Multimed Tools Appl (2016) 75:15849–15880
DOI 10.1007/s11042-015-2893-x

Modelling attacks on self-authentication watermarking

Hussain Nyeem[1] · **Wageeh Boles**[2] · **Colin Boyd**[3,4]

Received: 15 September 2014 / Revised: 6 July 2015 / Accepted: 17 August 2015 /
Published online: 1 September 2015

Abstract Although the Self-Authentication Watermarking (SAW) schemes are promising to tackle the multimedia information assurance problem, their unknown security level seems to impair their potential. In this paper, we identify three new counterfeiting attacks on those schemes and present their countermeasure. We develop, analyse, and validate the models of the identified attacks followed by the development of a new SAW model to resist those attacks. The identified attack models generalize three main security levels that capture all the possible counterfeiting instances. We focus on the block-wise dependent fragile watermarking schemes, and their general weaknesses. Experimental results successfully demonstrate the practicality and consequences of the identified attacks in exploiting those weaknesses to maliciously and undetectably alter valid watermarked images. To resist the identified attacks, we further determine a set of general requirements for SAW schemes and illustrate their attainment in developing an extended SAW model. While the identified attack models can be used as a means to systematically examine the security levels of similar SAW

✉ Hussain Nyeem
h.nyeem@kuet.ac.bd

Wageeh Boles
w.boles@qut.edu.au

Colin Boyd
colin.boyd@item.ntnu.no

[1] Department of Electronics and Communication Engineering (ECE), Khulna University of Engineering and Technology (KUET), Khulna 9203, Bangladesh

[2] School of Electrical Engineering and Computer Science (EECS), Queensland University of Technology (QUT), Brisbane, QLD 4001, Australia

[3] Department of Telematics, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway

[4] School of EECS, QUT, Brisbane, QLD 4001, Australia

schemes, the extended SAW model may lead to developing their more secure variants. Our study has also revealed some open challenges in the development and formal analysis of SAW schemes.

Keywords Counterfeiting attack · Fragile watermarking · Information assurance · Multimedia security · Self-authentication

1 Introduction

Information assurance has attracted worldwide attention as a major and challenging multimedia security problem [3, 18, 22, 38, 39, 48]. Recently, we have witnessed a tremendous growth in multimedia (e.g., image, video, audio, etc.) applications, which is dramatically changing our life and continually impacting our research, business and economy. This growth, however, is also raising serious security concerns at the same time. An immediate threat, posed by the ready availability of sophisticated multimedia processing tools, is the diminishing trustworthiness of multimedia information. As a result, digital watermarking has been proposed as an enabling data-hiding technology leading to developing many self-authentication watermarking (SAW) schemes [4, 8, 14, 25, 27, 30, 41, 42, 45, 46].

The SAW schemes, as a general form of multimedia authentication tool, authenticate the semantic content of multimedia information such as images and videos using self-embedded watermark(s), with localisation and recovery of any possible alteration. There are different flavours in their construction (e.g., content authentication, self-embedding, self-recovery schemes) and application areas (e.g., image and video). In this paper, however, we mainly focus on the SAW schemes that are based on the block-wise dependent fragile watermarking principle and their applications to digital images. Although we develop and present the general model for these schemes in Section 3, the basic idea is that an input image is divided into non-overlapping blocks, and a watermark for each block is embedded into its mapped block. A mapping transform is used to generate the block-mapping sequence for a given set of block indexes. In the detection process, any possible alteration in an image is detected by comparing the embedded watermark(s) with the regenerated watermark(s). For a match, a detector authenticates the input image, otherwise it marks the image as tampered and attempts to localise and recover the tampered blocks.

Despite the continuing interest in developing new SAW schemes, disregard for their security analysis seems to impair their potential for the multimedia applications. One reason behind this disregard perhaps is the wrong consideration of active attacks [33, 35]. SAW schemes are usually based on the fragile watermarking, where active attacks that directly alter image contents are usually ignored. It is considered by the fragile watermarking property that the watermarks would be invalid for minimum possible changes in a watermarked image, and thus those attacks can be detected. This consideration, however, leaves an opportunity for the attackers to counterfeit a detector, by keeping the embedded watermarks valid for the alterations. We call those active attacks *counterfeiting attacks*. Consequences of those attacks, though can be unarguably severe, have not been properly realised yet. As a result, the security level of many SAW schemes remains undetermined.

The primary contributions presented in this article are three new counterfeiting attack models and an extended SAW model to facilitate the systematic development and formal security analysis of the SAW schemes. We start with reviewing the weaknesses and existing counterfeiting attacks (Section 2) and develop a general SAW model (Section 3). In developing the identified attack models, we then show how several adversary actions may apply and win in different levels of counterfeiting instances by exploiting the weaknesses of SAW schemes (Section 4). These attack models generalize all possible counterfeiting instances in three main security levels (Section 5). We present examples and experimental results to validate the practicality of the identified attacks and thereby to demonstrate how a SAW scheme may violate a systematic definition of security (Section 6). To avoid the weaknesses and counteract the identified attacks, we further outline a set of requirements and discuss some general guidelines for the SAW schemes. This finally leads us to developing an extended SAW model (Section 7). We also discuss a few open challenges in the development and formal analysis of SAW schemes based on the proposed SAW model (Section 8).

2 Review of counterfeiting weaknesses and attacks

In this section, we review weaknesses of the SAW schemes and relevant counterfeiting attacks. A SAW scheme may also have some performance issues like “inefficient” localisation and “poor quality” restoration. For example, tampering of one image block may lead to its mapped block to be marked as tampered (in addition to the real tampered block). Additionally, limited embedding capacity may often result in poor quality restoration of the tampered image blocks. However, in this paper, we are more interested in some of the schemes’ general weaknesses that can be exploited for counterfeiting attacks.

2.1 Counterfeiting weaknesses

The SAW schemes are promising for image content authentication and integrity verification with possible localisation and recovery of tampered image blocks. Without rigorous security analysis, their present development aims at improving localization-accuracy and restoration-quality. Consequently, they seem to have a number of weaknesses, which we now discuss.

Weak block-mapping transform SAW schemes usually employ a block-mapping sequence to obtain block-wise dependence property. This property helps avoid vector quantisation (VQ) weaknesses, which will be discussed below in Section 2.2. For the block-mapping sequence generation, although a non-linear transform is recently studied in [20]; a common approach is to use a linear transform such as using fixed offset [12], 1D-transformation [8, 24, 27, 46], or 2D-transformation [28]. The block-mapping weakness of a linear block-mapping transform mainly stems from choosing a key from a key-space of the range of block indexes. For example, for an image of size 512 × 512 and a block of size 4 × 4, the range of the block indexes is $[1, 16384]$ (i.e., $[1, \frac{512}{4} \times \frac{512}{4}]$). Such an “incongruously” small key space speeds up the process of block-mapping sequence recovery to only a fraction of a second (considering a typical key search time for the key-space using a *Brute-force attack* [36]).

Lack of collision resistance SAW schemes generally consider that any alteration in a valid watermarked image makes the embedded watermark invalid, as mentioned in Section 1. However, a detector can be deceived, if the embedded watermark remains valid for any possible alteration. Various local features (e.g., average intensity, transform or quantisation coefficients) of an image are used for the authentication watermark generation. Although these features facilitate the recovery process, they posses no or little "collision resistance".[1] Such SAW schemes therefore may no longer be reliable and in a more strict sense, seem to violate the systematic definition of security.

2.2 Counterfeiting attacks

Security aspects of the SAW schemes have attracted very limited attention in research. A few works [5, 15, 19, 21] have studied some counterfeiting possibilities for earlier schemes as follows.

The *vector quantisation attack* [21] (or *VQ attack*) and *collage attack* [15] are the initial counterfeiting attacks studied on some SAW schemes that embed watermarks into the host images in a block-wise independent fashion. Holliman and Memon [21] showed that there exists equivalence classes for each block containing a similar watermark for a given key, and thus the block is susceptible to the VQ attack. The *collage attack* is based on the same principle but assumes that an attacker has only a set of (large number) valid images watermarked with the same key and watermark. Thus an attacker replaces a set of valid image blocks with a set of collage blocks (i.e., a set of chosen blocks from the equivalence class) and wishes to validate those collage blocks for the key and original watermarks. Although the equivalence class principle is the key idea of the VQ and collage attacks, the principle is considered inapplicable, if the watermarks are block-wise dependent, and thus those attacks become invalid [5, 19]. (We will show, however, in Section 4.3 that the equivalence class principle can also be partially extended to the block-wise dependent watermarks).

Therefore, to avoid the VQ and collage attacks, SAW schemes are later designed to have the *block-wise dependence*; however, counterfeiting weaknesses in those schemes have also been reported. He et al. [19] showed the possibility of unauthorised recovery of the mapping sequence and secret key by using *verification device attack* and *exhaustive key search*, respectively. Subsequently, Chang et al. [5] proposed a *four-scanning attack* to obtain the secret mapping sequence. In a *verification device attack*, the attacker tampers with the embedded watermark of a block, and observes corresponding location of the mapped block as detected tampered. Consequently, the corresponding mapped image block is marked as tampered and the attacker comes to know which block it is mapped to, for a given block. The attacker thus continues verification for a set of input blocks of a valid watermarked image to recover their mapping sequence. In an *exhaustive key search*, an attacker tries all possible keys to find the correct mapping sequence, for which the regenerated watermark of each block will match with its original watermark embedded in the respective mapped block. In a *four-scanning attack*, like the *exhaustive key search*, an attacker applies an exhaustive search, but not for the secret key; rather the attacker aims to recover the mapping sequence.

[1] Collision resistance is mainly studied for cryptographic hash functions [36]. We can informally define here a watermark as *collision resistant* if for a given image block, it is "hard" to find another image block, which will have the same watermark. This is a notion of "weak" collision resistance, whereas for a watermark being "strong" collision resistant, it is "hard" to find two image blocks for a given watermark. However, consideration of these different notions of collision resistance may depend on the requirements of an application scenario (see Section 7).

In addition to those secret recovery attacks, some particular counterfeiting scenarios have also been studied. He et al. [19] illustrated a counterfeiting scenario called *synchronous attack*, where with the knowledge of the secret mapping sequence, an attacker modifies the chosen block(s) of a valid watermarked image keeping their original watermark(s) valid for the modification. Similarly, Chang et al. [5] presented a counterfeiting scenario called *constant-average attack*, which first modifies a block and then adjusts the pixels of the block such that their average intensity matches that of the original block.

In summary, the above studies [5, 19] mainly aimed at the unauthorised recovery of secret parameters (i.e., key or mapping sequence) exploiting the weaknesses of some SAW schemes. However, neither the weaknesses themselves nor the recovery of secret parameters demonstrate their practical consequences in an application. Although, those studies also illustrated particular counterfeiting instances, there can be many other counterfeiting possibilities for their studied schemes [11, 27, 47] and other similar schemes as well. Therefore, to generalise all possible counterfeiting instances at three levels of modification of a valid watermarked image, we studied three new counterfeiting attacks, developed their models, and defined their win conditions. We presented our early results for those attacks in [34].

In this paper, we extend our previous work [34] by developing a SAW model. We present three counterfeiting attacks valid for the SAW model and later extended the model to avoid the counterfeiting weaknesses. We also substantially revise our previous work with further technical details and clarification. Particularly, we demonstrate the counterfeiting consequences for the SAW schemes with new experimental results in different image applications. A set of requirements for the SAW schemes is determined, which was not studied and presented in the literature, as we will argue in Section 7. Our conjecture is, this gap in the literature is the main source of many protocol weaknesses and security problems of the SAW schemes including what we will discuss in this paper. Because, if the requirements are not properly considered for an application scenario, a scheme may be able to achieve the intended goals, but it can be still vulnerable to the attacks (as will be demonstrated in Section 6). Finally, considering those requirements, we extend the SAW model to resist the counterfeiting attacks.

3 Developing a general SAW model

In this section, we develop and describe a general model for the SAW schemes, which are based on the the block-wise dependent fragile watermarking principle (as mentioned in Section 1). The developed model, illustrated in Model 1, thus simplifies the realisation of a case of SAW schemes, which our identified attacks are valid for. For the model, we adopt necessary notations from the formal model of digital watermarking in [31, 32, 35].

We define a SAW scheme with three basic functions: watermark generation, $G(\cdot)$, watermark embedding, $E(\cdot)$, and watermark detection, $D(\cdot)$. The generation function generates the watermark: $w = G(i) = \{0, 1\}^{+}$. The embedding function embeds the watermark, w in an input image, i with a secret key, k, and thus outputs a watermarked image, $\bar{i}$ such that $E_k(i, w) = \bar{i}$. The detection function, on the other hand, verifies $\bar{i}$ with the detection key, k such that $D_k(\bar{i}) \neq \perp$, where '$\perp$' denotes a failure. $D(\cdot)$ also performs a few additional tasks (e.g., tampering localisation and recovery as showed in Step 3 and 7 in Model 1).

A SAW scheme operates on image blocks. The scheme divides a given input image, i into non-overlapping blocks, $\{B_n\}$ and sub-blocks, $\{B_n^l\}$ such that $i = \{B_n\} = \{B_n^l\}$.

Model 1 A General SAW Model

Watermark Generation, $G(\cdot)$

```
Input: an input image, $i = \{B_n^l\}$, where $n \in \{1, 2, \cdots, N_b\}$ and $N_b$ is
    the total no. of blocks.
Output: watermark, $w = \{w_n^l\}$.
Begin
    $\{w_n^l\} \leftarrow G : \{B_n^l\}$
End
```

Watermark Embedding, $E(\cdot)$

```
Input: (i) an input image, $i = \{B_n^l\}$; (ii) key, $k$; (iii) $Map(\cdot)$, and
    (iv) watermark, $w = \{w_n^l\}$.
Output: (i) watermarked image, $\bar{i} = \{\bar{B}_n^l\}$
Begin
    $\{w_q^l\} \leftarrow Map\left(\{w_n^l\}, k\right)$
    $\{\bar{B}_n^l\} \leftarrow E : \{B_n^l\} \times \{w_q^l\} = E\left(\{B_n^l\}, Map(\{w_n^l\}, k)\right)$
End
```

Watermark Detection, $D(\cdot)$

```
Input: (i) a watermarked image, $\bar{i} = \{\bar{B}_n^l\}$; (ii) key, $k$; (iii) $G(\cdot)$,
    (iv) $E^{-1}(\cdot)$; (v) $Verify(\cdot)$; and (vi) $Recover(\cdot)$.
Output: (i) a pass, $\top$ or the tampering localized and recovered
    images, $\bar{\bar{i}} = \{\bar{\bar{B}}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$ and $\tilde{i} = \{\tilde{B}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$,
    respectively, where $\{n_1\} \subseteq \{n\}$.
Begin
    $\{\tilde{w}_n^l\} \leftarrow E^{-1}\left(\{\bar{B}_n^l\}, Map(\cdot), k\right)$    ▷ watermark extraction
    $\{wnew_n^l\} \leftarrow G : \{\bar{B}_n^l\}$    ▷ watermark regeneration
    $\{\bar{\bar{B}}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\} \leftarrow Verify\left(\{\bar{B}_n^l\}, \{\tilde{w}_n^l\}, \{wnew_n^l\}\right)$    ▷ tampering detection
    if $\{n_1\} = \phi$ then
        return a pass, $\top$    ▷ the image is authentic and un-tampered
    else
        $\{\tilde{B}_{n_1}^l\} \leftarrow Recover\left(\{\bar{\bar{B}}_{n_1}^l\}, \{\tilde{w}_{n_1}^l\}\right)$    ▷ tampering recovery
        return $\bar{\bar{i}} = \{\bar{\bar{B}}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$ and $\tilde{i} = \{\tilde{B}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$.
    end if
End
```

Similarly, $w = \{w_n\} = \{w_n^l\}$, and $\bar{i} = \{\bar{B}_n\} = \{\bar{B}_n^l\}$. Here, l and n denote the indexes of the sub-blocks and blocks respectively. For example, $l \in \{1, \ldots, 4\}$ and $n \in \{1, 2, \ldots, N_b\}$ are the indexes of the 4×4 sub-block and 8×8 image block, where $N_b = \left(\frac{M}{8} \times \frac{N}{8}\right)$ and $M \times N$ is the image size. Note here that not all SAW schemes operate on the sub-blocks, where $l = 1$ and an image is thus simply a set of image blocks, i.e., $i = \{B_n\}$ or $\bar{i} = \{\bar{B}_n\}$.

In embedding, a mapping function, $Map(\cdot)$ is used to achieve the block-wise dependence. This function generates a mapping sequence and rearranges its input blocks according to the generated mapping sequence such that i.e., $\{w_q^l\} \leftarrow Map\left(\{w_n^l\}, k\right)$ or $\{B_q^l\} \leftarrow Map\left(\{B_n^l\}, k\right)$. For generating a mapping sequence, different mapping transform can be used in $Map(\cdot)$, as discussed in Section 2.1. However, for the considered case of SAW schemes, the mapping sequence, $\{q\}$ is generated for the block indexes, $\{n\}$, using the secret key, k, such that $q = [(k \times n)\, mod\, N_b] + 1$ for all n, where k is a prime number and usually chosen from the range of $[2, N_b]$. Finally, block-wise dependence is achieved by embedding the mapped blocks' watermarks into the input image blocks, i.e., $E : \left\{B_n^l\right\} \times \left\{w_q^l\right\} \rightarrow \left\{\bar{B}_n^l\right\}$.

To verify $\bar{i}$, $D(\cdot)$ regenerates watermarks and extracts their original version from $\bar{i}$ such that $G : \left\{\bar{B}_n^l\right\} \rightarrow \left\{wnew_n^l\right\}$, and $E^{-1}\left(\{\bar{B}_n^l\},\ Map(\cdot), k\right) \rightarrow \left\{\tilde{w}_n^l\right\}$, for all l and n. Here, $\left\{wnew_n^l\right\} = wnew$ and $\left\{\tilde{w}_n^l\right\} = \tilde{w}$ denote the regenerated and extracted versions of w. Ideally, $wnew = \tilde{w} = w$, but assuming a few possible bit errors in $\tilde{w}$ (usually addressed by some *error correction code*) and possible adversary actions leading to a different $wnew$, they are differently denoted. The extraction function, $E^{-1}(\cdot)$ is the inverse of E such that $E^{-1}(\cdot)$ extracts the bits (considering them as watermark bits) from the same embedding locations in $\bar{i}$. As shown in Model 1, $D(\cdot)$ then block-wise authenticates $\bar{i}$ using $Verify(\cdot)$ as follows: for all l and n,

$$Verify\left(\bar{B}_n^l, \tilde{w}_n^l, wnew_n^l\right) = \begin{cases} \bar{B}_n^l, & \text{for } \tilde{w}_n^l = wnew_n^l \\ \bar{\bar{B}}_n^l, & \text{otherwise} \end{cases} \tag{1}$$

Any tampered blocks $\left\{\bar{\bar{B}}_{n_1}^l\right\}$, where $\{n_1\} \subseteq \{n\}$, are then recovered using the recovery function, $Recover(\cdot)$ such that $\{\tilde{B}_{n_1}^l\} \leftarrow Recover\left(\{\bar{\bar{B}}_{n_1}^l\}, \{\tilde{w}_{n_1}^l\}\right)$. If no tampered blocks are found, $D(\cdot)$ returns a pass, $\top$ indicating the input image is authentic. Otherwise, the tampering localized and recovered images, $\bar{\bar{i}} = \{\bar{\bar{B}}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$ and $\tilde{i} = \{\tilde{B}_{n_1}^l\} \cap \{\bar{B}_{n-n_1}^l\}$ are output, respectively.

Therefore, in SAW schemes, $D(\cdot)$ performs verification in two phases: authentication, and tampering localization and recovery. For the security analysis, however, we only consider here the authentication phase, where an attacker is particularly interested to break in. For an authentic and un-tampered watermarked image, $\bar{i}$, thus there exists a match between $\left\{\tilde{w}_n^l\right\}$ and $\left\{wnew_n^l\right\}$ such that $D_k : \left\{\bar{B}_n^l\right\} \neq \perp$ or simply $D_k\left(\bar{i}\right) \neq \perp$. With satisfying this property of $D(\cdot)$, we will show in the following sections, how an attacker may modify $\bar{i}$ in different counterfeiting scenarios.

4 New counterfeiting attacks

In a general counterfeiting scenario, a valid watermarked image is maliciously manipulated to get undetectably verified. Consider an attacker outputs an attacked image, $\bar{i}_a$ (which is a maliciously modified version of a valid watermarked image, $\bar{i} = E_k\{i, w\}$) and wishes to verify $\bar{i}_a$ as authentic. We note here that, $\bar{i}_a$ and $\bar{i}$ may or may not be "perceptually similar" to each other depending on the intended use of $\bar{i}_a$. (Perceptual similarity is a watermarking property that defines the minimum distances or dissimilarities between the perceptual content of two images. For more precise definition, see ref. [31, 35].) Additionally, $\bar{i}_a$ may have either an original or new watermark, w or w_a, respectively. As a detector authenticates $\bar{i}$ with $D_k\left(\bar{i}\right) \neq \perp$, we can also define a general *win condition* (irrespective of the attacker's capability) to determine a successful counterfeiting attack.

Definition 1 (Win condition) An attacker outputs an attacked image, $\bar{i}_a$ for a SAW scheme, and wins with $D_k\left(\bar{i}_a\right) \neq \perp$.

An attacker's capability and intention, however, play an important role in counterfeiting attacks. Attackers of different capabilities (e.g., to choose input image(s) with/without watermark(s), to access to component functions or to know the secret parameters of a scheme) and intentions (e.g., what the attacked image is to be used for) may output an attacked image in different ways to satisfy the win condition. In practice, it is reasonable in attack modelling to assume the expected capabilities of an attacker. While a strong attacker may have access to all watermarking functions and can choose a watermark or a set of watermarked images, a weak attacker may only work on a single (or more) watermarked image(s) and with any disclosed secret information.

Depending upon the capability and intention, an attacker may recover the secret parameters (i.e., key or mapping sequence) using different methods: *exhaustive key search* [19], *verification device attack* [19], or a *four-scanning attack* [5] as discussed in Section 2.2. We propose here another effective approach, $Getmap\ (\cdot)$ for the mapping sequence recovery by combining the *four-scanning attack* with the *verification device attack*. In the $Getmap\ (\cdot)$, the exhaustive search principle of *four-scanning attack* is used to generate initial mapping sequence, and then the *verification device attack* principle is used to correct the sequence. The main difference between the $Getmap(\cdot)$ and those above mentioned methods is that $Getmap(\cdot)$ can operate on a set of watermarked images (watermarked with the same key), instead of only one watermarked image for mapping sequence recovery.

Moreover, the output obtained from $Getmap(\cdot)$ can be exploited in different ways to modify a (valid) watermarked image and to satisfy the win condition. To demonstrate this, we propose three new counterfeiting attacks; namely, *Counterfeiting Attack* 1, *Counterfeiting Attack* 2, and *Counterfeiting Attack* 3. We discus the $Getmap(\cdot)$ and the identified attacks, and develop their models below. (As discussed in [31, 32, 35], we use $X \approx Y$ to denote that two images X and Y are perceptually similar, and $X \not\approx Y$ to denote that they are not perceptually similar).

4.1 The getmap function

The block mapping sequence of a SAW scheme can be recovered without authorisation (i.e., without knowing the secret key). To this, $Getmap(\cdot)$ is developed, which outputs a complete (or partial) set of mapped block indexes using $G\ (\cdot)$ and the inverse of the modified embedding function, $Embed(\cdot)$. The watermark is generated using $G(\cdot)$ such that $G(\cdot)$: $\left\{\bar{B}_u^l\right\} \rightarrow \left\{wnew_u^l\right\}$, where $wnew$ is the regenerated version of the w. Here, $\{u\} \subseteq \{n\}$ is the set of indexes of the selected blocks $\left\{B_u^l\right\}$ that the attacker wants to modify. Further, $Embed^{-1}(\cdot)$ is used to extract the original watermarks, w embedded in the blocks $\left\{\bar{B}_u^l\right\}$ such that $Embed^{-1}$: $\left\{\bar{B}_u^l\right\} \rightarrow \left\{\tilde{w}_u^l\right\}$, where $\tilde{w}$ is the extracted version of the w. It is worth noting here that, unlike $E_k(\cdot)$, $Embed(\cdot)$ embeds the watermark(s) directly into the block(s) without the secret mapping sequence (and thus without the key) such that $Embed$: $\left\{B_u^l\right\} \times \left\{w_u^l\right\} \rightarrow \left\{\bar{B}_u^l\right\}$. So, the extracted watermark(s) using $Embed^{-1}(\cdot)$ remains in the order as they were embedded, which suggests that the match between two versions of block-wise watermarks, $\left\{wnew_u^l\right\}$ and $\left\{\tilde{w}_u^l\right\}$ may lead to the secret mapping sequence. $Getmap(\cdot)$ also attempts to correct any ambiguous pairs (i.e., an index pair is ambiguous to another pair, when they have a common mapped block index). This is illustrated in Model 2.

We show $Getmap(\cdot)$ here as one of a few possible options to recover the mapping sequence. As a distinctive feature, $Getmap(\cdot)$ allows an attacker to use a set of watermarked images to obtain an unambiguous mapping sequence. In Model 2, the basic $Getmap(\cdot)$ model is shown to operate on a single watermarked image, $\bar{i}$. However, note that it can also operate on a set of V–valid watermarked images, $\left\{\bar{i}_v : \bar{i}_v = E_k\left(i_v, w_v\right)\right\}$ for all $v \in \{1, \cdots, V\}$. In other words, when an attacker has a set of images watermarked using the same embedding key, $Getmap(\cdot)$ can operate on the images $\left\{\bar{i}_v\right\}$ to get the mapped indexes of the selected image blocks more efficiently. Once the key or mapping sequence is known, there is an open opportunity for an attacker to output successful attacked images with not only malicious, but also “meaningful” modifications. (A “meaningful” modification roughly means that the modification has visual semantic in the application context).

4.2 Counterfeiting attack 1

An attacker, without any specific use of the attacked image in mind, may wish to modify a watermarked image simply with manipulating the pixel locations. The image pixels can be rearranged such that original watermarks remain valid for the new orientation of original pixels. An attacker modifies neither any pixels nor their watermarks to output such an attacked image. However, the attacked image may be perceptually different from the input (valid watermarked) image for the new orientation of original pixels. The adversary actions in this scenario form our *Counterfeiting Attack* 1. Two cases can be studied here.

Entire block swap In this case, an attacker is interested in all image blocks and swaps all of them with their mapped blocks. Thus, the pixels of an image (or image blocks) remain the same as the watermarked image but with different orientation (i.e., their original block

Model 2 *Getmap* $(\cdot)$

```
Input: (i) watermarked image, $\bar{i} = \{\bar{B}_n^l\}$ with watermark,
    $w = \{w_n^l\}$; (ii) set of selected block indexes,
    $\{u\} \subseteq \{n\}$; (iii) generation function, $G(\cdot)$; and
    (iv) extraction function, $Embed^{-1}(\cdot)$.
Output: set of mapped blocks' indexes, $\{uu\} \subseteq \{q\}$.
Begin
    $\{wnew_u^l\} \leftarrow G : \{B_u^l\}$            ▷ watermark regeneration
    $\{\tilde{w}_u^l\} \leftarrow Embed^{-1} : \{\bar{B}_u^l\}$        ▷ watermark extraction
    for all $index \in \{u\}$ do   ▷ computing mapping sequence
        if $\tilde{w}_{index}^l = wnew_u^l$ then
            $uu \leftarrow index$
        end if
    end for
    for all $index \in \{u\}$ do        ▷ correcting ambiguous pair
        if there exists $\{ambiguous\_index\} \subset \{uu\}$ then
            for all $ambiguous\_index$ do
                if $wnew_{ambiguous\_index}^l = \tilde{w}_{another\_index}^l$
    where, $another\_index \in \{n\} : another\_index \neq index$
    then
                    $\{uu\} \leftarrow \{uu\} \backslash ambiguous\_index$
                end if
            end for
        end if
    end for
    return set of mapped block indexes, $\{uu\}$
End
```

indexes are now their mapped indexes, and *vice versa*). We call this swap of all image blocks as *entire block swap*.

Selected block swap In this case, an attacker is interested in a particular set of image block(s) rather than all the blocks in an image. Here, an attacker chooses a set of blocks to swap, which requires correction of the orientation of swapped blocks' watermarks to remain valid [34]. We call this *selected block swap*.

Model 3 Counterfeiting Attack 1

```
Input: (i) watermarked image, $\bar{i} = \{\bar{B}^l_n\}$; (ii) embedding
    function, $Embed(\cdot)$; (iii) extraction function,
    $Embed^{-1}(\cdot)$; (iv) $\{u\}$—set of indexes of the selected
    blocks, $\{B^l_u\}$ ; (v) $\{uu\}$—set of the mapped block
    indexes of $\{u\}$; (vi) $\{uux\}$—set of the mapped block
    indexes of $\{uu\}$.
Output: attacked image, $\bar{i}_a \not\approx \bar{i}$.
Begin
1: $\{\tilde{w}^l_n\} \leftarrow Embed^{-1} : \{\bar{B}^l_n\}$          ▷ watermark extraction
2: if $\{u\} \cap \{n\} \neq \phi$ then                    ▷ selected block swap
3:     $\{\bar{B}^l_{ua}\} \leftarrow Embed : \{\bar{B}^l_u\} \times \{\tilde{w}^l_{uu}\}$          ▷ watermark correction
       $\{\bar{B}^l_{uua}\} \leftarrow Embed : \{\bar{B}^l_{uu}\} \times \{\tilde{w}^l_{uux}\}$
       $\{\bar{B}^l_{uuxa}\} \leftarrow Embed : \{\bar{B}^l_{uux}\} \times \{\tilde{w}^l_u\}$
4:     swap the blocks $\{\bar{B}^l_{ua}\}$ with their mapped blocks
       $\{\bar{B}^l_{uua}\}$
5:     return the attacked image,
       $\bar{i}_a \leftarrow \left( \{\bar{B}^l_n\} \setminus \left( \{\bar{B}^l_u\} \cup \{\bar{B}^l_{uu}\} \cup \{\bar{B}^l_{uux}\} \right) \right) \cup \{\bar{B}^l_{ua}\} \cup \{\bar{B}^l_{uua}\} \cup \{\bar{B}^l_{uuxa}\}$
6: else                                     ▷ entire block swap
7:     swap the blocks $\{\bar{B}^l_u\}$ with their mapped blocks,
       $\{\bar{B}^l_{uu}\}$
8:     return the attacked image, $\bar{i}_a \leftarrow \{\bar{B}^l_u\}|_{u=n}$
9: end if
End
```

Model 3 illustrates the general steps of the *Counterfeiting Attack* 1. For an entire block swap, the attacker simply interchanges all the blocks, $\{\bar{B}^l_u\}|_{u=n}$ with their respective mapped blocks, $\{\bar{B}^l_{uu}\}|_{uu=q}$ (see Step 7 and 8 of the model). On the other hand, Steps 3–5 describe a selected block swap. Here, as mentioned above, watermarks embedded in $\{\bar{B}^l_u\}$ —the selected blocks, $\{\bar{B}^l_{uu}\}$—the mapped blocks of $\{\bar{B}^l_u\}$, and $\{\bar{B}^l_{uux}\}$ —the mapped blocks of $\{\bar{B}^l_{uu}\}$ need watermark correction along with the interchange between $\{\bar{B}^l_u\}$ and $\{\bar{B}^l_{uu}\}$. Finally, in both cases, an attacker outputs an attacked image, $\bar{i}_a$. In this attack scenario, an attacker can also shuffle all the pixels in a selected block to introduce a more "meaningful" modification with $\bar{i}_a$, keeping respective watermarks' locations unchanged. Considering the inputs of the model 3, the *Counterfeiting Attack* 1 represents a "weak" counterfeiting attack. Here, the attacker's capability may only include a set of watermarked images and access to the embedding function.

4.3 Counterfeiting attack 2

In a more sophisticated counterfeiting scenario, an attacker may wish to modify some (or all) watermarked image blocks for a more meaningful outcome. Here, a set of selected blocks may either be modified directly or be replaced with another set of chosen blocks. Unlike *Counterfeiting Attack* 1, where no pixels and watermarks were modified (but their locations), in this counterfeiting scenario, the original watermarks remain unchanged and valid for the replaced blocks. This is defined as our *Counterfeiting Attack* 2 and illustrated in Model 4.

An attacker first outputs a set of blocks perceptually similar to the set of chosen blocks. These output blocks must have the same watermark as the selected (original) blocks to remain valid. The output blocks then replace the selected blocks in the watermarked image. We define the construction of the perceptually similar blocks as a function $Sim(\cdot)$, which outputs a set of blocks, $\left\{A_u^l\right\}$ for the set of chosen blocks, $\left\{C_u^l\right\}$ such that $Sim : \left\{C_u^l\right\} \times \left\{\bar{B}_u^l\right\} \rightarrow \left\{A_u^l\right\} \cup \{\perp\}$, where $\left\{A_u^l\right\} \approx \left\{C_u^l\right\} \not\approx \left\{\bar{B}_u^l\right\}$, and $\left\{w_u^l\right\} \leftarrow G : \left\{A_u^l\right\}$. Here, $\perp$ is a *failure* and $\left\{\bar{B}_u^l\right\}$ are the selected blocks to be replaced with the blocks $\left\{\bar{A}_u^l\right\}$. As shown in Model 4, once $Sim(\cdot)$ outputs $\left\{A_u^l\right\}$, an attacker extracts the watermarks embedded in the selected blocks, and embed that extracted watermarks in $\left\{A_u^l\right\}$. Finally, the watermarked blocks, $\left\{\bar{A}_u^l\right\}$ replace the selected blocks, $\left\{\bar{B}_u^l\right\}$ to output an attacked image, $\bar{i}_a$ as shown in the attack model.

A successful *Counterfeiting Attack* 2, therefore, mainly depends on the success of the function $Sim(\cdot)$. With the output of $Sim(\cdot)$, the attacker may output an attacked image that satisfies the win condition. With this additional requirement of $Sim(\cdot)$, this attack presents a "stronger" notion of counterfeiting attack than the *Counterfeiting Attack* 1. A simple $Sim(\cdot)$, for example, can replace the pixels in a selected input block with their average intensity or pixel value (leaving their LSBs— least significant bits intact that carry watermark bits). For the output (modified) blocks, the watermarks remain valid as the modification in the output blocks does not affect the watermark. We have used this simple construction of $Sim(\cdot)$ for *Counterfeiting Attack* 2 implementation and will discuss it in Section 6. We note here that the principle of keeping average intensity of a block unchanged for a modified block is the main idea of a *constant-average attack* [5].

However, the *constant-average attack* differs from the above example of $Sim(\cdot)$ construction, where all pixels of a modified block will have the average intensity value of the block. In *constant-average attack*, the pixels of a modified block are usually different but their average intensity remains the same as that of the original block, as mentioned in Section 2.2. In other words, an attacker attempts to adjust the pixels of an already modified block further so that their average intensity equals that of the original block. Thus, the *constant-average attack* representing a case of our *Counterfeiting Attack* 2, employs a $Sim(\cdot)$ different from the one we used in this paper.

Moreover, $Sim(\cdot)$ generally extends the equivalence class principle (of the VQ attack [21]) for the block-wise dependent watermarking schemes, as pointed out in Section 2.2. Once the $Getmap(\cdot)$ (or any other secret recovery method) outputs the mapping sequence (or key), the block-wise dependence property is actually lost. Consequently, $Sim(\cdot)$ outputs a block from an equivalence class, which will give the same watermark as the original block and valid for the secret key (used for the original watermarked image). However, unlike the VQ equivalence principle, $Sim(\cdot)$ has an additional requirement of perceptual similarity and thus has to output a block perceptually similar to the chosen input blocks. It is worth noting here that, with a very strict perceptual similarity requirement, $Sim(\cdot)$ may not work effectively, and may output a failure.

Model 4 Counterfeiting Attack 2

```
Input: (i) watermarked image, $\bar{i} = \{\bar{B}_n^l\}$; (ii) $Sim(\cdot)$
    (iii) embedding function, $Embed(\cdot)$; (iv) extraction
    function, $Embed^{-1}(\cdot)$; (v) $\{u\}$—set of indexes of the
    selected blocks, $\{B_u^l\}$ ; (vi) $\{uu\}$—set of the mapped
    block indexes of $\{u\}$; and (vii) chosen blocks $\{C_u^l\}$.
Output: attacked image, $\bar{i}_a \not\approx \bar{i}$.
Begin
  $\{A_u^l\} \cup \{\perp\} \leftarrow Sim : \{C_u^l\} \times \{\bar{B}_u^l\}$    ▷ new blocks' computing
  if $Sim(\cdot) \neq \perp$ then
      $\{\tilde{w}_{uu}^l\} \leftarrow Embed^{-1} : \{\bar{B}_{uu}^l\}$ ▷ watermark extraction
      $\{\bar{A}_u^l\} \leftarrow Embed : \{A_u^l\} \times \{\tilde{w}_{uu}^l\}$    ▷ watermark re-embedding
      replace $\{\bar{B}_u^l\}$ with $\{\bar{A}_u^l\}$
      return the attacked image,
      $\bar{i}_a = \left(\{\bar{B}_n^l\} \setminus \{\bar{B}_u^l\}\right) \cup \{\bar{A}_u^l\}$
  else
      return a failure, $\perp$
  end if
End
```

4.4 Counterfeiting attack 3

As a notion of a more stronger attacker, we illustrate another counterfeiting scenario that introduces the highest level of modification into a watermarked image. Unlike the other counterfeiting scenarios discussed above, here an attacker can choose new blocks and generate their watermarks to output an attacked image. This means that this attacker's capability include the access to the watermark generation and embedding functions. We call this counterfeiting scenario *Counterfeiting Attack* 3. The severity of this attack is that an attacker with the access to all watermarking functions can make a more meaningful modifications than the above counterfeiting attacks.

The general steps of the *Counterfeiting Attack* 3 model are shown in the Model 5. An attacker starts with choosing a set of new blocks, $\{C_u^l\}$ and extracting the embedded watermarks, $\{w_u^l\}$ from the selected blocks, $\{B_u^l\}$. Having access to the watermark generation and embedding functions, an attacker may embed the extracted watermark in the chosen blocks. The chosen blocks' watermarks, $\{w_{ua}^l\}$ are also generated and required to be embedded in the selected blocks' mapped blocks, $\left\{B_{uu}^{l}\right\}$. Finally, the chosen blocks replace the selected blocks to output an attacked image.

5 Practicality of the identified attacks

We have developed and presented the counterfeiting attack models in last section. To demonstrate their practicality, we now discuss how the identified attacks can be mounted on the SAW schemes. Although the attack models theoretically apply to the schemes that follow the general SAW model presented in Section 3, two typical SAW schemes [8, 46] are studied here that capture the medical and other image applications. The Zain and Fauzi scheme (or ZF scheme) [46] is a variant of the prominent Lin et al. scheme [27], and later applied in a potential medical imaging environment [26]. The Edupuganti, Shih, and Chang scheme (or ESC scheme) [8] is recently proposed for tampering localisation and recovery of digital images. Below, we briefly review those schemes and discuss the implementation of the identified attacks.

Model 5 Counterfeiting Attack 3

```
Input: (i) watermarked image, $\bar{i} = \{\bar{B}_n^l\}$; (ii) generation
    function, $G(\cdot)$ (iii) embedding function, $Embed(\cdot)$;
    (iv) $\{u\}$—set of indexes of the selected blocks, $\left\{B_u^l\right\}$;
    (v) $\{uu\}$—set of the mapped block indexes of $\{u\}$; and
    (vi) chosen blocks $\left\{C_u^l\right\}$.
Output: attacked image, $\bar{i}_a \not\approx \bar{i}$.
Begin
  $\left\{wnew_{uu}^l\right\} \leftarrow G : \left\{\bar{B}_{uu}^l\right\}$    ▷ watermark regeneration
  $\left\{w_{au}^l\right\} \leftarrow G : \left\{C_u^l\right\}$    ▷ new watermark generation
  $\left\{\bar{C}_u^l\right\} \leftarrow Embed : \left\{C_u^l\right\} \times \left\{wnew_{uu}^l\right\}$    ▷ watermark correction
  $\left\{\bar{B}_{uua}^l\right\} \leftarrow Embed : \left\{\bar{B}_{uu}^l\right\} \times \left\{w_{au}^l\right\}$
  replace $\left\{\bar{B}_u^l\right\}$ with $\left\{\bar{C}_u^l\right\}$
  return the attacked image, $\bar{i}_a = \left(\left\{\bar{B}_n^l\right\} \setminus \left\{\bar{B}_u^l\right\}\right) \cup \left\{\bar{C}_u^l\right\}$
End
```

5.1 The ZF and ESC schemes

The ZF Scheme [46] operates on 8×8 non-overlapping blocks and their 4×4 sub-blocks of an image of size $M \times N$. In order to get the mapping sequence for the image block indexes, an 1D linear transformation is used. This transform uses a secret key, which is a prime number chosen from the range of 1 to the total number of the blocks, which limits the key-space to $\left[2, \left(\frac{M}{8} \times \frac{N}{8}\right)\right]$. ZF scheme avoids the VQ weaknesses and has good localisation ability. For higher recovery rate of tampered pixels and their better restoration quality, this scheme considers average intensity of individual sub-blocks as their recovery watermarks. However, in addition to the common weakness of the small key-space, ZF scheme uses the watermarks generated from local image properties, which have not been justified for image authentication and integrity verification.

On the other hand, the ESC scheme [8] operates on 2 × 2 non-overlapping blocks of an image of size $M \times M$, where M is a multiple of 2. A lookup table is generated containing the mapped indexes of the image block from the set of block indexes, $\{1, \cdots, N\}$ by using a secret key, where $N = \left\{\frac{M}{2} \times \frac{M}{2}\right\}$. The secret key is chosen as a prime number from the range of the block indexes, $[2, N - 1]$. Similar to ZF scheme, a liner transform is used in the ESC scheme to obtain an initial mapping sequence. But, this mapping sequence is modified in ESC scheme using a “block-shift” operation to construct the final lookup table. The dual watermarking principle, 5-bit image block feature, and use of CRC-2 and lookup table make the ESC scheme attractive. However, the ESC scheme suffers from various weaknesses that may cause security problems in a target application. Like ZF scheme, this scheme has a small key-space (of $[2, N - 1]$). Further, use of feature bits, lookup table and CRC-2 is not justified for any expected security problems.

5.2 Implementation of the identified attacks

Our identified counterfeiting attacks are accomplished in two parts: *secret recovery* and *forgery*. In the first part, an attacker tries to recover the secret parameters (e.g., key, mapping sequence). The general steps of this part are already shown in $Getmap(\cdot)$ model (Model 2) and discussed in Section 4.1. We note that the computation of this part may vary depending on the design of the target SAW scheme.

We implement the $Getmap(\cdot)$ to demonstrate the relative computation time for an attacker to obtain the mapping sequence of both the ZF and ESC schemes. However, in order to implement our attacks on ZF and ESC schemes, we assume that the attacker has the secret keys. Since the key space of both schemes is too small, it is not difficult to obtain the key at all, even for an attacker having limited computational power. For example, for a typical image of size 512 × 512, the maximum key size of the ZF and ESC schemes are 13-bit and 15-bit respectively. Theoretically, compared with cryptographic keys, these key lengths do not provide any protection [16].

In the second part, an attacker has to output a forgery using the secret key or mapping sequence obtained in the first part. The output is valid for the embedded watermark (to satisfy the win condition), and is different from any previous outputs of the SAW scheme. In other words, an attacker outputs a new watermarked image (with new pixels or watermarks, or both), which remains valid for a given key. Here, an attacker of different capabilities (discussed in the beginning of Section 4) may output forgeries in different levels: *change of pixel locations only*, *change of original pixels only*, and *change of original pixels and watermarks*, as shown in Table 1. Attacker’s capabilities are generally classified here to indicate their relative notions of strength. We implement the identified attacks that individually represent different levels of counterfeiting scenarios (see Table 1).

Therefore, the identified attacks address the counterfeiting scenarios at three levels of modifications, and we argue that any counterfeiting scenarios (i.e., any possible ways of modifying a valid watermarked image) can be described from one of these three levels. In other words, our identified counterfeiting attacks capture all possible counterfeiting scenarios at the three levels. In fact, an attacker may have different ways to modify a watermarked image at a particular counterfeiting level. However, we implement a few of them to demonstrate the practicality and consequences of modifying a valid watermarked image at each counterfeiting level. All necessary simulation and implementation were carried out using MATLAB (R2012a-7.14.0.739) and an Intel Core i5 3.2GHz CPU.

Table 1 Counterfeiting attack levels

Levels	Counterfeiting attack scenarios	Objectives	Attacker's capabilities
1	Change of pixel locations only (*Counterfeiting Attack* 1)	To output a successful attacked image with original image pixels and watermarks, but their locations are changed	Low
2	Change of original pixels only (*Counterfeiting Attack* 2)	To output a successful attacked image with new pixels keeping the original watermarks	Medium
3	Change of original pixels and watermarks (*Counterfeiting Attack* 3)	To output a successful attacked image with both new pixels and their watermarks	High

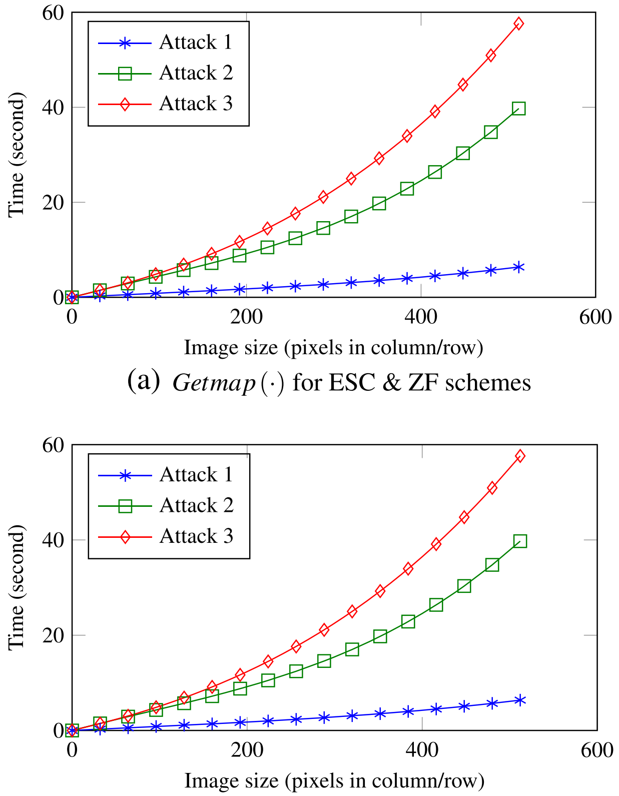
6 Experimental results

In this section, we present our experimental results to validate the effectiveness and to demonstrate possible consequences of the identified counterfeiting attacks. Several experiments were conducted with a set of medical and other images. We analyse the computation time for the effectiveness, and present a set of attacked images for illustrating the possible consequences, of the identified attacks on the ESC scheme [8] and ZF scheme [46]. (The reason for choosing those schemes are discussed in Section 5).

The $Getmap(\cdot)$ computation time, illustrated in Fig. 1, is obtained for the increasing number of image blocks up to the image size of 512×512. As expected, finding the mapping sequence for the ZF scheme is computationally less expensive than the ESC scheme. Further, the average attack computation time of both schemes (shown in Fig. 1) for yielding attacked images for the identified attacks are obtained. To output an attacked image with any level of modifications, it took less than a minute for an input image of size 512×512. We note that these computation times are relative, and depend not only on the computing power of the operating machine, but also on the image and block sizes, number of blocks to modify, underlying design of the schemes, etc. Here, we used a total of 113 (medical and other) images of size 512×512, and varied their sizes to observe the influence of varying image size on the computation time.

A set of examples of the attacked images from our experimental results are shown in Figs. 3 and 4 for the ESC and ZF schemes, respectively. The set of corresponding original watermarked images are shown in Fig. 2. The modified regions (unless the entire image is modified) of the attacked images are indicated by a (red) dotted-ellipse. All the attacked images in complete block swap of *Counterfeiting Attack* 1 are completely distorted as illustrated in Figs. 3 and 4 (from top, first rows). Although these images may have no practical implication, they are verified as authentic and un-tampered by the detector.

As expected, the attacked images in selected block swap, shown in Figs. 3 and 4 (from top, second rows), are not completely distorted. Unlike the ZF scheme, ESC scheme embeds two copies of a watermark (for each block) into two halves of the input images. As a result, it is evident in Fig. 3 (from top, second row) that the selected block swap has symmetric visual artefacts in the two halves of the output attacked images. Since for the selected block



(a) *Getmap*$(\cdot)$ for ESC & ZF schemes

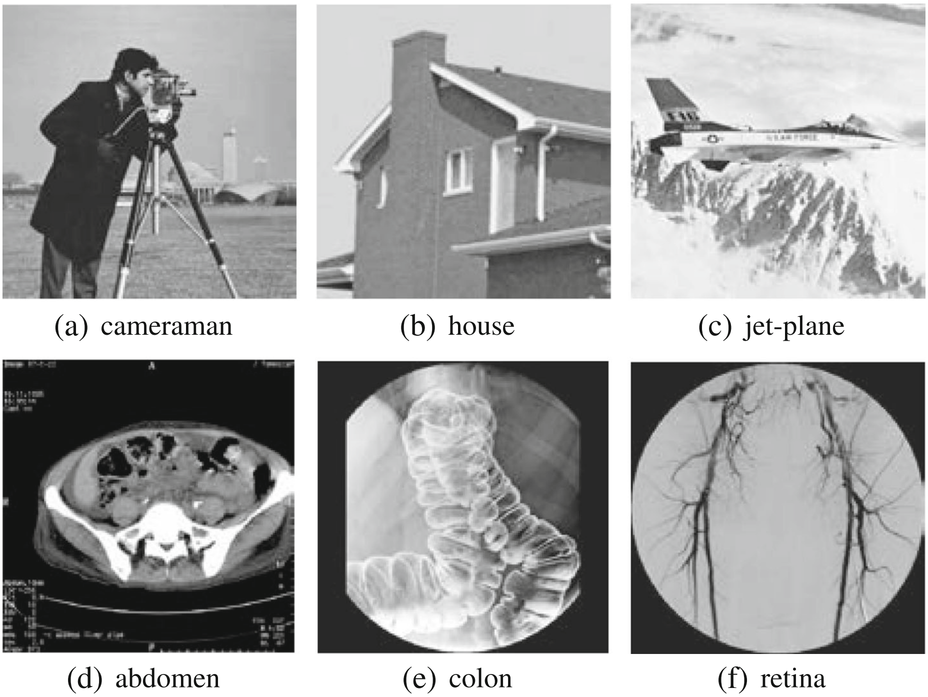
(b) Our identified attacks

Fig. 1 Average computation time for the images (size up to 512 × 512)

swap, we arbitrarily chose a set of block indexes, the output images had no or little practical significance. However, satisfying the win condition with these attacked images suggests that an attacker may succeed with modifying a valid watermarked image having more significant implications. For example, location of a tumour in a Head MRI may be moved in another region of interest, using the selected block swap.

Unlike the *Counterfeiting Attack* 1, the attacked images (shown in Figs. 3 and 4, from top, third rows) for *Counterfeiting Attack* 2, are almost similar to the original watermarked images (in Fig. 2). This is because that the function $Sim(\cdot)$ is designed here to compute a new block using the average intensity of the selected block pixels as described in Section 4.3. Although this example represents a particular case in this counterfeiting level like *constant-average attack*, there can be many other ways to design $Sim(\cdot)$. Further, instead of entire blocks manipulation, an attacker may also consider a selected block scenario for this attack, requiring an additional watermark correction process as mentioned for *Counterfeiting Attack* 1 in Section 4.2.

Furthermore, the attacked images shown in Figs. 3 and 4 (from top, fourth rows) for the *Counterfeiting Attack* 3 illustrate how an attacker outputs a successful forgery with the highest level of modification. An attacker may select a set of arbitrary blocks of a valid watermarked image to replace with a set of chosen blocks. Win with such a modification



(a) cameraman

(b) house

(c) jet-plane

(d) abdomen

(e) colon

(f) retina

Fig. 2 Original set of watermarked images: (**a**–**c**) ESC scheme and (**d**–**f**) ZF scheme. (Original test images for (**a**–**c**) and (**d**–**f**) are downloaded from: [6] and [2], respectively)

leads an attacker to making a complete practical sense for an attacked image in many possible ways, which demonstrates the severity of this attack.

Attacked images of identified attacks, although are perceptually different from the original watermarked images, are not clear for all cases in Figs. 3 and 4 (as shrunk to fit in the page size). Therefore, to observe the difference between the attacked images and respective original watermarked images, we present their PSNR and MSSIM values in Table 2. However, we stress here that the modifications in attacked images are random, and depend on attacker's objectives. So the performance of the attacks and pattern of consequences cannot be determined from the qualitative measures (e.g., PSNR or MSSIM).

Both the ZF and ESC schemes accept all the attacked images (including the images in Figs. 3 and 4) as authentic, where clearly they are not. The implications of the attacked images can be more severe if the attacks are applied in a more meaningful way. However, the presented examples in this paper reasonably show that *Counterfeiting-Attack* 1, -*Attack* 2, and -*Attack* 3 render the schemes invalid for their intended purpose. They also suggest that there would be similar consequences for other SAW schemes based on similar watermarking principle.

7 Countermeasure

Many SAW schemes (including the ZF and ESC schemes) irrespective of their technical differences, do not consider the required properties of the watermarks explicitly. This also

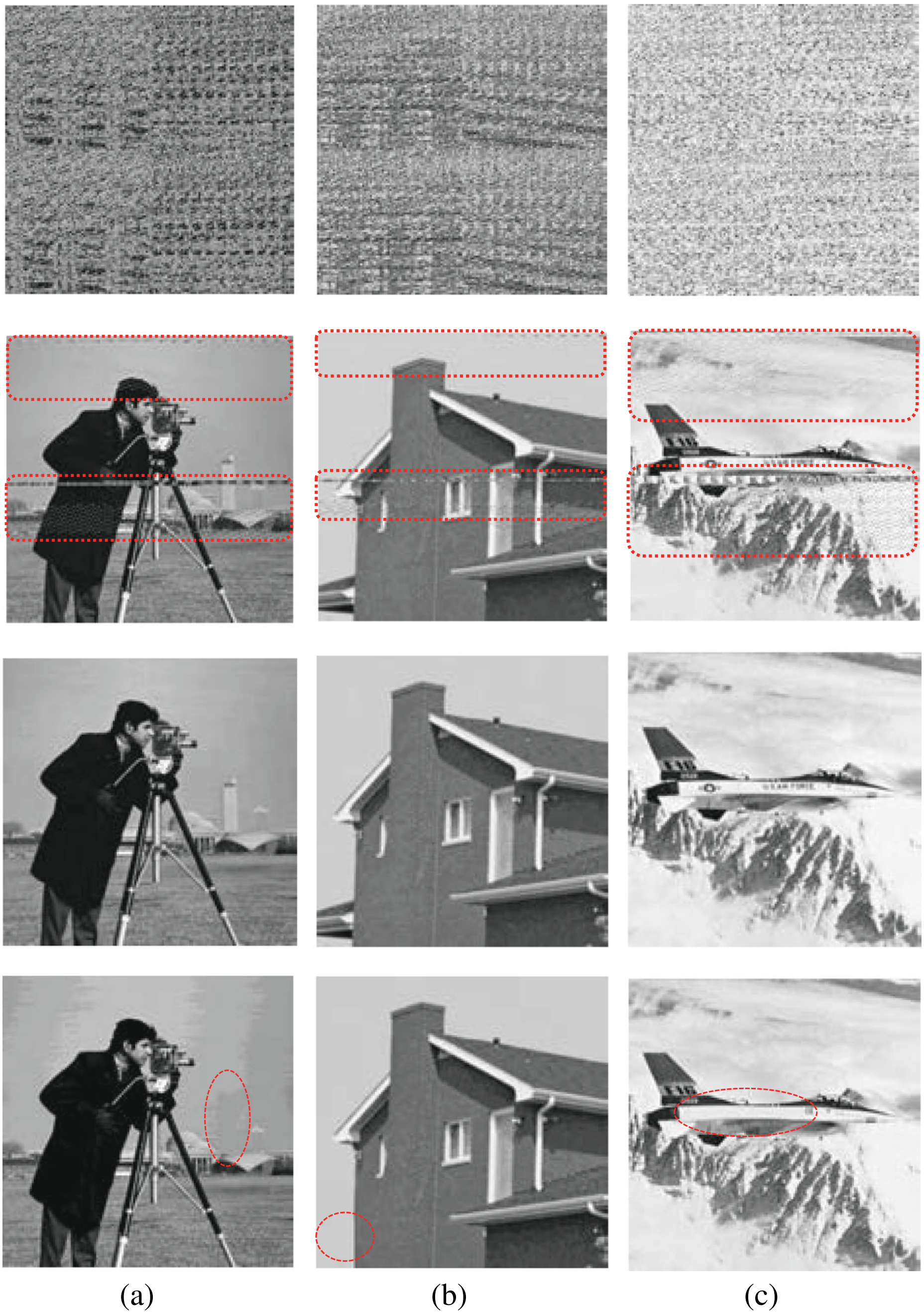


Fig. 3 Attacks on the ESC scheme watermarked images: (**a**) cameraman, (**b**) house and (**c**) jet-plane. From top, 1st row: Counterfeiting Attack 1 (entire blocks); 2nd row: Counterfeiting Attack 1 (selected blocks); 3rd row: Counterfeiting Attack 2; and 4th row Counterfeiting Attack 3

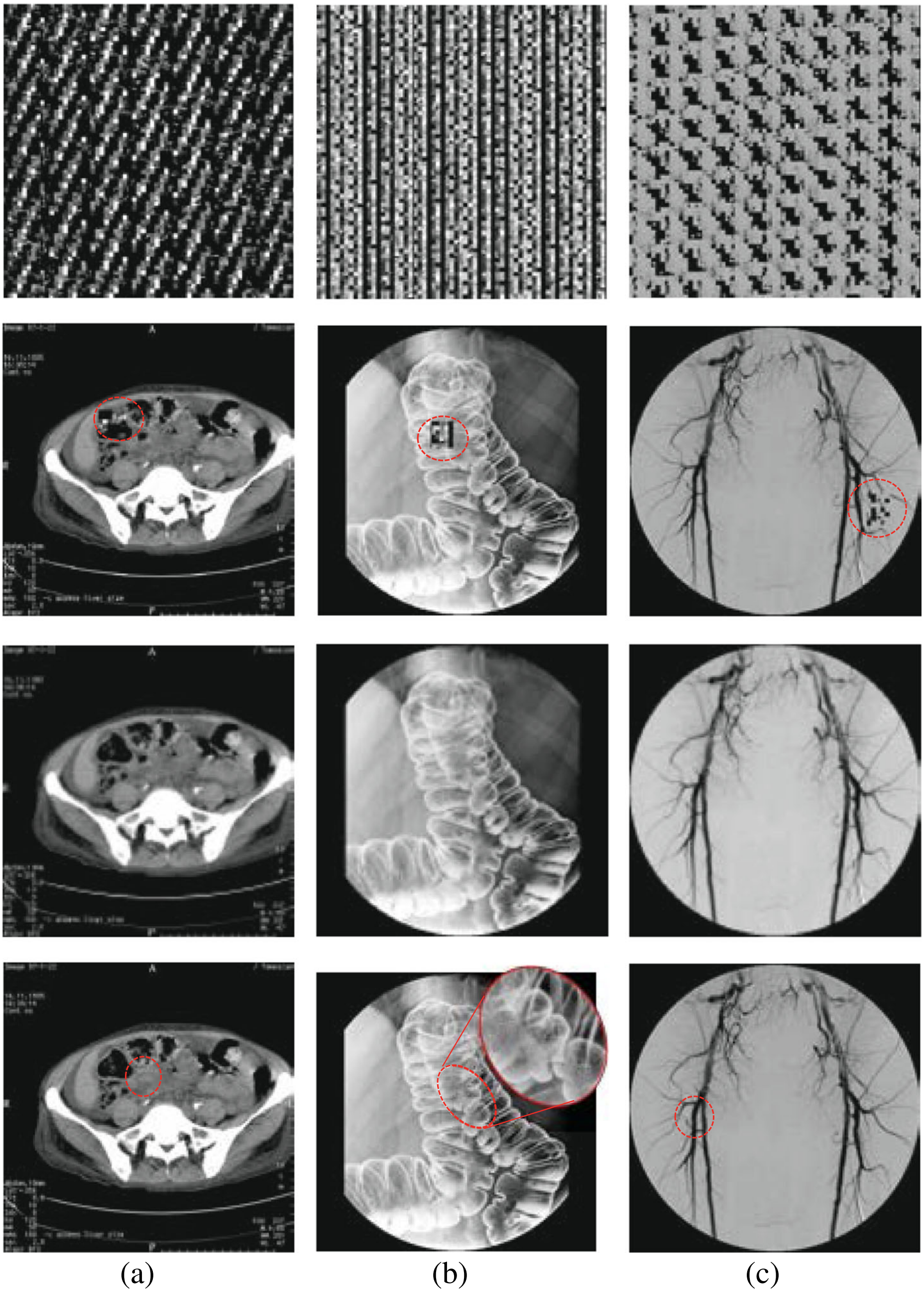


Fig. 4 Attacks on the ZF scheme watermarked images: (**a**) abdomen, (**b**) colon and (**c**) retina. From top, 1st row: Counterfeiting Attack 1 (entire blocks); 2nd row: Counterfeiting Attack 1 (selected blocks); 3rd row: Counterfeiting Attack 2; and 4th row Counterfeiting Attack 3

Table 2 Perceptual differences between output and input images of the identified attacks

Test Image	Measure	Attack 1 (entire blocks)	Attack 1 (selected blocks)	Attack 2	Attack 3
Cameraman (ESC)	PSNR (dB)	9.32	23.97	30.09	39.99
	MSSIM	0.0158	0.8529	0.9897	0.9921
House (ESC)	PSNR (dB)	10.21	28.54	33.45	25.29
	MSSIM	0.0186	0.9241	0.996	0.9824
Jet-plane (ESC)	PSNR (dB)	6.36	11.38	10.54	27.72
	MSSIM	0.3016	0.6682	0.6689	0.9916
Abdomen (ZF)	PSNR (dB)	7.5	29.93	19.29	38.01
	MSSIM	0.0207	0.9879	0.8683	0.9958
Colon (ZF)	PSNR (dB)	6.64	27.1	24.24	45.35
	MSSIM	0.0069	0.9863	0.8965	0.9981
Retina (ZF)	PSNR (dB)	10.55	32.61	26.92	49.14
	MSSIM	0.0538	0.9887	0.9042	0.9988

means that the requirements for a SAW scheme either have not been completely studied yet or are not well understood, which is possibly the main source of several security problems as discussed in Section 2.2. Addressing this, we outline here a set of general requirements for SAW schemes below. We also discuss some guidelines to meet the requirements using existing authentication tools. We particularly illustrate, with extending the SAW Model presented in Section 3, how some of the tools can be employed to achieve the requirements and thus to avoid the counterfeiting weaknesses.

7.1 General requirements for the SAW schemes

General requirements of image (and other multimedia) authentication are well known [3, 17, 18, 38]. However, a SAW scheme, in general, has additional requirements from the typical image authentication, which we call here SAW requirements. We attempt to determine a set of requirements for the SAW schemes in view of the standard authentication tools (e.g., message authentication code, digital signature, etc.) and existing image authentication schemes. The general SAW requirements thus can be: (*i*) authenticity, (*ii*) integrity, (*iii*) unforgeability, (*iv*) non-repudiation (*v*) localisation accuracy, (*vi*) recovery quality, (*vii*) perceptual similarity, (*viii*) embedding capacity, (*ix*) efficiency, and (*x*) reliability. These requirements are discussed below. It is worth noting that, for simplicity, we do not formulate these requirements explicitly.

Authenticity The presence of a valid watermark in a watermarked image implies that the content is deliberately watermarked by the embedder. It is important for a SAW scheme to establish the image content is genuine and was watermarked by an embedder possessing the proper embedding (and/or generation) key (used in watermark embedding and/or generation).

Integrity A valid watermark also ensures the image content is not undetectably modified in an unauthorised way. This further requires the following properties for the watermark:

Fragile A valid watermark embedded in an image is required to be invalid for any smallest changes in the image.

Block-wise collision resistance For a given image block, it is hard to find another image block, which will have the same watermark. (This is a notion of weak collision resistance; however a strong collision resistance can be considered as discussed in Section 2.1).

Unforgeability A valid watermark can only be generated and embedded by a valid generator and embedder (i.e., possessing the proper generation and/or embedding key(s)), respectively. In other words, it is to be computationally "hard" to forge a valid watermark. Here, a watermark may require the following properties: *block-wise dependence* and *block-wise collision resistance*. Block-wise dependence can be of two types:

Intra-block dependence An image block is to be used as an input for its watermark generation. This is required to be *copy attack* resistant (where an attacker directly copies a valid watermark to illicitly embed that in a chosen image which is later verified as authentic for the given key).

Inter-block dependence Image blocks are to be mutually watermark dependent (i.e., watermark of one block is embedded into its mapped block) for the VQ attack resistance.

Non-repudiation A watermarked image must be verifiable to resolve a dispute arising either from a deceitful entity trying to repudiate the watermarked image or from a fraudulent claimant.

Localisation accuracy In case of a tampered image, the localisation of the tampered pixels must come with an optimum accuracy considering computational cost and time.

Recovery quality In case of a tampered image, the localised image pixels must be recovered with an "acceptable" image quality. The notion of "acceptable" image quality may vary with the applications.

Perceptual similarity A watermarked image must be perceptually similar, which ensures an "acceptable" level of distortion in the image, and thus the image remains usable for its intended application.

Embedding capacity A SAW scheme must have the required capacity to accommodate the payload (i.e., the watermark plus any side information). This requirement however may conflict with the restoration quality and perceptual similarity requirements, and thus a necessary trade-off is to be made.

Efficiency A SAW scheme must be computationally efficient to generate, embed and detect (with optimum tampering localisation accuracy and recovery quality) a watermark for a given image. Although the computational effort depends upon the size of input image, the "work" should not grow rapidly with the image size.

Reliability A SAW scheme must be reliable to perform objectively (i.e., to attain the above specified requirements) under given conditions and over a specified period of time.

7.2 How can the SAW requirements be met?

Meeting the above mentioned SAW requirements is a challenging task, which naturally poses a fundamental question: *can the conventional authentication tools meet the SAW requirements?* Addressing this question, we discuss some general principles of using conventional tools as building blocks in SAW schemes. We outline their capabilities and limitations to meet the requirements for the following objectives of a SAW scheme: (*i*) content authentication, and (*ii*) tampering localisation and recovery. This is discussed below and summarised in Table 3.

Using Encryption Encryption is a cryptographic tool generally used to preserve confidentiality of information. Therefore, a direct use of encryption may not help meet the SAW requirements. Encryption of a “suitable” image feature (either by a shared or public key) may help achieve the requirements of integrity, authenticity, etc. to a certain extent [4, 43]. (The suitability of an image feature may depend on several factors; namely, feature length in bits, its computation and uniqueness for image blocks, etc.) For a SAW scheme, the encrypted image-block-features can be used for tampering localisation and recovery. Thereby, some SAW requirements such as integrity, authentication, localization accuracy, etc., can be attained, whereas meeting unforgeability, efficiency, etc., can still be challenging.

Using Message Authentication Code (MAC) A MAC (or a *keyed hash function*) is a cryptographic tool that generates and later verifies an authentication code (or checksum) using a symmetric (or shared) key [36]. For the SAW schemes, block-wise integrity and authentication can be achieved by computing the MAC for each given block (or its unique features) and embedding it into a mapped block. This will help achieve the security level of the used MAC scheme. However, similar to using encryption, the computation time and payload size may grow with the increasing size of the input image and its block, resulting in an efficient SAW scheme. Using MAC also seems incapable of tampering recovery.

Using Digital Signature (DS) DS is another cryptographic tool widely used today, which offers many security services [36]; for example, integrity, authenticity, non-repudiation, unforgeability, etc. Similar to MAC, DS can be block-wise used for the SAW schemes. Such a use of DS can offer tampering localisation, although it still lacks tampering recovery capability. Besides, as DS is usually slower than MAC [36], it can be more computationally expensive for the block-wise embedding principle. It also requires trusted certificates that may incur an additional cost.

Using Perceptual Hash Function (PHF) PHF (also known as *visual* or *robust* (image) *hashing*) is a keyed and content-based hash function that uses image features robust to content-preserving manipulation (e.g., file-format conversion, compression, etc.) and fragile to content-modifying manipulation (e.g., change of objects, background, etc.) [10]. Similar to MAC, it can be used for image integrity and authentication. But, for some special requirements such as access to search a large database of pre-computed hashes, PHF can be more computationally expensive than the other tools. Presumably, the security levels of PHF are also not well known as the cryptographic tools, and thus using PHF without any security proof can be vulnerable in a SAW scenario. Its tampering recovery capability is also unknown.

Table 3 Attainment of SAW Requirements

Tools	Features/ advantages	Limitations
Encryp-tion	Can provide (strict) integrity, authenticity, non-repudiation, etc., if used with hash functions [36] Faster than Digital Signature Can be cryptographically secure Can be robust to content preserving manipulations, if a suitable image feature is used Tampering localisation and recovery, if encrypted image feature is used as a watermark	High capacity may be required and can be computationally expensive, for block-wise embedding
MAC	Integrity (strict) Authenticity (strict) Much faster than Digital Signature Based on block-cipher or cryptographic hash function Can be cryptographically secure Tampering localisation	No non-repudiation High capacity may be required and can be computationally expensive, for block-wise embedding No tampering recovery No robustness to content preserving manipulations
DS	Integrity (strict) Authenticity (strict) Time stamp Non-repudiation Unforgeability Tampering localisation Can be cryptographically secure	Trusted certificates are required High capacity may be required and can be computationally expensive, for block-wise embedding No tampering recovery Much slower than MAC No robustness to content preserving manipulations
PHF	Integrity (selective) Authenticity (selective) Robustness to content preserving manipulations Tampering localisation	High capacity may be required and can be computationally expensive, for block-wise embedding Tampering recovery capability is not known Possibly slower than MAC and DS as access to the database of (large set) pre-computed hash values is required Key recovery weakness for linear randomisation method Learning for appropriate quantisation threshold is required Complete security analysis is required
PDS	Integrity (selective)	High capacity may be required and can be computationally expensive, for block-wise embedding

Table 3 (continued)

Tools	Features/ advantages	Limitations
	Authenticity (selective) Unforgeability? Non-repudiation? Tampering localisation Robustness to content preserving manipulations Possibly faster than PHF	Tampering recovery capability is not known Possibly slower than MAC due to its “robust” feature computation process Complete security analysis is required

Using Perceptual Digital Signature (PDS) PDS (also known as *visual* or *content-based digital signature*) uses the content-preserving-manipulation-invariant features (like PHF) and public key schemes [7]. PDS has potential to provide several security services like DS, considering the PDS’s security levels are known for the application. Generally speaking the performance of a PDS mainly depends upon the image features and their extraction processes. However, for the block-wise embedding principle, PDS is probably faster than PHF as PDS usually does not require any database access and learning process like PHF. Similar to PHF, tampering recovery capability of PDS is also not known.

Whither are the Above Conventional Tools Leading? It is obvious that the above tools distinguish their two different notion of security services: *strict* and *selective*. Cryptographic tools like MAC and DS are intended to serve the strict security services, where a single bit change can be detected. Whereas, the multimedia content-based tools like PHF and PDS are robust to *content-preserving manipulation* like compression, file-format conversion, etc., and thus provide the selective security services. However, since defining the notion of selective security, in general, is more than challenging for different applications, the use of cryptographic tools in SAW schemes can be relatively secure, efficient, and straight-forward. The above considerations and their summary in Table 3 lead us to a conclusion that any individual tool is not sufficient for the attainment of SAW requirements, and thus to considering their combined use.

7.3 An extended SAW model

We extend the construction of the SAW model developed in Section 3. This extended model incorporates the novel approach of employing conventional authentication tools. Use of those tools are not new for image (or other multimedia) authentication, for example, in [1, 4, 7, 9, 13, 17, 18, 29, 40, 43, 44], where authenticity and integrity verification of the visual semantics of multimedia information is mainly addressed. However, as mentioned above, the SAW schemes as a general form of multimedia authentication, also have an additional tampering localisation and recovery objective. Combined use of some of those conventional authentication tools thus seem to be a better option. To this, we consider two different image features, *global* and *local*. The global feature is computed over the whole image and the local feature is computed block-wise. Those features are used for content authentication, and tampering localisation and recovery objectives, respectively.

For simplicity, we partitioned the SAW objectives into two classes: (*i*) *primary* (i.e., content authentication) and (*ii*) *secondary* (i.e., tampering localisation and recovery). We consider the tampering localization and recovery as the secondary objective, since it logically comes after the content authentication (i.e., once the image integrity is found compromised). For the primary objective, a suitable signature scheme can be used for the global feature. Whereas, for the secondary objective, a private key encryption scheme can be used for the local features. The choice of the private key encryption here is made based on the following facts: (*i*) it is simpler and faster than the public key encryption, and (*ii*) using digital signature (for the content authentication), which uses public key, would complement any security need that the used private key encryption does not provide. With this setting, we extend the SAW model, where authenticity and integrity of a watermarked image can be verified publicly (using a public key) and if found tampered, tampering can be localized and recovered using a private key. Before presenting the proposed SAW model, we discuss its component functions below.

Key generation function, $\boldsymbol{KeyGen}(\cdot)$ On the given security parameter τ, $KeyGen(\cdot)$ generates a set of keys: $\{(K_S, K_P), K_R\}$. The pair of public and private keys, K_S and K_P, are used for the signature scheme to sign and to verify the signature, respectively. A private key, K_R is used for both the encryption scheme (which is symmetric and thus shares the same key for encryption and decryption) and mapping function, $Map(\cdot)$.

Feature extraction function, $\boldsymbol{Feature}(\cdot)$ This function takes any (watermarked or unwatermarked) image and outputs its global and local features, f^{pri} and f^{sec}, used for primary and secondary objectives, respectively. Note that f^{pri} is computed over the whole input image and f^{sec} is computed block-wise, i.e., $f^{sec} = \{f_n^{sec}\}$, where an input image i is divided into total N_b non-overlapping blocks such that $i = \{B_n\}$ and $n \in \{1, 2, \cdots, N_b\}$.

Signature scheme $(\boldsymbol{Sign}(\cdot), \boldsymbol{SigVerify}(\cdot), \boldsymbol{K_S}, \boldsymbol{K_P})$ The signing function $Sign(\cdot)$ outputs a signature, w^{pri} on the primary feature, f^{pri} and private signing key, K_S. This signature is embedded as a watermark and extracted in detection to be verified using $SigVerify(\cdot)$ and its public key, K_P. Thus the signature scheme can serve the primary objective. Recall that the $Verify(\cdot)$ in the general SAW model (Section 3) simultaneously verify the image blocks' authenticity, tampering localization and recovery. However, for security reasons and more logical construction, as those tasks have been separated in terms of primary and secondary objectives, the $SigVerify(\cdot)$ is used here to only declare the whole image's authenticity and integrity. If this verification fails, tampering localization and recovery is attempted.

Encryption scheme $(\boldsymbol{Encrypt}(\cdot), \boldsymbol{Decrypt}(\cdot), \boldsymbol{K_R})$ The local feature f^{sec} is block-wise encrypted using encryption function $Encrypt(\cdot)$ and its private key K_R. The encrypted features are block-wise embedded as another watermark for the secondary objective. If an image fails the signature verification, the regenerated watermark, $\{w_n^{newsec}\}$ of the tampered image are compared with the extracted watermark $\{\tilde{w}_n^{sec}\}$. For a mismatch, a block $\bar{B}_n$ is marked as a tampered block $\bar{\bar{B}}_n$, which is recovered by the recovery function, $Recover(\cdot)$.

Model 6 An Extended SAW Model

```
$\{(K_S, K_P), K_R\} \leftarrow KeyGen(1^{\tau})$  ▷ key generation for the security parameter, $\tau$

Watermark Generation, $G(\cdot)$

Input: (i) $i = \{B_n\}$, where $n \in \{1, 2, \cdots, N_b\}$ (ii) $(K_S, K_R)$ (iii) $Feature(\cdot)$ (iv) $Sign(\cdot)$ and (v) $Encrypt(\cdot)$.
Output: (i) $w^{pri}$; and (ii) $w^{sec}$
Begin
  $\left(f^{pri}, \{f_n^{sec}\}\right) \leftarrow Feature(i)$  ▷ feature extraction
  $w^{pri} \leftarrow Sign\left(f^{pri}, K_S\right)$  ▷ primary watermark generation
  $w^{sec} = \{w_n^{sec}\} \leftarrow Encrypt(\{f_n^{sec}\}, K_R)$  ▷ secondary watermark generation
End

Watermark Embedding, $E(\cdot)$

Input: (i) $i = \{B_n\}$ (ii) $w^{pri}$ and $w^{sec}$ (iii) $K_R$ (iv) $Epri(\cdot)$ (v) $Esec(\cdot)$ and (vi) $Map(\cdot)$.
Output: (i) watermarked image, $\bar{i}$.
Begin
  $\bar{i}^{sec} \leftarrow Esec(i, w^{sec}, K_R, Map(\cdot))$  ▷ here, $\{w_q^{sec}\} \leftarrow Map(\{w_n^{sec}\}, K_R)$ for all $n$
  $\bar{i} \leftarrow Epri\left(\bar{i}^{sec}, w^{pri}\right)$
End

Watermark Detection, $D(\cdot)$

Input: (i) $\bar{i} = \{\bar{B}_n\}$ (ii) $K_P$ and $K_R$ (iii) $Epri^{-1}(\cdot)$ (iv) $Esec^{-1}(\cdot)$ (v) $SigVerify(\cdot)$ (vi) $Map(\cdot)$ (vii) $Encrypt(\cdot)$ (viii) $Decrypt(\cdot)$ (ix) $Feature(\cdot)$ and (x) $Recover(\cdot)$.
Output: (i) a pass, $\top$ or the tampering localized and recovered images: $\bar{\bar{i}}$ and $\tilde{i}$.
Begin
  $\tilde{w}^{pri} \leftarrow Epri^{-1}(\bar{i})$
  $\top$ or $\bot \leftarrow SigVerify\left(\bar{i}, \tilde{w}^{pri}, K_P\right)$  ▷ primary objective
  if $SigVerify(\cdot)$ outputs a failure, i.e., $SigVerify(\cdot) \to \bot$ then  ▷ secondary objective
    $\tilde{w}^{sec} \leftarrow Esec^{-1}(\bar{i}, K_R, Map(\cdot))$
    $(\sim, \{f_n^{newsec}\}) \leftarrow Feature(\bar{i})$
    $w^{newsec} = \{w_n^{newsec}\} \leftarrow Encrypt(\{f_n^{newsec}\}, K_R)$
    for all image block index, $n_1 \in \{n\}$ do
      if $w_{n_1}^{newsec} \neq \tilde{w}_{n_1}^{sec}$ then
        $\bar{\bar{B}}_{n_1} \leftarrow \bar{B}_{n_1}$  ▷ tampering localization
        $\tilde{f}_{n_1}^{sec} \leftarrow Decrypt\left(\tilde{w}_{n_1}^{sec}, K_P\right)$
        $\tilde{B}_{n_1} \leftarrow Recover\left(\bar{\bar{B}}_{n_1}, \tilde{f}_{n_1}^{sec}\right)$  ▷ tampering recovery
      end if
    end for
    return $\bar{\bar{i}} = \left\{\bar{\bar{B}}_{n_1}\right\} \cap \{\bar{B}_{n-n_1}\}$ and $\tilde{i} = \{\tilde{B}_{n_1}\} \cap \{\bar{B}_{n-n_1}\}$
  else
    return a pass, $\top$  ▷ indicates $\bar{i}$ is authentic and un-tampered
  end if
End
```

Recovery function, *Recover*$(\cdot)$ This is a component function of the detection, $D(\cdot)$, which outputs the recovered block, $\tilde{B}_n$ for a given tampered block, $\bar{\bar{B}}_n$, using the extracted and decrypted local feature of the block, $\tilde{f}^{sec}_{n_1}$.

Embedding functions, *Epri*$(\cdot)$ and *Esec*$(\cdot)$ In embedding, $E(\cdot)$, two separate embedding functions, namely $Epri(\cdot)$ and $Esec(\cdot)$, are used for embedding the separate watermarks, w^{pri} and w^{sec} respectively. Unlike $Epri(\cdot)$ that embeds w^{pri} over the whole image, $Esec(\cdot)$ is used to block-wise embed w^{sec} using $Map(\cdot)$ and its key K_R. Both embedding functions operate on input images without interfering with each other (e.g., embedding regions are different), and cannot distinguish whether the input images are watermarked or not. To extract the embedded watermarks, their respective inverse functions, $Epri^{-1}(\cdot)$ and $Esec^{-1}(\cdot)$ are used. As discussed in Section 3, the notion of being inverse of the embedding function lies in the fact that these inverse embedding functions extract the bits considering them as the watermark bits.

Mapping function, *Map*$(\cdot)$ A mapping function, $Map(\cdot)$ is used in block-wise embedding of the encrypted local features. As mentioned in Section 3, for the general SAW model, $Map(\cdot)$ uses a linear mapping transform (i.e., $q = [(k \times n)\, mod\ N_b] + 1$ for all n). However, we stress here that a pseudo-random-number-generator based mapping transform can be used to avoid the discussed mapping weakness (Section 2.1).

Model 6 presents the construction of our SAW model based on the above principle and functions. The use of signature and encryption schemes are shown there to achieve the primary and secondary objectives, respectively. Using a signature scheme, the authenticity and integrity of a watermarked image is publicly verifiable (with K_P). Additionally, for a tampered image, tampering can be localised and recovered privately (with K_R) as a secondary objective if required (e.g., for digital forensic processing). We note here that the above extended model, although aimed at capturing all the necessary construction details of a SAW scheme, is not completely general. There are always ways to include additional options depending on the application scenarios, which will be briefly outlined in the following section

8 Future challenges

The desirable notion of security of the SAW schemes may vary and depends on the application scenario. Because even if the system (or non-security) requirements (e.g., perceptual similarity, embedding capacity, etc.) remain the same in different applications, the required security goal and attackers' capabilities may significantly vary. Until we know which scheme is the best for a particular application, developing the new schemes may be left detached from their practical use despite their validation for a partial set of requirements. Although we have studied the case of SAW schemes, our study has revealed some fundamental challenges for the broad range of SAW schemes. These challenges, given below, should essentially be addressed in future research.

Development of a scheme based on the extended SAW model This requires further study on: (*i*) the local and global feature extraction processes; (*ii*) the required properties of the features for different objectives; (*iii*) user key management; and (*iv*) overall security and performance analysis of the scheme, for an application. Choice of the suitable mapping transform, and the embedding, signature, and encryption schemes should also be clearly justified for the application.

Formal treatment of SAW schemes as a watermarking primitive This includes formally defining a SAW scheme and its requirements, analysis of the existing state-of-the-art constructions, developing attack models for broad application scenarios, etc. This will help generate a methodological knowledge to identify the similarities or differences among the variants of SAW scheme such as the self-embedding and self-recovery schemes. As a result, knowing the strength and weakness of a scheme, determining its security level, and thereby choosing an appropriate scheme for an application would be easier and systematic.

Development of quantitative measure for SAW requirements In addition to the above challenges, another question may naturally arise; can we quantify how well a SAW scheme meets the requirements? Since not all applications will have the similar (level) requirements, it can be a further challenge to determine/develop such measures that help verify the attainment of those requirements, for the SAW schemes. Note that, to assess the performance of robust watermarking schemes that are mainly used for copyright protection and fingerprinting, a number of benchmarks (e.g., StirMark [37], Fair benchmark [23], etc.) have been proposed. However, due to having different properties and application requirements, SAW schemes require further development in this area.

9 Conclusions

We have developed a SAW model for the block-based fragile watermarking schemes. We then identified three counterfeiting attacks, developed their models and validated them for the SAW model. We observed that neither the weaknesses of a SAW scheme nor their exploiting in the secret recovery demonstrate how they can affect a target application. In fact, there can be many counterfeiting instances for the SAW schemes in different application scenarios. It is more than difficult (and may not be necessary too) to individually consider every possible counterfeiting instance for developing a SAW scheme. Our identified attacks individually represent the counterfeiting instances in three levels of modifications of a valid watermarked image: (*i*) change of pixel locations only, (*ii*) change of original pixels only, and (*iii*) change of original pixels and watermarks. We, therefore, have argued that the identified attacks generalise all possible counterfeiting instances in those three levels of modification. Experimental results have successfully demonstrated their practical consequences and showed how a SAW scheme can violate the systematic definition of security.

In order to resist the counterfeiting attacks, we have extended the SAW model. Since the model is based on the block-based fragile embedding principle, the state-of-the-art fragile watermarking technique can be used in a block-wise fashion. We have partitioned the objectives of the SAW schemes into primary (i.e., content authentication) and secondary (i.e., tampering localisation and recovery). We have then determined a set of general requirements and presented guidelines for their attainment using conventional authentication tools as building blocks of SAW schemes. We observed that none of the conventional tools can individually help to completely achieve the SAW requirements. These efforts have led us to a logical extension of the SAW model that employs the digital signature and encryption for attaining the primary and secondary objectives, respectively.

Additionally, our study has revealed some fundamental challenges in systematic development and formal analysis of the SAW schemes; namely: (*i*) development of a scheme based on the extended SAW model, (*ii*) formal treatment of SAW schemes as a watermarking primitive, and (*iii*) development of quantitative measure for SAW requirements. We have particularly stressed on formalising the concept of the SAW schemes to systematically determine their security levels.

As a final remark, the presented contributions can be useful in the development and security analysis of SAW schemes. The identified attack models can be used as a means to systematically examine the security levels of similar schemes. Additionally, the extended SAW model with an appropriate consideration of the identified requirements may lead to developing more secure variants of SAW scheme. As this study has demonstrated, failure to consider the security levels and requirements can render a SAW scheme vulnerable for its intended application. In other words, identifying the security levels and the properties of a SAW scheme can help not only to justify the merit of the scheme, but to also show any potential security holes for similar schemes.

Acknowledgments The assistance of Dr Frank Gaillard (Administrator, Radiopedia, http://radiopaedia.org/) for the test medical images, Professor Anthony Maeder, University of Western Sydney, http://www.uws.edu.au/staff_profiles/uws_profiles/professor_anthony_maeder for making useful suggestions regarding medical image datasets and their web-links, and Dr Beat Schmutz, IHBI, Queensland University of Technology, http://staff.qut.edu.au/staff/schmutz/ for providing MR and CT test images, are noted with gratitude.

References

1. Ahmed F, Siyal MY, Uddin Abbas V (2010) A secure and robust hash-based scheme for image authentication. Sig Process 90(5):1456–1470
2. Barré S Medical image samples. http://barre.nom.fr/medical/samples/ (2003). [Online; last accessed 12-Dec-2013]
3. Bartolini F, Tefas A, Barni M, Pitas I (2001) Image authentication techniques for surveillance applications. Proc IEEE 89:1403–1418
4. Celik MU, Sharma G, Saber E, Tekalp AM (2002) Hierarchical watermarking for secure image authentication with localization. IEEE Trans Image Process 11(6):585–595
5. Chang CC, Fan YH, Tai WL (2008) Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recog 41(2):654–661
6. The USC-SIPI image database. http://sipi.usc.edu/database/ (1977). [Online; last accessed 23-Nov-2013]
7. Dittmann J, Steinmetz A, Steinmetz R (1999) Content-based digital signature for motion pictures authentication and content-fragile watermarking. Proc ICMCS'99 2:209–213
8. Edupuganti VG, Shih FY, Chang IC (2012) An Efficient Block-Based Fragile Watermarking System for Tamper Localization and Recovery. CRC Press
9. Fei C, Kundur D, Kwong RH (2006) Analysis and design of secure watermark-based authentication systems. IEEE Trans on Information Forensics and Security 1(1):43–55
10. Fridrich J (2000) Visual hash for oblivious watermarking. In: Electronic Imaging, pp 286–294. SPIE
11. Fridrich J, Goljan M (1999) Images with self-correcting capabilities. In: Proceedings of ICIP'99, vol 3, pp 792–796. IEEE
12. Fridrich J, Goljan M (1999) Protection of digital images using self embedding. In: Symposium on Content Security and Data Hiding in Digital Media. Newark, NJ, USA
13. Fridrich J, Goljan M (2000) Robust hash functions for digital watermarking. In: Proceedings of ITCC'00, pp 178–183. IEEE
14. Fridrich J, Goljan M, Du R (2002) Lossless data embedding-new paradigm in digital watermarking. EURASIP Journal on Applied Signal Processing:185–196
15. Fridrich J, Goljan M, Memon N (2002) Cryptanalysis of the yeung-mintzer fragile watermarking technique. Journal of Electronic Imaging 11:262–274
16. Giry D Cryptographic key length recommendation. http://www.keylength.com/en/ (2013). [Online; last accessed 21-Nov-2013]

17. Han SH, Chu CH (2010) Content-based image authentication: current status, issues, and challenges. Int J Inf Secur 9:19–32
18. Haouzia A, Noumeir R (2008) Methods for image authentication: A survey. Multimedia Tools and Applications 39:1–46
19. He H, Zhang J, Wang H (2006) Synchronous counterfeiting attacks on self-embedding watermarking schemes. Int Journal of Computer Science and Network Security 6(1B):251–257
20. He HJ, Zhang JS, Chen F (2009) Adjacent-block based statistical detection method for self-embedding watermarking techniques. Signal Process 89(8):1557–1566
21. Holliman M, Memon N (2000) Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Trans Image Process 9:432–441
22. Katzenbeisser S, Liu H, Steinebach M (2013) Challenges and solutions in multimedia document authentication
23. Kutter M, Petitcolas FA (1999) Fair benchmark for image watermarking systems. In: Electronic Imaging'99, pp 226–239. SPIE
24. Lee TY, Lin SD (2008) Dual watermark for image tamper detection and recovery. Pattern Recog 41(11):3497–3506
25. Lie WN, Lin TI, Cheng SL (2006) Dual protection of jpeg images based on informed embedding and two-stage watermark extraction techniques. IEEE Trans on Information Forensics and Security 1(3):330–341
26. Liew SC, Zain JM (2010) Experiment of tamper detection and recovery watermarking in picture archiving and communication systems. J Comput Sci 6:794–799
27. Lin PL, Hsieh CK, Huang PW (2005) A hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recognit 38(12):2519–2529
28. Lin PL, Huang PW, Peng AW (2004) A fragile watermarking scheme for image authentication with localization and recovery. In: Proceedings of MSE'04, pp 146–153. IEEE
29. Lu CS, Liao HY (2003) Structural digital signature for image authentication: an incidental distortion resistant scheme. IEEE Trans Multimedia 5(2):161–173
30. Mobasseri BG, Sieffert MJ, Simard RJ (2000) Content authentication and tamper detection in digital video. In: Proceedings of ICIP'00, vol 1, pp 458–461. IEEE
31. Nyeem H (2014) A digital watermarking framework with application to medical image security. Ph.D. thesis, QUT, School of Electrical Eng. and Computer Science, Australia
32. Nyeem H, Boles W, Boyd C (2011) Developing a digital image watermarking model. In: Proceedings of DICTA'11, pp 468–473. IEEE, Piscataway
33. Nyeem H, Boles W, Boyd C (2012) On the robustness and security of digital image watermarking. In: Proceedings of ICIEV'12. IEEE, Piscataway
34. Nyeem H, Boles W, Boyd C (2013) Counterfeiting attacks on block-wise dependent fragile watermarking schemes. In: Proceedings of SIN'13, pp 86–93. ACM
35. Nyeem H, Boles W, Boyd C (2014) Digital image watermarking: its formal model, fundamental properties and possible attacks. EURASIP Journal on Advances in Signal Processing 2014(1):135. doi:10.1186/1687-6180-2014-135
36. Paar C, Pelzl J (2010) Understanding cryptography: a textbook for students and practitioners. Springer
37. Petitcolas FA Stirmark benchmark 4.0. http://www.petitcolas.net/fabien/Watermarking/stirmark/ (2004). [Online; last accessed 26-Mar-2014]
38. Rey C, Dugelay JL (2002) A survey of watermarking algorithms for image authentication. EURASIP Journal on Applied Signal Processing 2002(1):613–621
39. Sencar H, Memon N (2008) Overview of state-of-the-art in digital image forensics. Algorithms, Architectures and Information Systems Security 3:325–348
40. Sun Q, Chang SF (2005) A secure and robust digital signature scheme for JPEG2000 image authentication. IEEE Trans Multimedia 7(3):480–494
41. Tian J Content authentication and recovery using digital watermarks (2008). US Patent 7,389,420
42. Weng L, Braeckman G, Dooms A, Preneel B, Schelkens P (2012) Robust image content authentication with tamper location. In: Proceedings of ICME'12, pp 380–385. IEEE
43. Wong PW, Memon N (2001) Secret and public key image watermarking schemes for image authentication and ownership verification. IEEE Trans Image Process 10:1593–1601
44. Xie L, Arce GR, Graveman RF (2001) Approximate image message authentication codes. IEEE Trans Multimedia 3(2):242–252
45. Yeung MM, Mintzer F (1997) An invisible watermarking technique for image verification. In: Proceedings of ICIP'97, vol 2, pp 680–683. IEEE
46. Zain JM, Fauzi AR (2006) Medical image watermarking with tamper detection and recovery. In: Proceedings of IEEE EMBS'06, pp 3270–3273. IEEE

47. Zhang Hb, Yang C (2004) Tamper detection and self recovery of images using self-embedding. Chin J Electron 32(2):196–199
48. Zhu BB, Swanson MD, Tewfik AH (2004) When seeing isn't believing [multimedia authentication technologies]. IEEE Signal Proc Mag 21(2):40–49

Hussain Nyeem received the B.Sc. degree in electronics and communication engineering from the Khulna University of Engineering & Technology (KUET), Bangladesh in 2007, and the Ph.D. degree in computational intelligence and signal processing from Queensland University of Technology (QUT), Australia in 2014.

Dr Nyeem is currently an Assistant Professor at KUET. He was also a Lecturer at KUET from 2007 to 2010, and a Doctoral research scholar at QUT from 2010-2014. His current research interests include image processing, digital watermarking, teleradiology and eHealth.

Wageeh Boles is currently a Professor at the Electrical Engineering and Computer Science School, Queensland University of Technology (QUT), Australia. Professor Boles was an Assistant Professor at Penn State University, USA, prior to joining QUT where he held several positions including Assistant Dean (Teaching and Learning).

Professor Boles has been successful in obtaining numerous competitive research and teaching development grants and has published widely in high impact journals and conferences. He conducted research on image processing techniques in applications such as biometric human identification using iris and palm images, object recognition and texture analysis and security.

Professor Boles has received many awards for excellence in teaching and leadership. While working at the University of Pittsburgh, PA, he was awarded two Outstanding Teaching Assistant Medals, in 1987–88. At QUT, he won the Faculty of Built Environment and Engineering teaching excellence award and a QUT outstanding academic contribution award in teaching and leadership in 1999. In 2004, he won the National

Engineers Australia and the Australasian Association for Engineering Education Award for Excellence in Teaching and Leadership. He also won the Vice Chancellor's Performance Award and was nominated for two Vice Chancellor's Excellence Awards in 2007. In December 2008, Wageeh won the Faculty of Built Environment and Engineering Dean's excellence award for outstanding contributions to student learning. He was awarded a 2007 Australian Learning and Teaching Council Associate Fellowship and a 2011 National Teaching Fellowship.

Professor Boles is a member of the Institute of Electrical and Electronics Engineers, IEEE, and the Australian Learning and Teaching Fellows.

Colin Boyd is a Professor of information security at the Norwegian University of Science and Technology (NTNU). Prior to moving to Norway in 2013, he held positions at Queensland University of Technology, University of Manchester and British Telecom Research Labs. His research interests are in information security with a particular focus on cryptographic protocols. He is an author of over 150 peer-reviewed articles and Google Scholar reports over 8000 citations of his work.