversary wins if $D_d\left(i_a, i, w_0\right) = \perp$ but there exists $w \neq w_0$ such that $D_d\left(i_a, i, w\right) \neq \perp$, as defined in Def. 5.3.

Definition 5.3 (Masking Attack)**.**

Input. *A watermarked image,* $\bar{i} = E_e\left(i, w_0\right)$*, where* $w_0 \in W$

Output. $i_a \in \bar{I}$ *such that* $i_a \approx \bar{i}$

Win Condition. $D_d\left(i_a, i, w_0\right) = \perp$*, but there exists* $w \neq w_0$ *such that* $D_d\left(i_a, i, w\right) \neq \perp$

5.1.4 Distortion Attack

In some masking attacks, an adversary applies some processing techniques uniformly over the watermarked image or some part of it, in order to degrade the watermark, so that the embedded watermark becomes undetectable or unreadable. This sub-class of masking attack has special merit in image processing and is referred to as distortion attack. De-noising attacks and synchronization attacks are two common attacks in this category.

Given a watermarked image, $\bar{i} = E_e\left(i, w_0\right)$, an adversary applies a processing technique, $q \in Q$ uniformly over the whole $\bar{i}$, or selected object/region of $\bar{i}$, and outputs $q\left(\bar{i}\right)$. According to Def. 5.4, the adversary wins if $D_d\left(q\left(\bar{i}\right), i, w_0\right) = \perp$ but there exists $w \neq w_0$ such that $D_d\left(q\left(\bar{i}\right), i, w\right) \neq \perp$. Q is the set of applicable processing techniques such that $Q \subset \mathbb{P}$.

Definition 5.4 (Distortion Attack)**.**

Input. *A watermarked image,* $\bar{i} = E_e\left(i, w_0\right)$*, and a processing technique,* $q\left(\cdot\right) \in Q$*. where* Q *is the set of applicable processing techniques such that* $Q \subset \mathbb{P}$

Output. *A processed image,* $q\left(\bar{i}\right)$

Win Condition. $D_d\left(q\left(\bar{i}\right), i, w_0\right) = \perp$ *but there exists* $w \neq w_0$ *such that* $D_d\left(q\left(\bar{i}\right), i, w\right) \neq \perp$

5.1.5 Forgery Attack

In a forgery attack, an adversary outputs an invalid watermarked image in the form of unauthorized embedding. An adversary with the ability to perform unauthorized embedding can be presumed able to cause the detector to falsely authenticate an invalid watermarked image.

Given access to $E_e\left(\cdot\right)$, an adversary chooses a new unwatermarked image, $i_a \in I$ and a new watermark, $w_a \in W$ to output the watermarked image, $\bar{i_a} \in \bar{I}$. As in Def. 5.5, the adversary wins with the output $\left(\bar{i_a}, i_a\right)$ if there exists $w_a \in W$ such that $D_d\left(\bar{i_a}, i_a, w_a\right) \neq \perp$, and also, possibly, there exists $\tilde{w_a} \in \tilde{W}$ such that $X_x\left(\bar{i_a}, i_a, \tilde{w_a}\right) \neq \perp$.

Definition 5.5 (Forgery Attack).

Input. *A new unwatermarked image, $i_a \in I$, a new watermark, $w_a \in W$, and the access to $E_e(\cdot)$*

Output. *A new watermarked image, $\bar{i_a}$*

Win Condition. *There exists $w_a \in W$ such that $D_d(\bar{i_a}, i_a, w_a) \neq \perp$*

This attack is accomplished in two parts. During the first part, the adversary has access to $E_e(\cdot)$. In the second part, the adversary has to output a forgery, which is different from all the outputs from $E_e(\cdot)$ in the first part. A stronger adversary may also have access to $G_g(\cdot)$ to obtain w_a (and possibly, choose m and j), and thus to output $\bar{i_a} = E_e(i_a, G_g(i_a, m, j))$ that makes the adversary more likely to win, specially over $X_x(\cdot)$.

5.1.6 Copy Attack

In a copy attack, an adversary outputs an invalid watermarked image as in a forgery attack. However, the adversary copies a watermark from one valid watermarked image into another to falsely authenticate an invalid watermarked image. In principle, an adversary initially tries to estimate the unwatermarked image from its watermarked version and then estimates the original watermark from the estimated unwatermarked image and the original watermarked image. Finally, the estimated watermark is embedded to a new unwatermarked image to get a forged watermarked copy.

Suppose an adversary is given a valid watermarked image, $\bar{i} = E_e(i, w_0)$ and the access to $E_e(\cdot)$. The adversary obtains the estimated original watermark, $\tilde{w_0}$, and chooses an unwatermarked image, i_a to output a new watermarked image, $\bar{i_a} = E_e(i_a, \tilde{w_0})$. Finally, as given in Def. 5.6, the adversary wins with output $(\bar{i_a}, i_a)$ if there exists $\tilde{w_0} \in W$ such that $D_d(\bar{i_a}, i_a, \tilde{w_0}) \neq \perp$. Also possibly, there exists $\tilde{\tilde{w_0}} \in \tilde{W}$ such that $X_x(\bar{i_a}, i_a, \tilde{\tilde{w_0}}) \neq \perp$, where $\tilde{\tilde{w_0}}$ is the estimate of $\tilde{w_0}$.

Definition 5.6 (Copy Attack).

Input. *A valid watermarked image, $\bar{i} = E_e(i, w_0)$, a new unwatermarked image, $i_a \in I$, and the access to $E_e(\cdot)$*

Output. *A new watermarked image, $\bar{i_a} = E_e(i_a, \tilde{w_0})$*

Win Condition. *There exists $\tilde{w_0} \in W$ such that $D_d(\bar{i_a}, i_a, \tilde{w_0}) \neq \perp$, where $\tilde{w_0}$ is the estimate of $\tilde{w_0}$*

An adversary can win with the copy attack if the original watermark, w_0 is independent of the image, i such that $w_0 = G_g(m, j)$. In addition, obtaining $\tilde{w_0}$ from $\tilde{i}$ and $\bar{i}$ can be easier for the adversary if the watermark embedding is simply additive. such that, $\tilde{w_0} \cong \left|\bar{i} - \tilde{i}\right|$. Thus, without having an access to $G_g(\cdot)$, the adversary can find $\tilde{w_0}$ and output a forged watermarked image, $\bar{i_a}$.

5.1.7 Ambiguity Attack

In a successful ambiguity attack, an adversary outputs a forgery, where a valid watermarked image is forged (*i.e.*, illegally watermarked) with a chosen watermark. The output forgery later can be verified as valid for the chosen (not for the originally embedded) watermark. Therefore, unlike a copy or forgery attack, it has a direct impact on the scheme.

Suppose a valid watermarked image, $\bar{i}$ and access to $E_e(\cdot)$ are given to an adversary. An ambiguity attack outputs a new watermarked image, $\bar{i_a} = E_e(\bar{i}, w_a)$ and the adversary wins if there exists $w_a \in W$ such that $D_d(\bar{i_a}, \bar{i}, w_a) \neq \perp$ (Def. 5.7). Also possibly, there exists $(\tilde{w_a}) \in \tilde{W}$ such that $X_x(\bar{i_a}, \bar{i}, \tilde{w_a}) \neq \perp$. Similar to forgery attack, a stronger adversary may have access to $G_g(\cdot)$ to obtain $w_a = G_g(i, m, j) \,|i = \bar{i}$.

Definition 5.7 (Ambiguity Attack).

Input. *Valid watermarked image, $\bar{i}$ and the access to $E_e(\cdot)$*

Output. *A new watermarked image, $\bar{i_a} = E_e(\bar{i}, w_a)$*

Win Condition. *There exists $w_a \in W$ such that $D_d(\bar{i_a}, \bar{i}, w_a) \neq \perp$*

5.1.8 Scrambling Attack

The objective of an adversary in applying a scrambling attack is similar to that of masking attack (*i.e.*, to falsify the detection of a valid watermarked image). However, in this attack, the samples of a watermarked image are scrambled prior to being presenting to the detector and subsequently descrambled. The type of scrambling can be a simple sample permutation or a more sophisticated pseudo-random scrambling [42]. A well-known scrambling attack is the mosaic attack, in which an image is broken into many small rectangular patches, each too small for reliable watermark detection. These image segments are then displayed in a table such that the segment edges are adjacent. The resulting table of small images is perceptually identical to the image prior to subdivision.

Definition 5.8 (Scrambling Attack).

Input. *A watermarked image,* $\bar{i} = E_e(i, w_0)$*, where* $w_0 \in W$*, and the access to 'suitable' scrambling and descrambling functions*

Output. *An image,* $\bar{i}_a \in \bar{I}$ *from scrambling the samples of* $\bar{i} \in \bar{I}$ *(before detection, and descrambles back to* $\bar{i} \in \bar{I}$ *after detection)*

Win Condition. $D_d(i_a, i, w_0) = \perp$ *but there exists* $w \neq w_0$ *such that* $D_d(i_a, i, w) \neq \perp$

Given input to an adversary includes a watermarked image, $\bar{i} = E_e(i, w_0)$, where $w_0 \in W$. The adversary outputs an image, $\bar{i}_a \in \bar{I}$ from scrambling the samples of $\bar{i} \in \bar{I}$ before detection, and descrambles back to $\bar{i}$ after detection such that $i_a \approx \bar{i}$. The adversary wins with a suitable scrambler and descrambler, if $D_d(i_a, i, w_0) = \perp$ but there exists $w \neq w_0$ such that $D_d(i_a, i, w) \neq \perp$, as in Def. 5.8.

5.2 Passive Attacks

Passive attacks can have different objectives such as detecting the presence of a valid watermark or knowing the associated information being carried by it. As mentioned in the beginning of this section, unlike active attacks, passive attacks do not attempt to alter the watermarking resources. However, a passive attack aims at knowing or exploiting the watermarking information and can have different level of consequences depending upon what it tries to achieve. We, therefore, define three different levels for the passive attacks considering their different objectives. We name these levels (to classify the passive attacks in each level) as *comprehensive detection* attack, *incisive detection* attack, and *detection only* attack.

In a comprehensive detection attack, an adversary wins by achieving all the three levels of target given in Def. 5.9. Similarly, to win an incisive detection attack, an adversary achieves the first two levels of target but fails to achieve target level 3. In the basic form of passive attack, a detection only attack, an adversary wins only with the target level 1.

Definition 5.9 (Passive Attacks)**.**

Level 1. (Detection only). *An adversary only detects the presence of valid watermark,* $w \in W$ *in a watermarked image,* $\bar{i} \in \bar{I}$*.*

Level 2. (Incisive detection). *An adversary distinguishes the watermark,* $w \in W$ *from that of other watermarked image(s),* $\bar{l} \in \bar{I} | \bar{l} \neq \bar{i}$*.*

Level 3. (Comprehensive detection). *An adversary obtains information at least partially (e.g., the message,* $m \in M$ *and other image data,* $j \in J$ *etc.) that the valid watermark,* $w \in W$ *carries, without modifying the watermarked image,* $\bar{i} \in \bar{I}$*.*

6 Conclusions

The study of digital watermarking is by no means new [81, 82]. Although it has received tremendous attention in different applications, formal concept in their systematic developments are yet to be established. Addressing the gap, in this paper, we have presented our work in three main parts: (*i*) a formal watermarking model (Sec. 3), (*ii*) definitions and uses of fundamental properties (Sec. 4), and (*iii*) possible attacks on the watermarking security (Sec. 5).

We have presented a formal generic watermarking model for digital image applications. Due to the high application variant properties of watermarking, we have focused on the image applications. We believe that our models can usefully be extended to other applications later. We determined a set of possible inputs, outputs, and component functions by studying the watermarking schemes proposed for different image applications. Thereby, we have initially constructed a basic watermarking model and later extended the model to a key-based model for completeness. Using the proposed model with suitable inputs, outputs, and functional properties, all possible variants of digital image watermarking schemes can be characterized and described (for example, to carry out the necessary computational analyses).

In addition, we have highlighted and defined a set of properties of watermarking with their practical interpretation in different image applications. Particularly, we defined the robustness and security properties of watermarking using the sets of (signal and image based) processing techniques and possible attacks, respectively. Although robustness can be interpreted as a security property, we believe our definition helps avoid any potential confusion between them in the signal and image processing contexts. Some other properties, such as computational complexity and cost, are important; however, in this paper, we have considered mainly those properties which can have varying interpretation with the application. Thus, addressing some hidden assumptions and associated confusions, we have presented the necessary corrections and clarifications with examples.

We have also defined a set of possible attacks with their win conditions using our model. Knowing the inputs, outputs, and win conditions helps one to visualize the necessary models of possible attacks, and thus helps conduct an application-specific security analysis more efficiently. Depending upon the application scenario and available data (*e.g.*, watermarked image, watermark) and tools (*e.g.*, embedding function), they can be defined for a stronger or weaker adversary. However, we mainly focused on a weaker adversary (as a notion of stronger security requirements) by classifying them into two categories: *active* and *passive*. Some active attacks, known as *system attacks*, aim at the protocols of the schemes. Two prominent system attacks, ambiguity and scrambling attacks, in addition to the common active attacks, are also defined. For passive attacks, we have defined three different levels (*i.e.*, detection only, incisive detection, and comprehensive detection attacks) to define the win conditions for an adversary. With all these attack definitions, we have shown how an adversary of different capabilities may win with different conditions.

As a final remark, we believe that the contributions presented in this paper are a first step towards a unified and intuitive theory for digital image watermarking. We also believe that the proposed model allows a unified treatment of all practically meaningful variants of digital image watermarking. Further, our considerations, definitions, and discussions on the fundamental defining properties and attacks can help to understand them while avoiding some potential confusions and taking a step forward towards the systematic development of watermarking schemes. We have supported our thesis with meaningful examples, necessary explanations, and comparative studies. The following, however, could be interesting topics for future research: (*i*) further development and a quantitative analysis of the proposed model; (*ii*) developing complete attack models (using the proposed model) and (*iii*) defining security levels (in terms of possible attacks), for different image (and other media such as audio and video) applications.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Electrical Eng. & Computer Science, Queensland University of Technology (QUT), Brisbane, Queensland, Australia. [2]Department of Telematics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

References

1. Li, Q., Memon, N., Sencar, H.T.: Security issues in watermarking applications - a deeper look. In: Proceedings of MCPS, pp. 23–28 (2007). ACM
2. Nyeem, H., Boles, W., Boyd, C.: Developing a digital image watermarking model. In: Proceedings of DICTA, pp. 468–473 (2011). IEEE
3. Nyeem, H., Boles, W., Boyd, C.: Counterfeiting attacks on block-wise dependent fragile watermarking schemes. In: Proceedings of the 6th International Conference on Security of Information and Networks, pp. 86–93 (2013). ACM
4. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding-a survey. Proceedings of the IEEE **87**, 1062–1078 (1999)
5. Jian, Z., Koch, E.: A generic digital watermarking model. Computers & Graphics **22**(4), 397–403 (1998)
6. Mittelholzer, T.: An information-theoretic approach to steganography and watermarking. In: Proceedings of Information Hiding, vol. 1768, pp. 1–16 (2000). Springer
7. Adelsbach, A., Katzenbeisser, S., Sadeghi, A.-R.: A computational model for watermark robustness. In: Proceedings of Information Hiding. LNCS, vol. 4437, pp. 145–160 (2007). Springer
8. O'Sullivan, J.A., Moulin, P., Ettinger, J.M.: Information theoretic analysis of steganography. In: Proceedings of Int. Symp. on Information Theory (1998). IEEE
9. Cohen, A.S., Lapidoth, A.: The gaussian watermarking game. IEEE Transactions on Information Theory **48**(6), 1639–1667 (2002)
10. Cachin, C.: An information-theoretic model for steganography. Information and Computation **192**(1), 41–56 (2004)
11. Cox, I.J., Miller, M.L., McKellips, A.L.: Watermarking as communications with side information. Proceedings of IEEE, 1127–1141 (1999)
12. Adelsbach, A., Katzenbeisser, S., Veith, H.: Watermarking schemes provably secure against copy and ambiguity attacks. In: Proceedings of Workshop on Digital Rights Management, pp. 111–119 (2003). ACM
13. Barni, M., Bartolini, F., Furon, T.: A general framework for robust watermarking security. Signal Processing **83**(10), 2069–2084 (2003)
14. Moulin, P., Mihcak, M.K., Lin, G.-I.: An information-theoretic model for image watermarking and data hiding. In: Proceedings of ICIP, vol. 3, pp. 667–670 (2000). IEEE
15. Moulin, P., O'Sullivan, J.A.: Information-theoretic analysis of information hiding. IEEE Transactions on Information Theory **49**(3), 563–593 (2003)
16. Holliman, M., Memon, N.: Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Transactions on Image Processing **9**, 432–441 (2000)
17. Nyeem, H., Boles, W., Boyd, C.: Utilizing least significant bit-planes of roni pixels for medical image watermarking. In: Proceedings of DICTA, pp. 1–8 (2013). IEEE
18. Barreto, P.S.L.M., Kim, H.Y., Rijmen, V.: Toward secure public-key blockwise fragile authentication watermarking. In: Proceedings of Vision, Image and Signal Processing, vol. 149, pp. 57–62 (2002). IEEE
19. Dai, H.K., Yeh, C.T.: Content-based image watermarking via public-key cryptosystems. In: Proceedings of ICCSA, vol. 1, pp. 937–950 (2007). Springer
20. Fridrich, J., Baldoza, A.C., Simard, R.J.: Robust digital watermarking based on key-dependent basis functions. In: Proceedings of IH, pp. 143–57 (1998). Springer
21. Wong, P.W., Memon, N.: Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Transactions on Image Processing **10**, 1593–1601 (2001)
22. Yu-Wen, D., Zi, L., Li, W.: A multipurpose public-key cryptosystem based image watermarking. In: Proceedigns of WiCOM, pp. 1–4 (2008). IEEE
23. Fridrich, J., Goljan, M.: Comparing robustness of watermarking techniques. In: Proceedings of SPIE, vol. 3657, pp. 214–225 (1999). SPIE
24. Tefas, A., Nikolaidis, N., Pitas, I.: Image watermarking: Techniques and applications. In: Al, B. (ed.) The Essential Guide to Image Processing (Second Edition), pp. 597–648. Academic Press, Boston (2009)
25. Chen, B., Wornell, G.W.: Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory **47**(4), 1423–1443 (2001)
26. Barni, M., Bartolini, F., Piva, A.: Improved wavelet-based watermarking through pixel-wise masking. IEEE Transactions on Image Processing **10**(5), 783–791 (2001)
27. Lin, C.-Y., Wu, M., Bloom, J.A., Cox, I.J., Miller, M.L., Lui, Y.M.: Rotation, scale, and translation resilient watermarking for images.

IEEE Transactions on Image Processing **10**(5), 767–782 (2001)

28. Bas, P., Chassery, J.-M., Macq, B.: Geometrically invariant watermarking using feature points. IEEE Transactions on Image Processing **11**(9), 1014–1028 (2002)
29. Luo, L., Chen, Z., Chen, M., Zeng, X., Xiong, Z.: Reversible image watermarking using interpolation technique. IEEE Transactions Information Forensics and Security **5**(1), 187–193 (2010)
30. Lin, C.-C., Tai, W.-L., Chang, C.-C.: Multilevel reversible data hiding based on histogram modification of difference images. Pattern Recognition **41**(12), 3582–3591 (2008)
31. Deng, C., Gao, X., Li, X., Tao, D.: A local tchebichef moments-based robust image watermarking. Signal Processing **89**(8), 1531–1539 (2009)
32. Guo, X., Zhuang, T.-G.: A region-based lossless watermarking scheme for enhancing security of medical data. Journal of Digital Imaging **22**(Compendex), 53–64 (2009)
33. Nikolaidis, N., Pitas, I.: Robust image watermarking in the spatial domain. Signal Processing **66**(3), 385–403 (1998)
34. Pan, W., Coatrieux, G., Cuppens, N., Cuppens, F., Roux, C.: An additive and lossless watermarking method based on invariant image approximation and haar wavelet transform. In: Proceedings of EMBC, pp. 4740–4743 (2010). IEEE
35. Qi, X., Qi, J.: A robust content-based digital image watermarking scheme. Signal Processing **87**(6), 1264–1280 (2007)
36. Rey, C., Dugelay, J.-L.: Blind detection of malicious alterations on still images using robust watermarks. In: Proceedings of the Seminar on Secure Images and Image Authentication, pp. 7–1 (2000). IET
37. Shih, F.Y., Wu, Y.-T.: Robust watermarking and compression for medical images based on genetic algorithms. Information Sciences **175**(3), 200–216 (2005)
38. Tsai, P., Hu, Y.-C., Yeh, H.-L.: Reversible image hiding scheme using predictive coding and histogram shifting. Signal Processing **89**(6), 1129–1143 (2009)
39. Lee, S., Yoo, C.D., Kalker, T.: Reversible image watermarking based on integer-to-integer wavelet transform. IEEE Transactions on Information Forensics and Security **2**(3), 321–330 (2007)
40. Luo, H., Yu, F.-X., Chen, H., Huang, Z.-L., Li, H., Wang, P.-H.: Reversible data hiding based on block median preservation. Information Sciences **181**(2), 308–328 (2011)
41. Nyeem, H., Boles, W., Boyd, C.: A review of medical image watermarking requirements for teleradiology. Journal of Digital Imaging **26**(2), 326–343 (2013). doi:10.1007/s10278-012-9527-x
42. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2nd ed. edn. Elsevier, Burlington : (2007)
43. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding-new paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing **2002**, 185–196 (2002)
44. De Vleeschouwer, C., Delaigle, J.-F., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Transactions on Multimedia **5**(1), 97–105 (2003)
45. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-lsb data embedding. IEEE Transactions on Image Processing **14**(2), 253–266 (2005)
46. Tian, J.: Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology **13**(8), 890–896 (2003)
47. Ni, Z., Shi, Y.-Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Transactions on Circuits and Systems for Video Technology **16**(3), 354–362 (2006)
48. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Transactions on Image Processing **13**(8), 1147–1156 (2004)
49. Kamstra, L., Heijmans, H.J.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing **14**(12), 2082–2090 (2005)
50. Caciula, I., Coltuc, D.: Capacity control of reversible watermarking by two-thresholds embedding. In: Proceedings of WIFS, pp. 223–227 (2012). IEEE
51. Kim, H.J., Sachnev, V., Shi, Y.Q., Nam, J., Choo, H.-G.: A novel difference expansion transform for reversible data embedding. IEEE Transactions on Information Forensics and Security **3**(3), 456–465 (2008)
52. Hu, Y., Lee, H.-K., Li, J.: DE-based reversible data hiding with improved overflow location map. IEEE Transactions on Circuits and Systems for Video Technology **19**(2), 250–260 (2009)
53. Kim, K.-S., Lee, M.-J., Lee, H.-Y., Lee, H.-K.: Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recognition **42**(11), 3083–3096 (2009)
54. Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.Q.: Reversible watermarking algorithm using sorting and prediction. IEEE Transactions on Circuits and Systems for Video Technology **19**, 989–999 (2009)
55. Thodi, D.M., Rodríguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE Transactions on Image Processing **16**(3), 721–730 (2007)
56. Li, X., Yang, B., Zeng, T.: Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Transactions on Image Processing **20**(12), 3524–3533 (2011)
57. Coltuc, D., Bolon, P., Chassery, J.M.: Fragile and robust watermarking by histogram specification. In: Proceedings of SPIE: Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 701–710 (2002)
58. Zhao, Z., Luo, H., Lu, Z.-M., Pan, J.-S.: Reversible data hiding based on multilevel histogram modification and sequential recovery. AEU-Int. Journal of Electronics and Communication **65**(10), 814–826 (2011)
59. Coatrieux, G., Montagner, J., Huang, H., Roux, C.: Mixed reversible and RONI watermarking for medical image reliability protection. In: Proceedings of EMBC, pp. 5653–5656 (2007). IEEE
60. Coatrieux, G., Le Guillou, C., Cauvin, J.M., Roux, C.: Reversible watermarking for knowledge digest embedding and reliability control in medical images. IEEE Transactions on Information Technology in Biomedicine **13**(2), 158–165 (2009)
61. Nyeem, H., Boles, W., Boyd, C.: On the robustness and security of digital image watermarking. In: Proceedings of ICIEV, pp. 1136–1141 (2012). IEEE
62. Piper, A., Safavi-Naini, R.: How to compare image watermarking algorithms. Transactions on Data Hiding and Multimedia Security **5510**, 1–28 (2009). Springer
63. Tefas, A., Nikolaidis, N., Pitas, I.: Image Watermarking: Techniques and Applications (Chapter 22), pp. 597–648. Academic Press, Boston (2009)
64. Zhang, F., Zhang, H.: Digital watermarking capacity analysis in wavelet domain. In: Proceedings of ICSP'04, vol. 3, pp. 2278–2281 (2004)
65. Yu, N., Cao, I., Fang, W., Li, X.: Practical analysis of watermarking capacity. In: Proceedings of the Int. Conf. on Communication Technology, vol. 2, pp. 1872–1877 (2003)
66. Wong, P.H.W., Au, O.C.: A capacity estimation technique for JPEG-to-JPEG image watermarking. IEEE Transactions on Circuits and Systems for Video Technology **13**(8), 746–752 (2003)
67. Barni, M., Bartolini, F., De Rosa, A., Piva, A.: Capacity of full frame DCT image watermarks. IEEE Transactions on Image Processing **9**(8), 1450–1455 (2000)
68. Moulin, P., Mihcak, M.K.: A framework for evaluating the data-hiding capacity of image sources. IEEE Transactions on Image Processing **11**(9), 1029–1042 (2002)
69. Harmsen, J.J., Pearlman, W.A.: Capacity of steganographic channels. IEEE Transactions on Information Theory **55**, 1775–1792 (2009)
70. Kalker, T., Willems, F.M.: Capacity bounds and constructions for reversible data-hiding. In: Proceedings of Int. Conf. on Digital Signal Processing, vol. 1, pp. 71–76 (2002). IEEE
71. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: theory and practice. IEEE Transactions on Signal Processing **53**(10), 3976–3987 (2005)
72. Kalker, T.: Considerations on watermarking security. In: Proceedings of Multimedia Signal Processing Workshop, pp. 201–206 (2001). IEEE
73. Voloshynovskiy, S., Pereira, S., Iquise, V., Pun, T.: Attack modelling: towards a second generation watermarking benchmark. Signal Processing **81**(6), 1177–1214 (2001)
74. Fridrich, J.: Security of fragile authentication watermarks with localization. In: Proceedings of SPIE- Security and Watermarking of

Multimedia Contents, vol. 4675, pp. 691–700 (2002). SPIE
75. Braci, S., Boyer, R., Delpha, C.: Security evaluation of informed watermarking schemes. In: Proceedings of ICIP, pp. 117–120 (2009). IEEE
76. Craver, S.A., Katzenbeisser, S.: Security analysis of public key watermarking schemes. In: Proceedings of Int. Symposium on Optical Science and Technology, pp. 172–182 (2001). SPIE
77. Wang, J., Liu, G., Lian, S.: Security analysis of content-based watermarking authentication framework. In: Proceedings of MINES, vol. 1, pp. 483–487 (2009). IEEE
78. Loukhaoukha, K., Chouinard, J.Y.: Security of ownership watermarking of digital images based on singular value decomposition. Journal of Electronic Imaging **19**(1) (2010)
79. Xiaomeng, C., Jie, S., Jianguo, Z., Huang, H.K.: Evaluation of security algorithms used for security processing on DICOM images. In: Proceedings of SPIE- Medical Imaging: PACS and Imaging Informatics, vol. 5748, pp. 548–56 (2005). SPIE
80. Li, Q., Memon, N.: Practical security of non-invertible watermarking schemes. In: Proceedings of ICIP, pp. 445–8 (2007). IEEE
81. Cox, I.J., Miller, M.L.: The first 50 years of electronic watermarking. EURASIP Journal on Applied Signal Processing **2002**(1), 126–132 (2002)
82. Hartung, F., Kutter, M.: Multimedia watermarking techniques. Proceedings of IEEE **87**, 1079–1107 (1999)