Towards Practical Self-Embedding for JPEG-compressed Digital Images

Paweł Korus, *Member, IEEE,* Jarosław Białas and Andrzej Dziech

***Abstract*—This paper deals with the design of a practical self-recovery mechanism for lossy compressed JPEG images. We extend a recently proposed model of the content reconstruction problem based on digital fountain codes to take into account the impact of emerging watermark extraction and block classification errors. In contrast to existing methods, our scheme guarantees high and stable level of reconstruction quality. Instead of introducing reconstruction artifacts, emerging watermark extraction errors penalize the achievable tampering rates. We introduce new mechanisms that allow for handling high-resolution and color images efficiently. In order to analyze the behavior of our scheme, we derive an improved model to calculate the reconstruction success probability. We introduce a new hybrid mechanism for spreading the reference information over the entire image, which allows to find a good balance between the achievable tampering rates and the computational complexity. Such an approach reduced the watermark embedding time from the order of several minutes to the order of single seconds, even on mobile devices.**

***Index Terms*—Content Authentication; Content Reconstruction; Self-Embedding; Digital watermarking;**

I. Introduction

Self-embedding is a pro-active digital image protection technique allowing for the reconstruction of maliciously tampered image fragments. It exploits an auxiliary *reconstruction reference*, embedded in the image by means of imperceptible digital watermarking [1]. Typically, the reconstruction reference takes the form of an encoded low-quality representation of the original image and is embedded along with hashes of the original content, which are used for tampering localization.

A variety of self-embedding schemes have been proposed so far, but none of them have successfully addressed all issues preventing practical application of this protection mechanism. Due to high requirements towards watermarking capacity, most of existing schemes use spatial domain least significant bit substitution (LSBS) for information embedding. Such an approach limits the applicability of self-embedding to lossless image representation formats. The key to its practical applicability is to provide support for commonly used lossy-compressed formats, in particular for the widely adopted JPEG. It is also necessary to handle high-resolution and color images efficiently. While there have been some attempts to address JPEG compatibility, the latter aspect has not received any attention. Compatibility with color images cannot be achieved by simple replication of the single-channel protection process, as chrominance channels are often treated differently from the luminance during compression. The issues related to high-resolution images and computational complexity are equally important for successful implementation on mobile image acquisition devices, e.g., smartphones or digital cameras [2].

The goal of this study is to address the above-mentioned issues. The adopted approach to handling lossy compression is different from existing solutions. It aims at delivering a stable and high level of reconstruction quality. We achieve this goal by ignoring damaged fragments of the watermark, which enabled us to maintain the desired reconstruction fidelity at the cost of achievable tampering rates. By contrast, other approaches involve a tolerant restoration procedure, where erroneous fragments of the reconstruction reference introduce reconstruction artifacts. In such schemes, the maximum tampering rate is determined implicitly by the humans' capability to recognize the content. Depending on the configuration, our scheme can perform successful reconstruction when even up to 67% of the image area becomes tampered. For the highest considered fidelity, the average peak signal to noise ratio (PSNR) reaches 33 dB on a dataset of 10,000 natural images.

One of our main goals is to optimize the computational complexity. Due to potential implementation on mobile devices, we focus on watermark embedding. To this end, we develop a new mechanism for spreading the reference information over the image. Previous studies have shown that it is essential for achieving optimal content reconstruction performance [3–5]. However, existing approaches proved computationally infeasible for high-resolution images. Our method combines the benefits of existing solutions; its adoption can reduce the embedding time from a dozen of minutes to a few seconds, even for large images. In this study, we theoretically analyze the efficiency of the proposed mechanism, and verify the obtained results in an exhaustive experimental evaluation. Due to inaccuracy of previous analysis [3] in the current conditions, we derive an improved model to calculate the probability of successful reconstruction.

The paper is organized as follows. In Section II we review the current state-of-the-art, and highlight the key concepts and techniques. The operation of the proposed scheme is described in Section III, and analyzed theoretically in Section IV. The performed experimental evaluation is described in Section V. We conclude in Section VII. Fragments of this work were presented during EUSIPCO'13 [6]. This paper is accompanied

Copyright (c) 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The research leading to these results received funding from the European Regional Development Fund under INSIGMA project no. POIG.01.01.02-00-062/09.

The authors are with the Department of Telecommunications, AGH University of Science and Technology, Kraków 30-059, Poland, (e-mail: pkorus@agh.edu.pl, bialas@kt.agh.edu.pl, dziech@kt.agh.edu.pl). Fax: +48 12 6342372. Telephone: +48 12 6173805 (P. Korus).

by supplementary materials with full resolution, images, more evaluation results, and a short video demonstration.

II. Related Work

The problem of self-recovery can be approached from different perspectives, ranging from simple application of forward error correction codes [7], up to sophisticated mechanisms based on compressive sensing [4]. The problem essentially comes down to communication of the reference information to a decoder over the tampered image. In this section, we review the works that addressed the problem of constructing efficient self-recovery mechanisms, evaluating their success bounds, and supporting lossy compression. We conclude with a brief summary of our contributions.

A. Achievable Limits of Self-Recovery

Most self-embedding schemes operate block-wise. The early ones used naive construction of the watermark, involving embedding a block's reference information into another block. This approach is clearly sub-optimal as it is crippled by inter-block dependencies. A block can be restored only if another block, carrying its reference information, is still authentic. It has been recently shown that for optimal performance it is necessary to spread the reference information on each block over the entire image, and reuse the remaining authentic content during recovery [3–5].

Spreading mechanisms can be constructed by combining reference information from many image fragments while constructing the watermark for each image block. The process is usually controlled by a pseudo-random matrix. Pixel-wise application of such a mechanism is computationally feasible only for very small images. So far, only two practical approaches have been described. First, the input image can be divided into a lot of smaller, randomly selected tasks [3, 4]. The spreading mechanism is applied to each of the tasks separately, and the resulting watermark is additionally scattered over the entire image. A considerable disadvantage of this approach is the drop of achievable tampering rates, e.g., to 25% [3] from the theoretically achievable 33.3% [5]. Another approach is to model the reconstruction problem as a communication over an erasure channel, and apply the protection mechanism to the whole image at once, but using M-ary symbols instead of individual pixels or bits (with symbol length corresponding to blocks' embedding capacity) [5]. Such schemes can be implemented in practice with the use of digital fountain codes (DFC) [8], and achieve the theoretical communication limits. However, neither of the two approaches is well suited for mobile devices. While reasonable computational efficiency is observed for common test images (0.25 Mpx), for high-resolution images the computations become prohibitively time-consuming. A C++ implementation of the DFC-based encoder for JPEG images [6] needs 6 minutes to protect (low quality mode - $\lambda = 2$) a 16 Mpx grayscale image on a typical desktop PC. For color images the time increases twofold.

In order to address this issue, we pseudo-randomly divide the image into sub-images, processed in separate tasks; the DFC-based spreading is performed for each of these tasks separately. This hybrid mechanism combines the approaches described in [3] and [5]. It involves an additional operation layer which controls the execution of the conventional protection mechanism. Such an approach has several benefits. First, the tasks can be executed in parallel which makes the method well suited for contemporary computing environments. Second, it gives more control over the computational complexity. Utilization of constant-size tasks guarantees linear growth of the protection time, instead of the polynomial growth for the single-task version. Finally, proper selection of the number of tasks allows for balancing the processing time and deterioration in success conditions. Specifically, we were able to reduce the watermark embedding time to the order of seconds, even for high-resolution images, without deteriorating the success bounds by more than 3% of the image area.

B. Tolerance for Lossy Compression

The reconstruction reference needs to describe the appearance of the image, and therefore incurs a requirement for high watermarking capacity. As a result, the dominant embedding technique is LSBS in the spatial domain. While it delivers high capacity with low embedding distortion, it fails to provide robustness against any image processing. As a result, it is best suited for fragile watermarking schemes.

Tolerance for selected image processing operations, known as semi-fragility, usually involves robustness against content-preserving operations, most importantly lossy compression. This implies the necessity to tolerate distortions of the reference information, which comes at a cost of abandoning compact image representations, and replacing them with more robust ones. As a result, the reconstruction quality is severely limited. Moreover, it is difficult to assess the efficiency of such schemes as there are no explicit success bounds. The bounds are implicitly determined by the humans' ability to recognize the content despite emerging restoration artifacts.

Only a few semi-fragile self-recovery schemes have been proposed in the literature so far [9–12]. One possible approach is to construct the reconstruction reference as a traditional binary watermark obtained by half-toning the sub-sampled image [9]. The authors quantize discrete wavelet transform (DWT) coefficients to embed the watermark. The reconstruction is performed by inverse half-toning of the extracted watermark. Emerging errors introduce noise into the restoration result, which quickly becomes indiscernible. When no errors are observed the scheme delivers reconstruction PSNR between 22 dB to 28 dB. Alternatively, the reconstruction can be performed by training a multilayer perceptron neural network to predict gray-scale values from the embedded half-tone image [13]. The authors report an improvement of $\approx$4 dB compared to Gaussian filtering-based inverse half-toning.

Content reconstruction can also be modeled as an irregular sampling problem [10]. The restoration is then performed by iterative projections onto convex sets. The reference data is obtained by logical exclusive disjunction on cosine transform coefficients' polarity and pseudo-random bit sequences. The watermark is embedded by modulating middle frequency components of the pinned sine transform. The scheme operates

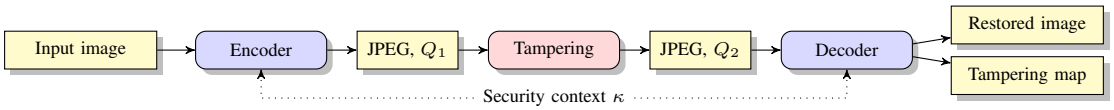


Fig. 1: Operation of the considered self-embedding scenario with prospective recompression to a quality factor $Q_2 \geq Q_1$.

on sub-blocks and macro-blocks, used for authentication and restoration, respectively. The scheme is robust against lossy JPEG compression, Gaussian filtering, unsharpening, and contrast changes. The main limitations include low reconstruction fidelity and low resistance to tampering. Provided that the tampered areas are sufficiently small, and that no global attacks are present, the PSNR in the restored regions can reach 27 dB, but typically varies between 12 dB and 24 dB.

There also exist format-specific schemes, e.g., dedicated to JPEG [11, 12]. As a reconstruction reference Lin and Chang use a down-sampled gray-scale version of the image [11], compressed to an equivalent of JPEG quality level 25. Such an approach severely limits the achievable fidelity. The reconstruction is possible if the tampering affects only a small area. A specific bound on the tampering rate is not reported. Wang et al. use linear regression to predict first 4 DCT coefficients of the tampered blocks from the embedded reference information [12]. Again, the maximum achievable reconstruction quality is low. The reported typical PSNR is ≈25 dB, and drops even further with JPEG quality. The scheme tolerates only limited tampering, but no specific bound is given.

A common disadvantage of the mentioned schemes is not only the relatively low reconstruction quality, but also its quick deterioration with increasing compression strength. Distortions in the reference information are escalated and introduce artifacts which ultimately render the image indiscernible. We propose to adopt a different approach where erroneous portions of the watermark are discarded and do not contribute to the reconstruction process. As a result, the reconstruction quality remains at the same level, and instead the emerging errors limit the tolerance for tampering. In this study, we define 4 quality levels, ranging from coarse fidelity with average PSNR of 28 dB, up to high fidelity with average PSNR of 33 dB. The achievable tampering rates vary between 67% and 20%, depending on the selected quality level.

III. Proposed Scheme

The considered application scenario is illustrated in Fig. 1. The encoder yields a protected JPEG image with quality factor Q_1. As a result of malicious tampering, the attacker yields a JPEG image with quality $Q_2 \geq Q_1$. The reconstruction will also work properly after conversion to a lossless image format, e.g., PNG. The protection process is controlled by a *security context* κ. The structure of the context will depend on the application, and will typically contain the time-stamp of the photograph, the necessary encryption keys, etc.

For the sake of presentation clarity, we first explain the operation of the proposed scheme for single-channel gray-scale images. Extension to color images is discussed separately. The adopted notation of symbols is summarized in Table I. Let I be the input image of size $w \times h$ px, divided into $4N$ blocks of size 8×8 px. Due to limited embedding capacity, the blocks are grouped into 16×16 px macro-blocks, which serve as authentication and reconstruction units. The i-th macro-block is denoted as I_i. The embedding capacity is $4B + 2H$ bits per macro-block. The number of reference bits $4\lambda B$ is identical for all macro-blocks and is controlled by the reference rate $\lambda \in \mathbf{N}^+$. In this study we consider 4 reconstruction fidelity levels, i.e., $\lambda \in \{1, 2, 3, 4\}$.

TABLE I: Summary of the adopted notation.

λ	rate of reference information (quality controlling parameter)
κ	security context
Q_1/Q_2	JPEG quality level of the protected/tampered image
N	number of macro-blocks in the image
N_t	number of tasks
L	number of macro-blocks in a single task
K	total number of tampered macro-blocks in the image
I/I_i	input image / i-th macro-block
r_i	reference information for the i-th macro-block
w/h	image width / height
$\tilde{\gamma}$	tampering rate
B	number of ref. bits that can be embedded in a 8×8 px block
H	number of hash bits that can be embedded in a 8×8 px block
$\mathbf{P}_\lambda$	precision allocation matrix for reconstruction quality level λ
$\mathbf{P}_{Q_1}$	maximum coefficient precision matrix
$\mathbf{E}_{Q_1}$	embedding capacity matrix
X_k	$4B$-bit RLF input symbol
Y_i	$4B$-bit RLF output symbol (ref. payload for i-th macro-block)
$c_i/\hat{c}_i$	i-th DCT coeff. in the current block (original / watermarked)
n_c	number of same-capacity channels in the image
f_p/f_n	false positive / negative block classification rate
f_0	hash collision probability
p_e	watermark symbol error rate
n_l	number of tested combinations during error compensation
d_m	total number of coefficients considered during compensation
d_c	number of coefficients that can be compensated at once
d_h	Hamming distance threshold for hash validation
P_t	distribution of the number of tampered blocks in a task
P_{so}	probability that a single task will be successful
P_S	probability that all tasks are successful
$q(i,j)$	probability that a random binary matrix is of insufficient rank

The restoration success is mainly determined by the reconstruction quality. Let $\tilde{\gamma} = 1 - \gamma$ denote the tampering rate, i.e., the number of tampered authentication units. Then, the restoration success condition becomes [5]:

$$\gamma \geq \lambda(1-\gamma) \Rightarrow \gamma \geq \lambda(\lambda+1)^{-1}. \quad (1)$$

The maximum achievable tampering rate is denoted as $\tilde{\gamma}_{\max}$.

A. Encoder

Operation of the encoder is shown in Fig. 2. The first step is to perform a standard JPEG compression with quality factor Q_1. The resulting JPEG image is then used to generate the reconstruction reference. The reference information for the i-th macro-block is denoted as r_i, and consists of concatenated bit-streams for its corresponding image blocks. Each block

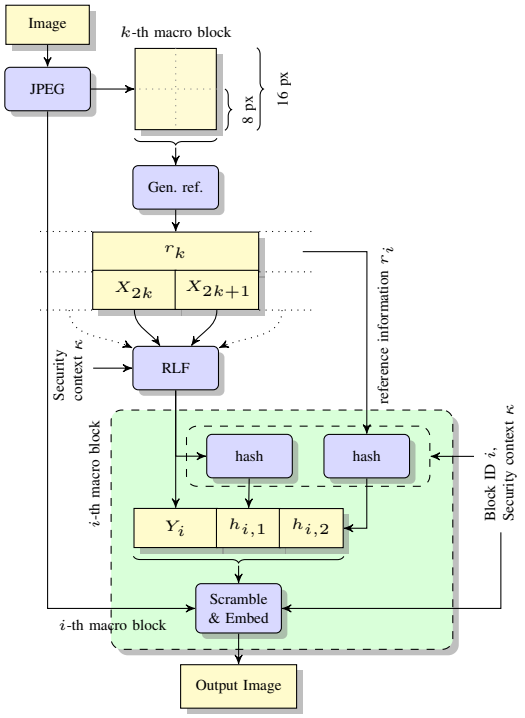


Fig. 2: Operation of the encoder for $\lambda = 2$.

is described by λB bits, allocated to individual coefficients according to an allocation matrix $\mathbf{P}_\lambda$. The component corresponding to the i-th coefficient is denoted as $\mathbf{P}_\lambda[i]$.

Let $\mathbf{E}_{Q_1}$ be a matrix of embedding capacity, and $\mathbf{D}_{Q_1}$ a matrix of maximal coefficient precision for Q_1. Then, the reference information for the i-th DCT coefficient c_i can be extracted as its sign and $\mathbf{P}_\lambda[i] - 1$ most significant bits from its $\mathbf{D}_{Q_1}[i]$-bit representation:

$$\text{round}\left(c_i \cdot 2^{\mathbf{P}_\lambda[i] - \mathbf{D}_{Q_1}[i]}\right). \tag{2}$$

Coefficients' magnitudes exceeding the precision defined by $\mathbf{D}_{Q_1}$ are saturated to $2^{\mathbf{D}_{Q_1}[i]} - 1$. In order to ensure that the extractable reference information is identical after watermark embedding, the following condition needs to be satisfied:

$$\forall_{i=1,2,\ldots,64}\ \mathbf{P}_\lambda[i] + \mathbf{E}_{Q_1}[i] \leq \mathbf{D}_{Q_1}[i]. \tag{3}$$

The complete reconstruction reference is then divided into $4B$-bit symbols $X_k : k \in \{1, \ldots, \lambda N\}$. A random linear fountain (RLF) code produces same-length embedding symbols $Y_i : i \in \{1, \ldots, N\}$ for N macro-blocks. Watermark symbols are then obtained by appending two H-bit hashes to validate the watermark payload, and the image content, respectively:

$$h_{i,1} = \text{hash}(Y_i, \kappa, i), \tag{4a}$$

$$h_{i,2} = \text{hash}(r_i, \kappa, i). \tag{4b}$$

The double-hash mechanism improves the reconstruction performance by enabling discrimination between corrupted block payload and content (Section IV).

The final step is to scramble the individual watermark symbols and embed them into their corresponding macro-blocks. In order to embed a message $m \in \{0, \ldots, 2^{\mathbf{E}_{Q_1}[i]} - 1\}$, the coefficients of the originally produced JPEG file are modified according to:

$$\hat{c}_i = \text{round}\left(c_i \cdot 2^{-\mathbf{E}_{Q_1}[i]}\right) 2^{\mathbf{E}_{Q_1}[i]} - 2^{\mathbf{E}_{Q_1}[i]-1} + m, \tag{5}$$

which can be seen as a variant of quantization index modulation (QIM) [14] or bit substitution.

Depending on the desired embedding strength, the coefficients might be bit-wise shifted before and after embedding. We use a 1-bit shift for $Q_1 \geq 92$. The embedding locations are defined individually for various quality levels Q_1 by means of the embedding capacity matrices $\mathbf{E}_{Q_1}$, obtained by selecting coefficients least vulnerable to rounding errors. Two example matrices for the luminance component are shown in (6). For the sake of notation compactness, only positive elements are shown. All matrices are of size 8×8.

$$\mathbf{E}_{85} = \begin{pmatrix} 222222 \\ 2\ \ 22 \\ 2\ 22 \\ 222 \\ 22 \\ 2 \end{pmatrix}, \mathbf{E}_{90} = \begin{pmatrix} 2\ \ 222 \\ 2\ 222 \\ 2222 \\ 222 \\ 22 \\ 2 \end{pmatrix}. \tag{6}$$

The allocation matrices $\mathbf{P}_\lambda$ for quality levels $\lambda = 1$ and $\lambda = 2$ are shown in (7).

$$\mathbf{P}_1 = \begin{pmatrix} 643 \\ 44 \\ 3 \end{pmatrix}, \mathbf{P}_2 = \begin{pmatrix} 7663 \\ 663 \\ 53 \\ 3 \end{pmatrix}. \tag{7}$$

B. Decoder

Operation of the decoder is shown in Fig. 3. The first step is to extract the watermark. For each watermarked coefficient $\hat{c}_i$ the embedded message m is extracted according to:

$$m = \hat{c}_i - \text{round}\left(\hat{c}_i \cdot 2^{-\mathbf{E}_{Q_1}[i]}\right) 2^{\mathbf{E}_{Q_1}[i]} - 2^{\mathbf{E}_{Q_1}[i]-1}. \tag{8}$$

The extracted symbols are then unscrambled and demultiplexed to yield the embedding payload $\hat{Y}_i$, and the hashes $\hat{h}_{i,1-2}$. Simultaneously, reference information is regenerated. Both hashes are then recalculated, and compared with their extracted counterparts. The resulting erasure and tampering maps identify image blocks which need to be restored, and watermark symbols which can be used for the restoration.

Due to prospective coefficient rounding errors resulting from recompression, a compensation step is employed. A precalculated error map indicates the coefficients, which are the most vulnerable. If a block is deemed tampered, the decoder attempts to match the hashes for a number of most likely rounding errors. We allow for ± 1 changes in the coefficients' magnitudes. Analogous compensation is used for the watermark payload, where bit-flips are considered. Additionally, since the hashes differ significantly even for the slightest changes in the input data, it is beneficial to increase the Hamming distance threshold to allow for a certain number of erroneous bits during hash comparison, provided that their locations match the most probable rounding errors. The number of trials should be chosen according to the desired false negative classification rate (Section IV).

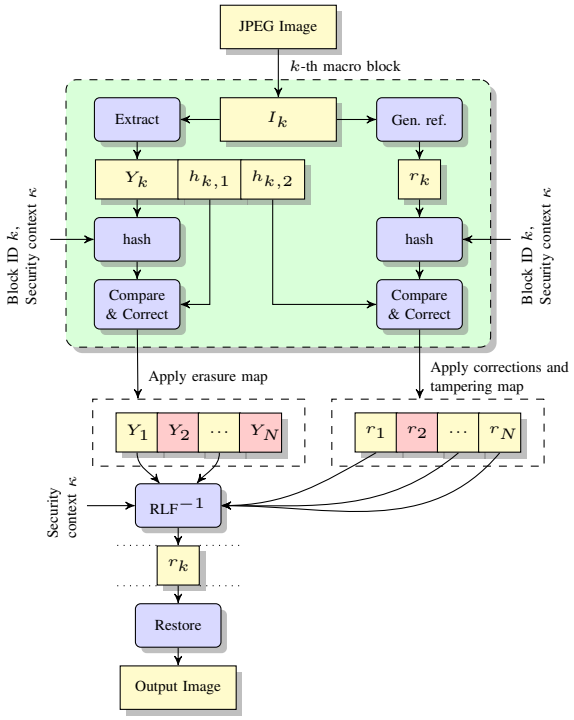


Fig. 3: Operation of the self-embedding decoder.

The corrected reference information regenerated from authentic image blocks is then used to remove their corresponding dependencies from the correctly extracted embedding symbols. The resulting simplified RLF code is then decoded to yield the reference information of the tampered image fragments. Their approximate original appearance is then restored using the recovered DCT coefficients.

C. Operation on Color Images

Extension to color images operates in one of two possible modes. Assuming gray-scale reconstruction is sufficient, the additional embedding capacity can be used to provide more redundancy for the luminance channel. We refer to this configuration as *extended grayscale reconstruction*. Then, disregarding rounding errors, the restoration condition can be obtained by adapting (1):

$$n_c\gamma \geq \lambda(1-\gamma) \Rightarrow \gamma \geq \lambda(n_c+\lambda)^{-1}, \tag{9}$$

where n_c denotes the number of available same-capacity channels. Due to more severe quantization of the chrominance components, their capacity is lower than for the luminance. However, the rate of reference information is also lower. We combine the Cb and Cr channels to obtain a single auxiliary channel of identical capacity as the luminance. Hence, $n_c = 2$ which translates to improvement of the achievable tampering rates by up to 17% of the image area (Table II).

Alternatively, the additional capacity can be used for the reconstruction of the chrominance channels. The reconstruction is performed independently from the luminance component, and the success bound for each component is (9) for $n_c = 1$.

TABLE II: Improvement of the supported tampering rates in the extended grayscale reconstruction mode.

	Max. tampering rate $\tilde{\gamma}_{\max}$ for			
	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 4$
$n_c = 1$	0.50	0.33	0.25	0.20
$n_c = 2$	0.67	0.50	0.40	0.33

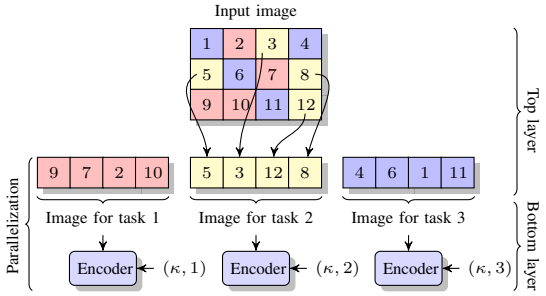


Fig. 4: Parallel processing of a high-resolution image.

D. Handling High-Resolution Images

In order to address the problem of excessive computational complexity, we propose a two-layer spreading mechanism (Fig. 4), which combines the approaches proposed in [5] and [3]. The top layer pseudo-randomly divides the input image into separate tasks, analogously as in [3]. Hence, the tasks contain macro-blocks that are scattered over the entire image. Each task is then independently processed at the bottom layer (Section III-A) which enabled their parallel execution. The task identifier needs to be included in the security context κ.

The number of tasks can be either specified directly, or indirectly via the desired task size. Such flexibility allows for more precise control over the computation time, which grows polynomially with the size of a single task. If the task size is fixed, the total processing time will increase linearly. The process can be configured to maintain achievable tampering rates close to the theoretical bounds, as well as significantly reduced processing time (Section IV-B).

IV. Theoretical Analysis

This section shows how the theoretical model proposed in [5] can be extended to take into account the impact of watermark extraction and block classification errors. We also calculate the reconstruction success probability for the proposed two-layer reference information spreading mechanism.

A. Impact of Block Classification Errors

Due to coefficient rounding errors during prospective image editing, unintentional bit flips either in the blocks' reference information, or the embedded payload, make it possible for authentic image blocks to be misclassified as tampered. Such blocks would be restored in the decoder, and would unnecessarily limit the achievable tampering rates. Given false positive

TABLE III: Success bounds in the presence of block classification and watermark extraction errors.

		$\tilde{\gamma}_{\max}$ [%] for symbol error rate p_e				
Mode	λ	0.0	0.01	0.05	0.10	0.15
Single-hash	1	50.0	49.5	47.4	44.4	41.2
Single-hash	2	33.3	32.7	29.8	25.9	21.6
Single-hash	3	25.0	24.2	21.1	16.7	11.8
Single-hash	4	20.0	19.2	15.8	11.1	5.9
False positive rate $f_p = 0.01$						
Double-hash	1	49.7	49.5	48.5	47.1	45.7
Double-hash	2	32.9	32.7	31.7	30.6	29.3
Double-hash	3	24.4	24.2	23.5	22.5	21.5
Double-hash	4	19.4	19.2	18.5	17.7	16.8
False positive rate $f_p = 0.05$						
Double-hash	1	48.7	48.5	47.4	45.9	44.4
Double-hash	2	31.0	30.8	29.8	28.6	27.3
Double-hash	3	22.1	21.9	21.1	20.0	18.9
Double-hash	4	16.7	16.5	15.8	14.9	14.0

classification rate f_p, the restoration condition becomes:

$$(1-f_p)\gamma \geq \lambda(1-\gamma) + \lambda\gamma f_p, \tag{10a}$$

$$\gamma \geq \lambda(1 - f_p + \lambda(1-f_p))^{-1}, \tag{10b}$$

for a self-embedding scheme with a single hash.

The introduced double-hash mechanism can distinguish tampered blocks from erased embedding symbols. If a block is authentic, yet contains invalid payload, it will not be reconstructed. Let p_e denote the watermark symbol error rate. Then, the reconstruction condition becomes:

$$(1-p_e)\gamma \geq \lambda(1-\gamma) + \lambda\gamma f_p, \tag{11a}$$

$$\gamma \geq \lambda(1 - p_e + \lambda(1-f_p))^{-1}. \tag{11b}$$

False positive classification errors are significantly less frequent than watermark symbol errors, i.e., $f_p \ll p_e$. Hence, adoption of the double-hash mechanism limits unnecessary block reconstructions, and leads to higher achievable tampering rates. Table III collects the theoretical tampering rate bounds for both the single and the double-hash configurations. For an example case of $f_p = 0.01$ and $p_e = 0.1$, the tampering rate bound for the reconstruction quality $\lambda = 2$ is 25.9% for the single, and 30.6% for the double-hash configurations.

False negative classification errors occur when a tampered block is by chance deemed authentic. The primary factor, which influences the collision probability f_0 is the length of the hash H, i.e., $f_0 \approx 2^{-H}$. The introduced hash tolerance and compensation mechanism increases the effective collision probability. By proper selection of the compensation parameters it is possible to maintain the desired error rate.

The compensation mechanism attempts to perform the most likely ± 1 adjustments of the coefficients' values. Given that up to d_c coefficients out of d_m most probable ones can be corrected at once, the number of tested combinations is:

$$n_l = \sum_{i=1}^{d_c} \left(\begin{array}{c} d_m \\ i \end{array} \right) 2^i. \tag{12}$$

Once the compensation attempts fail, the decoder compares the Hamming distance between the hashes against a threshold d_h. The number of possible valid hashes is:

$$\sum_{i=1}^{d_h} \left(\begin{array}{c} H \\ i \end{array} \right). \tag{13}$$

Finally, the false negative probability can be estimated from the Bernoulli trials:

$$f_n \approx 1 - (1-f_0)^{n_l} + (1-f_0)^{n_l} \sum_{i=1}^{d_h} \left(\begin{array}{c} H \\ i \end{array} \right) f_0. \tag{14}$$

B. Impact of the Proposed Two-Layer Spreading Mechanism

Division of the content-reconstruction problem into smaller tasks has two main consequences. First, it gives more control over the computational complexity. In particular it allows for linear growth of the computation time with the image size, and for parallel execution of the tasks. Second, successful recovery becomes less probable as the number of tasks increases. Several reconstruction problems need to be solved for the whole reconstruction to be successful which leads to increased overall failure probability. Additionally, the efficiency of RLF coding deteriorates for smaller problems, as the relative overhead of the code increases. Combined with the requirement for multiple tasks to be successful, this may severely deteriorate the achievable reconstruction success bounds.

As a result, there is a trade-off between the computational complexity and the achievable success bounds. We will show that it is possible to balance the two aspects by properly choosing the number of tasks. Without loss of generality, we assume same size of the tasks. Each of the tasks experiences various extent of tampering, oscillating around the known tampering rate $\tilde{\gamma}$ for the whole image. Originally, this phenomenon was modeled with the use of Bernoulli trials under the assumption that the tasks are fully independent [3]. Then, when considering a single task, the number of tampered macro-blocks has the following distribution:

$$P_t(m) = \binom{L}{m} \cdot \tilde{\gamma}^m \cdot (1-\tilde{\gamma})^{L-m}, \tag{15}$$

where L is the number of macro-blocks within each task, and $\tilde{\gamma}$ is the tampering rate, corresponding to failure probability in the Bernoulli-based model. The probability of successful reconstruction of a single task is then:

$$P_{so} = \sum_{m=0}^{L} P_t(m)(1 - q(L-m, \lambda m)). \tag{16}$$

The term $1 - q(i,j)$ denotes the probability that a random binary matrix of size $i \times j$ has sufficient rank. Based on boundary analysis [15] we approximate it as:

$$q(i,j) \approx \begin{cases} 1, & \text{if } j > i, \\ 2^{-i}, & \text{if } j = 1, \\ 0.712, & \text{if } j = i, \\ 2^{j-i}, & \text{otherwise.} \end{cases} \tag{17}$$

Let $K = \tilde{\gamma}N$ be the number of tampered macro-blocks in the whole image, divided into $N_t = N/L$ tasks. The overall

reconstruction success probability is $P_S^{(\text{leg})} = P_{so}^{N_t}$. We refer to this Bernoulli-based model as a *legacy model*. While its accuracy is good for higher numbers of tasks, for lower ones it is no longer satisfactory. By examining the case of two tasks, it becomes clear that the tasks are dependent as they are constrained by the given number of tampered blocks in the whole image. When the tampering rate is near the theoretical bound, the increasing probability of the first task's success makes the second task's success impossible.

In order to address this issue, we derive an improved model, which accurately reflects the mentioned phenomena. The number of combinations of m tampered macro-blocks is $\binom{K}{m}$ and the number of combinations of choosing the complementing authentic macro-blocks is $\binom{N-K}{L-m}$. The total number of combinations is $\binom{N}{L}$. Let $m^{(1)}$ denote the number of tampered macro-blocks in the first task. Then the probability mass function of the possible outcomes has the following form:

$$P_{t_1}(m^{(1)} = m) = \rho(m, N, L, K) = \frac{\binom{K}{m}\binom{N-K}{L-m}}{\binom{N}{L}}, \quad (18)$$

where $m \in \Omega_m^{(1)} = \{\max(0, L+K-N), \ldots, \min(K, L)\}$. The reconstruction success probability for a single task can now be calculated as:

$$P_s = \sum_{m \in \Omega_m^{(1)}} P_{t_1}(m^{(1)} = m)(1 - q(L-m, \lambda m)). \quad (19)$$

The formula in (19) can also be interpreted as the expected *successful tasks rate*. By estimating the overall success probability as $P_S^{(\text{sim})} = P_{LI}^{N_t}$, we obtain a *simplified model*. Such an approximation is reasonable in the high probability region, but looses accuracy near the theoretical tampering rate bound. In order to calculate the overall success probability we need to take into account how the probability distribution in (18) changes for successive tasks.

The probability that the second task has m tampered macro-blocks, given that the first task has $m^{(1)}$ macro-blocks is:

$$P(m^{(2)} = m | m^{(1)}) = \frac{\binom{K-m^{(1)}}{m}\binom{N-K+m^{(1)}}{L-m}}{\binom{N-L}{L}} = \quad (20a)$$

$$= \rho(m, N-L, K-m^{(1)}, L). \quad (20b)$$

Let use define $K^{(i)}$ as the number of tampered macro-blocks, not yet assigned to any of previously considered $i-1$ tasks. Analogously $N^{(i)} = N - (i-1)\cdot L$ denotes the total number of not yet considered macro-blocks. Then:

$$P(m^{(2)} = m | m^{(1)}) = \rho(m, N^{(2)}, K^{(2)}, L). \quad (21)$$

This leads to the following recursive formula for the probability of all possible combinations:

$$P(m^{(1)}, \ldots, m^{(N_t)}) = P(m^{(1)}) \cdot \quad (22a)$$

$$\cdot P(m^{(2)} \mid m^{(1)}) \cdot \quad (22b)$$

$$\cdot P(m^{(3)} \mid m^{(1)}, m^{(2)}) \cdot \quad (22c)$$

$$\cdot \ldots \cdot \quad (22d)$$

$$\cdot P(m^{(N_t)} \mid m^{(1)}, \ldots, m^{(N_t-1)}). \quad (22e)$$

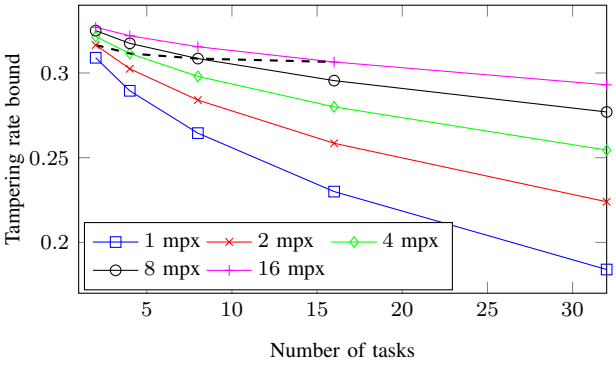


Fig. 6: Theoretical success bounds for the proposed reference spreading mechanism ($\lambda = 2$); failure threshold 10^{-3}.

The last term is always equal to 1 since there are only $N_t - 1$ degrees of freedom. The formula can be written in a more compact form:

$$P(m^{(1)}, \ldots, m^{(N_t)}) = \prod_{i=1}^{N_t - 1} \rho(m^{(i)}, N^{(i)}, K^{(i)}, L), \quad (23)$$

and the success probability can be easily derived as:

$$P_S^{(\text{full})} = \sum_{\substack{m^{(i)} \in \Omega_m^{(i)}, \\ i=1,\ldots,N_t}} \prod_{i=1}^{N_t} \rho(m^{(i)}, N^{(i)}, K^{(i)}, L) \cdot q(L - m^{(i)}, \lambda m^{(i)}), \quad (24)$$

where $\Omega_m^{(i)}$ denote the sets of feasible ranges of $m^{(i)}$, i.e.,:

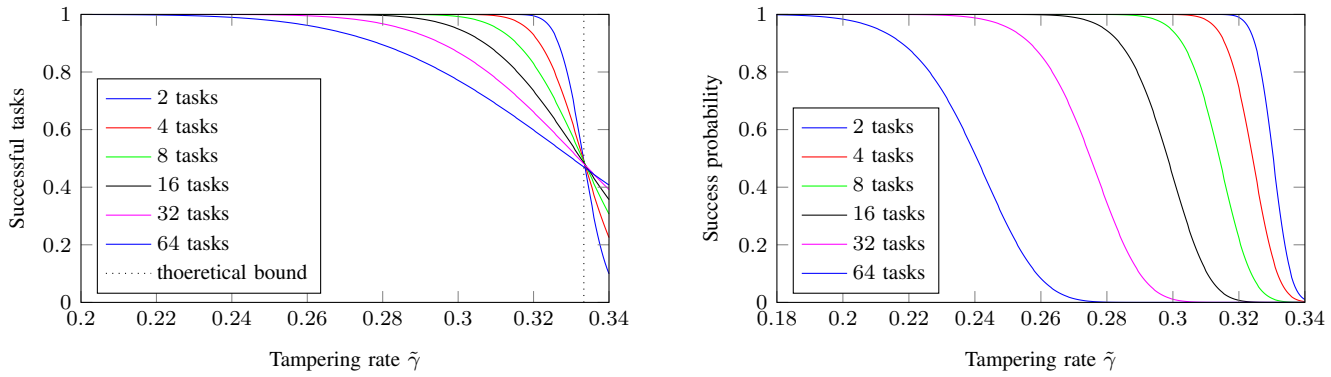
$$\Omega_m^{(i)} = \{\max(0, L + K^{(i)} - N^{(i)}), \ldots, \min(K^{(i)}, L^{(i)})\}. \quad (25)$$

We refer to this formula as the *full model*. Due to excessive computational complexity, for some configurations we use the reasonably accurate simplified model (Appendix A).

Fig. 5 shows the expected successful tasks rate P_s and the reconstruction success probability P_S vs. tampering rate for a 2 Mpx image, $\lambda = 2$ and various numbers of tasks. It is clearly visible that the success probability decreases with increasing number of tasks. The curve of the successful tasks rate becomes flatter. However, it is always nearly 0.5 for the theoretical upper bound $\tilde{\gamma}_{\max}$ (marked with a dotted line).

By applying a threshold on the reconstruction success probability it is possible to define a bound on the achievable tampering rate. Fig. 6 shows the obtained bounds for $\lambda = 2$ and failure threshold 10^{-3}. While the success bounds deteriorate considerably with increasing number of tasks, it is possible to remain close to the theoretical limits by carefully choosing the task size. Based on the performed analysis, we can conclude that a reasonable strategy is to define a desired size of a single task. For 1 Mpx tasks, the tampering rate bound does not fall below 30% of the image area from the theoretical limit of 33.3%. This configuration is marked with a thick dashed line in Fig. 6. The impact on watermark embedding time will be addressed in detail in Section V-E.

The tampering rate bound can also be defined differently. If it is not necessary to guarantee successful recovery for all of the tasks, it might be beneficial to define the bounds based



(a) Expected number of successful tasks

(b) Success probability

Fig. 5: Theoretical curves for the expected number of successful tasks and for the overall success probability versus the tampering rate for a 2 Mpx image and various numbers of tasks.

on the expected successful tasks rate. Irrecoverable tasks can be approximated from the surrounding blocks (Section V-F).

V. Experimental Evaluation

In the performed experiments we use $H = 24$ bit hashes, and embedding symbols of length $4B = 96$. In each 8×8 px block, we embed 36 bits, divided into $6+6$ bits for the hashes $h_{i,1-2}$, and 24 bits for the reconstruction reference. Hence, the amount of reference information per macro-block is 96λ bits. Except for the protection time assessment for high-resolution images, the experiments were performed on 512×512 px natural gray-scale images from the BOWS2 data set [16].

A. Block Classification Errors

The goal of this experiment was to assess the false positive classification rate, i.e., how often authentic, but recompressed image blocks are deemed as tampered. The first step was to produce a protected image with quality $Q_1 \in [85; 100]$. After recompression to $Q_2 \in [Q_1; 100]$, the decoder attempted to authenticate the image. The experiment was repeated with 10 distinct seeds for the PRNG, and a subset of 120 representative natural images. Fig. 7 shows the average rates of correctly classified blocks and extracted watermark symbols. The highest observed false classification rate f_p is 0.0003%. The highest observed symbol error rate p_e is 0.0608%.

To validate the theoretical estimate of the false negative probability (14), we tested 6,243 unwatermarked images. Since $H = 24$, the rank of the principal false negative rate is $\log_{10} f_0 = -7.22$. Compensation of the reconstruction reference uses $d_m = 24$ and $d_c = 2$; from (14) the rank of the false negative rate increases to $\log_{10} f_n = -4.16$. The compensation has dominant influence on f_n and on the basis of (14) we can allow for $d_h = 2$ without significantly deteriorating f_n. Then, the expected $\log_{10} f_n = -4.06$. From the total of 6,392,832 blocks, exactly 608 were classified as authentic. Hence, the empirical false negative classification rate falls into range $\log_{10} f_n \in [-4.058; -3.989]$ with 95% confidence.

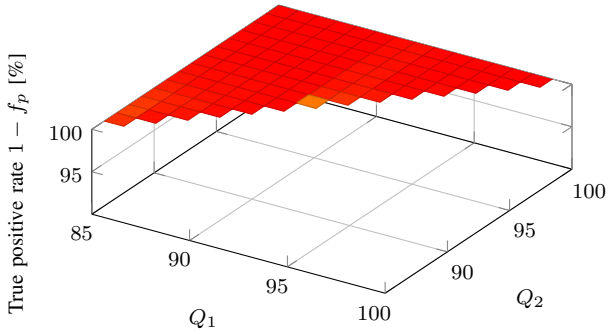
The payload compensation mechanism allows to consider $d_m = 32$ most probable coefficients. Hence, after allowing for up to $d_h = 2$ different bits in the hash vectors, the rank of the false negative rate increases to $\log_{10} f_n = -3.854$. Exactly 834 blocks were identified as carrying a valid watermark payload. Hence, the empirical false negative classification rate falls into range $\log_{10} f_n \in [-3.915; -3.856]$ with 95% confidence.

B. Reconstruction Success Bound Validation

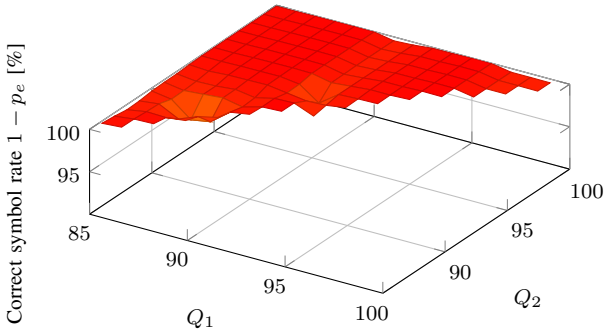
1) Impact of the Problem Sub-Division: In order to verify the derived theoretical models of the reconstruction success probability, we performed repeated reconstruction attempts for various tampering rates for images of size of 1 Mpx, 2 Mpx and 4 Mpx. Each attempt was initialized with a different security context κ. The images were pseudo-randomly divided into separate tasks, and then tampered by damaging a given number of macro-blocks. For each tampering rate, we counted successful reconstruction attempts and the number of successful tasks. We present the results for an example configuration of $\lambda = 2$, for which $\tilde{\gamma}_{\max} = \frac{1}{3}$ (1). The tampering rates were chosen to cover the range between overall success probability 10^{-2} and $1 - 10^{-3}$, based on the theoretical curves. The confidence intervals were calculated as Wilson intervals [17] at 95% confidence level.

Fig. 8a shows the average successful tasks rate. The theoretical curves were calculated according to the simplified model (19), that prooved to be in perfect agreement with the obtained empirical results. The overall reconstruction success probability is shown in Fig. 8b. In this case, the simplified model is accurate only for higher probabilities and higher numbers of tasks. Hence, we used the full model whenever the computations were feasible. The impact of model accuracy is addressed in more detail in Appendix A. Again, the full model is in perfect agreement with the obtained empirical results.

2) Impact of Recompression: The goal of this experiment was to confirm the theoretical success bound under image recompression (11b). A selected image was protected (1 task, $\lambda = 2$) with quality $Q_1 = 87$. Then, the image was randomly tampered, and recompressed to $Q_2 = 90$. We measured the number of successful reconstruction attempts for increasing tampering rates. The experiment was repeated 600 times for each tampering rate; each time with a different security context. An additional step with recompression only yielded more

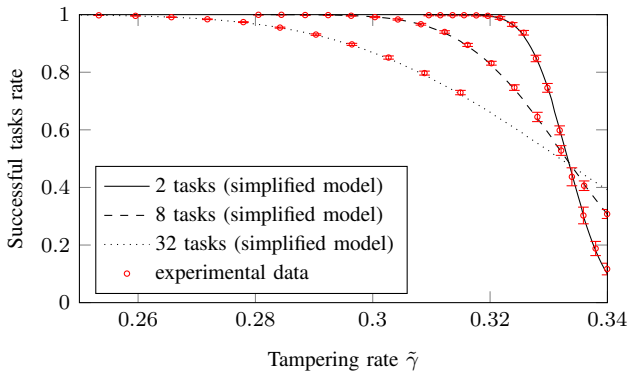


(a) average rate of true positives

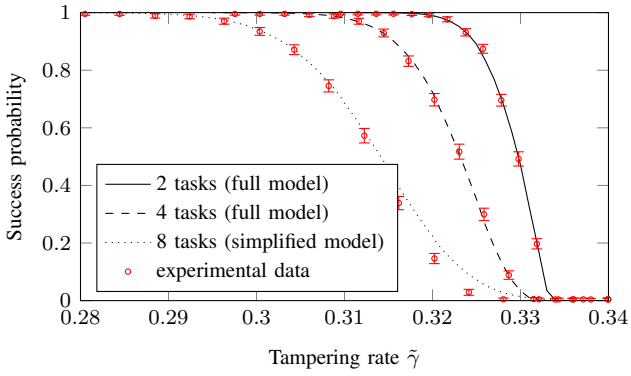


(b) average rate of correctly extracted symbols

Fig. 7: Impact of recompression on the authentication and watermark extraction performance ($\lambda = 1$).



(a) average successful tasks rate



(b) overall success probability

Fig. 8: Verification of the derived models of the overall success probability and successful tasks rate.

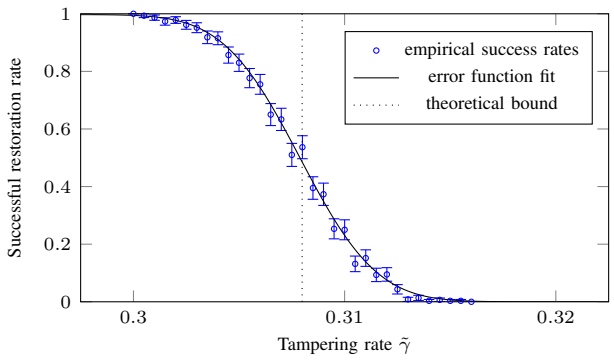


Fig. 9: Successful reconstruction attempts in the presence of recompression; 95% confidence intervals.

TABLE IV: Results of objective reconstruction quality assessment for 10,000 natural grayscale images.

	PSNR [dB] for $\lambda =$				SSIM for $\lambda =$			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Mean	27.8	30.0	31.7	33.2	0.75	0.83	0.88	0.91
Quantile 0.9	32.4	35.2	36.9	38.0	0.87	0.92	0.94	0.95
Quantile 0.1	23.5	25.4	27.0	28.6	0.62	0.73	0.81	0.86

accurate estimates of the applicable error rates: $f_p = 0.003$, and $p_e = 0.104$. Hence, from (11) the expected success bound is $\tilde{\gamma} = 0.308$. Fig. 9 shows the measured reconstruction success rate vs. the tampering rate.

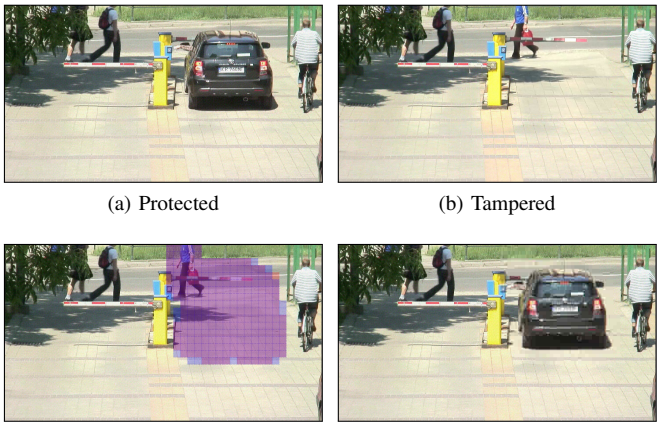
C. Image Quality

The embedding distortion changes with Q_1, since the JPEG quantization table indirectly impacts the embedding strength. Objective measurement of PSNR with respect to uncompressed PNG images yields values between $\approx$ 33 dB and 40 dB - increasing as Q_1 grows from 85 to 92 and then again for $Q_1 \geq 93$ (due to higher embedding strength). Objective measurement of the structural similarity index (SSIM) [18] reveals the same behavior with values from $\approx$ 0.87 to 0.96.

The reconstruction quality is controlled by the reference rate λ. Table IV shows the results of objective quality assessment on 10,000 grayscale natural images converted during protection to JPEG with quality $Q_1 = 90$. The distortion was measured using PSNR and SSIM with the original uncompressed image as a reference. Reconstruction quality reveals minor variations with Q_1, with peak difference of approx. 0.2 dB for $\lambda = 1$ and 0.6 dB for $\lambda = 4$.

Fig. 10 shows example images produced by the proposed scheme. The original $Q_1 = 90$ JPEG is shown in Fig. 10a. Fig. 10b shows the protected image, processed in the color reconstruction mode, i.e., with both luminance and chrominance channels watermarked. The PSNR of the protected image is 34.8 dB (SSIM = 0.85). Reconstruction results are shown in Fig. 10c-f for various reference rates λ.

Fig. 11 shows an example application of the proposed scheme to recover a removed car from a protected image. The



(a) Protected (b) Tampered

(c) Detected tampering maps (d) Restored, coarse quality

Fig. 11: Example application of the proposed scheme; color protection mode with coarse reconstruction quality.

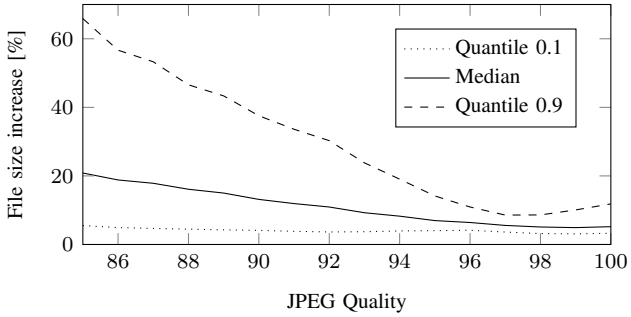


Fig. 12: Increase of the JPEG file size due to the proposed self-embedding scheme.

original image was protected in the color mode with quality $\lambda = 1$ to a $Q_1 = 90$ JPEG. The image was then tampered and recompressed to $Q_2 = 92$. The decoder successfully detects the tampering locations (Fig. 11c) for both the tampering (red) and erasure (blue) maps. The erasure map reveals a greater rate of damaged macro-blocks than the tampering map.

D. File Size Increase

Embedding the watermark increases the size of the JPEG files. The increase will negatively correlate with the amount of texture in the image. In this experiment we measured the size of the protected JPEGs for $Q_1 = \{85, \ldots, 100\}$, relative to unprotected JPEGs of the same quality. We tested 500 natural grayscale images. Fig. 12 shows the percentiles of the obtained distributions. The dotted line corresponds to the 10th percentile, the solid line to the median, and the dashed line to the 90th percentile. The observed increase tends to be higher for lower quality levels, typically reaching 20% for $Q_1 = 85$.

E. Processing Time

The goal of this experiment was to assess the protection time and confirm the expected impact of the mentioned task size selection strategies. We focused on the protection time due to its potential implementation on mobile devices. The experiments were performed with an optimized C++ implementation based on thread building blocks (TBB). A thread pool matching the number of cores was used to prevent preemption from the operating system, and Mersenne Twister was used for pseudo-random number generation. The following platforms were used for evaluation (all mobile devices were running Android Kit Kat - stock firmware for Note 8.0 and Z1 compact and CyanogenMod for S3):

- desktop PC with a Core 2 Duo E8500 processor (dual core, 3.16 GHz) running 64-bit Debian Wheezy;
- desktop PC with a Core i7 4771 processor (quad core, 3.50 GHz) running 64-bit Debian Jessie;
- Samsung Galaxy Note 8.0 with a Samsung Exynos 4412 processor (ARM Cortex-A9, quad-core, 1.6 GHz);
- Samsung Galaxy S3 with a Samsung Exynos 4412 processor (ARM Cortex-A9, quad core, 1.4 GHz);
- Sony Xperia Z1 compact with Qualcommm Snapdragon 800 processor (Krait 400, quad code, 2.15 GHz).

In the first part of the experiment we measured the time needed to protect ($\lambda = 2$) grayscale images of various sizes. In order to better illustrate the behavior of computational complexity, parallel computations were disabled. This part of the experiment was performed on the E8500-based PC. Fig. 13a shows the obtained results for a constant number of tasks. The top curve corresponds to a single task. The observed polynomial complexity leads to prohibitively long processing times for high-resolution images. The protection time of a 16 Mpx image reached 6 minutes. By increasing the number of tasks, we were able to decrease the time to 22 s and even to 7 s for 16 and 64 tasks, respectively.

Fig. 13b shows the results for constant task size. Just as expected, the watermark embedding time changes linearly with image resolution. Such an approach allows for balancing the computational complexity and the achievable success bounds. The use of 1 Mpx tasks delivers reasonable computation performance with success bound penalty between 1.4% and 3% of the image area (Table V). For 0.5 Mpx tasks, the success bound penalty varies between 2% and 4.4%, and for 0.25 Mpx tasks between 3.6% and 6.5%.

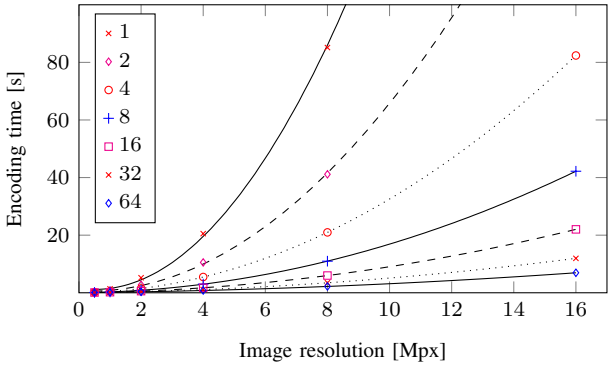
Further improvement of the processing performance can be obtained with parallel task execution. In order to evaluate practically achievable processing times, the second part of this experiment involves comprehensive evaluation on all considered platforms. In this experiment we used the color reconstruction mode and draw *uint32* primitives from the Mersenne Twister. The obtained results are collected in Table V.

Adoption of the proposed two-layer spreading mechanism allowed to reduce the protection time to the order of seconds, even on mobile devices. While more demanding configurations with higher reconstruction quality or bigger tasks still take too much time, many of them can already be computed efficiently. For contemporary smartphones input/output (I/O) operations constitute a major bottleneck, especially for larger images. Table VI shows an execution profile with the most time consuming operations for the Z1 compact smartphone. For the 16 Mpx image I/O operations take over 3 times as long as encoding. We expect that in a few years, mobile devices will be capable of efficiently handling the protection process for all relevant configurations.

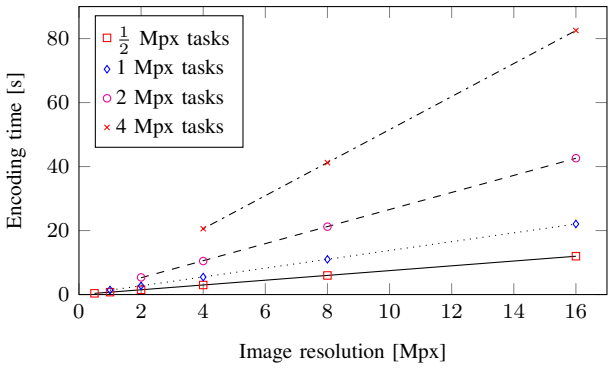
(a) Original (b) Protected, 34.8 dB (0.85) (c) Restored - Coarse, 24.9 dB (0.72)

(d) Restored - Low, 26.9 dB (0.81) (e) Restored - Medium, 28.6 dB (0.86) (f) Restored - High, 30.0 dB (0.89)

Fig. 10: Fragments of the original (a) and the protected image (b) and the corresponding restoration results (c-f) for the considered reconstruction quality levels in the color operation mode; SSIM scores in parentheses follow PSNR measurements.



(a) constant number of tasks



(b) constant task size

Fig. 13: Image encoding time on a common PC for grayscale reconstruction mode with disabled parallel computations.

TABLE VI: Execution time of the most time-consuming steps of the image protection procedure for a Sony Xperia Z1 compact smartphone; protection ($\lambda = 1$) of color images of various size with a constant task size of 1 Mpx.

Operation	**Average execution time [s] (4 repetitions)**				
	1 Mpx	2 Mpx	4 Mpx	8 Mpx	16 Mpx
Reading JPEG	0.062	0.071	0.139	0.308	0.749
Gen. of rec. ref.	0.067	0.070	0.113	0.117	0.124
Gen. of RLF code [1]	0.010	0.009	0.009	0.010	0.011
Encoding [1]	0.357	0.328	0.422	0.473	0.517
Embedding [1]	0.120	0.156	0.286	0.256	0.353
Writing JPEG	0.074	0.121	0.279	0.484	0.895

[1] - average for all tasks from both luminance and chrominance channels

F. Mitigating Partial Reconstruction Attempts

Successful recovery of all tasks is a restrictive requirement, which significantly reduces the achievable tampering rates. However, since the images are divided into tasks in a pseudo-random manner, prospective reconstruction failures are expected to yield missing blocks scattered over the whole image. Hence, even incomplete reconstruction might give enough information about the tampered area. Missing data can be approximated with a variety of error concealment, inpainting, or image completion techniques [19].

In this experiment we aimed to demonstrate the impact of inpainting on the achievable reconstruction fidelity. We protected a grayscale 4 Mpx image ($\lambda = 2$, $N_t = 8$). Then, we tampered the image by removing a car with its nearest surroundings. Finally, we performed content reconstruction with simulated failures in a given number of tasks. The missing blocks were concealed with an open-source implementation (from OpenCV) of a popular inpainting algorithm [20].

Fig. 14 shows corresponding restoration results before and after inpainting. While small image details cannot be recovered in this manner, the obtained reconstruction result is still meaningful, even for half of the tasks finishing with failure. Hence, for some applications it might be reasonable to define the tampering rate bound based on the expected successful tasks rate. Table VII collects the achievable reconstruction success bounds assuming the acceptable successful tasks rate is 0.75. By comparing the results with Table V, we can observe that the gap between the theoretical and achievable success bounds has shrunk to approximately 1% of the image area, compared to up to 6.5% for the conventional bound.

TABLE V: Reconstruction success bounds and encoding time in color protection mode for constant task size. Encoding time includes the necessary JPEG read and write operations.

Quality	Image size [Mpx]														
	1 Mpx tasks					0.5 Mpx tasks					0.25 Mpx tasks				
	1	**2**	**4**	**8**	**16**	**1**	**2**	**4**	**8**	**16**	**1**	**2**	**4**	**8**	**16**
Tampering rate bound [%] for error probability 10^{-3}															
$\lambda = 1$	50.0[1]	48.2	47.6	47.3	47.1	47.4	46.6	46.2	45.9	45.6	45.3	44.6	44.2	43.8	43.5
$\lambda = 2$	33.3[1]	31.6	31.1	30.8	30.6	30.9	30.2	29.8	29.5	29.3	29.0	28.4	28.0	27.7	27.4
$\lambda = 3$	25.0[1]	23.5	23.0	22.7	22.5	22.8	22.2	21.8	21.6	21.4	21.1	20.6	20.2	19.9	19.7
$\lambda = 4$	20.0[1]	18.6	18.2	17.9	17.7	18.0	17.4	17.1	16.9	16.7	16.4	15.9	15.6	15.4	15.2
Color image protection time [s] - quad core PC (i7-4771)															
$\lambda = 1$	0.40	0.45	0.57	0.94	1.65	0.19	0.27	0.41	0.66	1.28	0.13	0.19	0.31	0.54	1.06
$\lambda = 2$	0.67	0.73	0.94	1.70	3.28	0.26	0.32	0.58	1.09	1.85	0.17	0.24	0.40	0.71	1.33
$\lambda = 3$	0.95	1.01	1.33	2.89	4.51	0.33	0.46	0.77	1.48	2.47	0.20	0.29	0.53	0.91	1.64
$\lambda = 4$	1.21	1.30	1.56	3.34	5.83	0.41	0.55	0.89	1.83	2.97	0.23	0.34	0.62	1.04	1.93
Color image protection time [s] - dual core PC (E8500)															
$\lambda = 1$	0.53	0.63	1.23	2.40	4.79	0.22	0.43	0.83	1.60	3.19	0.18	0.33	0.63	1.21	2.42
$\lambda = 2$	0.94	1.05	2.06	4.05	8.09	0.33	0.65	1.27	2.52	4.92	0.24	0.45	0.86	1.66	3.47
$\lambda = 3$	1.34	1.46	2.89	5.70	11.40	0.44	0.86	1.69	3.37	6.60	0.30	0.56	1.12	2.16	4.45
$\lambda = 4$	1.75	1.88	3.72	7.37	14.74	0.56	1.07	2.11	4.20	8.71	0.35	0.67	1.31	2.66	5.48
Color image protection time [s] - Galaxy Note 8.0															
$\lambda = 1$	1.57	1.87	3.11	6.09	15.23	0.69	1.01	1.97	3.99	8.03	0.52	0.85	1.70	3.06	6.50
$\lambda = 2$	2.78	3.45	7.75	16.09	36.71	1.03	1.59	3.39	8.22	19.00	0.67	1.23	2.31	4.92	12.17
$\lambda = 3$	4.12	5.22	12.40	25.45	63.72	1.42	2.70	6.27	13.96	32.49	0.84	1.44	3.43	7.55	19.77
$\lambda = 4$	5.48	7.72	17.02	38.81	87.47	1.91	4.12	8.74	20.31	50.76	0.97	2.01	4.73	11.06	28.02
Color image protection time [s] - Galaxy S3															
$\lambda = 1$	2.24	2.72	4.34	9.07	17.80	0.99	1.56	2.95	5.63	11.35	0.73	1.31	2.56	5.72	10.50
$\lambda = 2$	3.97	4.73	9.86	20.41	45.62	1.50	2.31	4.92	10.55	26.38	0.90	1.70	3.44	8.14	17.24
$\lambda = 3$	5.77	7.66	15.44	36.42	73.22[2]	2.06	3.54	8.00	16.24	42.35	1.08	2.07	4.55	11.91	25.82
$\lambda = 4$	7.72	10.55	22.96	44.28	N/A	2.54	5.63	12.21	25.53	61.61	1.32	2.85	6.21	16.47	36.91
Color image protection time [s] - Xperia Z1 compact															
$\lambda = 1$	1.13	1.45	2.42	4.86	10.14	0.59	1.11	2.04	4.26	8.18	0.58	1.09	1.99	3.94	7.97
$\lambda = 2$	1.93	2.36	3.91	10.38	22.74	0.84	1.60	3.15	6.40	13.64	0.66	1.32	2.75	5.73	11.64
$\lambda = 3$	2.63	3.23	5.94	15.40	32.85	1.05	1.85	3.90	8.81	18.80	0.80	1.57	3.30	6.90	16.34
$\lambda = 4$	3.60	4.29	8.09	23.91	49.09	1.27	2.47	5.49	12.23	26.79	0.95	1.95	3.87	8.43	21.36

[1] - theoretical limit; [2] - occasional out of memory problem; N/A - out of memory problem.

TABLE VII: Success bounds for $\frac{1}{4}$ Mpx tasks with success criterion defined as the expected successful task rate of 0.75.

Quality	Image size [Mpx]				
	1	**2**	**4**	**8**	**16**
$\lambda = 1$	49.0	49.0	48.9	48.9	48.9
$\lambda = 2$	32.4	32.4	32.3	32.3	32.3
$\lambda = 3$	24.2	24.1	24.1	24.1	24.1
$\lambda = 4$	19.2	19.2	19.2	19.2	19.1

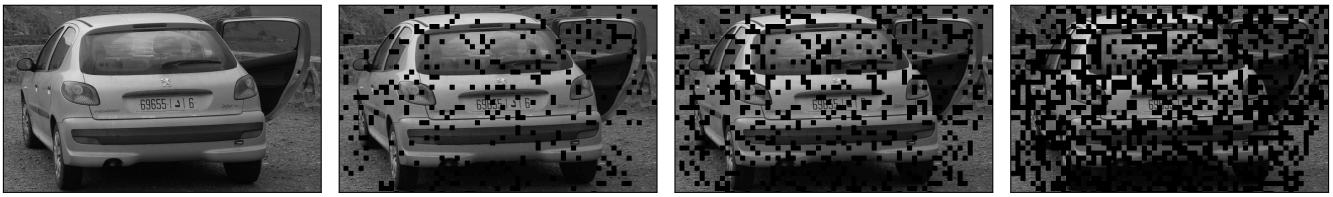
VI. Applicability and Limitations

The proposed scheme was tailored to the JPEG file format. It handles image recompression by considering the most likely distortions of the embedded watermark that occur when the image is recompressed to a different quality. However, these distortions are too big when the new compression level is higher. This incurs an applicability limit of $Q_2 \geq Q_1$. Similar limitation applies when the quantization matrix is changed to a completely different one. While such modifications can be easily detected with conventional methods, it might make the proposed scheme unsuitable for some applications.

The proposed scheme aggregates both chrominance channels into a single communication channel with embedding capacity matching the luminance component. This is possible if chrominance sub-sampling is disabled (4:4:4 mode). If it is enabled during recompression, the scheme will automatically fallback to use the luminance component only.

VII. Conclusions

The main goal of our work was to address the issues related to practical implementation of self-recovery. We extended a recently proposed model of the content reconstruction problem to take into account block classification and watermark extraction errors, inherent to lossy-compressed images. An important aspect of our study was related to handling high-resolution color images. We combined two known methods of spreading the reference information over the image. The obtained mechanism gives better control over the reconstruction performance and allows to balance its trade-offs. We managed to significantly reduce the image protection time, with minimal impact on the achievable success bounds. The protection time decreased to the order of seconds, even on mobile devices.



(a) Original image (b) Partial recovery (7/8) (c) Partial recovery (6/8) (d) Partial recovery (4/8)

(e) Full recovery, 30.4 dB (f) Concealed (7/8), 28.7 dB (g) Concealed (6/8), 27.1 dB (h) Concealed (4/8), 24.6 dB

Fig. 14: Concealment of partial recovery artifacts with inpainting for various task failure rates; 4 Mpx image; $\lambda = 2$.

TABLE VIII: Comparison of achievable reconstruction performance; experimental results obtained on 10,000 grayscale images from the BOWS2 data-set (500 for the file size evaluation).

Scheme	**Embedding distortion**	**File size increase**	**Reference quality under JPEG**			**Reference quality under random tampering**				**Success conditions**
			85	**90**	**95**	**5%**	**10%**	**20%**	**40%**	
Cheddad [9][1]	36.3 dB 0.90	≈16%	24.7 dB 0.61	25.4 dB 0.64	25.5 dB 0.65	23.3 dB 0.55	21.8 dB 0.48	19.5 dB 0.39	16.4 dB 0.28	Lossy compression, low pass filtering, noise, limited tampering
Wang [12]	35.3 dB 0.81	6 - 9 %	18.9 dB 0.44	21.35 dB 0.52	23.5 dB 0.62	20.46 dB 0.52	19.8 dB 0.51	18.5 dB 0.49	16.4 dB 0.45	Lossy JPEG compression, limited tampering
Proposed	33.0 - 39.4 dB 0.87 - 0.96	5 - 20%	27.8 dB 0.75	27.8 dB 0.75	27.8 dB 0.75	27.8 dB 0.75	27.8 dB 0.75	27.8 dB 0.75	27.8 dB 0.75	Recompression to higher quality, tampering up to 50% (grayscale)

[1] We changed the embedding method to QIM since it yields better results and gives more control over the embedding distortion

In order to verify the efficiency of the proposed scheme, we derived an improved theoretical model for calculating the reconstruction success probability. Based on the performed exhaustive experimental validation, we can conclude that it accurately represents the behavior of the adopted reference information spreading mechanism.

We have shown that it is possible to construct a self-embedding scheme which trades-off the success bounds in order to maintain the required reconstruction fidelity. This constitutes a paradigm shift compared to existing JPEG-compatible schemes. Table VIII summarizes the performance of the proposed ($\lambda = 1$) and two alternative schemes [9, 12]. The reported results were obtained experimentally on 10,000 grayscale images. While alternative schemes suffer from rapid quality deterioration with emerging tampering, the proposed scheme retains constant restoration fidelity. However, it typically incurs a larger overhead on the file size and does not allow for any post-processing of the protected images.

ACKNOWLEDGEMENTS

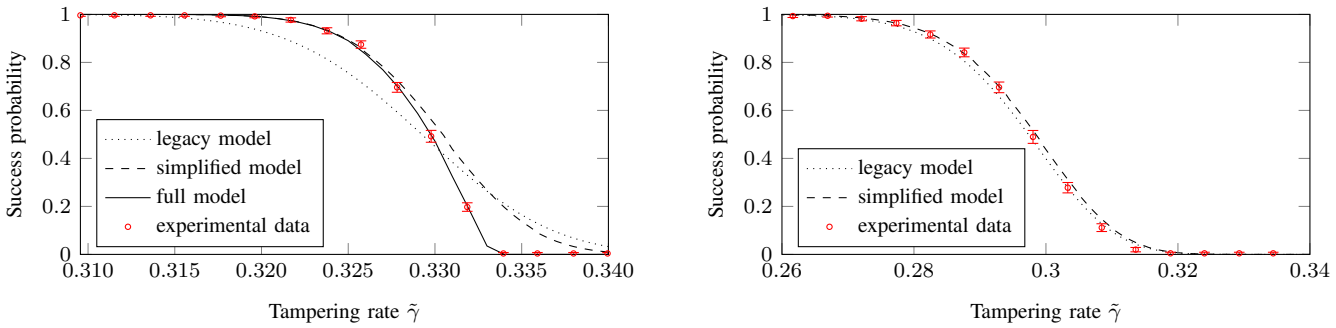
The authors would like to thank anonymous reviewers for insightful comments and also Hui Wang, Anthony T. S. Ho, and Xi Zhao for providing source code of their scheme.

APPENDIX

Due to insufficient accuracy of a previous theoretical model, in particular for low numbers of tasks, we derived a new model of the reconstruction success probability. While it is perfectly accurate, it is computationally infeasible if the number of tasks is greater than 6. We also derived a simplified model which can be efficiently computed. It overestimates the success probability near the theoretical limit for lower task counts, but always gives accurate values for the expected successful tasks rate (Fig. 8a). In this section, we aim to quantitatively assess the accuracy of all of the considered models.

The calculated theoretical curves and their corresponding empirical verification results for a 2 Mpx image are shown in Fig. 15. It is clearly visible that for low task count (e.g., 2 in Fig. 15a) both the legacy and the simplified models are inaccurate. For higher task counts (e.g., 16 in Fig. 15b) the results tend to converge to the actual behavior. For a 2 Mpx image divided into 2 tasks, the mean squared error (MSE) between the empirical data and theoretical estimates is $9 \cdot 10^{-3}$ for the legacy model, $5 \cdot 10^{-3}$ for the simplified model, and $5 \cdot 10^{-5}$ for the full model. For 16 tasks, the MSE is $21 \cdot 10^{-5}$ for the legacy model, and $13 \cdot 10^{-5}$ for the simplified model.

When comparing the tampering rate bounds, the obtained theoretical estimates are not very different. For small error rates the models deliver similar results. For instance, for $\lambda = 2$ and a 2 Mpx image both the full model and the simplified models yield a tampering rate bound of 0.316, compared to 0.309 according to the legacy model. This constitutes estimation error of approx. 2.2%. For higher number of tasks this error drops, reaching 1.35% for 4 tasks and 0.89% for 32 tasks. The highest error (3.15%) was observed for the lowest considered resolution and the lowest task count.



(a) 2 tasks

(b) 16 tasks

Fig. 15: Comparison of the accuracy of the considered overall success probability models for 2 and 16 tasks.

REFERENCES

[1] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proc. of IEEE Int. Conf. on Image Proc.*, 1999.

[2] P. Blythe and J. Fridrich, "Secure digital camera," in *Proc. of Digital Forensic Research Workshop*, 2004, pp. 1–12.

[3] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reference sharing mechanism for watermark self-embedding," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 485–495, 2011.

[4] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction," *IEEE Trans. Inf. Forens. Security*, vol. 6, no. 4, pp. 1223–1232, 2011.

[5] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 1134–1147, March 2013.

[6] P. Korus, J. Białas, and A. Dziech, "High-quality self-embedding for jpeg-compressed digital images," in *Proc. of European Signal Process. Conf. (EUSIPCO)*, Marrakech, 2013.

[7] J. Lee and C. S. Won, "Authentication and correction of digital watermarking images," *Electronics Letters*, vol. 35, no. 11, pp. 886–887, 1999.

[8] D. J. MacKay, "Fountain codes," *IEE Proceedings Communication*, vol. 152, no. 6, 2005.

[9] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," *Signal Process.*, vol. 89, pp. 2324–2332, December 2009.

[10] X. Zhu, A. T.S. Ho, and P. Marziliano, "A new semi fragile image watermarking with robust tampering restoration using irregular sampling," *Signal Process., Image Comm.*, vol. 22, no. 5, 2007.

[11] C. Y. Lin and S. F. Chang, "Sari: self-authentication-and-recovery image watermarking system," in *ACM Int. Conf. Multimedia*, 2001.

[12] H. Wang, A. T. S. Ho, and X. Zhao, "A novel fast self-restoration semi-fragile watermarking algorithm for image content authentication resistant to jpeg compression," in *Proc. of the 10th Int. Conf. Digital Forensics and Watermarking*, 2012, pp. 72–85.

[13] J.A Mendoza-Noriega, B.M. Kurkoski, M. Nakano-Miyatake, and H. Perez-Meana, "Halftoning-based self-embedding watermarking for image authentication and recovery," in *Proc. IEEE Int. Midwest Symposium on Circuits and Systems*, Aug 2010, pp. 612–615.

[14] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[15] R. P. Brent, S. Gao, and A. G. B. Lauder, "Random Krylov spaces over finite fields," *SIAM Journal on Discrete Mathematics*, vol. 16, no. 2, pp. 276–287, Feb. 2003.

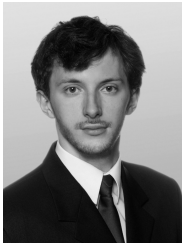
[16] "The dataset from the 2nd bows contest," http://bows2.ec-lille.fr/, 2007, Visited on 26 March 2012.

[17] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. pp. 209–212, 1927.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600 –612, april 2004.

[19] C. Guillemot and O. Le Meur, "Image inpainting : Overview and recent advances," *Signal Process. Mag., IEEE*, vol. 31, no. 1, pp. 127–144, 2014.

[20] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. of IEEE Conf. Computer Vision Patt. Recogn.*, 2001.



Paweł Korus received his M.Sc. and Ph.D. degrees in telecommunications (both with honors) from the AGH University of Science and Technology in 2008, and in 2013, respectively. He is currently an assistant professor at the Department of Telecommunications of the AGH University of Science and Technology, Krakow, Poland.

His research interests include multimedia security, digital watermarking, information hiding, and information coding techniques.

Jarosław Białas received his M.Sc. degree from the Department of Automatics of the AGH University of Science and Technology in 2011. He is currently a Ph.D. student at the Department of Telecommunications, AGH University of Science and Technology.

His research interests include signal processing, biometric systems and multimedia security. He has actively participated in two research projects INDECT and INSIGMA.

Andrzej Dziech holds the position of a full professor at the Department of Telecommunications of AGH University of Science and Technology in Krakow, Poland.

He received his M.Sc. and Ph.D. degrees from the Institute of Electrical Engineering in Saint Petersburg in 1970 and 1973, respectively, and the D.Sc. from Technical University of Poznan in 1978. He is an author of 6 books and nearly 180 publications. He was a supervisor of 18 Ph.D. students.

His fields of interest are related to digital communication, image and data processing, data compression, information and coding theory, random signals, computer communications networks and signal processing. He was awarded 4 times for research achievements by the Ministry of Education of Poland. Professor Dziech actively participated in numerous international research projects, e.g., Tempus, Knixmas, Calibrate. Currently, he is coordinating a European Union FP7 integrated project INDECT.