Matthias Kirchner
Dipak Ghosal (Eds.)

LNCS 7692

Information Hiding

14th International Conference, IH 2012
Berkeley, CA, USA, May 2012
Revised Selected Papers

Springer

Lecture Notes in Computer Science 7692

Matthias Kirchner Dipak Ghosal (Eds.)

Information Hiding

14th International Conference, IH 2012
Berkeley, CA, USA, May 15-18, 2012
Revised Selected Papers

Springer

Volume Editors

Matthias Kirchner
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704, USA
E-mail: kirchner@icsi.berkeley.edu

Dipak Ghosal
University of California Davis
Department of Computer Science
Davis, CA 95616, USA
E-mail: dghosal@ucdavis.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-36372-6 e-ISBN 978-3-642-36373-3
DOI 10.1007/978-3-642-36373-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013931274

CR Subject Classification (1998): E.3, K.6.5, D.4.6, E.4, H.5.1, I.4

LNCS Sublibrary: SL 4 – Security and Cryptology

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 14th edition of the Information Hiding (IH) Conference, since the inaugural 1996 workshop in Cambridge, UK, was held in Berkeley, CA, during May 15–18, 2012. With conference locations alternating between Europe and North America, it was our particular pleasure to host IH in California again after the 2008 meeting in Santa Barbara and stops in Darmstadt (2009), Calgary (2010), and Prague (2011).

IH 2012 once more attracted researchers from many different areas of information hiding, including steganography and steganalysis, digital watermarking, fingerprinting codes, anonymity and privacy, covert channels, multimedia forensics and counter-forensics, as well as theoretical aspects of information hiding and detection. Since its inception, the conference series has been a premier forum for publishing research in these areas. This year, the Program Committee reviewed 40 papers using a double-blind system, with at least three reviewers assigned to each paper. A discussion phase helped to reach consensus in controversial cases. In the end, 18 papers were accepted for presentation at the conference. This volume contains the revised versions of all accepted papers, incorporating the comments from members of the Program Committee. Shepherds were assigned to three of the papers to advise the authors and to ensure high-quality proceedings.

Two invited lectures completed the technical program of the conference. Hany Farid (Dartmouth College) presented his recent work on measuring the strength and impact of photo retouching. Venkat Anantharam (UC Berkeley) gave an overview of information-theoretic methods in information hiding.

We would like to the thank all those who contributed to the organization of a successful and interesting conference. Rennie Archibald and Diana Böhme ensured a smooth running of the conference, Tomas Filler was always available for valuable advice, ICSI Berkeley and UC Davis lent organizational support. The conference would also not have been possible without the generous financial backing by Technicolor and Civolution, as well as the organizers of IH 2011 in Prague. Their support was particularly welcomed in times of corporate sponsorship budget cuts.

Finally, we are indebted to all external reviewers and shepherds for voluntarily investing time and thoughts, as much as we thank all authors, presenters, and attendees for their support of the conference.

August 2012

Matthias Kirchner
Dipak Ghosal

Organization

14th Information Hiding Conference	May 15–18, 2012, Berkeley, CA

Executive Committee

General Chair	Dipak Ghosal (University of California Davis, USA)
Program Chair	Matthias Kirchner (ICSI Berkeley, USA)

Program Committee

Ross Anderson	University of Cambridge, UK
Mauro Barni	University of Siena, Italy
Patrick Bas	CNRS, France
Rainer Böhme	University of Münster, Germany
François Cayre	GIPSA-lab/Grenoble INP, France
Ee-Chien Chang	Naional University of Singapore
Christian Collberg	University of Arizona, USA
Ingemar Cox	University College London, UK
Scott Craver	SUNY Binghamton, USA
George Danezis	Microsoft Research, UK
Gwenaël Doërr	Technicolor, France
Tomáš Filler	Digimarc Corp., USA
Jessica Fridrich	SUNY Binghamton, USA
Neil Johnson	Booz Allen Hamilton and JJTC, USA
Stefan Katzenbeisser	TU Darmstadt, Germany
Andrew Ker	University of Oxford, UK
Darko Kirovski	Microsoft Research, USA
John McHugh	RedJack LLC and University of North Carolina, USA
Ira Moskowitz	Naval Research Lab, USA
Tomáš Pevný	Czech Technical University, Czech Republic
Ahmad-Reza Sadeghi	TU Darmstadt, Germany
Rei Safavi-Naini	University of Calgary, Canada
Berry Schoenmakers	TU Eindhoven, The Netherlands
Kaushal Solanki	Eyenuk LLC, USA
Kenneth Sullivan	Mayachitra Inc., USA
Paul Syverson	Naval Research Lab, USA

Organizing Team

Local Organizers	Rennie Archibald, Tracy Liu
External Advice	Tomáš Filler
Volunteers	Diana Böhme

External Reviewers

Frederik Armknecht
Paul Cotae
Matthew Edman
Marco Fontani
Teddy Furon
Miroslav Goljan
Dominik Heider
Vojtěch Holub
Jan Kodovský
Minoru Kuribayashi
Shiyue Lai
Hoi Le
Shujun Li
Jasvir Nagra
Angela Piper
Scott Russell
Pascal Schöttle
Nicholas Sheppard
Shankar Shivappa
Haya Shulman
Boris Škorić
Christian Wachsmann
Sascha Zmudzinski

Shepherds

Jessica Fridrich	SUNY Binghamton, USA
Matthias Kirchner	ICSI Berkeley, USA
Paul Syverson	Naval Research Lab, USA

Sponsoring Institutions

Technicolor R&D France Snc
Civolution

Local Support

ICSI Berkeley
UC Davis

Table of Contents

Multimedia Forensics and Counter-Forensics

Blind Median Filtering Detection Using Statistics in Difference Domain 1
Chenglong Chen, Jiangqun Ni, Rongbin Huang, and Jiwu Huang

Robustness of Color Interpolation Identification against Anti-forensic Operations 16
Wei-Hong Chuang and Min Wu

Steganalyis

Steganalysis of LSB Replacement Using Parity-Aware Features 31
Jessica Fridrich and Jan Kodovský

Statistical Detection of LSB Matching Using Hypothesis Testing Theory 46
Rémi Cogranne, Cathel Zitzmann, Florent Retraint, Igor Nikiforov, Lionel Fillatre, and Philippe Cornu

Textural Features for Steganalysis 63
Yun Q. Shi, Patchara Sutthiwan, and Licong Chen

JPEG-Compatibility Steganalysis Using Block-Histogram of Recompression Artifacts 78
Jan Kodovský and Jessica Fridrich

Data Hiding in Unusual Content

Hiding a Second Appearance in a Physical Relief Surface 94
Yi-Liu Chao and Daniel G. Aliaga

Blind Robust Watermarking Mechanism Based on Maxima Curvature of 3D Motion Data 110
Ling Du, Xiaochun Cao, Muhua Zhang, and Huazhu Fu

Steganography

A Game-Theoretic Approach to Content-Adaptive Steganography 125
Pascal Schöttle and Rainer Böhme

Key-Efficient Steganography 142
Aggelos Kiayias, Alexander Russell, and Narasimha Shashidhar

Covert Channels

Secret Agent Radio: Covert Communication through Dirty Constellations ... 160
Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker

Anonymity and Privacy

Lower Bounds for Private Broadcast Encryption ... 176
Aggelos Kiayias and Katerina Samari

The Dangers of Composing Anonymous Channels ... 191
George Danezis and Emilia Käsper

Watermarking

A New Measure of Watermarking Security Applied on QIM ... 207
Teddy Furon and Patrick Bas

Non-Integer Expansion Embedding for Prediction-Based Reversible Watermarking ... 224
Shijun Xiang

Walsh-Hadamard Transform in the Homomorphic Encrypted Domain and Its Application in Image Watermarking ... 240
Peijia Zheng and Jiwu Huang

Fingerprinting

Asymptotic Fingerprinting Capacity in the Combined Digit Model ... 255
Dion Boesten and Boris Škorić

Bias Equalizer for Binary Probabilistic Fingerprinting Codes ... 269
Minoru Kuribayashi

Author Index ... 285

Blind Median Filtering Detection Using Statistics in Difference Domain

Chenglong Chen[1], Jiangqun Ni[1,⋆], Rongbin Huang[2], and Jiwu Huang[1]

[1]School of Information Science and Technology, Sun Yat-Sen University
Guangzhou, China
[2]School of Information and Telecommunication Engineering, BUPT
Beijing, China
{c.chenglong,huangrongbin19}@gmail.com,
{issjqni,isshjw}@mail.sysu.edu.cn

Abstract. Recently, the median filtering (MF) detector as a forensic tool for the recovery of images' processing history has attracted wide interest. In this paper, we focus on two topics: 1) an analysis of the statistics in the difference domain of median filtered images; 2) a new approach based on the statistical characterization in difference domain to overcome the shortages of the prior related works. Specifically, we derive the cumulative distribution function (CDF) of first order differences based on simplifying assumptions, and also study the behavior of adjacent difference pairs in the difference domain for original non-filtered images, median filtered images and average filtered images. We then present a new MF detection scheme based on the statistics in the difference domain of images. Extensive simulations are carried out, which demonstrates that the proposed MF detection scheme is effective and reliable for both uncompressed and JPEG post-compressed images, even in the case of low resolution and strong JPEG compression.

Keywords: Median Filtering, Digital Image Forensics, Difference Domain.

1 Introduction

Exposing the processing history of a digital image is an important objective for forensic analysis. In order to determine if an image has undergone any form of manipulation, the possible use of a wide variety of operations must be tested for. Existing image forensic works involve the detection of median filtering (MF) [1] [2] [3], resampling [4], JPEG compression [5], amongst others.

This work concentrates on the median filter, a widely used and well-known nonlinear denoising operator. Due to its non-linearity and complex statistical properties, also counter-forensic techniques show special interest in this operation [6] [7]. Therefore, the median filtering detector becomes an important forensic tool for the recovery of the processing history of an image, or for exposing possible counter-forensic operations.

[⋆] Corresponding author.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 1–15, 2013.

Several prior related schemes for the detection of MF in digital images have been presented so far [1] [2] [3]. In [1], streaking artifacts and subtractive pixel adjacency matrix (SPAM) features are employed to detect MF in bitmap and JPEG post-compressed images, respectively. In [2], the probability of zero values in the first-order difference image in textured regions is considered as statistical fingerprint to detect the MF operation. Recently, Yuan [3] presented the *median filtering forensics* (MFF) feature sets based on order statistics for the forensic analysis of MF.

In this paper, we provide a new approach for reliable MF detection in digital images based on the statistical characterization of digital signals in the difference domain. Making a number of simplifying assumptions, we carry out a theoretical analysis of the statistical behavior in the difference domain for different image sources, such as original non-filtered images, median filtered images and average filtered images. With these results, we then analytically characterize the statistical artifacts in the difference domain of different image sources. Finally, we introduce two new feature sets and describe our new scheme for MF detection in digital images. The effectiveness of the proposed scheme is extensively evaluated with a composite image database of 9,000 images.

The rest of this paper is organized as follows. The theoretical analysis of statistics in the difference domain for different image sources is given in Section 2. Based on the analysis, two new feature sets in the difference domain are introduced in Section 3. Section 4 details our experimental methodology and presents experimental results, including a comparison with previous art. Finally the conclusion is drawn in Section 5.

2 Statistical Characterization in the Difference Domain

In this section, we carry out an analysis of the statistical behavior of different image sources in the difference domain to clarify the motivation of feature sets construction for MF detection. First, we derive the analytic cumulative distribution function (CDF) for different image sources in first order difference under some simplifying assumptions. Second, we study the behavior of adjacent difference pairs in the difference domain. Our analysis is supported by strong evidence from extensive experiments with natural images.

2.1 Analysis of Median Filtering

The median filter is a well-known non-linear filter based on order statistics. Given a $H \times W$ grayscale image $X_{n,m}$ with $(n,m) \in \{1,2,\ldots,H\} \times \{1,2,\ldots,W\}$, a 2-D median filter is defined as

$$\hat{X}_{n,m} = \text{median}\{X_{n',m'} : (n',m') \in W(n,m)\}, \tag{1}$$

where $\hat{X}_{n,m}$ is the output of the median filter and $W(n,m)$ is the 2-D filter window centered at image coordinates (n,m). Throughout the rest of this paper, we concentrate on filter with square windows of odd size without loss of generality.

It is noted that neighboring pixels in median filtered images are correlated to some extent because they originate from overlapping windows of the original signal. With the inherent nature of the median filter, it is expected that pixels in median filtered images are correlated to their neighbors in a specific way. We study this behavior in terms of the k-th order difference of the filtered image, which is defined as

$$\varDelta_k^{(p,q)}(n,m) = \varDelta_{k-1}^{(p,q)}(n,m) - \varDelta_{k-1}^{(p,q)}(n+p, m+q), \tag{2}$$

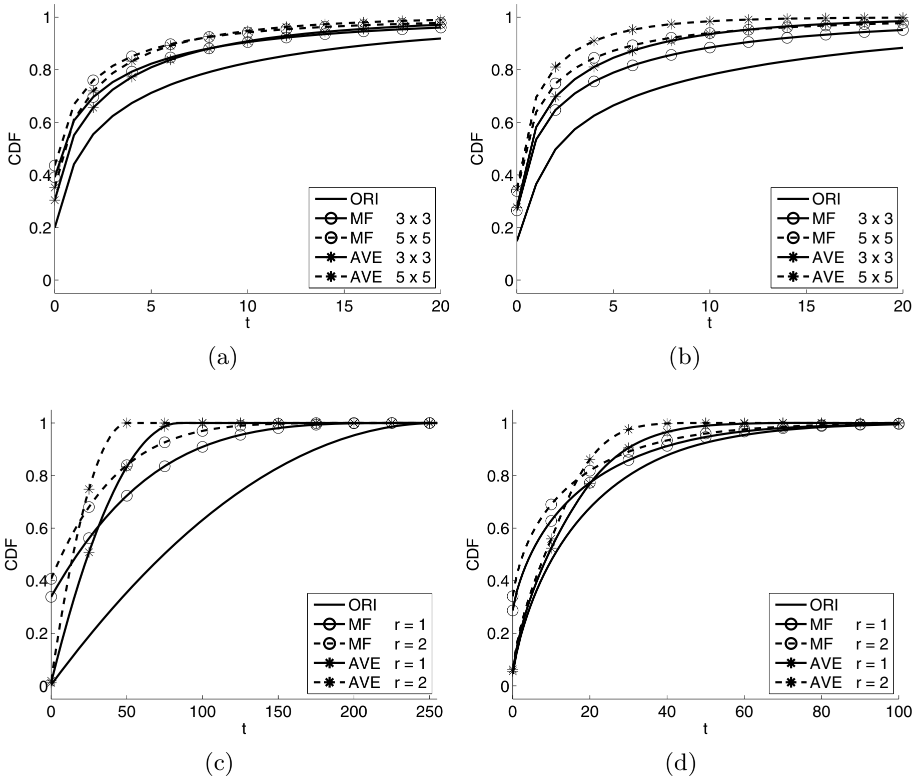
where $\varDelta_0^{(p,q)}(n,m) = \hat{X}_{n,m}$ and $(p,q) \in \{-1,0,1\}^2$ $(|p|+|q| \neq 0)$.

To give a preliminary look at this problem, Figs. 1(a) and (b) depict empirical CDFs of $|\boldsymbol{\varDelta}_1^{(0,1)}|$ and $|\boldsymbol{\varDelta}_2^{(0,1)}|$, respectively, where $|\bullet|$ is the operation to calculate the absolute value. These two CDFs are estimated using 3,000 8-bit grayscale natural images from the BOWS2 database [12] with 2-D windows. Apparently, the CDF curves of median filtered images (MF) are substantially different from the ones of original (ORI) and average filtered images (AVE). To get a further insight, we carry out an analysis on the CDF of first order difference and the behavior of adjacent difference pairs in the difference domain in Section 2.2 and Section 2.3, respectively.

2.2 Cumulative Distribution Function of First Order Differences

Overlapping windows and highly non-linear behavior make the theoretical analysis of two-dimensional median filtering a complex and cumbersome task. Following prior art [1] [2] [8], we thus make two simplifying assumptions. More specifically, we only consider one-dimensional filters and assume independent identically distributed (i.i.d.) uniform input signals, i.e., $\boldsymbol{X} \sim U[0, N-1]$.

For the sake of a focused presentation, the detailed derivations of the CDFs of $\boldsymbol{\varDelta}_1$ and $|\boldsymbol{\varDelta}_1|$, i.e. $F_1(r,t)$ and $F_{|1|}(r,t)$, respectively, for different image sources are carried out in the Appendix. Here, r relates to the filter window size w, $w = 2r+1$ (for original non-filtered image, we take $r=0$) and t is integer. Even under the given simplifying assumptions, the statistical characterization in the difference domain is already quite cumbersome. We also note that the results certainly do not yield a complete description of the problem. However, they are strongly indicative. To demonstrate this, we plot the analytical curves of $F_{|1|}(r,t)$ in Fig. 1(c), and also report the empirical curves of $F_{|1|}^{(0,1)}(r,t)$, estimated from natural images in the BOWS2 database using 1-D horizontal windows, in Fig. 1(d). Specifically, Fig. 1(d) is obtained by excluding untextured or smooth pixels satisfying $\sigma < \tau$, where σ is the standard deviation of local surrounding pixels in a square region of size $d \times d$. Note that the curves in Fig. 1(d) do not exactly match those in Fig. 1(c), however, similar effects can still be observed. Therefore, although our theoretical results are obtained under some simplifying assumptions, we believe that it still make sense to use them to investigate statistical artifacts in the difference domain of different image sources.



(a)

(b)

(c)

(d)

Fig. 1. (a) Empirical CDFs of $|\Delta_1^{(0,1)}|$ from BOWS2 with 2-D window, (b) Empirical CDFs of $|\Delta_2^{(0,1)}|$ from BOWS2 with 2-D window, (c) Analytical curves of $F_{|1|}(r,t)$ for i.i.d. uniform input, (d) Empirical curves of $F_{|1|}^{(0,1)}(r,t)$ from BOWS2 with 1-D window, excluding untextured pixels with $d = 7$ and $\tau = 20$.

Filtered Signal vs. Original Non-filtered Signal. The theoretical curves of $F_{|1|}(r,t)$ in Fig. 1(c) demonstrate the principal effect of the considered filters that, for median or average filtered signals, $|\boldsymbol{\Delta}_1|$ tends to take small values. Note that smaller and larger values in the first order difference correspond to the low and high frequency components in spatial domain, respectively. In other words, some of the high frequency components in filtered images are removed, which is indeed the case when we consider the low pass property of such filters. Although the median filter, in a strict sense, is not a low pass filter, it has been observed to have low-pass effect to some extent [8].

Median Filtered Signal vs. Average Filtered Signal. Fig. 1(c) also suggests a clear distinction between the difference domain CDFs of median filtered and average filtered signals. For median filtered images and small t, e.g. $t \leq 10$, $F_{|1|}(r,t)$ is much larger than that for original and average filtered images. This is related to the inherent effect of the median filter known as *streaking artifacts* [9].

Streaking means that the median filter tends to produce regions of constant or nearly constant intensities, which leads to the large probability in first order difference for small t.

It is also observed in Fig. 1(c) that the curves of median filtered images rise slowly while the curves of average filtered images rise rapidly with increasing t. This indicates that, compared to average filtered images, median filtered images retain more high frequency components, such as image edges. This effect is also related to another inherent nature of median filter known as *good edge preservation* [10]. It is noted that the median filter preserve edges better than the average filter, owing to its statistical and robustness properties [8] [10].

In general, also the CDFs of higher-order differences ($k > 1$) vary considerably between different image sources (cf. Fig. 1(b)). Therefore, we will exploit such statistics as fingerprint for MF detection in Section 3.1 for the construction of our global probability feature set (GPF).

2.3 The Behavior of Adjacent Difference Pairs

As shown in Fig. 1(b), the CDFs of $|\boldsymbol{\Delta}_2^{(0,1)}|$ also indicate that, the correlation between adjacent difference pair $(\Delta_1^{(0,1)}(n,m), \Delta_1^{(0,1)}(n,m+1))$ varies between different image sources. We investigate this further by adopting the joint probability $\Pr(m, m+l)$ of adjacent difference pairs $(\Delta_1^{(0,1)}(n,m), \Delta_1^{(0,1)}(n,m+l))$ as another metric besides the CDF of $|\boldsymbol{\Delta}_2^{(0,1)}|$. Fig. 2 shows the joint probability for different image sources, which are estimated using the same images from the BOWS2 database as above with 1-D horizontal windows of width $w = 3$. In these figures, the x axis represents $\Delta_1^{(0,1)}(n,m)$, and y axis represents, from left to right, adjacent difference $\Delta_1^{(0,1)}(n,m+l)$ with $l \in \{1,2,3\}$, respectively. Intensity values at (x,y) represent the probabilities at the logarithmic scale, whereas darker points refer to larger probabilities.

As shown in Fig. 2, $\Pr(m, m+l)$ varies considerably between different image sources for specific adjacent pairs. More specifically, both median filter and average filter employ sliding windows across the whole image. This introduces local correlations to the filtered image with respect to the position where the window is centered. This also holds for the difference domain of filtered images. Fig. 2 illustrates that the distribution of adjacent difference pairs from filtered images differs from original images, but also differs between median and average filtered images, respectively. For instance, adjacent difference pairs $(\Delta_1^{(0,1)}(n,m), \Delta_1^{(0,1)}(n,m+1))$ tend to cluster in the first and third quadrant with high probability, which becomes most evident for median filtered images. Since difference pairs with large value $(|\Delta_1^{(0,1)}(n,m)|, |\Delta_1^{(0,1)}(n,m+1)|)$ in the second and fourth quadrants are mainly related to impulse noise, the distribution $\Pr(m, m+1)$ for median filtered images can be explained by the effectiveness of the median filter to remove such noise [8]. Similar effects are also observed for high-order differences and two dimensions, indicating that the correlations between different adjacent difference pairs can be used as fingerprint for MF

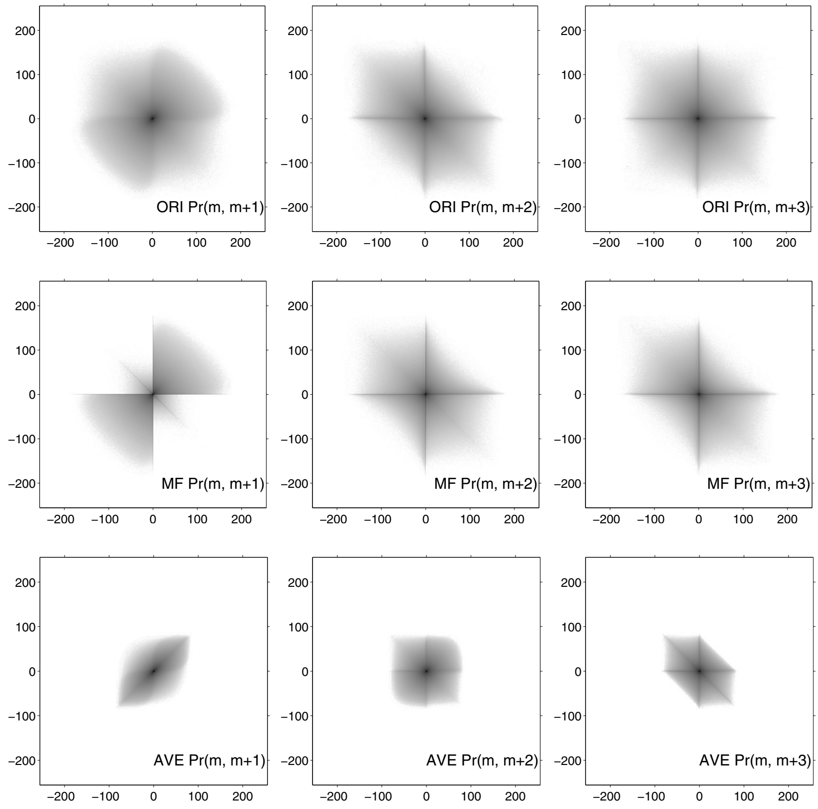


Fig. 2. Distribution of different adjacent difference pairs for different image sources

detection. We will exploit this observation in Section 3.2, where we construct a corresponding local correlation feature set (LCF).

Different characteristics of $\Pr(m, m+1)$ between different image sources, to some extent, also explain the feasibility of SPAM features for the forensic analysis of MF [1]. Although originally introduced for steganalysis, SPAM features are also very useful to analyze the conditional joint distribution of first-order differences [11]. Therefore SPAM effectively captures the correlation of adjacent difference pairs in the first order difference domain and achieves high performance for MF detection [1].

3 New Feature Sets in the Difference Domain

The analysis in Section 2 revealed that median filtered images inevitably exhibit distinctive statistical artifacts in the difference domain. This section presents feature sets in the difference domain to capture such artifacts in two different ways.

3.1 Global Probability Feature Set (GPF)

As analyzed in Section 2.2, $F_{|k|}(r,t)$ $(k \geq 1)$ varies considerably between different image sources, indicating a valuable resource for MF detection. With the k-th order difference array $\boldsymbol{\Delta}_k^{(p,q)}$, the estimated $F_{|k|}^{(p,q)}(t)$ (as we don't know the size of the filter window for a given image, so we drop the r term) is computed by

$$F_{|k|}^{(p,q)}(t) = \frac{\sum_{n=1}^{H_k}\sum_{m=1}^{W_k}\delta(t, |\Delta_k^{(p,q)}(n,m)|)}{H_k W_k}, \tag{3}$$

where $H_k = H - |p|k$ and $W_k = W - |q|k$ is the height and width of $\boldsymbol{\Delta}_k^{(p,q)}$ $(k \geq 1)$, respectively, and $\delta(x,y)$ is defined in the Appendix as Eqn (A.12). Confining $0 \leq t \leq T$ $(T \in \mathbb{Z})$, we define feature $\boldsymbol{P}_k^{(p,q)}$ as

$$\boldsymbol{P}_k^{(p,q)} = \{\, F_{|k|}^{(p,q)}(t) \mid 0 \leq t \leq T \,\}. \tag{4}$$

To reduce the feature dimensionality, we adopt the assumption in [11] that spatial statistics in natural images are symmetric with respect to mirroring and flipping. Thus, we separately average matrices with lags $|p| + |q| = i$ $(i = 1, 2)$ to form the final feature $\boldsymbol{P}_k$

$$\boldsymbol{P}_k^i = \frac{1}{4}\sum_{|p|+|q|=i} \boldsymbol{P}_k^{(p,q)}, \tag{5}$$

$$\boldsymbol{P}_k = [\boldsymbol{P}_k^1, \boldsymbol{P}_k^2]. \tag{6}$$

Concatenating all the $\boldsymbol{P}_k (k = 1, 2, \ldots, K)$ leads to a $2(T+1) \times K$-D global probability feature set (GPF)

$$\boldsymbol{F}_{\text{GPF}} = [\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_K]. \tag{7}$$

3.2 Local Correlation Feature Set (LCF)

As discussed in Section 2.3, the local correlations between different adjacent difference pairs can be modeled by the SPAM features as the transition probability of a higher-order Markov chain. Different from the transition probability measurement in SPAM, we construct our local correlation feature set using the normalized cross correlation (NCC). For random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, the NCC coefficient is defined as

$$\gamma = \frac{\text{cov}(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{\text{cov}(\boldsymbol{x}, \boldsymbol{x})\text{cov}(\boldsymbol{y}, \boldsymbol{y})}}, \tag{8}$$

where $\text{cov}(\boldsymbol{x}, \boldsymbol{y}) = E[(\boldsymbol{x} - E[\boldsymbol{x}])(\boldsymbol{y} - E[\boldsymbol{y}])]$ is the covariance and $E[\bullet]$ is the expectation operator. To deal with the denominator of Eqn (8) being zero, we simply define $\gamma = 0$ in this case. For discrete sample vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ with L points, $\text{cov}(\boldsymbol{x}, \boldsymbol{y})$ can be estimated by

$$\text{cov}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{L}\sum_{i=0}^{L-1}(x_i - \bar{\boldsymbol{x}})(y_i - \bar{\boldsymbol{y}}), \tag{9}$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ denote the arithmetic mean of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

To construct the local correlation feature on the 2-D difference array $\boldsymbol{\Delta}_k^{(p,q)}$, we first scan it with a 1-D window of width B along the same direction as it is computed and rearrange all B pixels of the i-th window into the i-th column of a matrix $\boldsymbol{Y}_k^{(p,q)}$. The scanning method is determined by the size of images, specially, if $H \times W > 64 \times 64$, *non-overlapping* scan is adopted, otherwise *overlapping* scan is preferred. We then eliminate the columns with all values zero and combine matrices $\boldsymbol{Y}_k^{(p,q)}$ with lags $|p| + |q| = 1$ to obtain $\boldsymbol{Y}_k^1$, i.e.,

$$\boldsymbol{Y}_k^1 = [\boldsymbol{Y}_k^{(0,1)}, \boldsymbol{Y}_k^{(0,-1)}, \boldsymbol{Y}_k^{(1,0)}, \boldsymbol{Y}_k^{(-1,0)}]^T. \tag{10}$$

Treating each column of $\boldsymbol{Y}_k^1$ as a variable which is related to the position in the filter window, we compute the NCC coefficient $\gamma_k^1(n, m)$ of the n-th and m-th column of $\boldsymbol{Y}_k^1$ $(n > m)$. The result is a feature vector $\boldsymbol{C}_k^1$,

$$\boldsymbol{C}_k^1 = \{\, \gamma_k^1(n, m) \mid n, m \in [1, \ldots, B], n > m \,\}. \tag{11}$$

Another feature vector $\boldsymbol{C}_k^2$ is computed from $\boldsymbol{Y}_k^{(p,q)}$ with lags $|p| + |q| = 2$ in a similar manner, resulting in the final LCF feature vector $\boldsymbol{C}_k$ for the k-th order difference arrays. That is

$$\boldsymbol{C}_k = [\boldsymbol{C}_k^1, \boldsymbol{C}_k^2]. \tag{12}$$

Concatenating all the $\boldsymbol{C}_k$ $(k = 1, 2, \ldots, K)$ together leads to a $(B^2 - B) \times K$-D local correlation feature set (LCF)

$$\boldsymbol{F}_{\mathrm{LCF}} = [\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_K]. \tag{13}$$

Combining GPF and LCF features, we obtain the final feature set (GLF) with $K[2(T + 1) + (B^2 - B)]$ elements for MF detection,

$$\boldsymbol{F}_{\mathrm{GLF}} = [\boldsymbol{F}_{\mathrm{GPF}}, \boldsymbol{F}_{\mathrm{LCF}}]. \tag{14}$$

While GPF relates to the estimated CDF in the difference domain (first-order statistics), LCF captures the correlation of adjacent difference pairs (second-order statistics). Thus, these two feature sets capture to some extent the statistical artifacts introduced by the median filter and other filters in a complementary way. As for practical implementation, the parameters of our feature sets, i.e., $\{T, B, K\}$, should be properly determined to give good detection capability with manageable computational complexity. According to our experimental study, $\{T, B, K\} = \{10, 3, 2\}$ is a reasonably default choice, which leads to 44 GPF features, 12 LCF features and 56 GLF features in total.

4 Experimental Study

In this section, we first describe the experimental methodology used in the experiments. Then, we compare our GLF-based MF detector to prior art, using both uncompressed images and JPEG post-compressed images, respectively.

4.1 Experimental Methodology

Employing a test setup similar to Yuan [3], this paragraph describes the experimental methodology to verify the effectiveness of our MF detector.

Image Database. Our experiments use the following three image databases:

- 3,000 images from the BOWS2 database. This database was used for the BOWS2 contest [12], and contains downsampled and cropped natural grayscale images of fixed size 512 × 512.
- 3,000 images from the NRCS Photo Gallery. This database is provided by the Department of Agriculture, United States [13], and includes scanned images from a variety of film and paper sources.
- 3,000 images from the Dresden Image Database (DID). This database is a collection of more than 14,000 images from 73 different digital cameras [14].

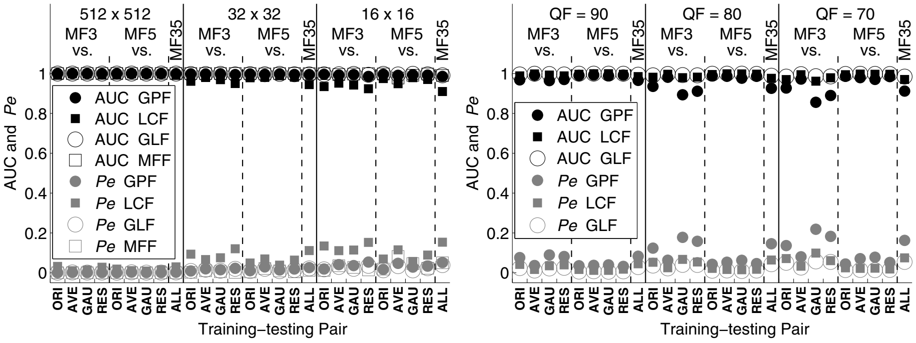
This results in a composite database of 9,000 images. Where necessary, images were converted to 8-bit grayscale. Moreover, only the center 512 × 512 part of the images from the NRCS and DID databases was used.

Training-Testing Pairs. Based on the above image database, we prepare 9 training-testing pairs as follows:

1. Process all images in the original database D^{ORI} to obtain 5 image sets, i.e., D^{MF3}, D^{MF5}, D^{AVE}, D^{GAU} and D^{RES}, which are generated with a 3×3 and a 5×5 median filter, an average filter with the filter window randomly set to 3×3 or 5×5, a Gaussian low-pass filter with σ randomly set to 0.5 or 0.8, and a rescale operation that is randomly composed of nearest or bilinear interpolation and scaling factors 1.1 or 1.2, respectively. These randomized parameter settings resemble practical use cases.
2. Separate the above 5 image sets into 8 training-testing pairs. Specifically, we use training sets $\{D^{\mathrm{MF}}(I), D^{\mathrm{ONE}}(I)\}$ and testing sets $\{D^{\mathrm{MF}}(\bar{I}), D^{\mathrm{ONE}}(\bar{I})\}$, where $\mathrm{MF} \in \{\mathrm{MF3}, \mathrm{MF5}\}$ and $\mathrm{ONE} \in \{\mathrm{ORI}, \mathrm{AVE}, \mathrm{GAU}, \mathrm{RES}\}$, I is a subset of the image indexes (randomly selected) and $\bar{I}$ is its complement.
3. Randomly select 50% from each of the median filtered image sets (D^{MF3} and D^{MF5}) to obtain D^{MF35}, and 25% from each of the 4 non-median filtered image sets to obtain D^{ALL}. Partition the image sets D^{MF35} and D^{ALL} into a training sets $\{D^{\mathrm{MF35}}(I), D^{\mathrm{ALL}}(I)\}$, and a testing sets $\{D^{\mathrm{MF35}}(\bar{I}), D^{\mathrm{ALL}}(\bar{I})\}$.

For all the constructed training-testing pairs described above, the size of the training set is set to be 40% of the database size.

Performance Evaluation. For each training-testing pairs, all detectors under investigation are implemented as binary classifier using C-SVM with RBF kernel. The inputs of the classifiers are the selected features under study, computed from all images in the respective training set. The best hyper-parameters (C_0, γ_0) are optimized over the parameter grid $(C, \gamma) \in \{(2^c, 2^g) | c, g \in \mathbb{Z}\}$ using five-fold cross-validation [15]. All images in the corresponding testing set are classified using the classifier trained on the training set after computing the selected features.



(a) (b)

Fig. 3. Classification results for (a) uncompressed images with varying image resolutions, (b) JPEG post-compressed images with varying QFs

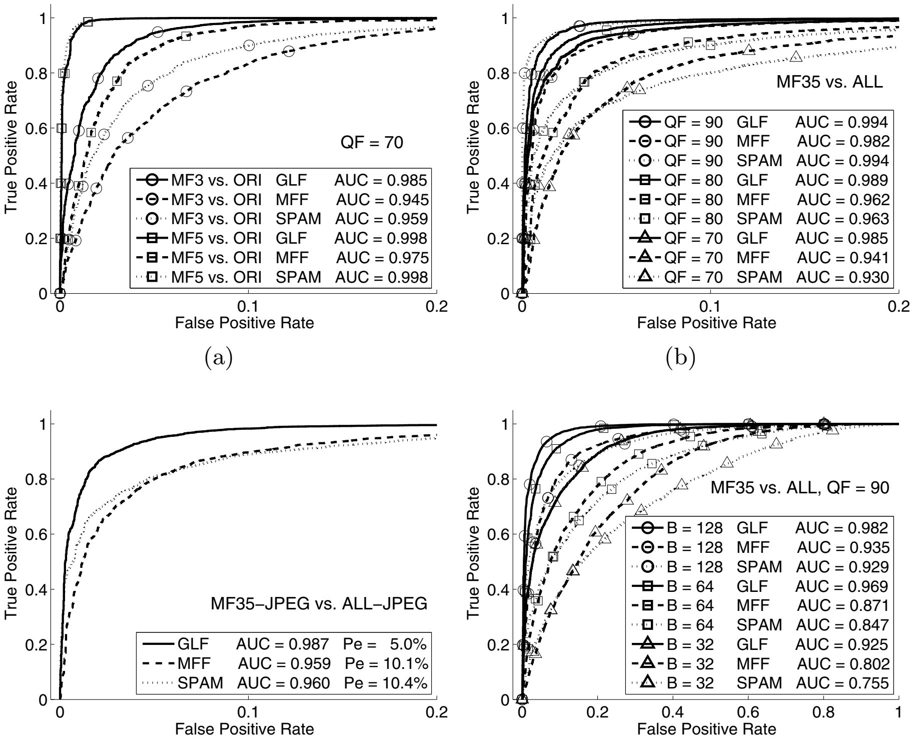
4.2 Comparison with Prior Art

MF Detection in Uncompressed Images. To evaluate the detection performance in this scenario, we take the MFF-based scheme as benchmark, which is known to give the best detection performance with uncompressed images [3]. We calculate two performance metrics, specifically, 1) AUC: the area under the ROC curve; 2) $P_e = \min 1/2(P_{\mathrm{FP}} + (1 - P_{\mathrm{TP}}))$: the minimum average decision error under the assumption of equal priors and equal costs, where P_{FP} and P_{TP} denote the false positive and true positive rates, respectively.

The results in Fig. 3(a) indicate that our scheme performs relatively comparable to the MFF-based method, and achieves nearly perfect classification performance even for image resolutions as low as 16 × 16. In general, GPF features perform better than LCF features to a certain extent, which becomes most evident for low resolution images. GLF features are generally superior to both GPF and LCF.

MF Detection in JPEG Post-compressed Images. To evaluate the robustness of the proposed MF detection scheme against JPEG compression, we use the JPEG versions of the training-testing pairs described in Section 4.1. More specifically, we test 3 different quality factors (QF), i.e., QF $\in \{70, 80, 90\}$.

From Fig. 3(b), we observe that the classification performance of GPF, LCF and GLF features decrease significantly with decreasing JPEG quality factor. As discussed in Section 2.2 (cf. Fig. 1(d)), excluding untextured pixels might increase the performance of GPF features since the statistical fingerprint of median filtering is not necessarily present in very smooth regions. However, this also poses a threat in the case of low image resolution, as there may not be enough pixels left to calculate the GPF features robustly. This applies to LCF features as well, although LCF features seem generally more robust to JPEG post-compression than GPF features. However, GLF features give the best performance again.



(a)

(b)

(c)

(d)

Fig. 4. Classification results of (a) post-compressed MF vs. ORI training-testing pair with QF = 70, (b) post-compressed MF35 vs. ALL training-testing pairs with varying QFs, (c) MF35-JPEG vs. ALL-JPEG training-testing pair, (d) post-compressed MF35 vs. ALL training-testing pair with QF = 90 for varying image resolutions

Fig. 4(a) depicts the results for post-compressed MF vs. ORI training-testing pairs with QF = 70, based on GLF, MFF and SPAM ($T = 3$) features. While all three schemes show satisfactory detection performance (note the scaling of the axes), it is observed that GLF considerably outperforms MFF and SPAM for 3×3 filter windows. For windows of size 5×5, GLF and SPAM yield comparable results (and are superior to the MFF features), whereas SPAM works in a feature space of considerably higher dimension (686-D for $T = 3$, compared to 56-D for our scheme). Moreover, the ROC curves of our scheme for both 3×3 and 5×5 median filtering detection are relatively closer to each other, which indicates a more consistent classification performance of our scheme. We note that a similar behavior is also observed on other MF vs. ONE training-testing pairs.

Fig. 4(b) depicts the results for post-compressed MF35 vs. ALL training-testing pairs with varying QFs. Our scheme outperforms the MFF and SPAM based schemes for QF $\in \{80, 70\}$, and yields comparable results with SPAM for QF = 90. For practical forensic analysis, however, the JPEG quality factor is generally unknown. Therefore the best strategy is to train the classifier with images of varying QFs. To this end, we randomly select 3,000 images from each of the 3 JPEG versions D^{MF35} to obtain $D^{\mathrm{MF35\text{-}JPEG}}$, and similarly obtain $D^{\mathrm{ALL\text{-}JPEG}}$ from D^{ALL}. The results shown in Fig. 4(c) indicate that proposed scheme also outperforms prior art in this practical setup. The result can be well explained with Fig. 4(b). For all QFs, the ROC curves of our scheme are relatively close to each other, indicating that the proposed scheme can achieve a more consistent classification performance in case of strong JPEG compression.

In our last experiment, we investigate the effect of the analyzed image (region) size on detection performance. We use post-compressed MF35 vs. ALL training-testing pairs with QF $= 90$ and crop 32×32, 64×64 and 128×128 center portions from each image of the composite database. As shown in Fig. 4(d), the detection performance of the SPAM-based scheme decreases rapidly with decreasing image resolution, which is also observed in [3]. On the contrary, our scheme and MFF-based scheme are more robust. Moreover, for each tested image resolution, our scheme outperforms both MFF and SPAM, indicating that the proposed scheme is more reliable for MF detection in low-quality images in terms of both low resolution and/or JPEG compression.

5 Conclusions and Future Work

In this paper, we investigated the blind forensics of median filtering (MF) in digital images. The contributions of the paper manifest in two main aspects. First, we presented a theoretical analysis of statistical characteristics of median filtered signals in the difference domain. Second, we proposed an effective and reliable scheme based on global probability features (GPF) and local correlation features (LCF) for MF detection in both uncompressed images and JPEG post-compressed images. Compared to prior art, we achieved considerable performance improvement for low resolution and strongly JPEG compressed images.

In our future work, we will extend our scheme to detect image tampering involving combinations of median filtering with non-median filtering. Preliminary experiment results demonstrate that our scheme is also reliable in this case.

Acknowledgments. The authors appreciate the support received from NSF of China under contract 60970145. They also thank Dr. M. Kirchner from UC Berkeley for thorough comments and suggestions, Dr. H. Yuan for providing the code of MFF-based scheme in [3] and the anonymous reviewers for their comments that helped to improve the manuscript.

References

1. Kirchner, M., Fridrich, J.: On detection of median filtering in digital images. In: Proceedings SPIE, Electronic Imaging, Media Forensics and Security II, vol. 7541, pp. 1–12 (2010)

2. Cao, G., Zhao, Y., Ni, R., Yu, L., Tian, H.: Forensic detection of median filtering in digital images. In: Proceedings of the 2010 IEEE International Conference on Multimedia and Expo (ICME), pp. 89–94 (2010)
3. Yuan, H.: Blind Forensics of Median Filtering in Digital Images. IEEE Transactions on Information Forensics and Security 6(4), 1335–1345 (2011)
4. Popescu, A.C., Farid, H.: Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing 53(2), 758–767 (2005)
5. Neelamani, R., de Queiroz, R., Fan, Z., Dash, S., Baraniuk, R.G.: JPEG compression history estimation for color images. IEEE Transactions on Image Processing 15(6), 1365–1378 (2006)
6. Kirchner, M., Böhme, R.: Hiding traces of resampling in digital images. IEEE Transactions on Information Forensics and Security 3(4), 582–592 (2008)
7. Stamm, M.C., Tjoa, S.K., Lin, W.S., Liu, K.J.R.: Undetectable image tampering through JPEG compression anti-forensics. In: Proc. IEEE Int. Conf. Image Process., pp. 2109–2112 (2010)
8. Pitas, I., Venetsanopoulos, A.N.: Order statistics in digital image processing. Proceedings of the IEEE 80(12), 1893–1921 (1992)
9. Bovik, A.C.: Streaking in median filtered images. IEEE Transactions on Acoustics, Speech and Signal Processing 35(4), 493–503 (1987)
10. Bovik, A.C., Huang, T., Munson, D.: The effect of median filtering on edge estimation and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(2), 181–194 (1987)
11. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
12. Bas, P., Furon, T.: BOWS-2, `http://bows2.gipsa-lab.inpg.fr`
13. United States Department of Agriculture (2002), Natural resources conservation service photo gallery, `http://photogallery.nrcs.usda.gov`
14. Gole, T., Böhme, R.: The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 2010 ACM Symposium on Applied Computing, March 22-26 (2010)
15. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification, `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`
16. Hall, M.: Combinatorial Theory, 2nd edn. John Wiley & Sons, Inc., Hoboken (1988)

Appendix: CDF of First Order Differences

With $\boldsymbol{X}$ being i.i.d., the statistics of $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ are symmetric with respect to mirroring and flipping. Therefore the following relations are maintained:

$$\Pr(\Delta_1(n) = t) = \Pr(\Delta_1(n) = -t), \tag{A.1}$$

$$F_1(r,t) = 1 - F_1(r, -t-1). \tag{A.2}$$

where $F_1(r,t) = \Pr(\Delta_1(n) \leq t)$, r relates to the filter window size w, $w = 2r+1$ (for original non-filtered image, we take $r = 0$) and t is integer. With this property, we can derive $F_1(r,t)$ for $t < 0$. For $t \geq 0$, the CDF follows directly from Eqn (A.2). With $F_1(r,t)$, the CDF of $|\boldsymbol{\Delta}_1|$ is then given by

$$F_{|1|}(r,t) = \Pr(|\Delta_1(n)| \leq t) = F_1(r,t) - F_1(r,-t-1). \tag{A.3}$$

Here, $F_{|1|}(r,t) = 0$ when $t < 0$.

Original Non-filtered Signals. $F_1(r,t)$ can be computed as follows

$$\begin{aligned} F_1(r,t) &= \sum_{x=0}^{N-1} \Pr(X_n \leq X_{n+1} + t \mid X_{n+1} = x) \cdot \Pr(X_{n+1} = x) \\ &= N^{-2} C_{N+t+1}^2, \end{aligned} \tag{A.4}$$

where C_{N+t+1}^2 is the binomial coefficient.

Median Filtered Signals. A one-dimensional median filter is defined as

$$\hat{X}_i = \text{median}\{X_{i-r}, \dots, X_i, \dots, X_{i+r}\}, \tag{A.5}$$

It is obvious that there exist common elements $\boldsymbol{Z}$ for computing $\hat{X}_n$ and $\hat{X}_{n+1}$,

$$\boldsymbol{Z} = \{\, X_i \mid i \in [n-r+1, n+r] \,\}, \tag{A.6}$$

with associated order statistics as below:

$$\boldsymbol{Z}_{(\cdot)} = \{\, Z_{(i)} \mid Z_{(i)} \leq Z_{(i+1)}, i \in [n-r+1, n+r] \,\}. \tag{A.7}$$

Inspired by [2], we further define three sets: $\bar{Z}_1 = [0, Z_{(r)}]$, $\bar{Z}_2 = (Z_{(r)}, Z_{(r+1)}]$, $\bar{Z}_3 = (Z_{(r+1)}, N-1]$. Then the filtered output $\hat{X}_n$ can be determined in terms of the relationship between X_{n-r} and the elements of $\boldsymbol{Z}_{(\cdot)}$. That is

$$\hat{X}_n = \begin{cases} Z_{(r)} & \text{if } X_{n-r} \in \bar{Z}_1 \\ X_{n-r} & \text{if } X_{n-r} \in \bar{Z}_2 \\ Z_{(r+1)} & \text{if } X_{n-r} \in \bar{Z}_3. \end{cases} \tag{A.8}$$

Similarly, $\hat{X}_{n+1}$ is related to X_{n+r+1} and $\boldsymbol{Z}_{(\cdot)}$. Now to derive $F_1(r,t)$, consider the following relation which holds by the law of total probability

$$\begin{aligned} F_1(r,t) = \sum_{z_1=0}^{N-1} \sum_{z_2=0}^{N-1} &\Pr(\Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &\cdot \Pr(Z_{(r)} = z_1, Z_{(r+1)} = z_2). \end{aligned} \tag{A.9}$$

As $t < 0$, the conditional probability can be broken into four event probabilities,

$$\begin{aligned} &\Pr(\Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &= \Pr(X_{n-r} \in \bar{Z}_1, X_{n+r+1} \in \bar{Z}_2, \Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &+ \Pr(X_{n-r} \in \bar{Z}_1, X_{n+r+1} \in \bar{Z}_3, \Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &+ \Pr(X_{n-r} \in \bar{Z}_2, X_{n+r+1} \in \bar{Z}_2, \Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &+ \Pr(X_{n-r} \in \bar{Z}_2, X_{n+r+1} \in \bar{Z}_3, \Delta_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2). \end{aligned} \tag{A.10}$$

Since $\boldsymbol{X}$ is i.i.d., each event probability can be easily calculated. For example, the first event probability is computed by

$$\begin{aligned} &= \Pr(X_{n-r} \in [0, z_1]) \cdot \Pr(X_{n+r+1} \in (z_1, z_2]) \cdot \Pr(z_1 - X_{n+r+1} \leq t) \\ &= N^{-2}(z_1 + 1) d P_1(t, d). \end{aligned} \tag{A.11}$$

Here, $d = z_2 - z_1$, $P_1(t, d) = (1 + d^{-1}(t + 1))\delta(t, -d)$ and

$$\delta(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{if } x < y. \end{cases} \tag{A.12}$$

The other three event probabilities are calculated in a similar manner. Then the final conditional probability is given by

$$\begin{aligned} &\Pr(\mathit{\Delta}_1(n) \leq t \mid Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &= N^{-2}[(z_1 + 1)dP_1(t, d) + (z_1 + 1)(N - z_2 - 1)\delta(t, -d) \\ &\qquad + d^2 P_2(t, d) + d(N - z_2 - 1)P_1(t - 1, d)], \end{aligned} \tag{A.13}$$

where $P_2(t, d) = 0.5d^{-2}(d + t)(d + t + 1)\delta(t, -d)$. The calculation of the joint probability in Eqn (A.9) is an exercise of combinatorics. In fact

$$\begin{aligned} &\Pr(Z_{(r)} = z_1, Z_{(r+1)} = z_2) \\ &= \begin{cases} N^{-2r} C_{2r}^{r} \left((z_1 + 1)^r - z_1^r\right)\left((N - z_2)^r - (N - z_2 - 1)^r\right) & \text{if } z_1 \neq z_2 \\ N^{-2r} \sum_{k_1=1}^{r} \sum_{k_2=1}^{r} C_{2r}^{k_1+k_2} C_{2r-k_1-k_2}^{r-k_1} z_1^{r-k_1} (N - z_2 - 1)^{r-k_2} & \text{if } z_1 = z_2. \end{cases} \end{aligned} \tag{A.14}$$

Average Filtered Signals. A 1-D moving average filter is defined as

$$\hat{X}_i = \text{round}\left\{(2r + 1)^{-1} \sum\nolimits_{j=-r}^{r} X_{i+j}\right\}. \tag{A.15}$$

where round$\{\bullet\}$ is the rounding operation. Let $S = \sum_{i=n-r+1}^{n+r} X_i$, we then have

$$\begin{aligned} F_1(r, t) = \sum_{x=0}^{N-1} \sum_{s=0}^{2r(N-1)} &\Pr(\mathit{\Delta}_1(n) \leq t \mid X_{n+r+1} = x, S = s) \\ &\cdot \Pr(X_{n+r+1} = x, S = s). \end{aligned} \tag{A.16}$$

By the i.i.d. assumption, the conditional probability follows as

$$\Pr(\mathit{\Delta}_1(n) \leq t \mid X_{n+r+1} = x, S = s) = N^{-1}(t_1 + 1)\delta(t_1, 0), \tag{A.17}$$

where $t_1 = (a + t)(2r + 1) - s + r$ and $a = \text{round}\{(2r + 1)^{-1}(s + x)\}$. Since X_{n+r+1} and S are independent, we have

$$\begin{aligned} \Pr(X_{n+r+1} = x, S = s) &= \Pr(X_{n+r+1} = x) \cdot \Pr(S = s) \\ &= N^{-1}\Pr(S = s). \end{aligned} \tag{A.18}$$

The calculation of $\Pr(S = s)$ can be solved by means of generating functions from the field of combinatorial mathematics [16]. In this case, we have

$$\Pr(S = s) = N^{-2r} \cdot \sum_{l=0}^{\infty} \sum_{k=0}^{2r} C_{2r+l-1}^{l} C_{2r}^{k} (-1)^k \mid_{kN+l=s}. \tag{A.19}$$

Robustness of Color Interpolation Identification against Anti-forensic Operations

Wei-Hong Chuang and Min Wu

University of Maryland, College Park, MD, USA
{whchuang,minwu}@umd.edu

Abstract. Color interpolation identification using digital images has been shown to be a powerful tool for addressing a range of digital forensic questions. However, due to the existence of adversaries who have the incentive to counter the identification, it is necessary to understand how color interpolation identification performs against anti-forensic operations that intentionally manipulate identification results. This paper proposes two anti-forensic techniques against which the robustness of color interpolation identification is investigated. The first technique employs parameter perturbation to circumvent identification. Various options that achieve different trade-offs between image quality and identification manipulation are examined. The second technique involves algorithm mixing and demonstrates that one can not only circumvent but also mislead the identification system while preserving the image quality. Additional discussions are also provided to enhance the understanding of anti-forensics and its implications to the design of identification systems.

1 Introduction

Recent years have witnessed the prevalence of digital images due to the advancement of affordable high-quality digital cameras and broadband Internet. However, as digital images are vulnerable to software editing and manipulations, concerns regarding their origin and authenticity have also been raised and received increasing attention. The study of digital image forensics aims at addressing these concerns by answering questions about the acquisition and processing history of a digital image, such as its source device and the post-processing operations that it has undergone since its creation.

One class of techniques in the forensic literature addresses the identification of the underlying color interpolation algorithm that a digital camera has used to create an image [3, 5, 11, 13]. Color interpolation is a common step in digital imaging and has a crucial impact on the quality of output images [8]. Different camera manufacturers compete with customized color interpolation modules to enhance the image quality, and it has been shown that important information about the color interpolation modules can be effectively learnt from detectable traces left in output images [13].

Similar to many other tasks regarding data trustworthiness, there exist adversaries who have incentives to perform anti-forensic operations to counter forensic

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 16–30, 2013.

identification [7, 12]. For example, consider the scenario of technology infringement that a company infringes another company's color interpolation technology via reverse engineering or industrial espionages. The pirate company has incentives to counteract the identification of color interpolation so that it can use the technology without being caught. It may be of further interest to the pirate company if it can mislead the identification toward a wrong direction. In the scenario of crime scene investigation [9], being aware that the camera information can be inferred from the color interpolation algorithm [13],a technology-savvy criminal can conceal the origin of a digital image by circumventing the identification.

These scenarios prompt a strong need for understanding the robustness and resilience of today's color interpolation technology against anti-forensic operations. Toward this end, we explore anti-forensic techniques and evaluate the performance of color interpolation identification against these anti-forensic operations. In principle, one can alter the image to weaken the evidence that may reveal the underlying color interpolation module. There exists, however, an inevitable trade-off between the strength of the trace concealment and the quality of the resulting image: if the strength is too weak, the identification is likely to remain effective, but if the strength is too strong, the image may suffer from serious distortions. We consider different situations of counter identification, and compare our proposed techniques in terms of their trade-offs between the quality of image and the manipulation of identification results.

To the best of our knowledge, the most relevant work to this paper is by Kirchner and Böhme in [6], whereby a method was presented to resynthesize a linear color interpolation relation in digital images and minimizes the image quality distortion. Compared to the work in [6], we propose a low-complexity methodology for counter identification, and our techniques are applicable to a large class of interpolation algorithms that cannot be simply modeled as linear.

The rest of the paper is organized as follows. Sec. 2 reviews color interpolation and its identification based on [13]. Sec. 3 proposes a generic methodology of parameter perturbation for circumventing the identification of a given color interpolation algorithm. Sec. 4 investigates how to mislead the identification toward an incorrect decision. Sec. 5 discusses extensions of the anti-forensic techniques and insights into our study. Sec. 6 concludes this paper.

2 Design and Evaluation of a Color Interpolation Identification System

In this section, we first review the process of color interpolation and the basic principles of its identification. We then describe in detail our design and evaluation of a color interpolation identification system, which will be used in subsequent sections for our anti-forensic study.

2.1 Principles of Color Interpolation Identification

In digital photography, light reflected from a real-world scene passes through the optical components and is then detected by an array of sensors. Most cameras in today's consumer market employ a color filter array (CFA) to filter the lights from the scene. The CFA selectively allows a certain color of light, commonly red, green, or blue, to pass through it to the sensors and be recorded. For each color plane, the lost pixel values are interpolated using its neighboring pixel values to form the interpolated image. In-camera post-processing, such as white balance or compression, follows the interpolation to enhance the overall image quality and/or to reduce storage demand.

Several previous works have studied how to identify the underlying color interpolation algorithm of a camera-generated image [3, 5, 11, 13].In this paper, we perform the identification of color interpolation based on the scheme proposed in [13]. This scheme is one of the earliest works that incorporates the concept of direction-adaptive interpolation and has been shown to have a promising identification performance We improve upon the scheme with refined directional classification for higher identification accuracy. Specifically, define $I_{x,y}$ as the sensor value at location (x, y). The local gradient profile along different directions can be found as:

$$\begin{cases} H_{x,y} &= |I_{x,y-2} + I_{x,y+2} - I_{x,y}|, \\ V_{x,y} &= |I_{x-2,y} + I_{x+2,y} - I_{x,y}|, \\ D_{x,y} &= |I_{x-2,y-2} + I_{x+2,y+2} - I_{x,y}|, \\ A_{x,y} &= |I_{x-2,y+2} + I_{x+2,y-2} - I_{x,y}|. \end{cases}$$

Each pixel at location (x, y) is classified into one of five directional regions according to its gradient profile using two preset thresholds T_1 and T_2. The partition of the gradient profile plane is illustrated in Fig. 1 and summarized in Table 1. By approximating the interpolated pixels as a linear weighted sum of the colors from directly-captured surrounding pixels, we can apply a least squares method to solve equations corresponding to each directional region in each color channel and obtain the linear interpolation coefficients that represents the interpolation algorithm.

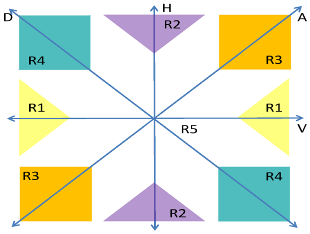


Fig. 1. Gradient profile plane

Table 1. Regions' gradient relations and meanings

Region	Gradient relation	Meaning
R_1	$V_{x,y} - H_{x,y} > T_1$	significant horizontal
R_2	$H_{x,y} - V_{x,y} > T_1$	significant vertical
R_3	$A_{x,y} - D_{x,y} > T_2$	significant diagonal
R_4	$D_{x,y} - A_{x,y} > T_2$	significant anti-diagonal
R_5	others	mainly smooth

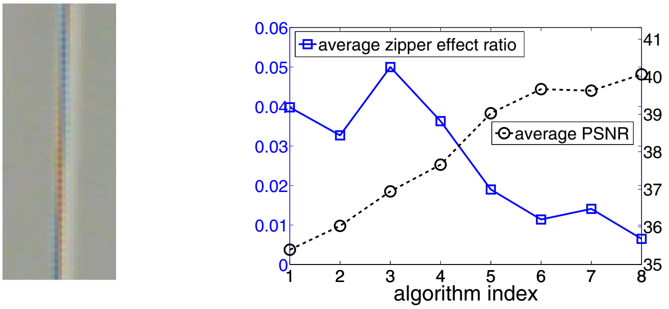
2.2 Experiment Setup and Performance Metrics

In this section, we describe our experiment setup and performance metrics for carrying out and evaluating our anti-forensic designs. Our goals here are to sample representative color interpolation algorithms used in our study, and to establish a testbed on which we can evaluate forensic and anti-forensic capabilities in terms of identification accuracy and the resulting image quality.

Color Interpolation Algorithms: Color interpolation has been an active research area in image processing. Detailed surveys and comparisons of color interpolation techniques can be found in [1, 8]. The algorithms in the literature range from non-adaptive ones with low complexity such as bilinear or bicubic interpolation to highly adaptive and complex ones that can better capture the underlying image structure and recover the lost color information. We investigate eight color interpolation algorithms. The first six have been well known in the literature for more than one decade, including bilinear, bicubic, smooth hue, median filter based, gradient based, and an adaptive color plane algorithm [1]. In recent years, significant progress has been made to improve the reconstruction quality of color interpolation. To reflect the advancement of the state of the art, we also include a recent algorithm based on local polynomial approximation (LPA) and intersection of confidence intervals (ICI) [10], which performs well in a comparative survey [8], and a latest algorithm that combines local directional interpolation (LDI) and nonlocal adaptive thresholding (NAT) [15].

We construct a dataset composed of images interpolated by the above eight algorithms. Specifically, we first take 50 high-resolution images with a variety of content by a high-end standalone camera. From each image, we extract the central portion of 1024×1024 pixels, which is prefiltered and down-sampled to 512×512 pixels in order to attenuate the traces of color interpolation and post-processing left by the camera. The resulting 512×512 "full-color" image is then sampled according to a given CFA pattern, and interpolated using each of the eight different interpolation algorithms to simulate in-camera processing.

Performance Metrics: We adopt the *full-reference* methodology [14] for image quality assessment. The quality of a color interpolated image is assessed with respect to a reference image. The 512×512 full-color image above is used for this purpose, which is justified in the same way as in [8] and we find that such reference images are visually pleasant. There are a handful of full-reference image quality metrics in the literature. The Peak Signal-to-Noise Ratio (PSNR) is probably the most well-known one. While it is still widely used, previous research has shown that PSNR may not always reflect the true signal fidelity [14]. The quality metric called Structural Similarity (SSIM) index [14] incorporates the similarity in image structure to capture the subjective quality perceived by human beings. One notable artifact in color interpolation is called *zipper effect*, which occurs if an interpolation algorithm fails to interpolate pixels along directional edges, as illustrated in Fig. 2(a). The extent of zipper effect can be quantified by the quality metric called *zipper effect ratio* [4, 15] , which measures the increase in spatial color discontinuity due to color interpolation. In order to provide a comprehensive assessment of image quality, it is beneficial to examine more than one quality metric. Fig. 2(b) compares the PSNR and the zipper effect ratio of each algorithm, averaging over all 50 images. In terms of



(a) (b)

Fig. 2. (a) an example of zipper effect (best viewed on screen); (b) PSNR and zipper effect ratio averaged over 50 images associated with different interpolation algorithms: (1) bilinear, (2) bicubic, (3) smooth hue, (4) median filter based, (5) gradient based, (6) adaptive color plane, (7) LPA-ICI, and (8) LDI-NAT

both metrics, algorithms with higher indices perform better. These algorithms are more sophisticated and represent the advancement of color interpolation technology.

Identification System: We construct a color interpolation identification system that uses the color interpolation coefficients as features. We use the 50 images described in Sec. 2 and their interpolated versions created by each of the eight interpolation algorithms. The total number of interpolated images is therefore $50 \times 8 = 400$. Half of the images are used for training an 8-class probabilistic Support Vector Machine (pSVM) classifier [13] with parameters selected by cross validation, and the remaining half are used for testing. The identification system takes an image as input, and outputs the identification confidence (or likelihood) of each of the eight algorithms. Maximum-likelihood classification yields an overall accuracy of 96.3%, suggesting the accuracy of color interpolation identification.

3 Circumventing Color Interpolation Identification via Parameter Perturbation

Our first anti-forensic goal is to circumvent the identification of a specific color interpolation algorithm when it is used for interpolation. We refer to such an algorithm as a *targeted interpolation algorithm*. We model a color interpolation algorithm as a combination of an *architecture* part that entails the algorithmic flow and the *parameter* part that consists of configurable settings. To circumvent the identification, perturbation can be introduced into a parameter part to alter the overall color interpolation algorithm, so that estimated color interpolation coefficients are changed and cannot be recognized by the identification system. As pointed out in Sec. 1, there is a trade-off between the resulting image quality and the manipulation of identification results. We will examine whether it is possible to reach a good balance between these two factors by wisely selecting the parameters for perturbation.

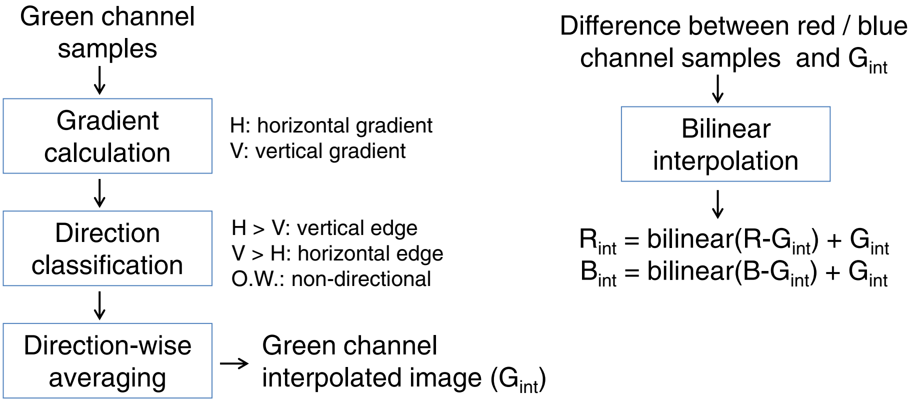


Fig. 3. Gradient-based color interpolation

3.1 Perturbing Gradient-Based Interpolation

We consider the 5th color interpolation algorithm reviewed in Sec. 2.2 as an targeted interpolation algorithm. This algorithm is based on a gradient-based partitioning of image pixels [1], and its architecture is shown in Fig. 3. We consider several options of parameter perturbation that are applicable to this algorithm. First, since the algorithm utilizes bilinear filtering in interpolating the difference between red/green and blue/green channels, one option is to perturb the kernel coefficients of bilinear filtering. Second, the targeted interpolation algorithm performs pixel averaging in the green channel according to the gradient direction (horizontal, vertical, and non-directional). A second option is hence to perturb the pixel averaging kernels in each direction. Finally, this algorithm takes two parameters, denoted as θ_1 and θ_2, to determine if a pixel falls on a horizontal edge, a vertical edge, or in a non-directional region, so a third option is to perturb the decision boundaries of individual directions. The three options are summarized as follows; the noise standard deviations are selected so that the trade-offs of different options can be compared more easily.

Option 1: Add Gaussian noise to the bilinear interpolation coefficient matrix. Noise standard deviation $\in \{0.16, 0.24, 0.3\}$. Note that the perturbation has to satisfy constraints on the coefficients' mutual relations. In particular, two coefficients at opposite horizontal/vertical positions, and four coefficients at opposite diagonal positions, must have a fixed sum of 1.

Option 2: Add Gaussian noise to the direction-wise averaging coefficients. Noise standard deviation $\in \{0.1, 0.3, 0.5\}$. Similar to Option 1, a fixed sum constraint must be imposed on the coefficients.

Option 3: Add Gaussian noise to the gradient decision threshold values θ_1 and θ_2. Noise standard deviation $\in \{0.1, 0.15, 0.2\}$. θ_1 and θ_2 must satisfy $\theta_1 + \theta_2 > 0$.

Table 2. Results of countering color interpolation identification for a gradient-based interpolation algorithm. PSNR is measured in dB. "Zipper" stands for the zipper effect ratio; "confidence" stands for the identification confidence.

	Uncompressed				JPEG compressed with QF=95			
	PSNR	SSIM	zipper	confidence	PSNR	SSIM	zipper	confidence
Option 1 (1)	38.83	0.96	0.02	0.81	39.50	0.96	0.02	0.81
(2)	38.33	0.96	0.03	0.63	38.99	0.96	0.03	0.70
(3)	37.89	0.95	0.03	0.46	37.16	0.94	0.04	0.66
Option 2 (1)	39.01	0.96	0.02	0.90	39.38	0.96	0.02	0.80
(2)	37.45	0.95	0.03	0.80	37.90	0.96	0.04	0.43
(3)	35.46	0.94	0.05	0.50	36.03	0.94	0.06	0.10
Option 3 (1)	39.02	0.96	0.02	0.53	39.02	0.96	0.03	0.49
(2)	38.66	0.96	0.03	0.30	38.80	0.96	0.03	0.28
(3)	38.41	0.96	0.03	0.18	38.42	0.96	0.04	0.16
Option 4 (1)	35.92	0.94	0.03	0.01	37.33	0.95	0.02	0.03
(2)	36.49	0.95	0.03	0.01	38.04	0.96	0.02	0.01
(3)	37.44	0.96	0.04	0.03	38.08	0.96	0.05	0.04
(4)	38.06	0.94	0.03	0.01	38.23	0.94	0.05	0.01
(6)	39.91	0.96	0.01	0.01	40.20	0.96	0.02	0.02
(7)	39.93	0.95	0.01	0.01	40.03	0.96	0.02	0.01
(8)	40.32	0.96	0.01	0.01	40.54	0.97	0.04	0.01
Option 5 (1)	37.24	0.93	0.02	0.64	38.55	0.95	0.03	0.45
(2)	35.41	0.91	0.03	0.08	37.02	0.94	0.04	0.11
Option 6 (1)	35.95	0.93	0.01	0.42	36.40	0.94	0.02	0.39
(2)	34.70	0.92	0.02	0.14	34.98	0.92	0.02	0.04

For comparison, we consider alternative options that do not involve parameter perturbation. For example, in the scenario of technology infringement, if the risk of being caught is high, one option that a pirate company has is to abandon the targeted interpolation algorithm and adopt another algorithm instead. Other alternative options include applying post-processing operations such as compression and filtering after color interpolation in order to conceal the trace of color interpolation. These three more options are summarized below:

Option 4 (i): Replace the gradient-based targeted interpolation algorithm, which is the 5th among those compared in Sec. 2.2, by another interpolation algorithm $i \in \{1, 2, 3, 4, 6, 7, 8\}$.

Option 5: JPEG compression after interpolation. Quality factor (QF) $\in \{95, 75\}$.

Option 6 (1): 3×3 median filtering after interpolation; (2): 3×3 average filtering after interpolation.

Comparison of Options: Table 2 shows the comparison of various options in terms of image quality and identification confidence. We present multiple image quality metrics to provide a more comprehensive quality assessment. This table consists of two parts. The left part of columns is the case when there is no post-processing following color interpolation. The right part of columns includes

JPEG compression as post-processing. Note that in the right part, the reference image is also compressed.

From the table, we can first see that parameter perturbation reduces the identification confidence at different costs in terms of image quality. Option 2 causes image quality degradation, but the identification confidence is kept relatively high. Note that we have imposed coefficient constraints on Option 1 and 2 to ensure that the perturbed coefficient matrices are still valid; otherwise the unconstrained perturbation would lead to much worse image quality-confidence reduction trade-offs than the reported values. Compared to Option 1 to 2, Option 3 achieves highest image quality and lowest identification confidence. In particular, Option 3 reduces the identification confidence by 40% with little reduction in image quality ((for example, PSNR decreases from 38.66dB to 38.41dB and there is nearly no reduction in other quality metrics).

We also compare Option 3 with options that do not involve parameter perturbation. If we replace the gradient-based targeted interpolation algorithm by any other interpolation algorithm as in Option 4, the identification confidence drops to near zero. This is expected since the 8-class pSVM is tailored to differentiate these algorithms. However, for Options 4 (1) to (4) that employ more rudimentary interpolation algorithms, the image quality is inferior to what Option 3 yields, which would be unacceptable as image quality is a crucial criterion in many imaging applications. Option 4 (6) to (8), which replace the gradient-based targeted interpolation algorithm by more sophisticated algorithms, outperform Option 3 in both image quality and identification confidence. This implies that, if a pirate company has more advanced technology, it should utilize such technology and there is no incentive to infringe other companies' technology.

Option 5 and 6 apply post-processing after color interpolation. These options reduce the identification confidence considerably, but none of them produce images with quality comparable to Option 3. Overall, Option 3 that perturbs decision threshold values is a simple yet effective choice for circumventing color interpolation identification with minimal reduction in image quality.

3.2 Perturbing Other Interpolation Algorithms

The proposed parameter perturbation methodology is readily applicable to other color interpolation algorithms. In particular, since a majority of interpolation algorithms are direction-adaptive based on local gradients, the options that perturb gradient-related parameters can also be employed. We have considered the adaptive color plane algorithm (6th in our list of interpolation algorithms), also known as Hamilton-Adams algorithm [2] and the LDI-NAT algorithm (our 8th algorithm) which is considered as the state-of-the-art progress in color interpolation [15]. Different from the gradient-based color interpolation algorithm that only involves intra-channel interpolation (*i.e.*, pixels are only interpolated using raw pixels of the same color), the adaptive color plane algorithm also performs inter-channel interpolation (*i.e.*, pixels can be interpolated using raw pixels of different colors). The LDI-NAT algorithm take a learning-based approach by first applying directional interpolation based on local gradients and then refining the interpolation results using nonlocal methods.

When applying the same set of options to perturb the parameters of these two interpolation algorithms, we obtain observations that are consistent with those from the gradient-based interpolation algorithm. Due to space limit, we omit details here. But overall, Option 3 that perturbs the gradient decision thresholds is found to be most effective for reducing the identification confidence while preserving the image quality. We will discuss in Sec. 5 more about the generality of our findings.

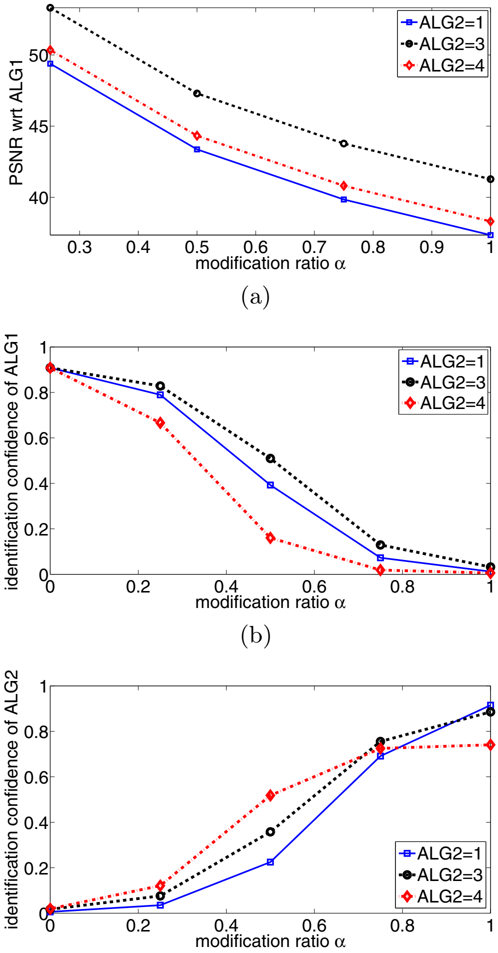
4 Misleading Color Interpolation Identification via Algorithm Mixing

So far, we have investigated ways to prevent the color-interpolation-based identification system from identifying a specific interpolation algorithm. We now study how to further mislead the identification system toward a wrong direction, namely, keeping the resulting image visually similar to the original version interpolated by a specific algorithm (referred to as ALG1), while making the identification system believe that the image is interpolated by a different algorithm (referred to as ALG2). This can be considered as a generalized scenario of the one described in Kirchner and Böhme's work [6], whereby ALG2 is the bilinear interpolation. For our study here, the similarity between two images is measured in terms of PSNR, but other metrics such as the SSIM can also be used for similarity measurement.

We examine the fusion of ALG1 and ALG2 per a given *modification ratio* $0 \leq \alpha \leq 1$. Specifically, we realize the fusion by mixing pixels generated by ALG1 and ALG2. There are multiple ways to carry out the mixing. One option is to mix pixels interpolated by ALG1 and ALG2 via linear averaging with weights $(1 - \alpha)$ and α, respectively. This is is also known as *alpha blending* in the literature of image editing. Alternatively, one can randomly select pixels from ALG1 and ALG2 with ratios $(1 - \alpha)$ and α, respectively. This method can be seen as non-linear mixing. We examine linear and random mixing methods for the case ALG1=5 and ALG2 $\in \{1, 3, 4\}$ (that is, ALG1 is the 5th algorithm and ALG2 are the 1st, 3rd, and 4th algorithms from Sec. 2.2), while similar results can be observed for other combinations of ALG1 and ALG2 as well.

Due to space limit, we only show the case of linear mixing in Fig. 4, but for both mixing methods, we see that when the modification ratio α increases, the resulting image becomes less similar to the original version by ALG1, the identification confidence of ALG1 decreases, and the identification confidence of ALG2 increases. The exact trade-offs between the visual similarity reduction and the identification manipulation depends on the choice of ALG2.

On the other hand, these two mixing methods themselves also differ in the trade-offs between visual similarity reduction and identification manipulation. For the illustrative case of ALG1=5 and ALG2=3, Fig. 5 shows the relation between the visual similarity to ALG1 and the identification confidence of ALG2.



(a)

(b)

(c)

Fig. 4. Algorithm mixing for misleading identification. (a) average PSNR ; (b) identification confidence of ALG1; (c) identification confidence of ALG2.

For a given modification ratio α, though these two mixing methods lead to similar identification confidences of ALG2, linear mixing yields a higher PSNR, meaning that the output of linear mixing remains more similar to the output of ALG1.

We also find that algorithm mixing can be employed as an option for circumventing the identification of a specific color interpolation algorithm (namely, the task in Sec. 3). For illustration, we perform algorithm mixing by choosing the gradient-based algorithm as ALG1 and the median filter based algorithm (the 4th in Sec. 2.2) as ALG2. Fig. 6 shows the resulting image quality and identification confidence of the targeted interpolation algorithm. Note that if linear mixing

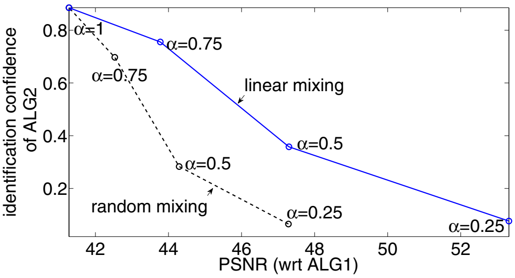


Fig. 5. PSNR w.r.t. ALG1 versus identification confidence of ALG2. ALG1=5, ALG2=3.

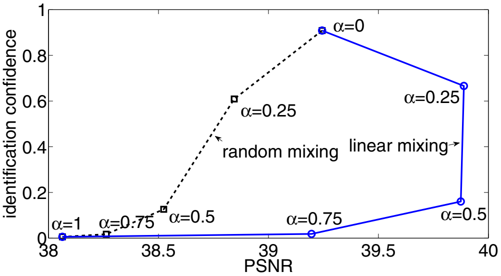


Fig. 6. Algorithm mixing for circumventing the identification of the gradient-based interpolation algorithm

is used, the PSNR increases when $0 < \alpha < 0.75$. A similar observation has also been reported in [8], and this can be potentially attributed to the independence of interpolation errors between different color interpolation algorithms. For the selected ALG1 and ALG2, both mixing methods achieve better balances between the image quality and the identification confidence as compared to the options considered in Sec. 3.1. For example, for a PSNR value of 38.41dB (the 3rd row associated with Option 3 in Table 2), the identification confidence yielded by Option 3 is 0.18, but the two mixing methods lead to even lower confidences of 0.09 and 0.01, respectively. While such a superior performance may not always be available for concealing other targeted interpolation algorithms (in particular, the performance of linear mixing depends on the validity of interpolation error independence and the choice of modification ratio), the algorithm mixing technique serves as an generic approach to circumventing identification. As a remark, it should be noted that algorithm mixing may require more processing and storage power in the camera since multiple color interpolation algorithms may need to be performed.

5 Extensions and Further Discussions

In this section, we provide additional discussions of the proposed anti-forensic techniques. First, we complement the randomized parameter perturbation by formulating and solving an optimization problem that incorporates image quality and identification confidence. We then compare this paper and a relevant prior work [6]. Finally, we look into the inherent issues and its implications of the state-of-the-art identification system.

5.1 Optimization Problem Formulation of Parameter Perturbation

As an illustrative example, we have applied in Sec. 3 randomized parameter perturbation to conceal the gradient-based color interpolation algorithm, and the performances in terms of the image quality and the identification confidence, are measured by averaging over all the test images. For some images, the identification confidence may remain high after the randomized perturbation, as shown in Fig. 7(a). In order to circumvent identification that is usually performed by an automated detector, it is necessary to make the identification confidence fall below a threshold set in the automated detector. Toward this end, we formulate parameter perturbation as the following optimization problem:

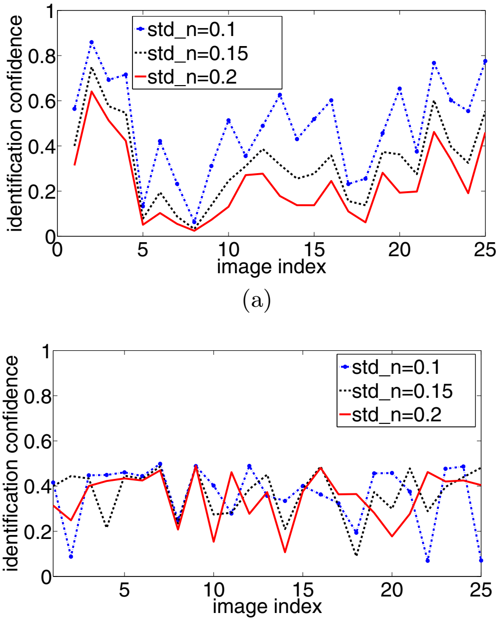
$$\max_{\theta_1,\theta_2} Q(I_p), \quad \text{subject to } C(I_p) \le C_t,$$

where I_p is the perturbed image, $Q(\cdot)$ is a quality metric of an image, $C(\cdot)$ is the identification confidence with respect to a targeted interpolation algorithm, and C_t is a preset threshold. Because the full-color reference image is not available during color interpolation, the image interpolated by the original gradient-based interpolation algorithm serves as an approximate reference image in the optimization. The PSNR with respect to this reference image is taken as the quality metric $Q(\cdot)$, and $C(\cdot)$ comes from the identification confidence of the gradient-based algorithm reported by the 8-class pSVM.

Since it is not always feasible to represent $Q(I_p)$ and $C(I_p)$ in a closed form, solving for the perturbation parameters θ_1 and θ_2 is a challenging optimization task. In this paper, we take a Monte-Carlo approach that applies Option 3 in Sec. 3.1 multiple times to perturb the image, and keep the result that satisfies the constraint on $C(I_p)$ with highest $Q(I_p)$. Compared to randomized perturbation, this solution is guided explicitly by the image quality and the identification confidence. We compare the results of Option 3 and the guided perturbation when $C_t = 0.5$ for three different noise strengths. Their average PSNR values are roughly equal. The identification confidences are shown in Fig. 7. It can be seen that the proposed approach suppresses the identification confidence for individual images while maintaining a high image quality; the results also suggest that the approximation of the reference image by the image interpolated using the gradient-based algorithm is effective.

5.2 Comparison with Kirchner and Böhme [6]

As reviewed in Sec. 1, the work by Kirchner and Böhme [6] is a related prior work that studies anti-forensic techniques for color interpolation identification. Despite the similar goals, the approaches adopted in [6] and the present paper differ substantially. Kirchner and Böhme's work tries to synthesize a linear



(a)

(b)

Fig. 7. Identification confidences as a result of randomized parameter perturbation (a) and the guided parameter perturbation (b). Identification confidence in (b) ≤ 0.5.

dependency among pixels in an image while minimizing the overall distortion. The authors proposed to search for a pre-filter that estimates raw samples acquired by the camera sensor array and applies the bilinear interpolation kernel to the estimated raw samples to reconstruct the entire image that satisfies the linear dependency. This approach can be viewed as altering the raw samples to counter the identification of color interpolation. In contrast, our proposed approaches leave the raw samples unchanged, but alter the color interpolation algorithms so that the output image either deviates from a target color interpolation algorithm or moves toward the algorithm. In a sense, it can be viewed that Kirchner and Böhme's method alters the color interpolation *after* the creation of an image, while our techniques alter the color interpolation *during* the creation of an image. Also notice that in Kirchner and Böhme's work, even for the case of bilinear interpolation, searching for the pre-filter (or equivalently, the virtual raw samples) is already computationally challenging, and it becomes even more difficult to generalize this method to more sophisticated color interpolation. In comparison, our techniques are less complex and exhibit a promising generalization capability. It will be an interesting future work to explore whether Kirchner and Böhme's work and our approaches can be properly fused for improved anti-forensic capability.

5.3 Reflections on Robustness of Color Interpolation Identification

As motivated in Sec. 1, a fundamental reason for studying anti-forensic operations against color interpolation identification is to understand the robustness and resilience of identification schemes in an adversarial environment against intentional manipulations of identification results. As demonstrated in the paper, properly configured parameter perturbation and algorithm mixing can circumvent and mislead the identification system while preserving image quality.

We have observed that by perturbing the decision boundaries of gradient directions, the identification confidence can be reduced with minimal reduction in image quality. The rationale of such effectiveness can be understood as follows. In order to capture the nature of direction adaptation in prevailing color interpolation algorithms (for example, the gradient-based, adaptive color plane, and LDI-NAT algorithms considered in this paper), today's color interpolation identification schemes [5, 13] are primarily based on direction classification of pixels and least-squares estimation of interpolation coefficients for each class. By perturbing the decision boundaries in color interpolation, we are essentially changing the ways some pixels are interpolated, and this directly makes the estimated color interpolation coefficients deviate from the typical values learnt from the original color interpolation algorithm, making the identification more difficult. In the meantime, pixels whose interpolation are more likely to be changed are those near the decision boundaries. These pixels are not coupled tightly with respective direction classes in the interpolation algorithm, and none of the classes is likely to interpolate these pixels particularly well. As such, the image quality does not seriously degrade when these pixels are interpolated by the methods associated with different direction classes. On the other hand, our investigation of algorithm mixing, especially linear mixing, suggests the possibility of manipulating identification results while potentially increasing the image quality. This can be attributed to the independence of interpolation errors caused by individual interpolation algorithms, and one could effectively counter the identification by properly selecting the modification ratio, given the validity of error independence. With our work raising the awareness of these inherent and common issues of color interpolation identification, forensic researchers could improve identification techniques accordingly to combat cost-effective anti-forensics.

6 Conclusions

Identification of color interpolation has been shown to be a promising approach to assisting answering forensic questions about imaging devices and content. However, in order to ensure the trustworthiness of forensic identification especially in an adversarial environment, it is necessary to understand how color interpolation identification performs against anti-forensic operations that intentionally manipulates identification results.

In this paper, we have proposed two techniques for countering color interpolation identification. For the technique of parameter perturbation, we have examined options that achieve different trade-offs between two important factors, the image quality and the reduction in identification confidence. We show that perturbing the decision threshold values for pixel classification is a simple yet effective option for circumventing the identification. For the technique of algorithm mixing that fuses results from multiple algorithms, we have quantitatively compared different mixing settings and shown that we can further mislead the identification system while preserving the image quality.

To complement the randomized parameter perturbation technique, we have formulated it as an optimization problem and proposed a Monte-Carlo approach that maximizes individual image quality with the identification confidence kept low. We also compare our proposed anti-forensics with the most relevant work [6], and find that our approach has the advantages of lower complexity and better generalization capability. Based on the analysis presented in this paper, we shed light on the inherent issues of the current identification system that has performed well. Forensic researchers could use the understanding developed in this paper as guidelines to design more robust identification systems.

References

1. Adams, J.: Interaction between color plane interpolation and other image processing functions in electronic photography. In: Proc. of SPIE Cameras and Systems for Electronic Photography & Scientific Imaging (1995)
2. Adams, J., Hamilton, J.: Adaptive color plane interpolation in single sensor color electronic camera (US Patent #5,506,619) (1996)
3. Bayram, S., Sencar, H.T., Memon, N., Avcibas, I.: Source camera identification based on cfa interpolation. In: Proc. of International Conf. Image Proc. (2005)
4. Buades, A., Coll, B., Morel, J.M., Sbert, C.: Self-similarity driven color demosaicking. IEEE Trans. on Image Processing 18(6), 1192–1202 (2009)
5. Cao, H., Kot, A.C.: Accurate detection of demosaicing regularity for digital image forensics. IEEE Trans. on Information Forensics and Security 4(4), 899–910 (2009)
6. Kirchner, M., Böhme, R.: Synthesis of color filter array pattern in digital images. In: Proc. of SPIE-IS&T Electronic Imaging: Media Forensics and Security (2009)
7. Kirchner, M., Böhme, R.: Counter-Forensics: Attacking Image Forensics. In: Digital Image Forensics. Springer (2012)
8. Li, X., Gunturk, B., Zhang, L.: Image demosaicing: A systematic survey. In: Visual Communications and Image Processing, Proc. of the SPIE, vol. 6822 (2008)
9. Mislan, R.: Cellphone crime solvers. IEEE Spectrum 47(7), 34–39 (2010)
10. Paliy, D., Katkovnik, V., Bilcu, R., Alenius, S., Egiazarian, K.: Spatially adaptive color filter array interpolation for noiseless and noisy data. International Journal of Imaging Systems and Technology 17, 503–513 (2007)
11. Popescu, A.C., Farid, H.: Exposing digital forgeries in color filter array interpolated images. IEEE Trans. on Signal Proc. 53(10), 3948–3959 (2005)
12. Stamm, M.C., Liu, K.J.R.: Anti-forensics of digital image compression. IEEE Trans. on Information Forensics and Security 6(3), 1050–1065 (2011)
13. Swaminathan, A., Wu, M., Liu, K.J.R.: Non intrusive component forensics of visual sensors using output images. IEEE Trans. on Infor. Forensics and Security 2(1), 91–106 (2007)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Processing 13(4), 600–612 (2004)
15. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Elec. Imag. (023016) (2011)

Steganalysis of LSB Replacement Using Parity-Aware Features

Jessica Fridrich and Jan Kodovský

Department of ECE, Binghamton University, NY, USA
{fridrich,jan.kodovsky}@binghamton.edu

Abstract. Detection of LSB replacement in digital images has received quite a bit of attention in the past ten years. In particular, structural detectors together with variants of Weighted Stego-image (WS) analysis have materialized as the most accurate. In this paper, we show that further surprisingly significant improvement is possible with machine–learning based detectors utilizing co-occurrences of neighboring noise residuals as features. Such features can leverage dependencies among adjacent residual samples in contrast to the WS detector, which implicitly assumes that the residuals are mutually independent. Further improvement is achieved by adapting the features for detection of LSB replacement by making them aware of pixel parity. To this end, we introduce two key novel concepts – calibration by parity and parity-aware residuals. It is shown that, at least for a known cover source when a binary classifier can be built, its accuracy is markedly better in comparison with the best structural and WS detectors in both uncompressed images and in decompressed JPEGs. This improvement is especially significant for very small change rates. A simple feature selection algorithm is used to obtain interesting insight that reveals potentially novel directions in structural steganalysis.

1 Introduction

Least Significant Bit (LSB) replacement, also colloquially called LSB embedding, is arguably the oldest data hiding method. According to the CEO of WetStone Technologies, Inc., as of December 1, 2011 in their depository containing 836 data hiding products, 582 (70%) of them hide messages using LSB embedding. To the same day, the IEEE Xplore database registered 182 conference and 22 journal articles on LSB embedding, which further underlines the enormous popularity of this topic among researchers.

The first accurate detector of LSB replacement was the heuristic RS analysis [10] published in 2001, serendipitously discovered during research on reversible watermarking. The simplest case of RS analysis, the Sample Pairs (SP) analysis, was analyzed and reformulated by Dumitrescu et al. [5] into a framework amenable to further generalization and great improvement [6,4]. The least-squares version of SP by Lu *et al.* [24] later inspired further significant development mostly due to Ker, who derived the detectors from parity symmetries of natural images,

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 31–45, 2013.

extended the framework to triples [14] and quadruples of pixels [15], and provided further valuable insight [17,16,18].

In 2004, a different kind of LSB detector was introduced [9] that was later dubbed Weighted Stego-image (WS) analysis and further improved in [19] by introducing moderated weights, a better pixel predictor, and a simpler yet more accurate bias correction. The WS detector differed from the structural detectors in that it did not utilize trace sets but instead incorporated the parity through a pixel predictor. The improved version of the WS detector was shown to outperform all other structural attacks in raw, never compressed images, while the triples analysis was identified as the most accurate for decompressed JPEGs. An unweighted version of WS equipped with a recompression predictor was shown to be very effective in decompressed JPEGs provided the quantization table can be estimated [2].

Recently, the WS detector was rederived [26] using invariant hypothesis testing by adopting a parametric model for the cover. An Asymptotically Universally Most Powerful (AUMP) test that seems to coincide with a generalized likelihood ratio was derived in [7]. This detector is a variant of the WS analysis with weights that give it Constant False Alarm Rate (CFAR) property, which allows threshold setting independent of the image source. Finally, we point out that with the exception of [3,7], all LSB replacement detectors mentioned above are quantitative in the sense that the detection statistic is an estimate of the change rate.[1]

Steganalysis of embedding operations other than LSB flipping went in a different direction due to the fact that parity symmetries are no longer useful even for rather trivial modifications of LSB embedding, such as LSB matching. For such embedding operations, the most accurate detectors today are built as classifiers using features obtained as sampled joint distributions (co-occurrence matrices) among neighboring elements of noise residuals [12,11,27,25,13]. These detectors perform equally well for both LSB replacement and LSB matching because features formed from noise residuals are generally blind to pixels' parity.

In contrast to modern steganalysis features (briefly outlined in Section 2), the WS method, which also works with noise residuals, makes an implicit assumption that adjacent residual samples are independent (Section 3). This suggests a potential space for improvement, which we confirm in Section 4 with a simple four-dimensional co-occurrence matrix obtained from the same noise residual that is typically used with WS analysis. With the help of feature selection, improvement over the state of the art (triples analysis) is achieved with as few as three co-occurrence bins for decompressed JPEGs. Besides better utilization of spatial dependencies through co-occurrences, we introduce calibration by parity and parity-aware residuals as two general methods to make features aware of pixel parity to further improve their sensitivity to LSB replacement. By scaling up the feature space complexity using rich models, the best results of this paper are reported in Section 5. The paper is summarized in Section 6.

[1] Since the relationship between the relative payload and change rate depends on the syndrome coding method employed (see, e.g., Chapter 8 in [8]), everywhere in this paper we strictly speak of change-rate estimators.

1.1 Notation

We use boldface symbols for vectors and capital-case boldface symbols for matrices or higher-dimensional arrays. The symbols $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \mathcal{I}^{n_1 \times n_2}$ and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$, $\mathcal{I} = \{0, \ldots, 255\}$, will always represent pixel values of 8-bit grayscale cover and stego images with $n = n_1 n_2$ pixels; $\mathbf{X}^{\mathrm{T}}$ denotes the transpose. We use $\mathbb{R}$ and $\mathbb{Z}$ for the set of real numbers and integers. The operation of rounding $x \in \mathbb{R}$ to an integer is $\mathrm{round}(x)$. Given $T > 0$, $\mathrm{trunc}_T(x) = x$ when $x \in [-T, T]$, and $\mathrm{trunc}_T(x) = T\mathrm{sign}(x)$ otherwise. We also define for $x \in \mathbb{Z}$, $\mathrm{LSB}(x) = \mathrm{mod}(x, 2)$, $\bar{x} = x + 1 - 2\mathrm{LSB}(x)$, which is x with its LSB "flipped." The symbol β stands for the change rate defined as the ratio between the number of embedding changes and the number of pixels. We reserve $\Pr(\mathrm{E})$ for the probability of event E.

1.2 Setup of All Experiments

All experiments in this paper are carried out on BOSSbase ver. 0.92 [1] and its JPEG compressed versions obtained using the Matlab `imwrite` command. The original database contains $9,074$ 512×512 images acquired by seven digital cameras in the RAW format (CR2 or DNG) and subsequently processed by resizing and cropping to the size of 512×512 pixels.

The classifiers we use are all instances of the ensemble proposed in [22,21] and available from `http://dde.binghamton.edu/download/ensemble`. It employs Fisher linear discriminants as base learners trained on random subspaces of the feature space. The out-of-bag estimate of the testing error on bootstrap samples of the training set is used to automatically determine the random subspace dimensionality and the number of base learners as described in [22]. The final classifier decision is obtained by fusing the decisions of its base learners. We train a separate classifier for each image source and payload.

The detection accuracy is evaluated in a standard fashion using the minimal total detection error under equal priors computed from the ROC from the testing set:

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{P_{\mathrm{FA}} + P_{\mathrm{MD}}(P_{\mathrm{FA}})}{2}, \tag{1}$$

where P_{FA} is the false alarm rate and P_{MD} is the missed detection rate. What is reported in all graphs and tables is the average value of this error, $\bar{P}_{\mathrm{E}}$, over ten random divisions of the database into equally-sized training and testing sets. The spread of the error over the database splits also includes the effects of randomness in the ensemble construction (e.g., formation of random subspaces and bootstrap samples). We measure this spread using Mean Absolute Deviation (MAD) defined as the mean of $|P_{\mathrm{E}}(i) - \bar{P}_{\mathrm{E}}|$, where $P_{\mathrm{E}}(i)$ is the testing error on the ith database split.

2 Steganalysis Features

Modern steganalysis features are built as co-occurrence matrices from noise residuals. Below, we summarize the approach taken in [11]. Denoting an estimate of the cover image pixel x_{ij} from its neighborhood $\mathcal{N}(\mathbf{Y},i,j)$ as $\mathrm{Pred}(\mathcal{N}(\mathbf{Y},i,j))$, the noise residual, $\mathbf{Z}=(z_{ij})$,

$$z_{ij}=y_{ij}-\mathrm{Pred}(\mathcal{N}(\mathbf{Y},i,j)), \tag{2}$$

is quantized with a quantization step $q>0$ and truncated to a finite dynamic range $\mathcal{T}=\{-T,-T+1,\ldots,T\}$:

$$r_{ij}\triangleq \mathrm{trunc}_T\left(\mathrm{round}(z_{ij}/q)\right). \tag{3}$$

The statistical properties of $\mathbf{R}=(r_{ij})$ are captured as joint probability mass functions (pmfs) or co-occurrence matrices of m neighboring residual samples in the horizontal and vertical direction. The horizontal co-occurrence for residual $\mathbf{R}$ is

$$\mathbf{C}^{(\mathrm{h})}_{\mathbf{d}}=\Pr(r_{ij}=d_1\wedge\ldots\wedge r_{i,j+m-1}=d_m),\quad \mathbf{d}=(d_1,\ldots,d_m)\in\mathcal{T}^m, \tag{4}$$

while the vertical matrix, $\mathbf{C}^{(\mathrm{v})}_{\mathbf{d}}$, is defined analogically. Both have $(2T+1)^m$ elements.

Most pixel predictors are realized as shift-invariant finite-impulse response linear filters captured by a kernel matrix. For example, the kernel

$$\mathbf{K}=\begin{pmatrix}-0.25 & 0.5 & -0.25\\ 0.5 & 0 & 0.5\\ -0.25 & 0.5 & -0.25\end{pmatrix}, \tag{5}$$

proposed in [19] predicts the value of the central pixel from its local 3×3 neighborhood using the operation of convolution: $\mathbf{K}\star\mathbf{Y}$.

Symmetries are conveniently utilized to further reduce the dimensionality of the co-occurrences and to make them better populated. Given $\mathbf{d}\in\mathcal{T}^m$, we assume that for natural images $\mathbf{C}_{\mathbf{d}}\approx\mathbf{C}_{-\mathbf{d}}$ and $\mathbf{C}_{\mathbf{d}}\approx\mathbf{C}_{\overleftarrow{\mathbf{d}}}$, $\overleftarrow{\mathbf{d}}=(d_m,d_{m-1},\ldots,d_1)$. Symmetrization by sign means merging the bins $\mathbf{C}_{\mathbf{d}}+\mathbf{C}_{-\mathbf{d}}$, while symmetrization by direction requires merging $\mathbf{C}_{\mathbf{d}}+\mathbf{C}_{\overleftarrow{\mathbf{d}}}$.

For example, for $T=2$ and $m=4$, which are the parameters solely used in this paper, the original co-occurrence matrix, $\mathbf{C}_{\mathbf{d}}$, with $(2\times 2+1)^4=625$ elements is reduced to 325 elements using the directional symmetry or 338 elements using the sign symmetry. When both symmetrizations are applied, the dimension is reduced to 169.

3 Motivation

We now provide heuristic arguments for why detectors that utilize joint statistics of neighboring residual samples are likely to outperform variants of the WS analysis. It is because the WS detector can be derived from the assumption that the individual residual values are independent. Detailed technical arguments appear in [7] and require proper treatment of quantization effects. The author derives a CFAR variant of the WS detector starting with the independence assumption imposed on residual samples obtaining the detector in an asymptotic limit of infinite pixel bit-depth.

Deriving the detector while considering dependencies among residuals would require tackling the difficult problem of estimating the covariance between residuals as well as higher-order moments from a rather limited data. Instead, in this paper we represent groups of neighboring residual samples with co-occurrence matrices and use machine learning rather than the likelihood ratio test. While this approach is suboptimal, it is tractable and, as shown below, greatly improves the accuracy of all variants of WS.

Researchers have been aware for quite a long time that by leveraging the dependencies among neighboring residual samples,[2] one can obtain quite substantial improvement in detecting steganographic changes. Steganalyzers working with features formed as joint or transition probability distributions as features were shown to outperform [27,25,12,11,13] all previously proposed attacks on LSB matching and the content-adaptive HUGO. In summary, it makes perfect sense to expect that the accuracy of the WS detector can be improved as well by considering higher-order statistical constructs from the residual.

4 Making Features Parity Aware

Features computed from noise residuals, which are outputs of linear filters, such as (5), "do not see" pixel parity as this information is lost when, for example, taking a difference between two pixel values. This means that such features will detect LSB matching and LSB replacement with approximately the same accuracy.

We now describe several ways how to make the features parity aware. To this end, we introduce the following notation. For image $\mathbf{X} \in \mathcal{I}^n$, we denote by $\dot{\mathbf{X}}$, $\tilde{\mathbf{X}}$, $\bar{\mathbf{X}}$ the image $\mathbf{X}$ after setting all its LSBs to zero, randomizing all LSBs, and flipping all LSBs, respectively. Formally,

$$\begin{aligned} \dot{x}_{ij} &= x_{ij} - \mathrm{LSB}(x_{ij}), && (6)\\ \tilde{x}_{ij} &= \dot{x}_{ij} + \varphi, \quad \varphi \text{ r.v. uniform on } \{0,1\}, && (7)\\ \bar{x}_{ij} &= x_{ij} + 1 - 2\mathrm{LSB}(x_{ij}). && (8) \end{aligned}$$

The residuals of $\mathbf{X}$, $\dot{\mathbf{X}}$,$\tilde{\mathbf{X}}$, and $\bar{\mathbf{X}}$ will be denoted correspondingly as $\mathbf{R}$, $\dot{\mathbf{R}}$,$\tilde{\mathbf{R}}$, and $\bar{\mathbf{R}}$. In general, a feature computed from a residual $\mathbf{R}$ will be denoted as $\mathbf{f}(\mathbf{R})$.

Borrowing the idea from the WS detector, we define the concept of a "parity-aware residual." Given a residual $\mathbf{R} = (r_{ij})$, its parity-aware version is

$$\mathbf{R}^{(\pi)} = (r_{ij}^{(\pi)}), \quad r_{ij}^{(\pi)} = (1 - 2\mathrm{LSB}(x_{ij}))\, r_{ij}. \tag{9}$$

To make a feature vector of image $\mathbf{X}$ parity aware, one can follow the idea of Cartesian calibration [20] and augment it with a reference feature computed from $\dot{\mathbf{R}}$, $\tilde{\mathbf{R}}$, or $\bar{\mathbf{R}}$. We call this "calibration by parity." Additionally, we can compute the feature from the parity-aware residual, $\mathbf{f}(\mathbf{R}^{(\pi)})$.

[2] The dependencies are due to in-camera processing, such as denoising, filtering, color interpolation, and also due to the traces of content in the residual.

Table 1. Average detection error $\bar{P}_{\mathrm{E}}$ for LSB replacement with change rate $\beta = 0.01$ in uncompressed and JPEG 80 BOSSbase. Six different feature sets and their symmetrizations are tested; the last five are parity aware. The last set, $\mathbf{f}^{(663)}$, is the 663-dimensional merger of $[\mathbf{f}(\mathbf{R}), \mathbf{f}(\dot{\mathbf{R}})]$ and $\mathbf{f}(\mathbf{R}^{(\pi)})$ symmetrized as explained in the text.

	Source	JPEG 80				Uncompressed			
	Symm.	None	Both	Dir	Sign	None	Both	Dir	Sign
1	$\mathbf{f}(\mathbf{R})$	0.0164	0.0162	0.0158	0.0159	0.3261	0.3282	0.3246	0.3305
2	$[\mathbf{f}(\mathbf{R}), \mathbf{f}(\dot{\mathbf{R}})]$	0.0114	0.0103	0.0103	0.0106	0.1958	0.1971	0.1959	0.2007
3	$[\mathbf{f}(\mathbf{R}), \mathbf{f}(\tilde{\mathbf{R}})]$	0.0139	0.0130	0.0141	0.0135	0.2534	0.2524	0.2497	0.2531
4	$[\mathbf{f}(\mathbf{R}), \mathbf{f}(\bar{\mathbf{R}})]$	0.0128	0.0123	0.0129	0.0128	0.2239	0.2281	0.2242	0.2286
5	$\mathbf{f}(\mathbf{R}^{(\pi)})$	0.0165	0.0398	0.0163	0.0388	0.1253	0.3456	0.1249	0.3480
6	$\mathbf{f}^{(663)}$	0.0086				0.1154			

4.1 Testing

In the remainder of this section, we test the above features on BOSSbase and its JPEG compressed version to investigate the efficiency of calibration by parity and the parity-aware residual as well as the effect of symmetrization on detection performance for both types of features. Since these experiments are investigative in nature, they will be carried out only for one type of residual $\mathbf{R}$ obtained using the predictor $\mathbf{K}$ (5). The basic (parity-unaware) feature is

$$\mathbf{f}(\mathbf{R}) = \mathbf{C}_{\mathbf{d}}^{(\mathrm{h})} + \mathbf{C}_{\mathbf{d}}^{(\mathrm{v})}, \tag{10}$$

obtained as sum of the horizontal and vertical co-occurrences[3] with parameters $T = 2$ and $m = 4$, and with total dimensionality of 625 in its non-symmetrized version.

Table 1 shows $\bar{P}_{\mathrm{E}}$ on BOSSbase and its version compressed with JPEG quality 80. The results are for a fixed change rate $\beta = 0.01$, six different feature sets, and four types of symmetrization. As expected, the detection error is significantly lower for decompressed JPEGs than for uncompressed images. The symmetrization also has a very different impact on the features. In general, features computed from the parity-aware residual, $\mathbf{R}^{(\pi)}$, should be symmetrized only directionally but not by sign. The symmetrization has a much lesser impact on features calibrated by parity, for which both the directional and sign symmetries can be applied. The best calibration by parity is by zeroing out the LSB plane, i.e., $[\mathbf{f}(\mathbf{R}), \mathbf{f}(\dot{\mathbf{R}})]$. For JPEG images, this type of calibration gives the best results while features computed from the parity-aware residual are the best for uncompressed images. Finally, combining calibration by zeroing-out the LSBs with parity-aware residual is beneficial as can be seen from the last row ($\mathbf{f}^{(663)}$) showing the 663-dimensional merger of $[\mathbf{f}(\mathbf{R}), \mathbf{f}(\dot{\mathbf{R}})]$ symmetrized by both direction and sign with $\mathbf{f}(\mathbf{R}^{(\pi)})$ symmetrized directionally.

[3] The symmetry of the kernel $\mathbf{K}$ allows us to add both co-occurrences.

The fluctuations over the ten database splits are all statistically insignificant as the MAD of $P_{\mathrm{E}}(i)$ over the runs (not shown) was between 5×10^{-4} on JPEGs and 4×10^{-3} for uncompressed images.

4.2 Analysis by Cover Source

In this section, we apply feature selection to reveal several interesting facts about the detection of LSB replacement using parity-aware features from Table 1.

The dimensionality of $\mathbf{f}(\mathbf{R})$ and $[\mathbf{f}(\mathbf{R}), \mathbf{f}(\dot{\mathbf{R}})]$ symmetrized using both symmetries is $d = 169$ and 338, respectively, while the directionally-symmetrized $\mathbf{f}(\mathbf{R}^{(\pi)})$ has dimensionality of $d = 325$. We use a simple forward feature selection (FFS) method in which the features are selected sequentially one by one based on how much they improve the detection w.r.t. the union of those already selected. We start with the feature with the lowest individual detection error estimated from the training set. Having selected $k \geq 1$ features, the $k + 1$st feature is selected as the one among the $d - k$ remaining features that leads to the biggest drop in the error estimate when the union of all $k + 1$ features is used. This strategy continuously utilizes feedback of the ensemble classifier as it greedily minimizes the detection error in every iteration, taking thus the mutual dependencies among individual features into account. This is an example of a wrapper [23], which is a feature selection method using the machine-learning tool as a black-box and is thus classifier-dependent.

Decompressed JPEGs. We start with the source of JPEG compressed images. Table 2 (left) shows the results of the FFS when applied to the 169-dimensional feature vector $\mathbf{f}(\mathbf{R})$. We used a larger change rate $\beta = 0.02$ to make the effects more pronounced. The most remarkable phenomenon is the large decrease in detection error when the second bin is supplied to the best individual bin. While the second bin by itself has a very poor performance almost equal to random guessing, it extremely well *complements* the first bin. The error drops further with added bins but does so rather gradually after the initial drop. Note that the first bin corresponds to a residual four-tuple with large differences among neighboring samples. Such a group of values seems to be much less frequent in decompressed JPEGs than in their stego versions (c.f. the last column in the table) because the compression smooths the covers and thus empties this bin while the embedding repopulates it. The second bin serves as a reference, which is approximately invariant to embedding, and the pair together facilitates a very accurate detection. In fact, *all four* next selected bins, $k = 2, 3, 4, 5$, have a rather poor individual performance, suggesting that they all serve as different references to the first bin.

Remarkably, after merging only the first *three* bins, the cumulative error of 0.0215 is already lower than for the triples analysis – the best prior art performer (see Table 5). When all 169 features are used, the error drops further to 0.005. We remind that this result is obtained for a feature vector that is *unaware* of the pixel parity! Applying the FFS to $\mathbf{f}(\mathbf{R})$ Cartesian-calibrated by parity, $\mathbf{f}(\dot{\mathbf{R}})$, returns the same first four bins as for $\mathbf{f}(\mathbf{R})$, which is why we are not showing the results. This also implies that the main power of the detection is drawn from

Table 2. Forward feature selection strategy with change rate $\beta = 0.02$ in JPEG 80: cumulative and individual $\bar{P}_{\mathrm{E}}$, selected bins, and average bin count in cover/stego images. Left: symmetrized $\mathbf{f}(\mathbf{R})$, dimension 169. Right: directionally symmetrized $\mathbf{f}(\mathbf{R}^{(\pi)})$, dimension 325. The last row is obtained when all features are used.

k	$\bar{P}_{\mathrm{E}}^{(\mathrm{cum})}$	$\bar{P}_{\mathrm{E}}^{(\mathrm{ind})}$	Bin	Bin count	$\bar{P}_{\mathrm{E}}^{(\mathrm{cum})}$	$\bar{P}_{\mathrm{E}}^{(\mathrm{ind})}$	Bin	Bin count
1	0.2986	0.2986	(-1 2 -1 0)	1509/2291	0.2226	0.2226	(-1 -1 -1 0)	2950/5730
2	0.0377	0.4798	(-1 -1 1 0)	4878/5061	0.0370	0.4660	(0 0 1 0)	10130/9470
3	0.0215	0.4582	(-2 0 0 0)	2939/2746	0.0261	0.4712	(0 -1 -1 0)	3930/4190
4	0.0190	0.4721	(-2 0 -1 1)	940/989	0.0209	0.4433	(0 0 0 0)	116120/91530
5	0.0149	0.4761	(-1 2 -2 0)	2155/2262	0.0117	0.4970	(1 0 -2 2)	650/650
169	0.0050	-	-	-	-	-	-	-

the singular property of the cover source (compression "empties out" certain bins) rather than the parity asymmetry of LSB replacement. This is additionally confirmed by the fact that LSB matching can be detected with the same feature vector $\mathbf{f}(\mathbf{R})$ equally reliably as LSB replacement.

Furthermore, the best individual bin $(-1, 2, -1, 0)$ seems to be universal across sources of images with suppressed noise, which immediately disperses any thoughts that the co-occurrence bins might somehow utilize JPEG compatibility for detection. We confirmed this by repeating the same experiment with the feature vector $\mathbf{f}(\mathbf{R})$ for BOSSbase images denoised using the 3×3 Wiener filter with noise variance $\sigma^2 = 2, 5, 10$ and for BOSSbase denoised using the 3×3 median filter.[4]

The 325-dimensional feature vector $\mathbf{f}(\mathbf{R}^{(\pi)})$ obtained from the parity-aware residual exhibits a similar initial phenomenon, see Table 2 (right). The best individually performing bin is now different than in images with suppressed noise, which only strengthens our interpretation above.

Uncompressed Images. The second experiment was carried out on the uncompressed BOSSbase. In Table 3 (left), we report the results for the best-performing bins obtained from the parity-aware residual. Although the cumulative error now falls off much slower than for decompressed JPEGs, we again observe a large initial drop – the best individual performer is supplied with a reference bin that is by itself a random guesser. Interestingly, the second selected bin is the negative of the first bin. In fact, the same is true for the first eight selected bin pairs! To obtain insight into why the bins pair up in this manner, realize that $E[r_{ij}^{(\pi)}] = 0$ for unchanged pixels, while $E[r_{ij}^{(\pi)}] = -1$ whenever the pixel ij was changed. Thus, while both bins, $\mathbf{d}, -\mathbf{d} \in \mathcal{T}^4$, occur equally likely in covers, in stego images the one with more negative values is more populated than its negative counterpart. The reason why the boundary bin $(-2, -1, 0, 0)$ was chosen as the best can be explained by its population. While there are other good individual performers with individual errors in the range $P_{\mathrm{E}} \approx 0.42 - 0.45$, they are less populated.

[4] We used Matlab commands `wiener2` and `medfilt2`.

Table 3. Forward feature selection strategy for $\mathbf{f}(\mathbf{R}^{(\pi)})$, dimension 325, for change rate $\beta = 0.02$: cumulative and individual $\bar{P}_{\mathrm{E}}$, selected bins, and average bin count in cover/stego images. Left: uncompressed images. Right: denoised images. The last row is obtained when all features are used.

k	$\bar{P}_{\mathrm{E}}^{(\mathrm{cum})}$	$\bar{P}_{\mathrm{E}}^{(\mathrm{ind})}$	Bin	Bin count	Wie 2	Wie 5	Wie 10	Med
1	0.4126	0.4126	(-2 -1 0 0)	1353/1536	0.3277	0.2988	0.2536	0.2729
2	0.2164	0.4954	(2 1 0 0)	1323/1320	0.1130	0.0709	0.0620	0.0474
3	0.1810	0.4866	(-2 -2 -1 -2)	1912/1976	0.0226	0.0491	0.0365	0.0111
4	0.1489	0.4910	(2 2 1 2)	1901/1868	0.0223	0.0354	0.0293	0.0092
5	0.1438	0.4915	(-2 0 -2 -2)	1503/1478	0.0222	0.0299	0.0236	0.0087
325	0.0384	-	-	-	0.0172	0.0133	0.0110	0.0021

About 30 features are enough to obtain a lower detection error than the best structural performer – the WS analysis with moderated weights with bias correction (see Table 5).

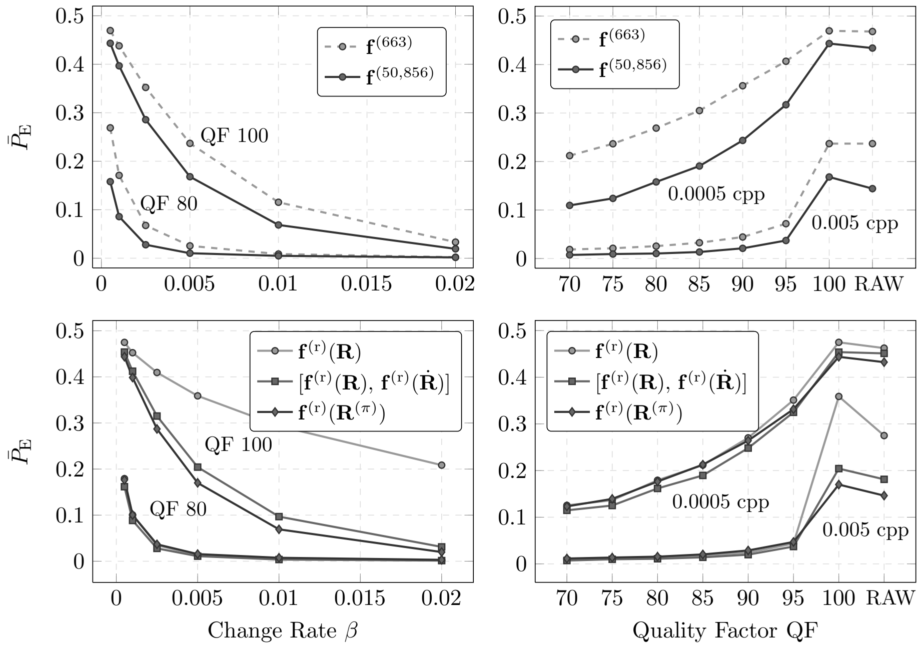


Fig. 1. Average detection error $\bar{P}_{\mathrm{E}}$ for different versions of the rich model (see text for details). Left: dependence on the change rate for two selected quality factors. Right: Dependence on the quality factor for two change rates.

Denoised Images. The last investigative experiment was carried out for four different versions of BOSSbase denoised using the 3 × 3 Wiener filter with

Table 4. Comparison of the average detection error $\bar{P}_{\mathrm{E}}$ for the best prior art detector, which is the triples analysis (Tr) and weighted stego-image with bias correction (WSb) marked by the symbol *, the feature $\mathbf{f}^{(663)}$ from Section 4, and the rich model $\mathbf{f}^{(50,856)}$

β	Det	70	75	80	85	90	95	100	UNCOMP.
0.0005	Tr/WSb	0.4022	0.4148	0.4190	0.4343	0.4464	0.4637	0.4767*	0.4776*
	$\mathbf{f}^{(663)}$	0.2121	0.2366	0.2689	0.3050	0.3563	0.4068	0.4695	0.4681
	$\mathbf{f}^{(50,856)}$	0.1095	0.1240	0.1579	0.1907	0.2435	0.3170	0.4433	0.4340
0.001	Tr/WSb	0.3168	0.3411	0.3521	0.3728	0.3961	0.4296	0.4547*	0.4536*
	$\mathbf{f}^{(663)}$	0.1270	0.1458	0.1709	0.2044	0.2558	0.3266	0.4380	0.4380
	$\mathbf{f}^{(50,856)}$	0.0610	0.0699	0.0858	0.1078	0.1439	0.2126	0.3968	0.3743
0.0025	Tr/WSb	0.1738	0.1973	0.2163	0.2372	0.2742	0.3350	0.3875*	0.3869*
	$\mathbf{f}^{(663)}$	0.0527	0.0575	0.0676	0.0816	0.1094	0.1704	0.3522	0.3522
	$\mathbf{f}^{(50,856)}$	0.0185	0.0245	0.0278	0.0365	0.0504	0.0869	0.2857	0.2512
0.005	Tr/WSb	0.0852	0.1014	0.1139	0.1283	0.1682	0.2346	0.2918*	0.2925*
	$\mathbf{f}^{(663)}$	0.0186	0.0211	0.0255	0.0325	0.0443	0.0718	0.2369	0.2369
	$\mathbf{f}^{(50,856)}$	0.0073	0.0092	0.0103	0.0134	0.0210	0.0371	0.1681	0.1441
0.01	Tr/WSb	0.0388	0.0464	0.0537	0.0628	0.0832	0.1341*	0.1697*	0.1662*
	$\mathbf{f}^{(663)}$	0.0045	0.0066	0.0086	0.0125	0.0186	0.0302	0.1154	0.1154
	$\mathbf{f}^{(50,856)}$	0.0027	0.0032	0.0049	0.0067	0.0113	0.0203	0.0686	0.0582
0.02	Tr/WSb	0.0199	0.0225	0.0268	0.0327	0.0430	0.0613	0.0675*	0.0664*
	$\mathbf{f}^{(663)}$	0.0009	0.0013	0.0021	0.0048	0.0079	0.0166	0.0332	0.0332
	$\mathbf{f}^{(50,856)}$	0.0010	0.0011	0.0017	0.0032	0.0066	0.0126	0.0193	0.0173

noise variance $\sigma^2 = 2, 5, 10$ and the 3×3 median filter. For the directionally-symmetrized $\mathbf{f}(\mathbf{R}^{(\pi)})$ we show in Table 3 (right) the cumulative detection error when selecting the five best bins using the FFS. The last row shows the detection error $\bar{P}_{\mathrm{E}}$ when using all 325 features $\mathbf{f}(\mathbf{R}^{(\pi)})$. The best performing bin was again $(-1, 2, -1, 0)$, as in case of decompressed JPEGs, with the exception of Wiener-filter images with $\sigma^2 = 2$ where the best bin was the same as the one found for uncompressed images. In all cases, we observed a sharp drop in detection error after the second bin is added to the best bin. Images processed by the median 3×3 filter appear to be particularly easy for detection of LSB replacement. For these four sources, the FFS did not seem to select the bins in pairs as observed for uncompressed images, which indicates that the detection utilizes the low level of noise of covers more than the singularity of LSB replacement.

5 Scaling Up the Image Model

In this section, we scale up our approach to the rich image model built in [11]. Due to the complexity of this model and the limited space in this paper, we cannot describe it here in detail and instead refer to the original publication. We use the predictors described in Section IV of [11] designed to better adapt to content around edges and in textures. The resulting set of 39 feature sets obtained with $T = 2$, $q = 1$, and $m = 4$ forms the rich model feature vector $\mathbf{f}^{(\mathrm{r})}$.

Table 5. Detection error $\bar{P}_{\mathrm{E}}$ for five structural detectors, six change rates, β, and eight cover sources: uncompressed BOSSbase (UNC) and its JPEG compressed versions using quality factors 70,75,. . .,100. Shaded in gray are the best results for each change rate. The acronyms are explained in Appendix A.

β	Det	70	75	80	85	90	95	100	UNC.
0.0005	SP	0.4725	0.4727	0.4752	0.4754	0.4792	0.4800	0.4849	0.4855
	WSb	0.4265	0.4323	0.4388	0.4477	0.4571	0.4642	0.4767	0.4776
	WS	0.4246	0.4240	0.4347	0.4422	0.4538	0.4635	0.4783	0.4768
	Tr	0.4022	0.4148	0.4190	0.4343	0.4464	0.4637	0.4853	0.4839
	AUMP	0.4564	0.4559	0.4620	0.4656	0.4698	0.4746	0.4805	0.4813
0.001	SP	0.4458	0.4448	0.4501	0.4510	0.4587	0.4626	0.4709	0.4719
	WSb	0.3717	0.3768	0.3879	0.3978	0.4124	0.4316	0.4547	0.4536
	WS	0.3580	0.3654	0.3768	0.3911	0.4086	0.4310	0.4548	0.4542
	Tr	0.3168	0.3411	0.3521	0.3728	0.3961	0.4296	0.4702	0.4673
	AUMP	0.4135	0.4139	0.4236	0.4317	0.4386	0.4514	0.4611	0.4614
0.0025	SP	0.3681	0.3768	0.3812	0.3854	0.3954	0.4066	0.4275	0.4255
	WSb	0.2639	0.2690	0.2809	0.2922	0.3124	0.3436	0.3875	0.3869
	WS	0.2356	0.2460	0.2630	0.2804	0.3069	0.3437	0.3878	0.3898
	Tr	0.1738	0.1973	0.2163	0.2372	0.2742	0.3350	0.4243	0.4185
	AUMP	0.3037	0.3056	0.3205	0.3392	0.3547	0.3812	0.4056	0.4044
0.005	SP	0.2766	0.2842	0.2909	0.2981	0.3106	0.3271	0.3595	0.3600
	WSb	0.1831	0.1838	0.1907	0.1990	0.2121	0.2386	0.2918	0.2925
	WS	0.1415	0.1563	0.1690	0.1848	0.2109	0.2392	0.2975	0.2939
	Tr	0.0852	0.1014	0.1139	0.1283	0.1682	0.2346	0.3548	0.3432
	AUMP	0.1962	0.2015	0.2153	0.2316	0.2494	0.2867	0.3256	0.3276
0.01	SP	0.1756	0.1802	0.1879	0.1949	0.2035	0.2195	0.2594	0.2576
	WSb	0.1083	0.1120	0.1164	0.1181	0.1251	0.1341	0.1697	0.1662
	WS	0.0730	0.0848	0.0935	0.1048	0.1232	0.1397	0.1770	0.1722
	Tr	0.0388	0.0464	0.0537	0.0628	0.0832	0.1377	0.2494	0.2383
	AUMP	0.1064	0.1081	0.1195	0.1316	0.1513	0.1818	0.2146	0.2162
0.02	SP	0.0916	0.0931	0.0989	0.0979	0.1094	0.1168	0.1447	0.1410
	WSb	0.0550	0.0565	0.0587	0.0592	0.0599	0.0613	0.0675	0.0664
	WS	0.0319	0.0359	0.0408	0.0494	0.0585	0.0676	0.0769	0.0714
	Tr	0.0199	0.0225	0.0268	0.0327	0.0430	0.0696	0.1392	0.1277
	AUMP	0.0498	0.0516	0.0563	0.0629	0.0790	0.1029	0.1231	0.1181

We test the following four versions of the rich model (the dimensionalities are in brackets):

1. $\mathbf{f}^{(\mathrm{r})}(\mathbf{R})$ symmetrized by both sign and direction (12,753);
2. $\mathbf{f}^{(\mathrm{r})}(\mathbf{R}^{(\pi)})$ symmetrized only directionaly (25,350);
3. $[\mathbf{f}^{(\mathrm{r})}(\mathbf{R}), \mathbf{f}^{(\mathrm{r})}(\dot{\mathbf{R}})]$ symmetrized by both sign and direction (25,506);
4. Merger of 2) and 3): $\mathbf{f}^{(50{,}856)} = [\mathbf{f}^{(\mathrm{r})}(\mathbf{R}), \mathbf{f}^{(\mathrm{r})}(\dot{\mathbf{R}}), \mathbf{f}^{(\mathrm{r})}(\mathbf{R}^{(\pi)})]$ (50,856).

Note that we do not symmetrize $\mathbf{f}^{(\mathrm{r})}(\mathbf{R}^{(\pi)})$ by sign as this would compromise its parity awareness as seen in Table 1.

Table 4 contrasts the performance of $\mathbf{f}^{(50,856)}$ with $\mathbf{f}^{(663)}$ and the best prior-art detectors from Table 5. The top two charts in Figure (1) show that the $\mathbf{f}^{(50,856)}$ model brings improvement over the 663-dimensional model especially for small change rates and high quality factors / uncompressed images. The two bottom charts inform us about the importance of making the feature vector $\mathbf{f}^{(\mathrm{r})}$ parity aware. The gain is the biggest for high-quality JPEGs and uncompressed images and it also increases with the change rate.

6 Conclusion

In 2005, the author of [14] expressed the following opinion about the state of the art in detection of LSB replacement: "... Because it makes full use of structural information, in some sense this framework [structural steganalysis] should be the last word on the detection of LSB replacement, although many practical questions remain open." In this paper, we challenge the supremacy of structural detectors and show that feature-based detectors with parity-aware features can significantly outperform all structural detectors as well as variants of WS analysis in both decompressed JPEG images and in uncompressed images. After all, it is only natural that the WS analysis with its limiting assumption of independent residual samples can be markedly improved as it has been shown in the literature before that utilizing dependencies in noise residual is quite important for detection of steganography.

Although the largest gain is demonstrated for high-dimensional rich models, state of the art can be outperformed using as few as three co-occurrence bins in decompressed JPEGs and thirty bins for uncompressed images. Our analysis shows that features built as co-occurrences of neighboring noise residuals are especially effective for detection in images with low level of noise, such as decompressed JPEGs or low-pass filtered images. In fact, here the detection strength is almost entirely in the peculiarity of the cover source rather than the asymmetry of the embedding operation (LSB replacement) as comparable detection accuracy can be obtained for LSB matching.

We introduce and study two general methods for making features parity aware – by calibration by parity (adding features computed from the image with zeroed-out LSBs) and by computing the features from a parity-aware residual. The latter is especially effective for steganalysis in uncompressed images.

Our approach has some obvious limitations imposed by the necessity to build a classifier. In particular, it is only feasible when sufficiently many images from a given source are available. For an unknown source, the accuracy of detection will undoubtedly be negatively affected by the mismatch between the training and testing data. Thus, for practical applications, quantitative LSB detectors and especially the CFAR detector of [7] will still be very important and useful tools. If the cover source is known, however, classifiers, such as those proposed here, offer a definitive advantage in terms of detection accuracy. The rich models, and in general any high-dimensional steganalysis, require extensive computing resources, which limits them to primarily off-line applications rather then real-time traffic monitoring. We note that the classifier training in high dimensions is quite feasible with tools, such as the ensemble classifier [22]. It is the time needed to compute the feature vector, that needs to be done for each analyzed image, that limits the practical use of such highly complex detectors.

Last but not least, our study seems to hint at new directions in structural steganalysis. We noticed a surprising universality across a wide spectrum of cover sources. Certain co-occurrence bins appear to be the overall best performers when accompanied with suitable reference features that by themselves are random guessers. In uncompressed images, bins of the parity-aware residual should be combined in mutually-negative pairs. A study with a simplified version of the residual, such as the second-order differences, may reveal well-defined flows between "trace sets" indexed by the residuals that might eventually lead to novel structural attacks. This work also reveals a possible way how to describe in a unified manner the WS analysis and structural detectors, which is a very exciting topic that we do not further elaborate on in this paper due to lack of space.

Acknowledgments. The work on this paper was partially supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of AFOSR or the U.S. Government. The authors would like to thank Vojtěch Holub, Miroslav Goljan, and Rainer Böhme for useful discussions, and Lionel Fillatre for help with implementing the AUMP detector.

A Prior Art

To establish a baseline and to identify the current most accurate LSB replacement detectors, we report here the results of five attacks that we consider state of the art: SP analysis [5], WS analysis with prediction kernel $\mathbf{K}$ (5) with moderated weights with (WSb) and without (WS) bias correction [19], triples analysis with $m, n \in \{-5, \ldots, 5\}$ (notation used as in [14]), and the AUMP detector [7] implemented with the recommended pixel block size $m = 16$, $q = 6$ (polynomial degree 5), and, per author's recommendation and in contrast to the paper, $\max\{1, \hat{\sigma}\}$ as an estimate of the standard deviation to assure numerical stability. The code for all detectors is available for download at: `http://dde.binghamton.edu/download/structural_lsb_detectors`.

Table 5 portrays triples analysis as the most accurate for decompressed JPEGs up to the quality factor of about 95 when it is outperformed by WSb, which is the best also for raw images. Our results for SP, WSb, WS, and triples seem compatible with previous art, at least as much as one can judge by results on different image sources. However, we observed a disturbingly large discrepancy between our results and what was reported on the *same image database* in [7] for WS as well as the SP. The author reports the entire ROC curves for relative payload $R = 0.05$, which corresponds to change rate $\beta = 0.025$ since the author is not considering any matrix embedding at the sender. Reading out the P_{E} from the ROC as the most distant point to the main diagonal in Fig. 5 in [7], the WS method and the weighted SP achieve $P_{\mathrm{E}} \approx 0.2$ and $P_{\mathrm{E}} \approx 0.45$, which is significantly worse than our results, $\bar{P}_{\mathrm{E}} = 0.0664$ and $\bar{P}_{\mathrm{E}} = 0.1410$, respectively, obtained for the change rate $\beta = 0.02$ (which is additionally slightly smaller than $R/2 = 0.025$).

References

1. Bas, P., Filler, T., Pevný, T.: "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 59–70. Springer, Heidelberg (2011)
2. Böhme, R.: Advanced Statistical Steganalysis. Springer, Heidelberg (2010)
3. Cogranne, R., Zitzmann, C., Fillatre, L., Retraint, F., Nikiforov, I., Cornu, P.: A Cover Image Model For Reliable Steganalysis. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 178–192. Springer, Heidelberg (2011)
4. Dumitrescu, S., Wu, X.: LSB steganalysis based on higher-order statistics. In: Proceedings of the 7th ACM Multimedia & Security Workshop, New York, August 1-2, pp. 25–32 (2005)
5. Dumitrescu, S., Wu, X., Memon, N.D.: On steganalysis of random LSB embedding in continuous-tone images. In: Proceedings IEEE, International Conference on Image Processing, ICIP 2002, Rochester, NY, September 22-25, pp. 324–339 (2002)
6. Dumitrescu, S., Wu, X., Wang, Z.: Detection of LSB steganography via sample pair analysis. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 355–372. Springer, Heidelberg (2003)
7. Fillatre, L.: Adaptive steganalysis of least significant bit replacement in grayscale images. IEEE Transactions on Signal Processing 60, 556–569 (2012)
8. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press (2009)
9. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI, San Jose, CA, January 19–22, vol. 5306, pp. 23–34 (2004)
10. Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in grayscale and color images. In: Proceedings of the ACM, Special Session on Multimedia Security and Watermarking, Ottawa, Canada, October 5, pp. 27–30 (2001)
11. Fridrich, J., Kodovský, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security (to appear, 2012)
12. Fridrich, J., Kodovský, J., Holub, V., Goljan, M.: Steganalysis of Content-Adaptive Steganography in Spatial Domain. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 102–117. Springer, Heidelberg (2011)
13. Gul, G., Kurugollu, F.: A New Methodology in Steganalysis: Breaking Highly Undetectable Steganograpy (HUGO). In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 71–84. Springer, Heidelberg (2011)
14. Ker, A.D.: A General Framework for Structural Steganalysis of LSB Replacement. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 296–311. Springer, Heidelberg (2005)

15. Ker, A.D.: Fourth-order structural steganalysis and analysis of cover assumptions. In: Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, January 16–19, vol. 6072, pp. 25–38 (2006)
16. Ker, A.D.: A Fusion of Maximum Likelihood and Structural Steganalysis. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 204–219. Springer, Heidelberg (2008)
17. Ker, A.D.: Optimally weighted least-squares steganalysis. In: Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, January 29-February 1, vol. 6505, pp. 6-1–6-16 (2007)
18. Ker, A.D.: Steganalysis of embedding in two least significant bits. IEEE Transactions on Information Forensics and Security 2, 46–54 (2007)
19. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA, January 27–31, vol. 6819, pp. 5-1–5-17 (2008)
20. Kodovský, J., Fridrich, J.: Calibration revisited. In: Proceedings of the 11th ACM Multimedia & Security Workshop, Princeton, NJ, September 7–8, pp. 63–74 (2009)
21. Kodovský, J., Fridrich, J.: Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In: Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics of Multimedia XIII, San Francisco, CA, January 23–26, vol. 7880, pp. OL 1–OL 13 (2011)
22. Kodovský, J., Fridrich, J., Holub, V.: Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security 7(2), 432–444 (2012)
23. Lal, T.N., Chapelle, O., Weston, J., Elisseeff, A.: Embedded Methods. In: Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. (eds.) Feature Extraction: Foundations and Applications. STUDFUZZ, vol. 207, pp. 137–165. Springer, Heidelberg (2006)
24. Lu, P., Luo, X., Tang, Q., Shen, L.: An Improved Sample Pairs Method for Detection of LSB Embedding. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 116–127. Springer, Heidelberg (2004)
25. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
26. Zitzmann, C., Cogranne, R., Retraint, F., Nikiforov, I., Fillatre, L., Cornu, P.: Statistical Decision Methods in Hidden Information Detection. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 163–177. Springer, Heidelberg (2011)
27. Zo, D., Shi, Y.Q., Su, W., Xuan, G.: Steganalysis based on Markov model of thresholded prediction-error image. In: Proceedings IEEE, International Conference on Multimedia and Expo, Toronto, Canada, July 9-12, pp. 1365–1368 (2006)

Statistical Detection of LSB Matching Using Hypothesis Testing Theory

Rémi Cogranne, Cathel Zitzmann, Florent Retraint,
Igor Nikiforov, Lionel Fillatre, and Philippe Cornu*

ICD, LM2S, Université de Technologie de Troyes, UMR STMR CNRS
12, rue Marie Curie, B.P. 2060, 10010 Troyes cedex, France
name.surname@utt.fr

Abstract. This paper investigates the detection of information hidden by the Least Significant Bit (LSB) matching scheme. In a theoretical context of known image media parameters, two important results are presented. First, the use of hypothesis testing theory allows us to design the Most Powerful (MP) test. Second, a study of the MP test gives us the opportunity to analytically calculate its statistical performance in order to warrant a given probability of false-alarm. In practice when detecting LSB matching, the unknown image parameters have to be estimated. Based on the local estimator used in the Weighted Stego-image (WS) detector, a practical test is presented. A numerical comparison with state-of-the-art detectors shows the good performance of the proposed tests and highlights the relevance of the proposed methodology.

1 Introduction and Contributions

Steganography and steganalysis form a cat-and-mouse game. On the one hand, steganography aims at hiding the very presence of a secret message by hiding it within an innocuous cover medium. On the other hand, the goal of steganalysis (in the wide sense) is to obtain any information about the potential steganographic system from an unknown medium. Usually, steganalysis focuses on exposing the existence of a hidden message in an inspected medium.

Many steganographic tools are nowadays easily available on the Internet making steganography within the reach of anyone, for legitimate or malicious usage. It is thus crucial for security forces to be able to reliably detect steganographic content among a (possibly very large) set of media files. In this operational context, the detection of a rather simple but most commonly found stegosystem seems more important than the detection of a very complex but rarely

* This work was partially supported by National Agency for Research (ANR) through ANR-CSOSG Program (Project ANR-07-SECU-004). With the financial support from the Prevention of and Fight against Crime Programme of the European Union European Commission - Directorate-General Home Affairs. (2centre.eu project).Research partially funded by Troyes University of Technology (UTT) strategic program COLUMBO.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 46–62, 2013.

encountered stegosystem. The vast majority of downloadable steganographic tools insert the secret information in the LSB plane. Consequently, substantial progress has recently been made in the detection of such steganographic algorithms, namely LSB replacement and LSB matching, also known as LSB ± 1 embedding (see [11,15,1] and the references therein). However, the steganalysis of LSB matching remains much harder than the steganalysis of LSB replacement. Indeed, if LSB matching is used instead of LSB replacement, the detection power of state-of-the-art detectors is significantly lower [25,5].

The recently proposed steganalyzers dedicated to LSB matching can be roughly divided into two categories. On the one hand, most of the latest detectors are based on supervised machine learning methods and use targeted [6,4] or universal features [17,23]. As in all applications of machine learning, the theoretical calculation of error probabilities remains an open problem [24]. On the other hand, the authors of [18] observed that LSB matching acts as a low-pass filter on the image Histogram Characteristic Function (HCF). This pioneering work lead to an entire family of histogram-based detectors [19,25].

In the operational context described above, the proposed steganalyzer must be immediately applicable without any training or tuning phase. For this reason, the use of a machine learning based detector is hardly possible. Moreover, the most important challenge for the steganalyst is to provide detection algorithms with an analytical expression for the false-alarm and missed-detection probabilities without which the "uncertainty" of the result can not be "measured." The proposed LSB matching steganalyzers are certainly very interesting and efficient, but these *ad hoc* algorithms have been designed with a very limited exploitation of statistical cover models and hypothesis testing theory. Hence, a few theoretical results exist and the only solution to measure their statistical performance is the simulation on large databases.

Alternatively, the first step in the direction of hypothesis testing has been made in [12,8,9] for LSB replacement to design a statistical test with known statistical properties. In the present paper, this statistical approach is extended to the case of detecting LSB matching. More precisely, the goal of this paper is threefold:

1. Define the most powerful (MP) test in the theoretical case when the cover image parameters are known, namely the expectation and noise variance of each pixel.
2. Analytically calculate the statistical performance of the MP test in terms of the false-alarm and missed-detection probabilities. More importantly, this result allows us to highlight the impact of the noise variance and quantization on the test performance [9].
3. Design a practical efficient implementation of this test based on a simple local estimation of expectation and variance of each pixel.

The paper is organized as follows. The problem of LSB matching steganalysis is casted within the framework of hypothesis testing in Section 2. Following the Neyman-Pearson approach, the MP Likelihood Ratio Test (LRT) is presented in Section 3 and its statistical performance is calculated in Section 4. Finally, the proposed practical implementation of the Generalized LRT (GLRT) is presented in Section 5. To show the relevance of the proposed approach, numerical results on large natural image databases are shown in Section 6. Section 7 concludes the paper.

2 Detection of LSB Matching Problem Statement

This paper mainly focuses on natural images but the extension of the presented results to any kind of digital media is immediate. Hence, the column vector $\boldsymbol{C} = (c_1, \dots, c_N)^T$ represents in this paper a cover image of $N = N_x \times N_y$ grayscale pixels. The set of grayscale levels is denoted $\mathcal{Z} = \{0; \dots; 2^{B-1}\}$ as pixels values are usually unsigned integers encoded with B bits. Each cover pixel c_n results from the quantization:

$$c_n = Q(y_n), \tag{1}$$

where $y_n \in \mathbb{R}^+$ denotes the raw pixel intensity recorded by the camera and Q represents the uniform quantization with a unitary step:

$$Q(x) = k \Leftrightarrow x \in [k - 1/2\,;\, k + 1/2[.$$

Seeking simplicity, it is assumed in this paper that the saturation effect is absent, *i.e.* the probability of excessing the quantizer boundaries $-1/2$ and $2^{B-1}+1/2$ is negligible. Indeed, taking into account the under or over-exposed pixels is rather simple but requires a much more complicated notation.
The recorded pixel value can be decomposed as [13,7]:

$$y_n = \theta_n + \xi_n, \tag{2}$$

where θ_n is a deterministic parameter corresponding to the mathematical expectation of y_n and ξ_n is a random variable representing all the noise corrupting the cover image during acquisition. As described in [13], ξ_n is accurately modeled as a realization of a zero-mean Gaussian random variable $\varXi_n \sim \mathcal{N}(0, \sigma_n^2)$ whose variance σ_n^2 varies from pixel to pixel. It thus follows from (1) and (2) that c_n follows a distribution $P_{\theta_n} = P_{\theta_n,\sigma_n} = (p_{\theta_n}[0], \dots, p_{\theta_n}[2^{B-1}])$ defined by:

$$\forall k \in \mathcal{Z}\ ,\ p_{\theta_n}[k] = \varPhi\left(\frac{k + {}^1\!/_2 - \theta_n}{\sigma_n}\right) - \varPhi\left(\frac{k - {}^1\!/_2 - \theta_n}{\sigma_n}\right), \tag{3}$$

with $\varPhi$ is the standard Gaussian cumulative distribution function (cdf) defined by $\varPhi(x) = \int_{-\infty}^{x} \phi(u)du$ and ϕ the standard Gaussian probability distribution function (pdf) $\phi(u) = \frac{1}{\sqrt{2\pi}} \exp(u^2/2)$. In virtue of the mean value theorem, (3) can be written as:

$$p_{\theta_n}[k] = \frac{1}{\sigma_n} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \phi\left(\frac{u - \theta_n}{\sigma_n}\right) du = \phi\left(\frac{k - \theta_n}{\sigma_n} + \epsilon\right), \tag{4}$$

where ϵ is a (small) corrective term [26].

To statistically model stego-image pixels from (3)–(4), the two following assumptions are usually adopted [12,14] : 1) the probability of insertion is equal for every cover pixel (independence between hidden bits and cover pixels) and 2) the message is assumed compressed and/or cyphered $\boldsymbol{M} = (m_1, \dots, m_L)^T$ before insertion. Hence, each hidden bit m_l is drawn from a binomial distribution $\mathcal{B}(1, 1/2)$, *i.e.* m_l is either 0 or 1 with the same probability. This situation is captured by denoting

$$\forall n \in \{0, \dots, N\}, \begin{cases} \mathbb{P}[s_n = c_n] = (1-R), \\ \mathbb{P}[s_n = c_n + \mathrm{ins}(m_n, c_n)] = R, \end{cases} \tag{5}$$

where $\boldsymbol{S} = \{s_1, \dots, s_N\}$ are the values of stego-image pixels, the embedding rate $R = L/N$ corresponds to the number of hidden bits per cover pixel and $\mathrm{ins}(m_n, c_n)$ represents the value added to c_n to insert the hidden bit m_n.

The particularity of LSB matching lies in its insertion function $\mathrm{ins} : \{0; 1\} \times \mathcal{Z} \mapsto \{-1; 0; 1\}$. Whenever the LSB of c_n is equal to m_n, *i.e.* when $\mathrm{lsb}(c_n) = c_n \mathrm{mod} 2 = m_n$, there is no need to change c_n, hence $\mathrm{ins}(m_n, c_n) = 0$. On the contrary, whenever $\mathrm{lsb}(c_n) \neq m_n$, the insertion must change the LSB of c_n, which is done by adding or subtracting 1 with the same probabilities:

$$\begin{cases} \mathbb{P}[\mathrm{ins}(b_s, c_n) = 1 \,|\, \mathrm{lsb}(c_n) \neq m_n] \;\;= 1/2 \\ \mathbb{P}[\mathrm{ins}(b_s, c_n) = -1 \,|\, \mathrm{lsb}(c_n) \neq m_n] = 1/2. \end{cases} \tag{6}$$

Since each hidden bit m_n follows the binomial distribution $\mathcal{B}(1, 1/2)$, a straightforward calculation finally shows that $\mathbb{P}[\mathrm{lsb}(c_n) = m_n] = \mathbb{P}[\mathrm{lsb}(c_n) \neq m_n] = 1/2$. Hence, as described in [18,25,6,10], it follows from (5)–(6) that for all $n \in \{1, \dots, N\}$, the pmf of the stego-pixel s_n after embedding at rate R with LSB matching is given by $Q_{\theta_n}^R = \left(q_{\theta_n}^R[0], \dots, q_{\theta_n}^R[2^b - 1]\right)$ with $\forall k \in \mathcal{Z}$:

$$q_{\theta_n}^R[k] = \frac{R}{4}\left(p_{\theta_n}[k-1] + p_{\theta_n}[k+1]\right) + \left(1 - \frac{R}{2}\right) p_{\theta_n}[k]. \tag{7}$$

3 Likelihood Ratio Test (LRT) for Two Simple Hypotheses

When analyzing an unknown medium $\mathbf{Z}$ the first goal of LSB matching steganalysis is to decide between the two following hypotheses:

$$\begin{array}{rl} & \mathcal{H}_0 = \{z_n \sim P_{\theta_n}, \forall n \in \{1, \dots, N\}\} \\ \text{vs} & \mathcal{H}_1 = \{z_n \sim Q_{\theta_n}^R, \forall n \in \{1, \dots, N\}\}. \end{array} \tag{8}$$

Let us start with the simplest case, when the embedding rate R and, for all n, the parameters θ_n and σ_n are known. In this case, the hypothesis testing problem (8) is reduced to a test between two simple hypotheses.

The goal is obviously to find a test $\delta : \mathcal{Z}^N \mapsto \{\mathcal{H}_0, \mathcal{H}_1\}$, such that hypothesis $\mathcal{H}_i$ is accepted if $\delta(\mathbf{Z}) = \mathcal{H}_i$ (see [22] for details about statistical hypothesis testing). However, as explained in the introduction, in an operational forensics context the most important challenge is first, to warrant a prescribed (very low) false-alarm probability and second, to maximize the detection power defined by:

$$\beta_\delta = \mathbb{P}_1[\delta(\mathbf{Z}) = \mathcal{H}_1],$$

where $\mathbb{P}_i(\cdot)$ stands for the probability under hypotheses $\mathcal{H}_i$, $i = \{0; 1\}$. Therefore, let $\mathcal{K}_\alpha$ be the class of tests with an upper-bounded false-alarm probability α_0 defined by

$$\mathcal{K}_\alpha = \{\delta : \mathbb{P}_0[\delta(\mathbf{Z}) = \mathcal{H}_1] \leq \alpha_0\} . \tag{9}$$

In virtue of the Neyman-Pearson lemma, see [22, Theorem 3.2.1], the most powerful (MP) test over the class $\mathcal{K}_{\alpha_0}$ (9) is the LRT given by the following decision rule:

$$\delta_R(\mathbf{Z}) = \begin{cases} \mathcal{H}_0 & \text{if } \Lambda_R(\mathbf{Z}) \leq \tau_{\alpha_0} \\ \mathcal{H}_1 & \text{if } \Lambda_R(\mathbf{Z}) > \tau_{\alpha_0}, \end{cases} \tag{10}$$

where τ_{α_0} is the solution of $\mathbb{P}_0[\delta(\mathbf{Z}) > \tau_{\alpha_0}] = \alpha_0$, to insure that $\delta_R \in \mathcal{K}_{\alpha_0}$, and the likelihood ratio (LR) $\Lambda_R(\mathbf{Z})$ is given, from the statistical independence between pixels, by:

$$\Lambda_R(\mathbf{Z}) = \prod_{n=1}^{N} \Lambda_R(z_n) = \prod_{n=1}^{N} \frac{R}{4} \frac{p_{\theta_n}[z_n - 1] + p_{\theta_n}[z_n + 1]}{p_{\theta_n}[z_n]} + \left(1 - \frac{R}{2}\right) . \tag{11}$$

It can be noted that $\Lambda_R(z_n)$ depends on pixel values z_n through the quantity:

$$\Lambda_2(z_n) = \frac{1}{2} \frac{p_{\theta_n}[z_n - 1] + p_{\theta_n}[z_n + 1]}{p_{\theta_n}[z_n]}, \tag{12}$$

which corresponds to the the likelihood ratio for the conceptual case of $R = 2$. In other words, Equation (12) corresponds to this test: $\mathcal{H}_0 : \{ \mathbf{Z} \text{ is a cover medium} \}$ vs $\mathcal{H}_1 : \{ \text{each pixel of } \mathbf{Z} \text{ is modified by} \pm 1 \}$. Indeed, considering the case $R\!=\!2$ permits us to clarify the present methodology, which is then extended to the more general case of $R \in]0; 1[$ in Section 4.2.

The exact expression for the LR $\Lambda_2(z_n)$ is complicated due to the corrective terms ϵ defined in (4). However, the calculation shows that these corrective terms are usually negligible, particularly when $\sigma_n > 1$. Therefore, it is proposed to neglect ϵ in order to obtain a simplified expression for the LR $\Lambda_2(z_n)$. From (4), this approximation permits us to write:

$$\begin{aligned} \frac{p_{\theta_n}[z_n - 1]}{p_{\theta_n}[z_n]} &= \exp\left(-\frac{1}{2\sigma_n^2}\right) \exp\left(\frac{\theta_n - z_n}{\sigma_n^2}\right), \\ \frac{p_{\theta_n}[z_n + 1]}{p_{\theta_n}[z_n]} &= \exp\left(-\frac{1}{2\sigma_n^2}\right) \exp\left(\frac{z_n - \theta_n}{\sigma_n^2}\right). \end{aligned} \tag{13}$$

Finally, using (13), the LR $\Lambda_2(z_n)$ can be written as:

$$\Lambda_2(z_n) = \frac{1}{4} \exp\left(\frac{-1}{2\sigma_n^2}\right) \left[\exp\left(\frac{z_n - \theta_n}{\sigma_n^2}\right) + \exp\left(\frac{\theta_n - z_n}{\sigma_n^2}\right)\right] . \tag{14}$$

The logarithm of the likelihood ratio (15) is usually preferred in order to replace the product in (11) with a sum. From (14), it immediately follows that:

$$\widetilde{\Lambda}_2(z_n) \overset{\text{def.}}{=} \log\left[\exp\left(\frac{z_n-\theta_n}{\sigma_n^2}\right)+\exp\left(\frac{\theta_n-z_n}{\sigma_n^2}\right)\right] \tag{15}$$
$$= \log\big(\Lambda_2(z_n)\big)+\log(2)+\frac{1}{2\sigma_n^2}.$$

Again, one can note that the terms log(4) and $\frac{1}{2\sigma_n^2}$ do not depend on the true hypothesis. That is why, for the same reasons as those discussed in connection with Equation (12), these terms do not play any role in solving the detection problem (8). For the sake of clarity, these terms are thus omitted from expression (15) of the log-LR $\widetilde{\Lambda}_2(z_n)$.

4 Statistical Performance of the LR Test

4.1 Case of Simple Hypotheses, When $R = 2$

In this section it is first proposed to study the statistical performance for the case of simple hypotheses, when $R = 2$. The results are then extended to the general case of $R \in]0;1[$ in Section 4.2. To easily calculate the statistical performance of the LR test δ_R (10), the asymptotic approach is of crucial interest. Moreover, the assumption that N grows to infinity is relevant in practice due to the very large number of pixels in typical images.

For the sake of clarity, let the mean expectation and the mean variance of $\widetilde{\Lambda}_2(z_n)$ under hypotheses $\mathcal{H}_i$ be defined as follows:

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_i\left[\widetilde{\Lambda}_2(z_n)\right] \quad \text{and} \quad \sigma_i^2 = \frac{1}{N}\sum_{n=1}^{N}\mathbb{V}ar_i\left[\widetilde{\Lambda}_2(z_n)\right], \tag{16}$$

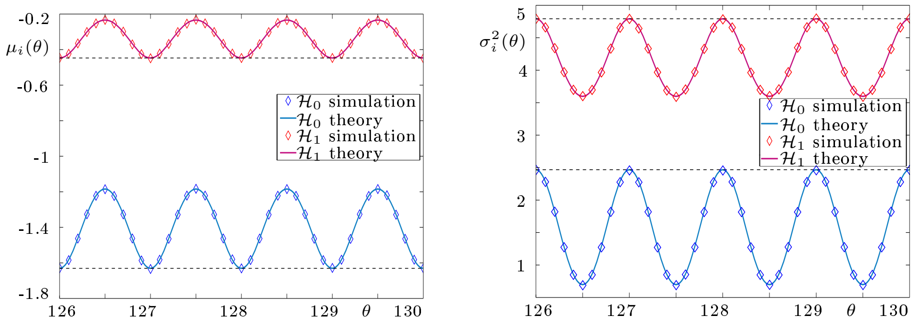
where $\mathbb{E}_i\big[\widetilde{\Lambda}_2(\mathbf{Z})\big]$ and $\mathbb{V}ar_i\big[\widetilde{\Lambda}_2(\mathbf{Z})\big]$ are respectively the expectation and the variance of $\widetilde{\Lambda}_2(z_n)$ under hypotheses $\mathcal{H}_i$, $i = \{0,1\}$.

The test $\widetilde{\delta}_2$ associated with the "normalized" log-LR $\widetilde{\Lambda}_2(\mathbf{Z})$ is defined as:

$$\widetilde{\delta}_2 = \begin{cases}\mathcal{H}_0 & \text{if} \quad \widetilde{\Lambda}_2(\mathbf{Z}) \le \widetilde{\tau}_{\alpha_0},\\ \mathcal{H}_1 & \text{if} \quad \widetilde{\Lambda}_2(\mathbf{Z}) > \widetilde{\tau}_{\alpha_0}.\end{cases} \quad \text{where} \quad \widetilde{\Lambda}_2(\mathbf{Z}) \overset{\text{def.}}{=} \frac{\sum\limits_{n=1}^{N}\widetilde{\Lambda}_2(z_n) - N\mu_0}{\sqrt{N\sigma_0^2}}, \tag{17}$$

It can noted that the random variables $\widetilde{\Lambda}_2(z_n)$ are assumed statistically independent and, for any $\sigma_n > 0$, have finite expectation and variance, which implies that the conditions necessary for application of the Lindeberg's central limit theorem [22, Theorem 11.2.5] are satisfied. These conditions can also be shown by using the fact that z_n are bounded because they can only take values in the set $\mathcal{Z}$. Therefore,

$$\widetilde{\Lambda}_2(\mathbf{Z}) \rightsquigarrow \begin{cases}\mathcal{N}(0\,,\,1) & \text{under} \quad \mathcal{H}_0\\ \mathcal{N}\left(\dfrac{\sqrt{N}(\mu_2-\mu_0)}{\sigma_0}\,,\,\dfrac{\sigma_2^2}{\sigma_0^2}\right) & \text{under} \quad \mathcal{H}_1.\end{cases} \tag{18}$$



(a) Expectations of the log-LR $\log\big(\Lambda_2(z_n)\big)$ as a function of expectation θ.

(b) Variances of the log-LR $\log\big(\Lambda_2(z_n)\big)$ as a function of expectation θ.

Fig. 1. Graphical representation of the two first moments of log-LR $\log\big(\Lambda_2(z_n)\big)$ (20) - (23). Presented results correspond to the case of i.i.d pixels with expectation $\theta_n \in [126; 130]$ and standard deviation $\sigma_n = 0.75$.

where $\rightsquigarrow$ represents the convergence in distribution as $N \to \infty$. From Equation (18), a short algebra establishes the following theorem.

Theorem 1. *For any given probability of false alarm $\alpha_0 \in]0;1[$, the decision threshold $\widetilde{\tau}_{\alpha_0}$ given by:*

$$\widetilde{\tau}_{\alpha_0} = \Phi^{-1}(1-\alpha_0) \tag{19}$$

where $\Phi^{-1}(\cdot)$ is the Gaussian inverse cumulative distribution, asymptotically warrants that the test $\widetilde{\delta}_2$ (17) is in $\mathcal{K}_{\alpha_0}$.

The main conclusion of Theorem 1 is that the decision threshold $\widetilde{\tau}_{\alpha_0}$ depends neither on the embedding rate R nor the image parameters θ_n and σ_n. Hence, by using the "normalized" log-LR $\widetilde{\Lambda}_2(\mathbf{Z})$, the same threshold permits us to respect a prescribed false-alarm probability α_0 whatever the analyzed image and the embedding rate are.

Equation (18) also implies that to asymptotically calculate the detection power of LR test $\widetilde{\delta}_2$ (17), one only needs to calculate the first moments of $\widetilde{\Lambda}_2(\mathbf{Z})$. The mean expectations used in the log-LR $\widetilde{\Lambda}_2(z_n)$ are given under hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ by

$$\mu_0 = \frac{1}{N}\sum_{n=1}^{N}\sum_{k\in\mathcal{Z}} p_{\theta_n}[k]\log\left(\exp\left(\frac{k-\theta_n}{\sigma_n^2}\right)+\exp\left(\frac{\theta_n-k}{\sigma_n^2}\right)\right), \tag{20}$$

$$\mu_2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{k\in\mathcal{Z}} q_{\theta_n}^{R}[k]\log\left(\exp\left(\frac{k-\theta_n}{\sigma_n^2}\right)+\exp\left(\frac{\theta_n-k}{\sigma_n^2}\right)\right), \tag{21}$$

where the probabilities $p_{\theta_n}[k]$ and $q^R_{\theta_n}[k]$ are respectively defined in (3) and (7). Similarly, the mean variances are by definition given under both hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ by:

$$\sigma_0^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{k\in\mathcal{Z}} p_{\theta_n}[k]\log\left(\exp\left(\frac{k-\theta_n}{\sigma_n^2}\right)+\exp\left(\frac{\theta_n-k}{\sigma_n^2}\right)\right)^2 - \mu_0^2, \tag{22}$$

$$\sigma_2^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{k\in\mathcal{Z}} q^R_{\theta_n}[k]\log\left(\exp\left(\frac{k-\theta_n}{\sigma_n^2}\right)+\exp\left(\frac{\theta_n-k}{\sigma_n^2}\right)\right)^2 - \mu_2^2. \tag{23}$$

The expectations μ_0 and μ_2 and the variances σ_0^2 and σ_2^2 as functions of θ_n are respectively drawn in Figures 1a and 1b. These figures highlight the fact that the pixel expectation θ_n can have a significant impact on the LR moments, and later on the detection power, particularly when $\sigma_n < 1$. However, a thorough study of equations (20)–(23) shows that this phenomenon rapidly tends to be negligible when $\sigma_n \gtrsim 1$.

Even thoug, the moments given in (20)–(23) have a rather complicated expression, their numerical calculation is straightforward as long as the parameters θ_n and σ_n are known.

From the asymptotic distribution (18) of the log-LR $\widetilde{\Lambda}_2(\mathbf{Z})$ and the expressions (20)–(23) of its two first moments, the detection power of the LR test $\widetilde{\delta}_2$ (17) is given by the following theorem.

Theorem 2. *For any $\alpha_0 \in]0;1[$, assuming that the parameters $\{\theta_n\}_{n=1}^N$ and $\{\sigma_n\}_{n=1}^N$ are known, the power function $\widetilde{\beta}_{\delta_2}$ associated with the test $\widetilde{\delta}_2$ (17) is asymptotically given, as $N \to \infty$, by:*

$$\widetilde{\beta}_{\delta_2} = 1 - \Phi\left(\frac{\sigma_0}{\sigma_2}\Phi^{-1}(1-\alpha_0) + \frac{\sqrt{N}(\mu_0-\mu_2)}{\sigma_2}\right). \tag{24}$$

Proof. Using the result (18), it asymptotically holds that for any $\widetilde{\tau}_{\alpha_0} \in \mathbb{R}$:

$$\alpha_0(\widetilde{\delta}_2) = \mathbb{P}_0\left[\widetilde{\Lambda}_2(\mathbf{Z}) > \widetilde{\tau}_{\alpha_0}\right] = 1 - \Phi\left(\widetilde{\tau}_{\alpha_0}\right).$$

Hence, because Φ is strictly increasing, one has:

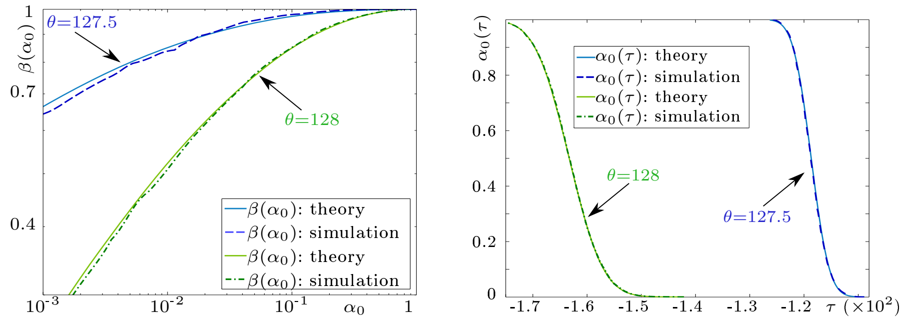
$$(1-\alpha_0(\widetilde{\delta}_2)) = \Phi(\widetilde{\tau}_{\alpha_0}) \Leftrightarrow \widetilde{\tau}_{\alpha_0} = \Phi^{-1}\left(1-\alpha_0(\delta_2)\right), \tag{25}$$

which proves Theorem 1.

It also follows from (18) that for any decision threshold $\widetilde{\tau}_{\alpha_0} \in \mathbb{R}$ the power of the test $\widetilde{\delta}_2$ (17) is given by:

$$\widetilde{\beta}_{\delta_2} = \mathbb{P}_1\left[\widetilde{\Lambda}_2(\mathbf{Z}) > \widetilde{\tau}_{\alpha_0}\right] = 1 - \Phi\left(\frac{\sigma_0}{\sigma_2}\left(\widetilde{\tau}_{\alpha_0} - \frac{\sqrt{N}(\mu_2-\mu_0)}{\sigma_0}\right)\right).$$

By substituting $\widetilde{\tau}_{\alpha_0}$ by the value given in Theorem 1, a short algebra leads to the relation (24). This proves Theorem 2 and concludes the proof.



(a) Detection power as a function of false alarm probability α_0 (ROC curves).

(b) False alarm probability α_0 as a function of threshold value τ_{α_0}.

Fig. 2. Illustration of LRT statistical performance, false-alarm probabilities and detection power, for $N = 1000$ pixels, $R = 0.1$, $\sigma_n = 0.5$ and $\theta = \{127.5; 128\}$. The empirical results were obtained with 5.10^4 realizations.

4.2 General Case of $R \in]0;1[$

The case for which the embedding rate R can take any value in $]0;1]$ is treated in a similar manner as the case $R = 2$. The problem of designing an optimal test has been shown to be particularly difficult in [26]. A thorough design a MP test uniformly with respect to the embedding rate lies outside of the scope of this paper which mainly studies the MP test for $R = 2$ and its practical implementation. Hence, it is proposed to use the test $\widetilde{\delta}_2$ (17) whatever the embedding rate R might be. Once again, the asymptotic distribution (18) is used to solve the decision problem (8).

The alternative hypothesis $\mathcal{H}_R$, that $\mathbf{Z}$ contains a stego-medium with embedding rate $R \in]0;1]$, can be considered as a combination of stego and cover pixels. Hence, the use of the law of total expectation and the law of total variance is relevant to calculate the two first moments of the log-LR $\widetilde{\Lambda}_2(\mathbf{Z})$. Using the moments given in (20)–(23), for the case $R = 2$, a short calculation gives:

$$\mu_R = \frac{R}{2}\mu_2 + \left(1 - \frac{R}{2}\right)\mu_0, \tag{26}$$

$$\sigma_R^2 = \frac{R}{2}(\sigma_2^2 + \mu_2^2) + \left(1 - \frac{R}{2}\right)(\sigma_0^2 + \mu_0^2) - \left(\frac{R}{2}\mu_2 + \left(1 - \frac{R}{2}\right)\mu_0\right)^2. \tag{27}$$

In other words, by using the test $\widetilde{\delta}_2$ (17) for any $R \in]0;1]$ only the detection power is impacted. Indeed, the null hypothesis does not change, hence, the asymptotic distribution (18) of the LR $\widetilde{\Lambda}_2(\mathbf{Z})$ under $\mathcal{H}_0$ as well as the decision threshold $\widehat{\tau}_{\alpha_0}$ (19) remain the same. This point is highlighted in the following theorem.

Theorem 3. *For any* $\alpha_0 \in]0;1[$*, assuming that the parameters* $\{\theta_n\}_{n=1}^N$ *and* $\{\sigma_n\}_{n=1}^N$ *are known, the power function* $\widetilde{\beta}_{\delta_R}$ *associated with the test* $\widetilde{\delta}_2$ *(17) is asymptotically given for any* $R \in]0;1]$ *by:*

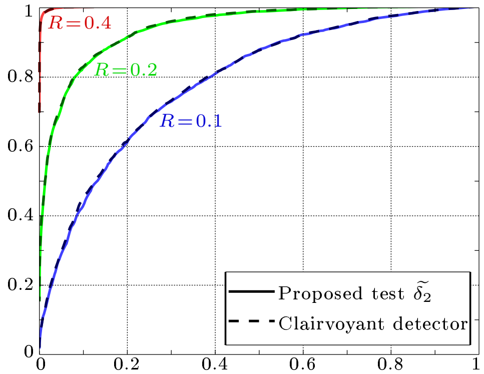


Fig. 3. Numerical comparison between Proposed LR test $\widetilde{\delta}_2$ (17), and the clairvoyant detector which knows the embedding rate $R = 0.1$ ans, thus, uses the LR test design for this rate. Results were obtained from a Monte-Carlo simulation with 5.10^4 realizations using *Lena* image cropped to 128×128 pixels and addition of a Gaussian white noise with $\sigma = 2$.

$$\widetilde{\beta}_{\delta_R} = 1 - \Phi\left(\frac{\sigma_0}{\sigma_R}\Phi^{-1}(1-\alpha_0) + \frac{R\sqrt{N}(\mu_0 - \mu_2)}{\sigma_R}\right). \tag{28}$$

The power functions $\widetilde{\beta}_{\delta_R}$ for $N\!=\!1000$, $R\!=\!0.1$, $\sigma_n\!=\!0.5$ and $\theta_n\!=\!\{127.5;128\}$ are drawn in Figure 2a. Once again, this figure highlights the potentially significant impact of pixel expectation on the performance of the test $\widetilde{\delta}_2$.
It should be highlighted that the most powerful property of the test $\widetilde{\delta}_2$ is difficult to prove for $R\in]0;1[$, see [9]. However, Figure 3 emphasizes the relevance of the proposed approach, which consists in designing a test for $R=2$ and extending its application to $R\in]0;1[$. Here, the power function of the proposed test is compared with the power function of the clairvoyant detector, that knows R. The numerical comparison present in Figure 3 shows that the loss of the power is negligible.

Finally, it can be noted that the detection power as given in Theorem 3 complies with the square root law of steganographic capacity [20]. Indeed, from (28), a short algebra immediately permits us to establish that:

$$\lim_{\sqrt{N}/L\to 0} \widetilde{\beta}_{\delta_R} = 1 \quad \text{and} \quad \lim_{\sqrt{N}/L\to \infty} \widetilde{\beta}_{\delta_R} = \alpha_0. \tag{29}$$

5 Practical Implementation of Proposed LR Test

In a practice, the application of the test $\widetilde{\delta}_2$ (17) is compromised because neither the expectation θ_n nor the variance σ_n^2 of pixels are known: their estimated values, denoted $\widehat{\theta}_n$ and $\widehat{\sigma}_n^2$, respectively, have to be used instead.

However, accurate estimation of the parameters θ_n and σ_n is a difficult problem but necessary to obtain a high detection performance. This problem also occurs in LSB replacement steganalysis. An efficient yet simple way to overcome this problem was introduced in the well-known Weighted Stego-image steganalysis (WS), initially proposed in [14]. The authors propose to locally estimate the parameter θ_n by filtering the inspected image so that $\widehat{\theta}_n$ correspond to the mean of the four surrounding pixels. Similarly, the local variance of the four surrounding pixels is used to estimate σ_n^2. The WS method has been studied thoroughly in [21] and two major improvements have been proposed. First, the authors have empirically enhanced the estimation of pixel expectations by testing different local filters. Second, the author proposed to use moderated weights $w_n = \widehat{\sigma}_n^2 + \alpha\,,\ \alpha > 0$ instead of the variance estimation $\widehat{\sigma}_n^2$.

In the present paper, it is proposed to use the WS filtering method to estimate the parameters θ_n and σ_n^2. Note that the proposed practical test is not optimal but intends to show the relevance of the proposed approach and feasibility to design a practical efficient test. Following the WS method, the practical implementation of the LR test $\widehat{\delta}_2$ proposed in this paper estimates each θ_n by filtering the inspected image with the kernel:

$$\frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & 0 & 2 \\ -1 & 2 & -1 \end{pmatrix}$$

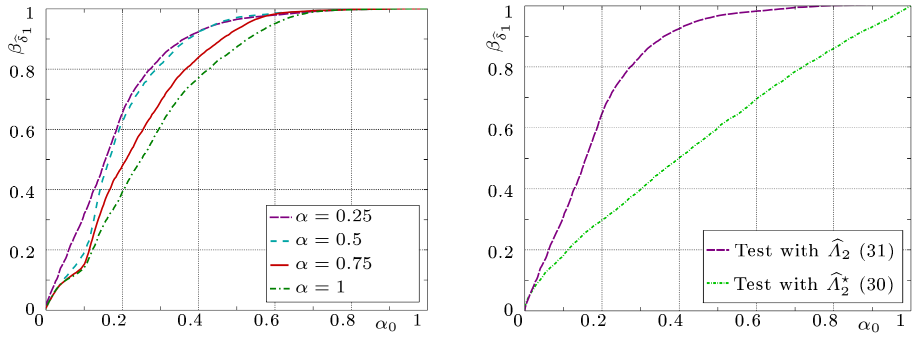
Contrary to what is suggested in [21], for the case of LSB replacement, our numerical experiments indicate that the detection performance tends to get worse when using the moderated weights instead of the estimated variance. Our interpretation of this phenomenon is as follows. The proposed LR test (17) essentially relies on the increase of pixels' variance due to insertion of hidden information. Hence, the use of moderated weights tends to fundamentally bias the test and deflates the performance results. Figure 4a offers an example of this phenomenon through a comparison of ROC curves obtained using 10 000 images from the BOSSbase database with $R = 1/2$ and $\alpha = \{1/4; 1/2; 3/4; 1\}$.

On the other hand, the direct use of the estimated variance $\widetilde{\sigma}_n^2$ may lead to numerical instability particularly in flat image areas. Hence, it was chosen to add $\alpha = 1/4$ to the estimated variance in our numerical experiments.

By using these estimated values in expression (15) the estimated log-LR $\widetilde{\Lambda}_2(z_n)$, see Equation (15) becomes:

$$\widehat{\Lambda}_2(z_n) = \log\left[\exp\left(\frac{z_n - \widehat{\theta}_n}{(\alpha + \widehat{\sigma}_n)^2}\right) + \exp\left(\frac{\widehat{\theta}_n - z_n}{(\alpha + \widehat{\sigma}_n)^2}\right)\right]. \qquad (30)$$

It should be highlighted that some difficult problems still remain open.
First, the normalization of the log-LR, suggested in Equation (17), requires the calculation of the expectation μ_0 and the variance σ_0^2 of the log-LR. Unfortunately, the estimates of the parameters σ_n are, in practice, not accurate enough to perform this normalization efficiently.



(a) ROC obtained with four different weight factor: $\alpha = \{1/4\,;\,1/2\,;\,3/4\,;\,1\}$.

(b) ROC curves obtained with the two statistics $\widehat{\Lambda}_2$ and $\widehat{\Lambda}_2^\star$.

Fig. 4. Impact of weights and calibration on proposed test performance. ROC curves obtained using the images from BOSS database [3] with $R = 0.5$.

Second, possibly the most difficult problem is that the statistical inference between the cover image and the hidden information should be taken into account. For instance it was proposed in [26] to remove the LSB plane in order to remove any potential stego-noise. For LSB matching this is not possible. Therefore, the impact of hidden information on estimators $\widehat{\theta}_n$ and $\widehat{\sigma}_n$ should be studied. Since the proposed test relies mainly on the slight increase of pixels' variance due to data hiding, the embedding changes may have an important effect on the estimates $\widehat{\sigma}_n$ and on the proposed test.

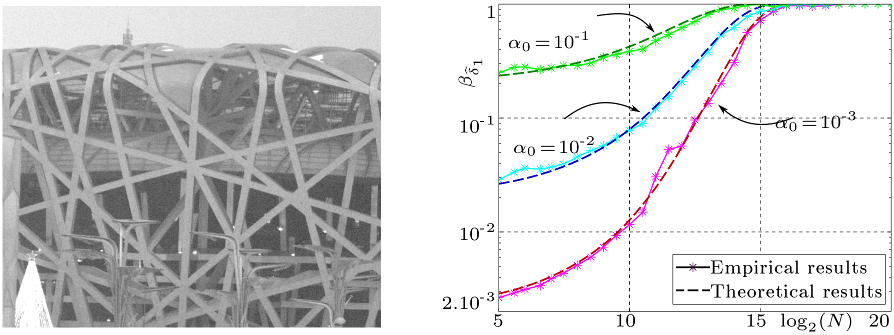
As explained above, proper normalization of the proposed test is critical in practice. Even though the proposed LR is very sensitive to hidden information, if its expectation can not be set to a fixed value under $\mathcal{H}_0$, the between-image-error described in [2] may negatively impact the test accuracy. Numerical simulations show that the expectation of the LR $\widehat{\Lambda}_2(z_n)$ can be roughly approximated by $-\log(2) - \frac{1}{4\widehat{\sigma}_n^2}$.

Therefore, the practical test proposed in the present paper is given as:

$$\widehat{\delta}_2 = \begin{cases} \mathcal{H}_0 & \text{if} \quad \widehat{\Lambda}_2^\star(\mathbf{Z}) \leq \widehat{\tau}_{\alpha_0}, \\ \mathcal{H}_1 & \text{if} \quad \widehat{\Lambda}_2^\star(\mathbf{Z}) > \widehat{\tau}_{\alpha_0}, \end{cases} \tag{31}$$

$$\text{with} \quad \widehat{\Lambda}_2^\star(\mathbf{Z}) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \widehat{\Lambda}_2(z_n) - \log(2) - \frac{1}{4(\alpha + \widehat{\sigma}_n)^2}. \tag{32}$$

One can note that, contrary to the LR statistically studied throughout Sections 4.1–4.2, the proposed decision statistic is not normalized. Indeed the variance of $\widehat{\Lambda}_2(z_n)$ is not taken into account in Equation 31. This is because the estimation of pixels' variance is particularly difficult and the method used in this paper is not accurate enough. In fact, normalization can even lower the detection performance. The most notable thing about the test (31) is that the expectation of the decision statistics $\widehat{\Lambda}_2^\star(\mathbf{Z})$ is always 0 under hypothesis $\mathcal{H}_0$.



(a) Digital image used for the Monte-Carlo simulations

(b) Power of the test $\widehat{\delta}_2$ (31) as a function of pixel number for different false-alarm probabilities: theory and simulation.

Fig. 5. Numerical verification of theoretical results through Monte-Carlo simulation based on natural image shown in Figure 5a

Figure 4b shows an example of the detection power obtained with the two tests based on the statistics (30) and (31).

6 Numerical Simulations

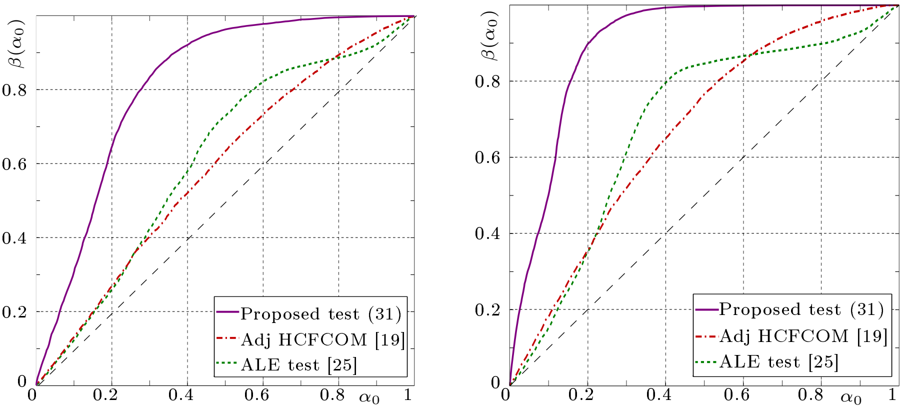
6.1 Theoretical Results on Simulated Data

Figure 5 presents a numerical verification of Theorem 3. The image shown in Figure 5a has been analyzed 5.10^4 times. Each run was preceded by the addition of a zero-mean Gaussian noise whose standard deviation was $\sigma = 1$. The embedded hidden information was drawn from a binomial distribution $\mathcal{B}(1, 1/2)$ with an embedding rate $R = 1$. The empirical power of the test $\widehat{\delta}_2$ is compared with the theoretical result given by Theorem 3 for three different false-alarm probabilities: $\alpha_0 = \{10^{-1}; 10^{-2}; 10^{-3}\}$. Observe that the obtained detection power almost perfectly corresponds to the theoretical results.

Note that it is crucial to use the same image for this Monte-Carlo simulation because the detection power of the proposed test depends on image parameters, namely on θ_n and particularly on σ_n^2. Hence, for a different image, the detection power may differ significantly as explained in Section 4. Moreover, the use of the same image artificially permits us to overcome the difficult problem of normalizing the log-LR and, thus, the effects of the between-image-error described in [2].

6.2 Comparison with the State of the Art on Real Images

Matlab source code of proposed test, as detailed in Equation (31), is available on the Internet at : `http://remi.cogranne.pagesperso-orange.fr/`.

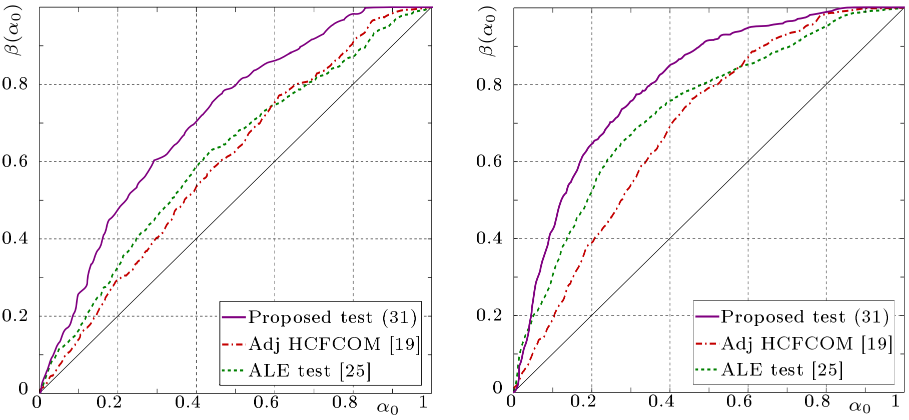


(a) ROC curves for $R = 0.5$.

(b) ROC curves for $R = 1$.

Fig. 6. Numerical comparisons of detectors performance using BOSS database [3]

One of the main motivations for this paper was to show that the hypothesis testing theory can be applied in practice to design an efficient LSB matching detector. This fact can only be shown by a numerical comparison with state-of-the-art detectors on large image databases. The potential competitors for LSB matching detection are not as numerous as for LSB replacement. As briefly described in the introduction, the operational context selected in this paper eliminates all prior-art detectors based on machine learning. Almost every other detector found in the literature is based on the image histogram. For the present comparison, two histogram-based detectors, namely ALE [25] and the adjacency HCF COM [19] detector, were used due to their high detection performance.



(a) ROC curves for $R = 0.25$.

(b) ROC curves for $R = 0.5$.

Fig. 7. Comparisons of detectors performance using Dresden database [16]

Figure 6 shows the results obtained with 10 000 images from BOSSbase contest database [3]. Each hidden bit was drawn from a binomial distribution $\mathcal{B}(1, 1/2)$. The embedding rate was $R = 0.5$ in Figure 6a and $R = 1$ in Figure 6b. Both figures show that the proposed test achieves a better detection power for any prescribed false-alarm probability.

Similarly, Figure 7 shows the results obtained with the 1488 raw images from the 'Dresden Image Database' [16]. Prior to our experiments, each image was converted to an unprocessed TIFF format (using dcraw) and only the red color channel was used. The embedding rate was $R = 0.25$ in Figure 7a and $R = 0.5$ in Figure 7b. The results presented in Figures 7a and 7b confirm that the proposed test has a better detection power for any prescribed false-alarm probability. Moreover by changing the embedding rate, the combined results of Figures 6 and 7 show that the proposed test also performs better than prior art for any R.

Note that, surprisingly, the detection power of the proposed test is slightly higher for the BOSSbase database than for the Dresden database for $R = 0.5$, see Figure 6a and 7b, respectively, whereas the Dresden database images are bigger. This phenomenon can be explained by the fact that the Dresden database images are RAW images that have not being further processed. In contrast, BOSSbase images have been downsampled, which may introduce correlations between neighboring pixels that implicitly make the filtering estimator more efficient.

7 Conclusion and Future Works

The first step to fill the gap between hypothesis testing theory and steganalysis was recently proposed in [12,7,26]. This paper extends this first step to the case of LSB matching. By casting the problem of LSB matching steganalysis in the framework of hypothesis testing theory, the most powerful likelihood ratio test is designed. Then, a thorough statistical study permits analytical calculations of its performance in terms of the false-alarm probability and detection power. To apply this test in practice, unknown image parameters have to be estimated. Based on a simple estimation of these unknown parameters, a practical test is proposed.

The relevance of the proposed approach is emphasized through numerical experiments. Compared to two leading histogram-based detectors, the proposed practical test achieves a better detection power.

However, the practical test presented in this paper relies on a simple yet efficient filtered version of inspected media to estimate pixel expectations and variances. In our future work, a more efficient model should be used to increase the detection power. Lastly, a thorough statistical study of the impact of this estimation on detection performance is desirable to complete the present work.

Acknowledgments. The authors would like to thank Jessica Fridrich for her contributions and stimulating discussions.

References

1. Böhme, R.: Advanced Statistical Steganalysis, 1st edn. Springer Publishing Company, Incorporated (2010)
2. Böhme, R., Ker, A.D.: A two-factor error model for quantitative steganalysis. In: Security, Steganography, and Watermarking of Multimedia Contents VIII. Proc. of the SPIE, vol. 6072 (2006)
3. BOSS contest: Break Our Steganographic System (2010), http://www.agents.cz/boss/
4. Cai, K., Li, X., Zeng, T., Yang, B., Lu, X.: Reliable histogram features for detecting LSB matching. In: 2010 17th IEEE International Conference on Image Processing, ICIP, pp. 1761–1764 (September 2010)
5. Cancelli, G., Doerr, G., Barni, M., Cox, I.: A comparative study of ±1 steganalyzers. In: 2008 IEEE 10th Workshop on Multimedia Signal Processing, pp. 791–796 (October 2008)
6. Cancelli, G., Doerr, G., Cox, I., Barni, M.: Detection of ±1 LSB steganography based on the amplitude of histogram local extrema. In: 15th IEEE International Conference on Image Processing, ICIP 2008, pp. 1288–1291 (October 2008)
7. Cogranne, R., Zitzmann, C., Fillatre, L., Retraint, F., Nikiforov, I., Cornu, P.: A Cover Image Model For Reliable Steganalysis. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 178–192. Springer, Heidelberg (2011)
8. Cogranne, R., Zitzmann, C., Fillatre, L., Nikiforov, I., Retraint, F., Cornu, P.: Reliable detection of hidden information based on a non-linear local model. In: Proc. of IEEE Workshop on Statistical Signal Processing, pp. 493–496 (2011)
9. Cogranne, R., Zitzmann, C., Fillatre, L., Retraint, F., Nikiforov, I., Cornu, P.: Statistical decision by using quantized observations. In: IEEE International Symposium on Information Theory, pp. 1135–1139 (2011)
10. Cogranne, R., Zitzmann, C., Nikiforov, I., Retraint, F., Fillatre, L., Cornu, P.: Statistical Detection of LSB Matching in the Presence of Nuisance Parameters. In: Proc. of IEEE Workshop on Accepted for Publication in Statistical Signal Processing (2012)
11. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2nd edn. Morgan Kaufmann (2007)
12. Dabeer, O., Sullivan, K., Madhow, U., Chandrasekaran, S., Manjunath, B.: Detection of hiding in the least significant bit. IEEE Transactions on Signal Processing 52(10), 3046–3058 (2004)
13. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.: Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE Transactions on Image Processing 17(10), 1737–1754 (2008)
14. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Security, Steganography, and Watermarking of Multimedia Contents VI. Proc. of the SPIE, vol. 5306 (2004)
15. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications, 1st edn. Cambridge University Press (2009)
16. Gloe, T., Böhme, R.: The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 25th Symposium On Applied Computing, ACM SAC 2010, vol. 2, pp. 1585–1591 (2010)
17. Goljan, M., Fridrich, J., Holotyak, T.: New blind steganalysis and its implications. In: Security, Steganography, and Watermarking of Multimedia Contents VIII. Proc. of the SPIE, vol. 6072 (2006)

18. Harmsen, J., Pearlman, W.: Higher-order statistical steganalysis of palette images. In: Security, Steganography, and Watermarking of Multimedia Contents V. Proc. of the SPIE, vol. 5020 (2005)
19. Ker, A.: Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters 12(6), 441–444 (2005)
20. Ker, A.D.: A capacity result for batch steganography. Signal Processing Letters 14(8), 525–528 (2007)
21. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Security, Forensics, Steganography, and Watermarking of Multimedia Contents X. Proc. of the SPIE, vol. 6819 (2008)
22. Lehman, E., Romano, J.: Testing Statistical Hypotheses, 2nd, 3rd edn. Springer (2005)
23. Lyu, S., Farid, H.: Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security 1(1), 111–119 (2006)
24. Scott, C.: Performance measures for Neyman-Pearson classification. IEEE Trans. Inform. Theory 53(8), 2852–2863 (2007)
25. Zhang, J., Cox, I., Doerr, G.: Steganalysis for LSB matching in images with high-frequency noise. In: IEEE 9th Workshop on Multimedia Signal Processing, MMSP 2007, pp. 385–388 (October 2007)
26. Zitzmann, C., Cogranne, R., Retraint, F., Nikiforov, I., Fillatre, L., Cornu, P.: Statistical Decision Methods in Hidden Information Detection. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 163–177. Springer, Heidelberg (2011)

Textural Features for Steganalysis

Yun Q. Shi[1], Patchara Sutthiwan[1], and Licong Chen[1,2]

[1] New Jersey Institute of Technology
Newark, NJ, USA
{shi,ps249}@njit.edu
[2] Fujian Normal Univeristy
Fuzhou, P.R. China
clcfz@fjnu.edu.cn

Abstract. It is observed that the co-occurrence matrix, one kind of textural features proposed by Haralick et al., has played a very critical role in steganalysis. On the other hand, the data hidden in the image texture area has been known difficult to detect for years, and the modern steganographic schemes tend to embed data into complicated texture area where the statistical modeling becomes difficult. Based on these observations, we propose to learn and utilize the textural features from the rich literature in the field of texture classification for further development of the modern steganalysis. As a demonstration, a group of textural features, including the local binary patterns, Markov neighborhoods and cliques, and Laws' masks, have been selected to form a new set of 22,153 features, which are used with the FLD-based ensemble classifier to steganalyze the HUGO on BOSSbase 0.92. At the embedding rate of 0.4 bpp (bit per pixel) an average detection accuracy of 83.92% has been achieved. It is expected that this new approach can enhance our capability in steganalysis.

1 Introduction

Steganography and steganalysis are a pair of modern technologies that have been moving ahead swiftly in the last decade. The conflicting between these two sides is a driving force for the rapid development. That is, each side learns from its counterpart. From the modern steganalysis point of view, the machine learning framework, consisting of statistical features and classifier, has been first utilized in [1]. In [2], the first four statistical moments of wavelet coefficients and their prediction errors of nine high frequency subbands from three-level decomposition are used to form a 72-dimensional (72-D) feature vector with the modern classifier SVM for steganalysis. The steganalysis method based on the mass center of histogram characteristic function has shown improved effectiveness in steganalysis [3]. A framework combining wavelet decomposition and moments of characteristic functions is reported in [4]. To break steganographic schemes with popularly used JPEG images as carriers, such as OutGuess, F5 and model-based steganographic schemes, a group of 23 features, including both the first and second order statistics, have been used together with a calibrate technique in [5]. Markov process has first been used in [6] for steganalysis. How to handle the high dimensionality of elements in the transition probability matrix

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 63–77, 2013.

resultant from the application of Markov process has been studied in [7], for the spatial-domain. In [8], both the first and the second order Markov models, called SPAM, have been established to detect the more advanced steganographic scheme known as LSB matching. As expected, there is no end in the competition between steganography and steganalysis just like mouse versus cat. A modern steganographic scheme, named HUGO [9], has been developed so as to fail the SPAM by taking high order difference into consideration in its data embedding. Steganalytic methods [10, 11, 12] have been reported to break HUGO. In [12], image features are extracted via applying high-pass filters to the image, followed by down-sampling, feature selection, and some optimization technique. Depending on the chosen parameters, the feature dimensionalities range from more than one hundred to more than one thousand; with a linear classifier, the detection accuracies of the generated features on BOSSbase 0.92 [13,14] range from 70% to more than 80%. In [10,11], the difference arrays from the first-order up to the sixth-order are all used for feature extraction in addition to other newly designed features, resulting in the total number of features as high as 33,963. Because of the high feature dimensionality, an ensemble classifier using Fisher's Linear Discriminant (FLD) has been developed and utilized. These novel measures result a detection rate of 83.9% on BOSSbase 0.92 [13, 14] (at the embedding rate of 0.4 bits per pixel bpp).

What described above is by no means a complete review of this active research field in steganalysis. For instance, the recent technologies of steganography and steganalysis in the JPEG domain have not been discussed here, which however have shown the same pattern of competition among these two areas. The observation from the above discussion is that the modern steganalysis has made rapid progress in the past decade, so does modern steganography.

1.1 Textural Features

In this paper, we take a different look at steganalysis from the texture classification point of view. According to the highly cited (as of February 2012, having been cited almost 7,000 times according to Google) paper by Haralick et al. [15] in 1973, "context, texture, and tone are always present in the image, although at times one property can dominate the other," "texture is an innate property of virtually all surfaces." In their paper, the co-occurrence matrix has been proposed as textural features for image classification. Since then it has been one of the most widely used statistical methods for various tasks in pattern recognition.

Now we extend this thought [15] further. The modern steganography hides data into a cover image. That means the original texture of cover image has been modified somehow after data embedding even though the change is small. Therefore many technologies developed for texture images classification are reasonably expected to be usable for steganalysis. In addition, it has been reported that the data hidden inside the texture images are difficult to be detected [e.g., 16], in other words, the texture images are suitable for steganography, consequently the steganalysis on texture images is challenging, and some efforts have been made [e.g., 17].

Therefore, it becomes clear that the technologies developed for texture images classification should be able to play an important role in modern steganalysis. That is, there are many tools developed for texture classification that we can borrow to use for steganalysts in addition to co-occurrence matrix (transition probability matrix can be shown equivalent to the co-occurrence matrix under certain condition which has been used in steganalysis). Specifically, by taking a close look at the techniques used in texture classification (e.g. according to [18]), we can find Markov random fields (MRF) and others which belong to the technologies suitable for stationary texture images. In the category of non-stationary texture images, there are Laws' masks, local binary patterns (LBP), and others.

These thoughts have led us to investigate new steganalysis technologies. We first examined the LBP technology [19, 20]. In this popular technology (as of February 2012 [20] has been cited almost 1,900 times according to Google), the pixels in the entire image (or in the area of interesting) are examines. For each considered pixel, the LBP opens, say, a 3×3 neighborhood surrounding it. Then the gray-value of each of the eight neighbor pixels is compared with that of the central pixel. If the gray-value of a neighbor pixel is smaller than that of the central pixel, a binary zero is recorded for this pixel; otherwise, a binary one is recorded; thus resulting a string of eight binary bits, each being either zero or one. This procedure is conducted for each pixel of the given image. If one chooses a sequencing among these eight binary bits assigned to the eight-neighbors, one then obtain a corresponding eight-bit binary number. Applying this procedure to all pixels, we end up with many eight-bit binary numbers, specifically one for each pixel of the image under consideration (with some treatment applied to the boundary pixels). Since any eight-bit binary number corresponds to a specific decimal number in a range from zero to 255, clearly, the histogram of all of the decimal numbers thus formulated consists of 256 bins. The distribution of this type of histogram bins' values is chosen to characterize the given image. Since it is obtained from each individual pixel through comparing it with its local neighbor pixels, this type of histogram is expected to be suitable for texture classification; in our case, for steganalysis.

Note that there are several different ways to generate the histogram. A popular way of LBP technology used in texture analysis ends up with only 59 bins for the 3×3 neighborhoods described above. That is, the statistics shows that there are many very sparse bins among the 256 bins. We can then merge them so as to result in only 59 bins without losing much information in classification. In order to achieve rotation invariance, the following procedures are taken. That is, we consider a unit circle from the central pixel with a radius being one, hence the gray-value of four corner pixels of this 3×3 block are determined by interpolation.

Furthermore, the LBP technology considers multi-resolutions. That is, in addition to a neighborhood of 3×3, one can also consider neighborhood of 5×5and or 7×7. It is shown in [20] that multi-resolution does help in texture classification. In addition to the linear binary patterns just discussed, the LBP scheme also considers "contrast" by introducing another quantity called variance. That is, if we consider the case of 3×3 square neighborhoods, we first calculate the mean average of the eight surrounding pixels' gray-value, and then we calculate the local variance with respect to the central pixel's gray-value. For detail of the LBP technologies, readers are referred to [19, 20].

As an exercise, we have applied these textural features to steganalyzing the above-mentioned HUGO stego dataset [13, 14] designed for the BOSS contest. We construct a steganalyzer with 22,153 features derived from the textural features. Instead of co-occurrence matrix we have used LBP features (59-D, corresponding to the above mentioned 59 bins, used for some filtered 2-D array, and 256-D (256 bins) used for others) and variance features derived from the multi-resolution way. In addition, we have used Laws mask and the mask and cliques associated with Markov Random fields [18]. The classifier utilized is the FLD-based ensemble classifier, reported in [10, 11]. The achieved average detection rate is 83.92% on BOSSbase 0.92 [13, 14] at the embedding rate of 0.4 bpp. Note that the stego images were generated by HUGO with default parameters. While our first-stage work has been positive, more works need to be done to further move our investigation ahead. It is hope that we have opened a different angle to view and handle steganalysis.

1.2 Rest of the Paper

The rest of this paper is organized as follows. In Section 2, the proposed textural feature framework to break HUGO is discussed. The experimental procedure and empirical validations are presented in Section 3. The discussion and conclusions are made in Section 4.

2 Textural Feature Framework

Advanced stegonographic schemes such as HUGO [9] tend to embed data into cover image locally into some regions so as to make the image statistical modeling difficult, especially into highly texture regions. Intuitively, this small local change should be efficiently captured by some image operators which emphasize on modeling microstructure image properties. In this paper, we would like to introduce the local binary pattern (LBP) operators [19, 20] which have been popularly used in texture classification arena, as a potential statistical image modeling for steganalysis.

2.1 Image Statistical Measures

Ojala et al. [19] proposed LBP to model the statistics of a texture unit defined within a neighborhood of, say, 3×3 pixels. Each of eight neighboring pixels of a 3×3 neighborhood is thresholded by the gray value of its central pixel to form an 8-bit binary pattern. Fig. 1 (a) depicts a 3×3 neighborhood employed in the calculation of the original LBP in which g_c is the center pixel and g_p, $p = 0,2,\ldots,P\text{-}1$, where P is the number of neighboring pixels and equal to 8 in this case, representing the neighboring pixels. In [20], Ojala et al. reported that LBP operators can achieve rotation invariant property after some manipulation. In this version of LBPs, the local neighborhood is circularly defined as shown in Fig. 1 (b) in which the pixel values of the neighbors falling outside the center of the pixel grids are estimated by interpolation. Rotation invariant and uniformity mappings are introduced. The authors classify LBP into two categories: “uniform” and “non-uniform” patters as shown in Fig. 1 (c). Uniform patterns have the number of binary transitions (between zero and one) over the whole neighborhood circle less than or equal to two while the patterns whose number of such transitions is greater than two are considered as non-uniform. In texture classification, uniform patterns often occupy the majority of the histogram which makes merging non-uniform patterns into the same bin legitimate. This pattern merging is simply called uniformity mapping (or u2 mapping), reducing the number of bins in a histogram from 256 to 59 bins. This type of LBP descriptor is denoted as $\mathrm{LBP}_{P,R}^{u2}$ where P defines the number of neighbor pixels, R the radius of the circular symmetric.

The authors also suggested a feasibility of enhancing texture classification performance by incorporating multi-resolution approach. Please be noted that while doing so we choose to always set P = 8 in order to keep feature dimensionality manageable and that the circular symmetric neighbor inscribed within 3×3 square neighborhood when R = 1, 5×5 when R = 2, and 7×7 when R = 3.

Generalized to different P values and correspondingly defined neighborhoods, Eq. (1) expresses the formulation of LBP mathematically.

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad (1)$$

where $s(x)$ equals one if the x is less than or equal to zero, or zero otherwise. Consequently, a histogram of 256 bins is formulated as a texture descriptor which represents vital information about spatial structure of image texture at microscopic level. We denote this basic LBP as LBP_8.

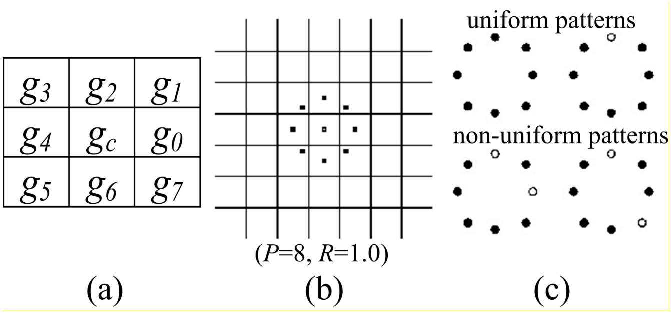


Fig. 1. (a) 3×3 neighborhood. (b) Example of circular symmetric neighborhood. (c) Examples of "uniform" and "non-uniform" local binary patterns. (b) and (c) are adapted from [20].

In some applications, the performance of LBP can be enhanced by the use of a local contrast measure [20]. In this paper, we measure local contrast in a 3×3 square neighborhood and as a result a variance image can thus be formed. We denote contrast measure on square 3×3 neighborhood as VAR_8. VAR_8 is defined as follows.

$$VAR_8 = \tfrac{1}{8}\sum_{p=0}^{7}(g_p - \mu_8), \quad \text{where } \mu_8 = \tfrac{1}{8}\sum_{p=0}^{7} g_p \qquad (2)$$

We found empirically that LBP features extracted from some variance images can enhance the detectability of our proposed steganalyzer.

To demonstrate the effectiveness of LBP operators in steganalysis, we constructed some simple testing scenarios to compare the performance of features derived from

LBP operators with those from co-occurrence matrix. Here we form a set of features on the first-order horizontal residual images generated by filtering images in BOSSbase 0.92 [13, 14] with the operator [-1 1].

Table 1. Comparative performance study of co-occurrence and LBP features from horizontal difference array

Feature Type	I	II	III	IV
TP	57.48%	56.61%	64.53%	61.56%
TN	51.46%	56.98%	65.20%	61.15%
AC	54.47%	56.80%	64.87%	61.36%
D	81	59	256	177

Feature type I stands for features derived by using co-occurrence matrix formulated along horizontal direction, II by $LBP_{8,1}^{u2}$, III by LBP_8 and IV by $LBP_{8,1}^{u2}$ $+LBP_{8,2}^{u2}$ $+LBP_{8,3}^{u2}$. TP (true positive rate) is the percentage of the stego images correctly classified, TN (true negative rate) being the percentage of cover images correctly identified. AC (accuracy rate) is percentage of stego and cover images correctly classified. D is feature dimensionality. Random data partitions are done 12 times, each with 8,074 pairs of image for training and the 1,000 left for testing.

To derive feature using co-occurrence matrix along horizontal direction, we first threshold the residual images with T = 4 which results in the feature dimensionality of 81 [7, 8] (first-order SPAM). The corresponding feature dimensionalities of LBP_8, $LBP_{8,1}^{u2}$ (i.e., as introduced, eight neighbor elements in total, radius being one, u2 mapping applied), $LBP_{8,2}^{u2}$ and $LBP_{8,3}^{u2}$ are 256, 59, 59, and 59, respectively. Fisher's Linear Discriminant (FLD) is employed.

The comparative performance is shown in Table 1. The statistics in Table 1 shows that: 1) features generated from LBP_8 are much more powerful than those from co-occurrence matrix but with a higher dimensionality; 2) features generated from $LBP_{8,1}^{u2}$ perform slightly better than those from co-occurrence matrix although they are of lower dimensionality; 3) multi-resolution approach improves the performance of $LBP_{P,R}^{u2}$ scheme while it keeps dimensionality manageable.

Instead of using co-occurrence matrix, in this paper we formulate statistical image features based solely on LBP operators. In so doing, we apply an LBP operator onto a set of residual images, each of which reveals artifacts associated with steganography in a different way. In the rest of this section, we describe a set of potential residual images to be used in our proposed image statistical model.

2.2 Content-Adaptive Prediction Error Image

Small perturbation to cover image caused by steganographic schemes may be considered as a high frequency additive noise; as a result, eliminating low-frequency representation of images before feature extraction process would make the resulting image features better represent the underlying statistical artifacts associated with steganography. With the modern steganographic schemes such as HUGO [9], it is intuitive that the prediction error images (also referred to as residual images) generated in a content adaptive manner would effectively reveal such artifacts caused by data embedding. Here we denote I as image, R as residual image, and Pred(I) as corresponding predicted image. Predicted images here are calculated based on some relationship within a predefined square neighborhood. Mathematically, R can be expressed as below.

$$R = I - \text{Pred}(I) \tag{3}$$

In this subsection, we propose to use the following two major kinds of content adaptive residual images. The first kind is generated based on our proposed prediction scheme modified based on [21], while the second kind is generated based on a collection of median filters.

Successive Prediction Error Image. We adopt a prediction scheme based on [21] to better reveal steganographic artifacts utilizing a 3×3 neighborhood to formulate the prediction error. Since our application is not coding, we are free to manipulate the prediction scheme. That is, the prediction scheme [21] is employed in a 2×2 neighborhood but in a different way; that is, with a fixed reference pixel (a pixel to be predicted), we rotate the 2×2 neighborhood four times to cover a 3×3 neighborhood, each rotation yielding one predicted value of the reference pixel. The final predicted value is the average of these four predicted pixel values. We found empirically that features extracted from residual images generated by this proposed scheme are more discriminative than those generated by the prediction scheme in [21]. Fig. 2 and Eq. 3 describe the prediction process.

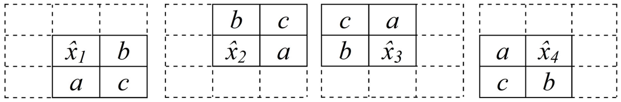


Fig. 2. Four 2×2 neighborhoods used predict the center pixel of a 3×3 neighborhood

$$\hat{x}_i = \begin{cases} \max(a,b) & c \le \min(a,b) \\ \min(a,b) & c \ge \max(a,b) \\ a+b-c & \text{otherwise} \end{cases} \tag{4}$$

Much of image content has been removed by the proposed scheme; however, the influence of image content can be further reduced by successive application of this scheme. In this paper, we denote PE_n as a prediction error image generated by applying the proposed scheme to the original input image for n multiple times.

Median-Filter-Based Prediction Error Images. Spatial filters have been widely used as low-pass filters. Much of their applications are for image denoising. It is therefore intuitive to generate residual images by using median filters to compute predicted images. That is, a median filtered image is subtracted from an original image, thus generating a prediction error image. In this paper, we use a set of median filters of three different sizes, 3×3, 5×5, and 7×7, to calculate predicted images. Pred(I) in Eq. (3) is defined by the output of applying a median filter defined here to a given input image *I*.

w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}

(a)

w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
w_{21}	w_{22}	w_{23}	w_{24}	w_{25}
w_{31}	w_{32}	w_{33}	w_{34}	w_{35}
w_{41}	w_{42}	w_{43}	w_{44}	w_{45}
w_{51}	w_{52}	w_{53}	w_{54}	w_{55}

(b)

w_{11}	w_{12}	w_{13}	w_{14}	w_{15}	w_{16}	w_{17}
w_{21}	w_{22}	w_{23}	w_{24}	w_{25}	w_{26}	w_{27}
w_{31}	w_{32}	w_{33}	w_{34}	w_{35}	w_{36}	w_{37}
w_{41}	w_{42}	w_{43}	w_{44}	w_{45}	w_{46}	w_{47}
w_{51}	w_{52}	w_{53}	w_{54}	w_{55}	w_{56}	w_{57}
w_{61}	w_{62}	w_{63}	w_{64}	w_{65}	w_{66}	w_{67}
w_{71}	w_{72}	w_{73}	w_{74}	w_{75}	w_{76}	w_{77}

(c)

Fig. 3. Symbolic representations of pixel locations used in the creation of median-filter-based prediction error images. (a) 3×3, (b) 5×5, and (c) 7×7 neighborhood.

Table 2. Configuration of Median Filters Employed in Generating Median-Filter-Based Prediction Error Images

Mask size	Filter number	Pixel locations used in computing median image
3×3	1	w_{11}, w_{13}, w_{22}, w_{31}, w_{33}
	2	w_{12}, w_{21}, w_{22}, w_{23}, w_{32}
5×5	1	w_{12}, w_{14}, w_{21}, w_{22}, w_{24}, w_{25}, w_{33}, w_{41}, w_{42}, w_{44}, w_{45}, w_{52}, w_{54}
	2	w_{11}, w_{13}, w_{15}, w_{31}, w_{33}, w_{35}, w_{51}, w_{53}, w_{55}
	3	w_{13}, w_{22}, w_{23}, w_{24}, w_{31}, w_{32}, w_{33}, w_{34}, w_{35}, w_{42}, w_{43}, w_{44}, w_{53}
7×7	1	w_{12}, w_{13}, w_{15}, w_{16}, w_{21}, w_{22}, w_{23}, w_{25}, w_{26}, w_{27}, w_{31}, w_{32}, w_{33}, w_{35} w_{36}, w_{37}, w_{44}, w_{51}, w_{52}, w_{53}, w_{55}, w_{56}, w_{57}, w_{61}, w_{62}, w_{63}, w_{65}, w_{66} w_{67}, w_{72}, w_{73}, w_{75}, w_{76}
	2	w_{14}, w_{22}, w_{24}, w_{26}, w_{34}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45}, w_{46}, w_{47}, w_{54}, w_{62} w_{64}, w_{66}, w_{74}
	3	w_{11}, w_{13}, w_{15}, w_{17}, w_{31}, w_{33}, w_{35}, w_{37}, w_{44}, w_{51}, w_{53}, w_{55}, w_{57}, w_{71} w_{73}, w_{75}, w_{77}
	4	w_{14}, w_{23}, w_{24}, w_{25}, w_{32}, w_{33}, w_{34}, w_{35}, w_{36}, w_{41}, w_{42}, w_{43}, w_{44}, w_{45} w_{46}, w_{47}, w_{52}, w_{53}, w_{54}, w_{55}, w_{56}, w_{63}, w_{64}, w_{65}, w_{74}

2.3 Residual Images Based on Laws' Masks

The residual images in this portion are computed by applying high-pass filters to the given image in the spatial domain. We also generate some residual images in this part in a content adaptive manner by incorporating two non-linear operators, minimum and maximum in order to catch the desired artifacts.

This part of image statistical features is formulated by two major set of 1-D spatial high-pass filters. The first set of high-pass filters is Laws' mask [18] which are of odd sizes (3,5, and 7), while the other set which contains even-tap high-pass filters (2,4, and 6) have been designed by us. As shown in Table 3, F4 and F6 were generated by convolving the mask [-1 1], popularly used in steganalysis and denoted by F2 in this paper, with S3 and E5, respectively, which are shown in Table 3.

Table 3. High-pass filters employed in the creation of residual images in Section 2.3

Category	Number of Taps	Name	Filter
Laws' Mask	3	Edge 3 (E3)	[-1 0 1]
		Spot 3 (S3)	[-1 2 -1]
	5	Edge 5 (E5)	[-1 -2 0 2 1]
		Spot 5 (S5)	[-1 0 2 0 -1]
		Wave 5 (W5)	[-1 2 0 -2 1]
		Ripple 5 (R5)	[1 -4 6 -4 1]
	7	Edge 7 (E7)	[-1 -4 -5 0 5 4 1]
		Spot 7 (S7)	[-1 -2 1 4 1 -2 -1]
		Wave 7 (W7)	[-1 0 3 0 -3 0 1]
		Ripple 7 (R7)	[1 -2 -1 4 -1 -2 1]
		Oscillation 7 (O7)	[-1 6 -15 20 -15 6 -1]
Even Taps	2	Filter 2 (F2)	[-1 1]
	4	Filter 4 (F4)	[1 -3 3 -1]
	6	Filter 6 (F6)	[1 -3 2 2 -3 1]

For a given filter, we possibly generate five different residual images as follows: 1) R_h by applying a filter in the horizontal direction; 2) R_v by applying a filter in the vertical direction; 3) R_{hv} by applying a filter in the horizontal direction and then in the vertical direction in a cascaded manner; 4) $R_{min} = \min(R_h, R_v, R_{hv})$; 5) $R_{max} = \max(R_h, R_v, R_{hv})$.

2.4 Residual Images Based on Markov Neighborhoods and Cliques

Markov Random Field (MRF) has been widely used in texture classification, segmentation and texture defect detection [18]. In MRF, a neighborhood, called a Markov neighborhood, can be constructed, into which the Markov parameters can be assigned as weights. These neighborhoods are characterized by a group of pixels with a variety of orientations often symmetrically inscribed within a square window of odd size. They are hence tempting choices for advanced steganalysis. Here our immediate application of Markov neighborhood is for high-pass filtering instead of texture classification. As a result, we do not strictly rely on Markov condition and parameters. Fig. 4 represents the masks we use to generate residual images described in this portion.

In addition to Markov neighborhoods, we propose to use cliques, portions of Markov neighbors, to high-pass filter images. The cliques used in this paper are shown in Fig. 5. The artifacts caused by steganalysis, reflected in residual images and obtained by applying these cliques are more localized than those caught by applying Markov neighborhood because of their small sizes. Thus, the detectability of our steganalysis scheme has been enhanced. Note that the masks in Fig. 4 (d), (h), (i), (j), (k), (l), and (m) and Fig. 5 (i), (j), (k), and (j), are created by us.

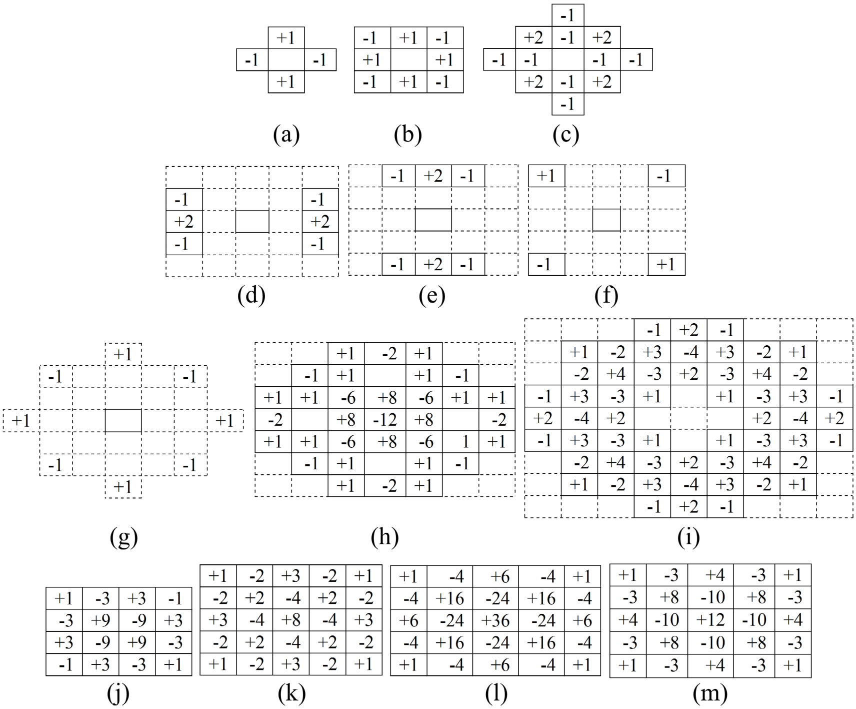


Fig. 4. High-pass filters based on Markov neighborhoods

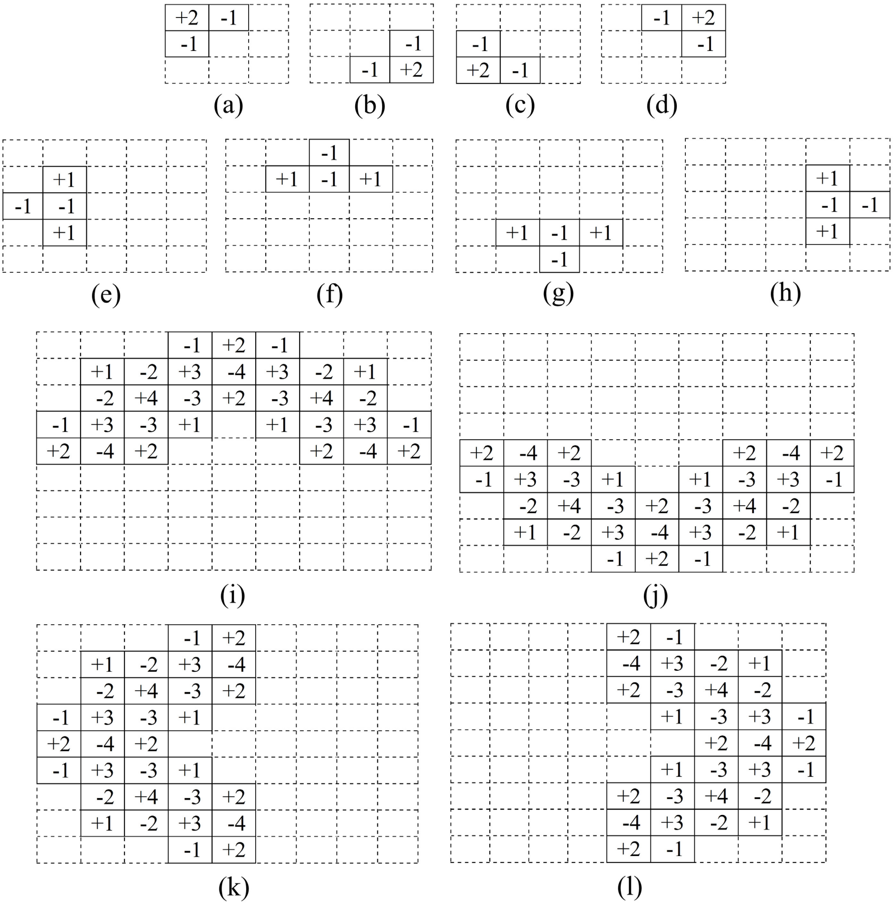


Fig. 5. High-pass filters based on cliques

3 Feature Construction and Experimentation

After the discussion of a variety of features made in the above section, one can observe that there are multiple ways to construct a feature set for steganalysis. An effective combination of features with a dimensionality of 22,153 is constructed based on the description in Section 2 to steganalyze HUGO at 0.4 bpp on BOSSbase 0.92 [13, 14]. We do not claim that this is the best possible combination of features in our framework. The details of the proposed combination are summarized in Table 4. The empirical validations on features from successive prediction error images and their variance images are shown in Table 5. Table 6 shows ensemble accuracies of each feature type. In order to validate whether or not each type of features are essential to the final accuracy of the whole feature set, the performances of the whole feature sets as well as the whole feature sets with each individual type of features dropped out are evaluated and shown in Table 7.

Table 4. The details of the proposed feature set

Features Described in Subsection	LBP Operators	Comments
2.2	Multi-resolution LBP: $LBP_{8,1}^{u2}$ +$LBP_{8,2}^{u2}$ +$LBP_{8,3}^{u2}$ (177-D features extracted from each residual image)	PEs denotes features generated from successive prediction error images PE_n (n=1 to 5).
		VARpe denotes features generated from variance images of successive prediction error images.
		MEDpe denotes features generated form median-filter-based prediction error images according to Table 2.
2.3		LMbased denotes features generated from residual images based on Laws' masks shown in Table 3.
2.4	The original LBP: LBP_8 (256-D features extracted from each residual image)	MN13 denotes features generated from 13 residual images based on Markov neighborhood filters shown in Fig. 4.
		CL12 denotes features generated from 12 residual images based on cliques shown in Fig. 5.

Table 5. Empirical validation on PEs and VARpe using FLD classifier

Residual	PE_1	PE_1-PE_2	PE_1-PE_3	PE_1-PE_4	PE_1-PE_5	$PEVAR_1$-$PEVAR_5$	
AC	61.29%	66.96%	68.49%	70.00%	70.78%	73.12%	76.55%
D	59	118	177	236	295	590	1,770
R	1	1	1	1	1	1	1, 2, 3

All the LBP operators used to construct features in Table 5 are based on uniformity mapping with P = 8 and different combination of R's. Note that the last column in Table 5 represents the multi-resolution setting of LBP operators ($LBP_{8,1}^{u2}$+ $LBP_{8,2}^{u2}$+ $LBP_{8,3}^{u2}$). In Table 5, PE_1-PE_5, and $PEVAR_1$-$PEVAR_5$ mean that PE_1 to PE_5, and PE_1 to PE_5 together with their variance images are used as inputs to LBP operators, respectively. The statistics shown in Table 5 reveals the successive applications of the prediction error schemes, contrast measure, and multi-resolution approach of LBP have all contributed to enhance the detection accuracy.

Table 6. Ensemble accuracies of each feature type

Feature Type	PEs	VARpe	MEDpe	LMbased	MN13	CL12
AC	75.58%	68.08%	66.43%	81.50%	74.88%	71.34%
D	885	885	1,593	12,390	3,328	3,072
d_{red}	300	300	600	2,600	1,000	1,000
L	101	89	45	49	101	43

Note that AC stands for accuracy, D for feature dimensionality, d_{red} for the dimensionality of random selected feature subset, L for the number of weak learners or ensembles. In all cases, we independently train and test classifiers for 12 times, with the same rule for data partition: randomly selected 8,074 pairs of cover and stego images for training and the 1,000 left for testing.

Table 7. Ensemble performance on feature elimination at d_{red} = 2,600

Feature Set	D	AC	L	Degradation
Whole	22,593	83.92%	50	0.00%
Whole - PEs	21,268	83.57%	46	-0.35%
Whole - VARpe	21,268	83.57%	57	-0.35%
Whole - MEDpe	20,560	83.67%	63	-0.25%
Whole - LMbased	9,763	82.72%	65	-1.20%
Whole - MN13	18,825	83.52%	45	-0.40%
Whole - CL12	19,081	83.67%	52	-0.25%

For the whole feature set, TP rate = 84.45%, TN rate = 83.40%, and AC = 83.92%. The statistics in Table 7 reveals that each type of the proposed features is essential to the final accuracy. That is, the final accuracy decreases upon the absence of each type of features. The degree of contribution among all types of features can be ranked in descending order as follows: LMbased, MN13, PEs (tied with VARpe), and MEDpe (tied with CL12). Note that it is very difficult to make a significant progress when more than 80% of detection accuracy has been attained. Therefore, only a fraction of percentage gained in the detection accuracy by some set of features matters in detection HUGO with high fidelity.

4 Discussion and Conclusions

In this paper we have reported our first-stage investigation on applying textural features for steganalysis. Specifically, we studied local binary patterns (LBP), which were inspired by the well-known co-occurrence matrix. In this LBP technique each pixel is compared with its neighbor pixels and thus binarized. This process is conducted for each pixel in a given image (or a region of interests). All of bins of the resultant histogram are used as LBP features. Furthermore, a multi-resolution structure can be constructed by using multi-size neighbor, e.g., 3×3, 5×5 and 7×7. In addition to the LBP, the variance generated from the above-mentioned local can also be used to characterize the contrast of the local region of, say, 3×3. Our work has verified that the LBP, variance and multi-resolution do work well in steganalysis. As to use the 256 bins or the 59 bins (the latter results from the so-called uniform mapping) in steganalysis, it depends. Our experimental works have demonstrated that the selection of 256 bins often perform better than 59 bins if the feature dimensionality is low. As the dimensionality increases, this may change. Hence in our work we use both 256 bins and 59 bins for different kinds of features and scenarios.

Prior to further summarizing this work, we would like to bring one point to readers' attention. That is, Avcibas et al. [22] proposed a steganalysis scheme which employs 18 binary similarity measures on the seventh and eighth bit planes in an image as distinguishing features. Instead of comparing, say, in a 3×3 neighborhood, the eight neighboring pixel values with the central pixel value to produce an eight-bit binary number so as to establish a histogram of 256 bins for classification, the authors [22] simply use the two least significant bit-planes in a given image without binarization. Furthermore, they include the bit corresponding to the central pixel position to formulate a nine-bit string, thus resulting in a histogram of 512, instead of 256, bins. One more difference is that we use the 59 and/or 256 features as suggested in the LBP technologies [19, 20], while they compute four binary similarity measures on the resulting 512-bin histograms [22] as features for steganalysis. Consequently, one should not consider the scheme in [22] as an application of the LBP technology.

Markov neighborhoods with Markov parameters utilized in Markov Random Field as shown in Figs. 3.53, 3.60 and some of their cliques shown in Fig. 3.68 in [18] have been studied in our work. Many of them with some of our additions as shown in Figs. 4 and 5 have been used in our steganalysis. They have contributed.

Among Laws' masks as shown in Figs. 4.126, 4.127 and 4.128 in [18], we eliminate all the masks that are considered low pass filters. Instead only the masks, which are considered high pass filters, are used. To construct the even-number masks to boost steganalysis capability, we use the well-known [-1,1] mask as two-tap mask to convolute the S3 (one kind of Laws' mask), i.e., [-1,2,-1] to form by our four-tap mask. The six-tap mask is formulated in the similar fashion. Our experimental works have verified the contribution made by these masks.

We have achieved an average detection accurate rate of 83.92% in the BOSSbase 0.92 [13, 14] (at payload 0.4 bpp) in our experiment after this initial study, which has indicated that our proposal to utilize techniques developed in the field of texture classification for steganalysis is valid. Hence, our future plan is to continue this approach to enhance the capability in modern steganalysis.

References

[1] Avcibas, M., Memon, N., Sankur, B.: Steganalysis Using Image Quality Metrics. In: SPIE, EI, Security and Watermarking of Multimedia Content, San Jose, CA (February 2001)

[2] Farid, H., Siwei, L.: Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 340–354. Springer, Heidelberg (2003)

[3] Harmsen, J.J.: Steganalysis of Additive Noise Modelable Information Hiding. Master Thesis of Rensselaer Polytechnic Institute, Troy, New York, advised by Professor W. A. Pearlman (2003)

[4] Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Zhang, Z., Chai, P., Chen, C.-H., Chen, W.: Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 262–277. Springer, Heidelberg (2005)

[5] Fridrich, J.: Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 67–81. Springer, Heidelberg (2004)

[6] Sullivan, K., Madhow, U., Chandrasekaran, S., Manjunath, B.S.: Steganalysis of Spread Spectrum Data Hiding Exploiting Cover Memory. In: SPIE 2005, vol. 5681, pp. 38–46 (2005)

[7] Zou, D., Shi, Y.Q., Su, W., Xuan, G.: Steganalysis Based on Markov Model of Thresholded Prediction-Error Image. In: IEEE International Conference on Multimedia and Expo., Toronto, Canada (July 2006)

[8] Pevny, T., Bas, P., Fridrich, J.: Stegabalysis by subtractive pixel adjacency matrix. In: ACMM MSEC Princeton, NJ, USA, September 7-8 (2009)

[9] Pevný, T., Filler, T., Bas, P.: Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 161–177. Springer, Heidelberg (2010)

[10] Fridrich, J., Kodovský, J., Holub, V., Goljan, M.: Breaking HUGO – The Process Discovery. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 85–101. Springer, Heidelberg (2011)

[11] Fridrich, J., Kodovský, J., Holub, V., Goljan, M.: Steganalysis of Content-Adaptive Steganography in Spatial Domain. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 102–117. Springer, Heidelberg (2011)

[12] Gul, G., Kurugollu, F.: A New Methodology in Steganalysis: Breaking Highly Undetectable Steganograpy (HUGO). In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 71–84. Springer, Heidelberg (2011)

[13] Bas, P., Filler, T., Pevný, T.: ”Break Our Steganographic System”: The Ins and Outs of Organizing BOSS. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 59–70. Springer, Heidelberg (2011)

[14] Filler, T., Pevný, T., Bas, P.: BOSS (July 2010), `http://boss.gipsa-lab.grenobleinp.fr/BOSSRank/`

[15] Haralick, R.M., Shanmugan, K., Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man and Cybernetics SMC 3(6), 610–621 (1973)

[16] Böhme, R.: Assessment of Steganalytic Methods Using Multiple Regression Models. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 278–295. Springer, Heidelberg (2005)

[17] Chen, C., Shi, Y.Q., Xuan, G.: Steganalyzing texture images. In: IEEE International Conference on Image Processing (ICIP 2007), Texas, US (September 2007)

[18] Petrou, M., Sevilla, P.G.: Image Processing Dealing with Texture. John Wiley & Sons Inc. (2006)

[19] Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29, 51–59 (1996)

[20] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)

[21] Weinberger, M., Seroussi, G., Sapiro, G.: LOCOI: A low complexity context-based lossless image compression algorithm. In: Proc. IEEE Data Compression Conf., pp. 140–149 (1996)

[22] Avcibas, I., Kharrazi, M., Memon, N., Sankur, B.: Image Steganalysis with Binary Similarity Measures. EURASIP Journal on Applied Signal Processing 17, 2749–2757 (2005)

JPEG-Compatibility Steganalysis Using Block-Histogram of Recompression Artifacts

Jan Kodovský and Jessica Fridrich

Department of ECE, Binghamton University, NY, USA
{fridrich,jan.kodovsky}@binghamton.edu

Abstract. JPEG-compatibility steganalysis detects the presence of embedding changes using the fact that the stego image was previously JPEG compressed. Following the previous art, we work with the difference between the stego image and an estimate of the cover image obtained by recompression with a JPEG quantization table estimated from the stego image. To better distinguish recompression artifacts from embedding changes, the difference image is represented using a feature vector in the form of a histogram of the number of mismatched pixels in 8×8 blocks. Three types of classifiers are built to assess the detection accuracy and compare the performance to prior art: a clairvoyant detector trained for a fixed embedding change rate, a constant false-alarm rate detector for an unknown change rate, and a quantitative detector. The proposed approach offers significantly more accurate detection across a wide range of quality factors and embedding operations, especially for very small change rates. The technique requires an accurate estimate of the JPEG compression parameters.

1 Introduction

When a JPEG image is decompressed to the spatial domain, the pixel values in each 8×8 block must be obtainable by decompressing an 8×8 block of quantized DCT coefficients. However, most steganographic algorithms change the pixels in a way that makes each block almost surely incompatible with the compression in the sense that no DCT coefficient block can decompress to such a modified block of pixels. This JPEG-compatibility attack was described for the first time in 2001 [6]. The assumption that the cover was originally stored as JPEG is not that unreasonable as the vast majority of images are stored as JPEGs and casual steganographers might hide data in the spatial domain in order to hide larger payloads or simply because their data hiding program cannot handle the JPEG format. In fact, while there are almost eight hundred publicly available applications that hide messages in raster formats, fewer than two hundred can hide data in JPEGs.[1]

The original JPEG-compatibility detection algorithm [6] strived to provide a mathematical guarantee that a given block was incompatible with a certain

[1] Statistics taken from a data hiding software depository of WetStone Tech.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 78–93, 2013.

JPEG quantization matrix, which required a brute-force search. With an increasing quality factor (decreasing value of the quantization steps), however, the complexity of this search rapidly increases making it impractically time consuming to use in practice. This prompted researchers to seek alternatives.

In 2008, a quantitative LSB replacement detector was proposed [1,2] as a version of the weighted stego-image (WS) analysis [5,7] equipped with uniform weights and a pixel predictor based on recompressing the stego image with a quantization table estimated from the stego image. This detector proved remarkably accurate and also fairly robust w.r.t. errors in the estimated quantization table as well as different JPEG compressors. Luo et al. [12] used the same recompression predictor but based their decision on the number of pixels in which the stego image and its recompressed version differed. This allowed detection of embedding operations other than LSB replacement.

The cover-image prediction based on recompression is fairly accurate for low quality factors. With decreasing size of the quantization steps, the quantization noise in the DCT domain becomes comparable to the quantization noise in the spatial domain and the recompression predictor becomes increasingly poor, preventing thus the detection of (or quantifying) the embedding changes. However, the recompression artifacts due to quantization in both domains cannot be completely arbitrary. In particular, it is highly unlikely that such artifacts would manifest as a single changed pixel or, in general, a small number of changed pixels. This motivated us in Section 4 to form a feature vector as the histogram of the number of mismatched pixels in 8×8 blocks after recompression. This 65-dimensional feature vector better distinguishes embedding changes from recompression artifacts and significantly improves the detection accuracy especially for low embedding rates. In Section 5, we report the detection accuracy of three types of detectors, interpret the results, and compare them to previous art. The paper is summarized in Section 7.

2 Notation and Preliminaries

We use the boldface font for matrices and vectors and the corresponding lower-case symbols for their elements. In particular, $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \mathcal{I}^{n_1 \times n_2}$, $\mathcal{I} = \{0, \ldots, 255\}$, and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$ will represent the pixel values of grayscale cover and stego images with $n = n_1 \times n_2$ pixels. For simplicity, we assume that both n_1 and n_2 are multiples of 8 and limit our exposition to grayscale images. This also allows us to use publicly available image datasets, such as the grayscale BOSSbase [4], which gives our results a useful context.

For convenience, images will also be represented by blocks, $\mathbf{X} = (\mathbf{X}^{(k)})$, $\mathbf{X}^{(k)} = (x_{ij}^{(k)})$, where now $i, j \in \{0, \ldots, 7\}$ index the pixels in the kth block, $k \in \{1, \ldots, n/64\}$, assuming, for example, that the blocks are indexed in a row-by-row fashion. For the purpose of this paper, we define the operator of JPEG compression on an 8×8 pixel block, $\mathbf{X}^{(k)}$, as $\mathrm{JPEG}_\theta(\mathbf{X}^{(k)}) = \mathbf{D}^{(k)} \in \mathcal{J}^{8 \times 8}$, where $\mathcal{J} = \{-1023, \ldots, 1024\}$ and $\mathbf{D}^{(k)}$ is the kth block of quantized Discrete Cosine Transform (DCT) coefficients. Here, θ stands for a vector parameter defining the compressor, such as the quantization table(s), the type of the JPEG compressor (e.g., Matlab `imwrite` or ImageMagick `convert`), and the implementation of the DCT, such as 'float', 'fast', 'slow'. The parameters related to the lossless compression in JPEG, such as the Huffmann tables, are not important for our problem.

Typically, the JPEG operator will be applied to the entire image in a block-by-block fashion to obtain an array of DCT coefficients of the same dimension, $\mathbf{D} \in \mathcal{J}^{n_1 \times n_2}$, as the original uncompressed image: $\text{JPEG}_\theta(\mathbf{X}) = \mathbf{D} = (\mathbf{D}^{(k)})$, $\text{JPEG}_\theta(\mathbf{X}^{(k)}) = \mathbf{D}^{(k)}$ for all k. We also define the JPEG decompression operator as $\text{JPEG}_\theta^{-1} : \mathcal{J}^{8\times 8} \to \mathcal{I}^{8\times 8}$. In short, $\text{JPEG}_\theta^{-1}(\mathbf{D}^{(k)})$ is the kth pixel block in the decompressed JPEG image $\text{JPEG}_\theta^{-1}(\mathbf{D})$. The decompression involves multiplying the quantized DCT coefficients by the quantization matrix, applying the inverse DCT to the resulting 8×8 array of integers, and quantizing all pixel values to $\mathcal{I}$. Note that JPEG_θ^{-1} is not the inverse of JPEG_θ, which is many-to-one. In fact, in general $\text{JPEG}_\theta^{-1}(\text{JPEG}_\theta(\mathbf{X})) \neq \mathbf{X}$; the difference between them will be called the recompression artifacts.

All experiments are carried out on the BOSSbase image database ver. 0.92 [4] compressed with Matlab JPEG compressor `imwrite` with different quality factors. The original database contains $9,074$ images acquired by seven digital cameras in their RAW format (CR2 or DNG) and subsequently processed by converting to grayscale, resizing, and cropping to the size of 512×512 pixels using the script available from [4].

3 Prior Art

In this paper, we compare to the WS detector adapted for decompressed JPEGs [1] and the method of Luo *et al.* [12]. Both methods output an estimate of the embedding change rate, β, defined as the ratio between the number of embedding changes and the number of all pixels.

3.1 WS Adapted for JPEG

Böhme's change-rate estimator of LSB replacement in decompressed JPEGs (WSJPG) is a version of the WS estimator:

$$\hat{\beta}_{\text{WSJPG}} = \frac{1}{n} \sum_{i,j=1}^{n_1,n_2} (y_{ij} - \bar{y}_{ij})(y_{ij} - \hat{y}_{ij}), \tag{1}$$

where $\bar{y} = y + 1 - 2 \bmod (y, 2)$ is y with its LSB "flipped,"

$$\hat{\mathbf{Y}} = (\hat{y}_{ij}) = \text{JPEG}_\theta^{-1}(\text{JPEG}_\theta(\mathbf{Y})), \tag{2}$$

is the recompression pixel predictor, and $\mathbf{R} = (r_{ij})$, $r_{ij} = y_{ij} - \hat{y}_{ij}$ is the residual. Note that both $\hat{\mathbf{Y}}$ and $\mathbf{R}$ depend on θ but we do not make this dependence explicit for better readability. The WSJPG estimator is limited to LSB replacement and will not work for other embedding operations, such as LSB matching.

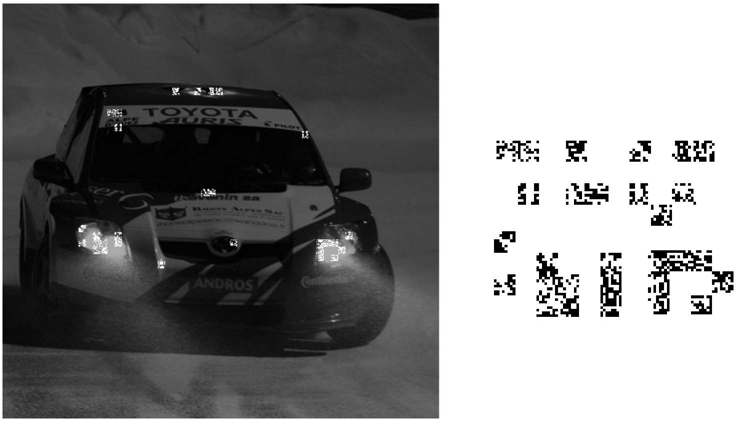


Fig. 1. Left: cover image '101.pgm' from BOSSbase compressed with quality factor 80. Right: close up of the recompression artifacts (grouped into a smaller region) with the same quality factor. The image contrast was decreased to better show the artifacts.

3.2 Detector by Luo *et al.*

The detector by Luo *et al.* [12] (which we abbreviate LUO) is also quantitative – it returns an estimate of the change rate as the detection statistic. It is computed from the relative number of differences between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$:

$$\triangle_\theta = \frac{1}{n}\left|\{(i,j)|r_{ij} \neq 0\}\right|. \tag{3}$$

In general, both the embedding changes as well as the recompression artifacts contribute to $\triangle_\theta$. Since the artifacts depend on θ, the authors further transform $\triangle_\theta$ to obtain an unbiased estimate of the change rate:

$$\hat{\beta}_{\mathrm{LUO}} = p_\theta(\triangle_\theta), \tag{4}$$

where $p_\theta(x)$ is a polynomial. The authors show that it is sufficient to consider a third degree polynomial, $p_\theta(x) = a_\theta + b_\theta x + c_\theta x^2 + d_\theta x^3$. Note that as long as the polynomial is monotone (as it seems to always be in [12]), $\triangle_\theta$ is an equivalent detection statistic, which is why we use it here for performance evaluation.

4 The Histogram Feature

Recompression artifacts manifest quite differently in the residual $\mathbf{R} = \hat{\mathbf{Y}} - \mathbf{Y}$ than the embedding changes. Figure 1 shows the cover image '101.pgm' from BOSSbase originally compressed with quality factor 80 together with the recompression artifacts. Although the artifacts typically occur in saturated areas, such as the overexposed headlights, they can show up in other regions with no saturated pixels (the car's hood and roof). The artifacts usually show up as a whole

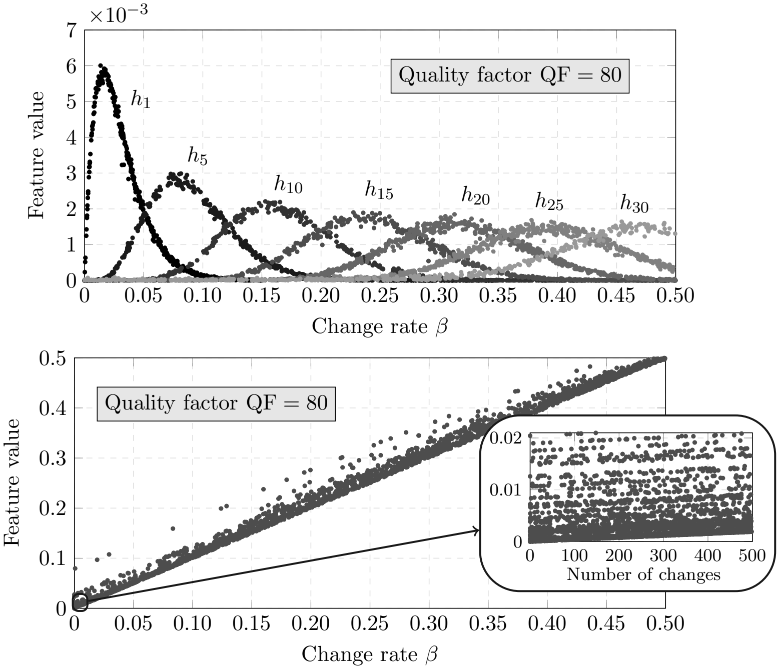


Fig. 2. Values of selected features h_i (top) and $\triangle_\theta$ (bottom) across 100 images and randomly selected change rates

pattern and almost never as individual pixels. Classifying them, however, would be infeasible as there are simply too many possible patterns and their number quickly increases with the quality factor. In fact, this is why the search in [6] is computationally intractable.

In this paper, we delegate the difficult task of distinguishing "legitimate" recompression artifacts from those corrupted by embedding changes to machine learning. To this end, each block, $\mathbf{R}^{(k)}$, of the residual is represented using a scalar – the number of pixels in $\mathbf{R}^{(k)}$ for which $r_{ij}^{(k)} \neq 0$. Denoting this number as $0 \leq \rho^{(k)} \leq 64$, $k = 1, \ldots, n/64$, each image will be mapped to a feature vector $\mathbf{h} = (h_m)$ obtained as the histogram of $\rho^{(k)}$:

$$h_m = \frac{64}{n} \left| \{k | \rho^{(k)} = m\} \right|, \quad m = 0, \ldots, 64. \tag{5}$$

This feature vector can be considered as a generalization of (3) because $\triangle_\theta = \frac{1}{64} \sum_{m=0}^{64} m h_m$ is a projection of $\mathbf{h}$ onto a fixed direction.

Using 100 randomly selected images and a large number of change rates, in Figure 2 (top) we show how the individual features h_m react to increasing change rate. Together, the features capture the effects of embedding much better than the scalar $\triangle_\theta$. For example, a small number of embedding changes affect primarily h_1 while the recompression artifacts typically disturb h_m with a much larger m. In contrast, $\triangle_\theta$ cannot distinguish embedding changes from recompression artifacts. Zooming in Figure 2 (bottom) around $\beta = 0$ reveals individual "lines" of dots corresponding to the 100 tested images. The vertical offset of the lines is due to recompression artifacts that introduce undesirable noise into $\triangle_\theta$, which prevents reliable detection (and estimation) of small change rates.

We close this section with one more remark. Detecting steganography using a binary classifier with a higher-dimensional feature is usually considered as less convenient or practical than alternative detectors that, for example, provide an estimate of the change rate. This is mainly because one needs to train the classifier on examples of cover (and stego) images from a given source. However, when images from a different source are tested, one may experience a loss of detection accuracy due to lack of robustness of today's classifiers to model mismatch (when one trains on one source but tests on another). In our case, however, the effect of the model mismatch is largely mitigated due to the fact that *all* JPEG-compatibility attacks require the knowledge of the JPEG parameter θ to apply in the first place. The source of JPEG images compressed with one quality factor is much more homogeneous than images in their uncompressed format because the compression suppresses the noise and thus evens out the source, making the issue with model mismatch less serious.

5 Experiments

This section contains all experiments and their interpretation. First, we measure the detection reliability of a clairvoyant detector (built for a specific change rate) across a wide spectrum of JPEG quality factors while comparing the results with WSJPG and LUO. Then, a single constant false-alarm rate (CFAR) detector is built to detect all change rates. Finally, we construct and test a quantitative version of the detector. All experiments are carried out under the assumption that the JPEG compressor parameter θ is correctly estimated, postponing the discussion of detector robustness to Section 6.

5.1 Classifier

The clairvoyant detector and the CFAR detector are instances of the ensemble [9,8] available from `http://dde.binghamton.edu/download/ensemble`. The ensemble reaches its decision using majority voting by fusing decisions of L individual base learners implemented as Fisher linear discriminants trained on random d_{sub}-dimensional subspaces of the feature space. The random subspace dimensionality, d_{sub}, and the number of base learners, L, are determined automatically by measuring the out-of-bag estimate of the testing error on bootstrap samples of the training set as described in [9].

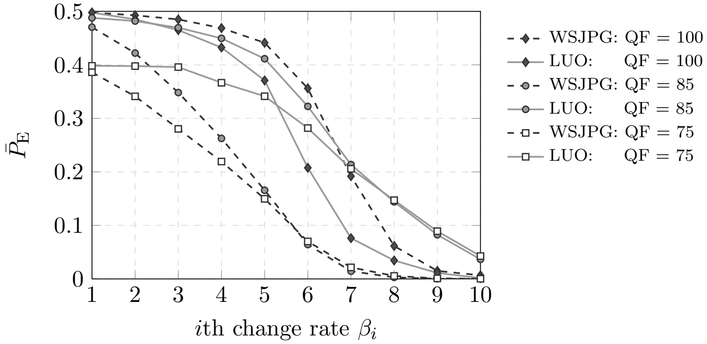


Fig. 3. Detection error $\bar{P}_\mathrm{E}$ for WSJPG (dashed lines) and LUO (solid lines) for all ten change rates $\beta_1, \dots, \beta_{10}$ and three selected quality factors 75, 85, and 100. Steganographic algorithm: LSB replacement.

5.2 Clairvoyant Detector

In this section, detection accuracy will be measured using the minimal total error under equal priors on the testing set:

$$P_\mathrm{E} = \min_{P_\mathrm{FA}} \frac{P_\mathrm{FA} + P_\mathrm{MD}(P_\mathrm{FA})}{2}, \tag{6}$$

where P_FA and P_MD are the false-alarm and missed-detection rates. We always report the mean value of P_E, denoted as $\bar{P}_\mathrm{E}$, over ten random splits of BOSSbase into equally-sized training and testing sets. Since the spread of the error over the splits, which includes the effects of randomness in the ensemble construction (e.g., formation of random subspaces and bootstrap samples), is typically very small, we do not show it in tables and graphs. We note that a separate classifier was trained for each β, which is why we call it clairvoyant.

First, we work with LSB replacement to be able to compare to the WSJPG detector. The focus is on detection of very small change rates:

$$\beta_i = \begin{cases} \frac{1}{n}(1, 10, 25, 50, 100) & \text{for } i = 1, \dots, 5, \\ 0.001, 0.0025, 0.005, 0.01, 0.02 & \text{for } i = 6, \dots, 10. \end{cases} \tag{7}$$

as this is where we see the biggest challenge in steganalysis in general. The actual embedding changes were always made pseudo-randomly and different for each image. The first five change rates correspond to making 1, 10, 25, 50, and 100 pseudo-randomly placed embedding changes. Note that the change rate $\beta_6 = 0.001$ corresponds to 261 embedding changes for BOSSbase images, continuing thus the approximately geometric sequence of $\beta_1, \dots, \beta_5$. Furthermore, β is the expected change rate when embedding 2β bits per pixel (bpp) if no matrix embedding is employed or the payload of $H^{-1}(\beta)$ bpp if the optimal binary coder is used ($H^{-1}(x)$ is the inverse of the binary entropy function on $x \in [0, 0.5]$).

Table 1. Mean detection error $\bar{P}_{\mathrm{E}}$ for the proposed method (shaded) versus WSJPG

QF	Number of changed pixels					Change rate (cpp)				
	1	10	25	50	100	0.001	0.0025	0.005	0.01	0.02
70	0	0	0	0	0	0	0	0	0	0
	0.3873	0.3468	0.2922	0.2295	0.1568	0.0763	0.0230	0.0057	0.0009	0.0003
75	0	0	0	0	0	0	0	0	0	0
	0.3861	0.3412	0.2804	0.2194	0.1497	0.0701	0.0216	0.0057	0.0010	0.0003
80	0	0	0	0	0	0	0	0	0	0
	0.4248	0.3761	0.3014	0.2295	0.1471	0.0625	0.0167	0.0037	0.0005	0.0003
85	0.0101	0	0	0	0	0	0	0	0	0
	0.4704	0.4220	0.3483	0.2626	0.1657	0.0642	0.0145	0.0029	0.0003	0.0002
90	0.0852	0.0046	0.0007	0.0010	0	0	0	0	0	0
	0.4899	0.4534	0.3950	0.3155	0.2197	0.0882	0.0183	0.0034	0.0005	0.0002
91	0.0798	0.0019	0.0001	0	0	0	0	0	0	0
	0.4913	0.4513	0.3882	0.3080	0.2076	0.0808	0.0167	0.0031	0.0004	0.0001
92	0.0893	0.0010	0	0	0	0	0	0	0	0
	0.4907	0.4505	0.3852	0.2981	0.1968	0.0722	0.0157	0.0032	0.0003	0.0001
93	0.4499	0.1017	0.0023	0	0	0	0	0	0	0
	0.4949	0.4727	0.4313	0.3673	0.2583	0.0936	0.0196	0.0040	0.0005	0.0001
94	0.4888	0.3885	0.2448	0.0906	0.0124	0.0003	0	0	0.0000	0
	0.4966	0.4802	0.4527	0.4094	0.3291	0.1482	0.0314	0.0081	0.0016	0.0003
95	0.4948	0.4472	0.3680	0.2538	0.0977	0.0025	0	0	0	0
	0.4972	0.4841	0.4611	0.4285	0.3589	0.1854	0.0372	0.0092	0.0028	0.0003
96	0.4973	0.4728	0.4320	0.3675	0.2509	0.0488	0.0018	0.0002	0.0001	0
	0.4975	0.4868	0.4680	0.4386	0.3797	0.2151	0.0499	0.0104	0.0028	0.0005
97	0.4983	0.4842	0.4595	0.4208	0.3438	0.1512	0.0178	0.0024	0.0003	0.0001
	0.4975	0.4877	0.4723	0.4433	0.3890	0.2316	0.0557	0.0108	0.0030	0.0007
98	0.4982	0.4795	0.4475	0.3936	0.3009	0.1744	0.0272	0.0034	0.0003	0.0001
	0.4980	0.4892	0.4725	0.4462	0.3911	0.2446	0.0587	0.0121	0.0024	0.0005
99	0.4988	0.4843	0.4602	0.4195	0.3398	0.1525	0.0161	0.0007	0	0
	0.4979	0.4899	0.4766	0.4588	0.4169	0.3016	0.1110	0.0226	0.0036	0.0007
100	0.4986	0.4855	0.4611	0.4251	0.3540	0.0942	0.0048	0.0006	0.0001	0.0001
	0.4978	0.4926	0.4849	0.4688	0.4413	0.3561	0.1920	0.0616	0.0151	0.0068

For such small β, the WSJPG method performed better than LUO with the exception of quality factor 100 (see Figure 3). Thus, in Table 1 we contrast the proposed method with WSJPG. The improvement is apparent across all quality factors and change rates and is especially large for the five smallest change rates. Remarkably, the clairvoyant detector allows reliable detection of a single embedding change for quality factors up to 92. Then the error abruptly increases. This is related to the first occurrence of '1' in the quantization table. With this quantization step, the rounding error in the spatial domain becomes comparable to the rounding error in the DCT domain and the recompression predictor no longer provides an accurate estimate of the cover. Despite this limitation, reliable detection of change rates $\beta_6, \ldots, \beta_{10}$ is still possible even for high quality factors. It appears that the least favorable quality factor is not 100 but 98 (for change rates $\beta_i, i > 5$). The detection error is not monotone w.r.t. the quality factor and one can observe "ripples" even at lower quality factors (e.g., from 90 to 91).

Table 2. Average detection error $\bar{P}_{\mathrm{E}}$ for HUGO

	Number of changed pixels					Change rate (cpp)				
QF	1	10	25	50	100	0.001	0.0025	0.005	0.01	0.02
80	.0213	.0017	.0022	.0016	.0018	.0017	.0013	.0007	.0006	.0004
90	.1235	.0160	.0065	.0035	.0049	.0035	.0023	.0024	.0024	.0012
95	.4953	.4627	.3974	.3306	.2415	.0859	.0286	.0191	.0076	.0023

We note that our feature vector **h** (5) as well as Luo's $\triangle_\theta$ work well for other steganographic methods than LSB replacement. Repeating the above experiment with LSB matching, we obtained identical values of $\bar{P}_{\mathrm{E}}$ well within its statistical spread. Interestingly, content-adaptive embedding appears to be slightly less detectable, which is most likely due the fact that recompression artifacts weakly correlate with texture/edges. The results for the content-adaptive HUGO [14] displayed in Table 2 should be contrasted with the corresponding rows of Table 1.[2]

5.3 CFAR Detector

In the previous experiment, a separate classifier was trained for each change rate and quality factor. However, in practice, the steganalyst will likely have no or little prior information about the payload and will face the more difficult one-sided hypothesis testing problem of deciding whether $\beta = 0$ or $\beta > 0$. For this purpose, we now construct a single CFAR classifier and report its performance for LSB replacement.

Following the recipe in [13], we first tried training on a uniform mixture of change rates from a certain range. This, however, caused the detector to be undesirably inaccurate for small change rates. There appears to be an interesting interplay between the design false-alarm rate, the ability to detect small change rates, and the detection rate. Through a series of experiments, we determined that the best results were obtained when training on a *fixed small* change rate for which the clairvoyant detector's P_{E} was neither too small or too big (a value in the range $P_{\mathrm{E}} \approx 0.2 - 0.3$ seemed to work the best). This makes an intuitive sense as $P_{\mathrm{E}} \approx 0.5$ would not allow accurate determination of the direction into which the features move with embedding, while easy detectability, $P_{\mathrm{E}} \approx 0$, is also bad as there exist many decision boundaries that are equally good but only some of them are useful for smaller change rates.

The performance of the detector for three quality factors is displayed in Figure 4. Three graphs show the detection rate $P_{\mathrm{D}}(\beta)$ for selected design P_{FA}. Overall, the false-alarm rates on the testing set agreed rather well with the design rates, which we show only for the quality factor 100 just as an example.

[2] To obtain the desired change rate β_i, we searched for the payload iteratively using the authors' embedding script.

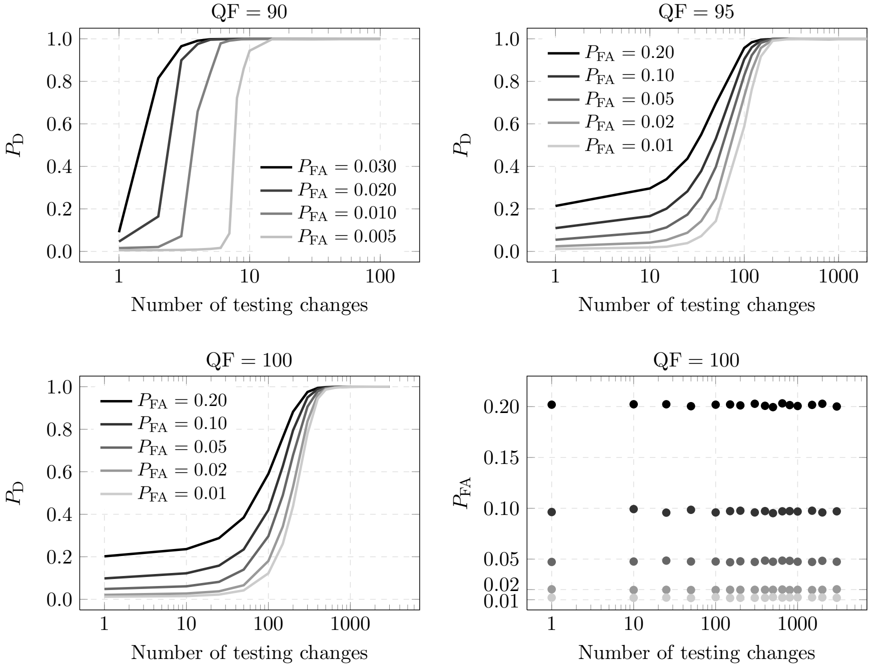


Fig. 4. Probability of detection P_D on the test set as a function of β for several design false alarm rates P_{FA} and three quality factors. For the highest quality factor, we also report the false alarm rate on test images. The CFAR classifier for quality factors 90, 95, and 100 was trained on 10, 25, and 50 changes, respectively.

For quality factor 90, even as few as six embedding change can be detected reliably with $P_{FA} = 0.01$. For quality factors 95 and 100, P_D experiences a sharp increase around 100 changes.

5.4 Quantitative Detector

Since WSJPG and LUO are both quantitative detectors, in this section we built a quantitative version of our detector using Support Vector Regression (SVR) and compare to previous art (tests carried out for LSB replacement).

Following the methodology described in [15], the BOSSbase was divided into two halves, one used to train the quantitative detector and the other used for testing. We used ν-SVR [16] with a Gaussian kernel whose hyper-parameters (kernel width, γ, cost, C, and the parameter ν which bounds the number of support vectors) were determined using five-fold cross-validation on $\mathcal{G}_\gamma \times \mathcal{G}_C \times \mathcal{G}_\nu$, where $\mathcal{G}_\gamma = \{2^k | k = -5, \ldots, 3\}$, $\mathcal{G}_C = \{10^k | k = -3, \ldots, 4\}$, and $\mathcal{G}_\nu = \{\frac{1}{10}k | k = 1, \ldots, 9\}$. We used a public SVM package libSVM [3].

The regressor was trained on images embedded with change rates chosen uniformly and pseudo-randomly from $[0, b]$. Its accuracy was measured on stego images from the testing set embedded with a fixed change rate β using relative bias, $B_r(\beta)$, and relative median absolute deviation (MAD) $M_r(\beta)$:

Table 3. Relative bias and median absolute deviation, $B_r(\beta) \pm M_r(\beta)$, as a function of β. Crosses correspond to failures (either B_r or M_r is larger than 50%). The best performance per change rate is highlighted. JPEG quality factor is 90.

β	Proposed scheme				Cascade
	$b = 0.0005$	$b = 0.005$	$b = 0.05$	$b = 0.5$	
$10/n$	-2.78 ± 4.84	×	×	×	-2.78 ± 4.84
$50/n$	$+0.64 \pm 2.34$	-9.04 ± 8.06	×	×	$+0.65 \pm 2.35$
$100/n$	-0.22 ± 2.00	-3.36 ± 4.13	-15.6 ± 28.5	×	-0.10 ± 2.02
0.001	-3.83 ± 1.72	-0.19 ± 1.75	-5.326 ± 10.9	×	-0.19 ± 1.75
0.0035	-16.4 ± 1.37	$+0.11 \pm 0.71$	-0.47 ± 3.06	×	$+0.13 \pm 0.71$
0.01	-43.7 ± 1.07	-0.90 ± 0.80	-0.00 ± 1.06	-16.3 ± 17.2	-0.00 ± 1.06
0.035	×	×	$+0.05 \pm 0.40$	-3.74 ± 4.68	$+0.07 \pm 0.40$
0.1	×	×	-21.1 ± 1.17	-1.17 ± 1.74	-1.27 ± 1.67
0.2	×	×	×	-0.57 ± 0.94	-0.57 ± 0.94
0.3	×	×	×	-0.26 ± 0.79	-0.24 ± 0.74
0.4	×	×	×	$+0.02 \pm 0.51$	$+0.04 \pm 0.47$
0.5	×	×	×	-0.90 ± 1.52	-0.96 ± 1.49

$$B_\mathrm{r}(\beta) = \frac{1}{\beta}(\mathrm{med}(\hat{\beta}) - \beta) \times 100\%, \tag{8}$$

$$M_\mathrm{r}(\beta) = \frac{1}{\beta}\mathrm{med}(|\hat{\beta} - \mathrm{med}(\hat{\beta})|) \times 100\%, \tag{9}$$

where $\hat{\beta}$ is the estimated change rate and the median $\mathrm{med}(\cdot)$ is always taken over all stego images in the testing set. Note that $B_\mathrm{r}(\beta)$ is the *percentual* inaccuracy in estimating β, while $M_\mathrm{r}(\beta)$ captures the statistical spread in the same units. These relative quantities are more informative when detecting change rates of very different magnitudes.

Table 3 shows $B_\mathrm{r}(\beta) \pm M_\mathrm{r}(\beta)$ when training on stego images embedded with change rates from $[0, b]$ for four values of b for JPEG quality factor 90. The detection was declared unsuccessful, and marked by a cross, when either $B_\mathrm{r}(\beta)$ or $M_\mathrm{r}(\beta)$ was larger than 50%. The table reveals that for small β, significantly better results could be obtained by training the regressor on a smaller range $[0, b]$, provided $\beta < b$. This is because a smaller interval yields a higher density of training change rates and allows the regressor to locally adjust its hyper-parameters.

This insight inspired us to construct the quantitative detector by cascading SVR detectors D_i trained on progressively smaller ranges $[0, b_i]$, $b_i > b_{i+1}$, $b_i \in [0, 0.5]$:

1. Set $\mathbf{b} = (b_1, \ldots, b_k)$, initialize $i = 1$.
2. Compute $\hat{\beta}_i$ using D_i. If $i = k$, terminate and output $\hat{\beta}_i$.
3. If $\hat{\beta}_i \leq b_{i+1}$, increment $i = i + 1$, go to Step 2.
4. Output $\hat{\beta}_i$.

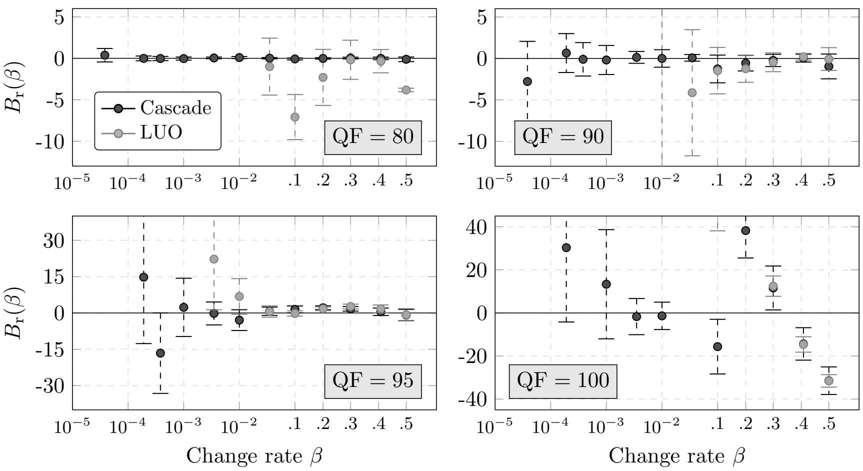


Fig. 5. Quantitative steganalysis of LSB replacement for 'Cascade' and LUO for different JPEG quality factors in terms of the relative median bias B_r; error bars depict M_r. Note the different ranges on y-axis.

The performance of this cascading regressor is reported in the last column of Table 3. As expected, it strongly benefits from its individual sub-detectors and consequently delivers superior performance across all change rates. To complete the picture, in Figure 5 we compare LUO with 'Cascade' for JPEG quality factors, 80, 90, 95, and 100. While both estimators become progressively inaccurate with increasing JPEG quality factor, 'Cascade' clearly outperforms LUO for small β in all cases while both estimators become comparable for larger β. We note that cascading the regressor for $\triangle_\theta$ by training on smaller intervals $[0, b]$ did not improve its performance. This is due to the low distinguishing power of $\triangle_\theta$ on smaller change rates (see Figure 2 bottom).

For quality factor 100 and $\beta \gtrsim 0.2$, neither of the two detectors can estimate the change rate reliably, and both begin outputting an estimate of $\hat{\beta} \approx 0.35$ (on average). This is because in this range the features are very noisy due to recompression artifacts – the quantization table consists solely of ones. Consequently, the regression learns the output that yields the smallest error on average.

5.5 Error Analysis

We now decompose the compound error of the proposed quantitative detector trained on $[0, 0.5]$ into the within-image error, E_W, and the between-image error, E_B, using the procedure described in [2].

The tails of the E_W distribution are analyzed by randomly selecting a single image from the testing set followed by 200 independent realizations of LSB embedding at a fixed change rate. Our experiments confirm that this error follows the Gaussian distribution. To estimate the between-image error, we compute

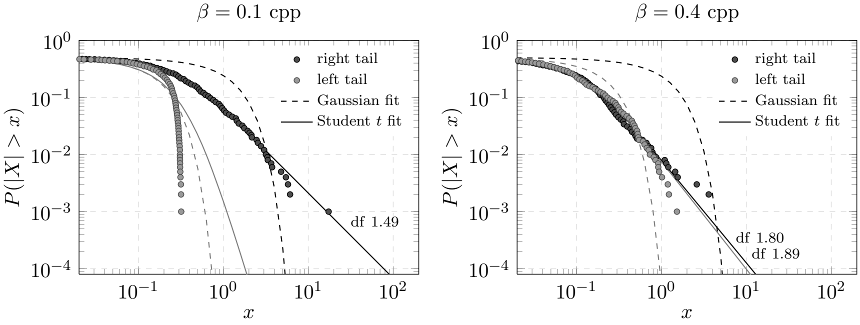


Fig. 6. Tail probability for the between-image error E_{B} for $\beta = 0.1$ and 0.4 with the Gaussian and the Student's t maximum likelihood fits. JPEG quality factor 90.

the change rate estimate for 1000 testing images by averaging estimates over 20 embedding realizations (for every image). The log-log empirical cdf plot of the resulting estimates is shown in Figure 6 for two selected values of β. While the the Student's t-distribution was generally a good fit for the right tail, we observed great variations in the distribution of the left tail based on the value of β. The tail could be extremely thin for some β, while for others it did follow the thick-tailed Student's t-distribution. We attribute these variations to the highly non-linear dependence of the feature vector on β seen in Figure 2.

6 Robustness to JPEG Compressor Parameters

The WSJPG detector appears to be quite resistant to incorrectly estimated quantization table or the JPEG compressor [2]. This is because stronger recompression artifacts due to improperly estimated compression parameter θ are not likely to manifest as flipped LSBs. In contrast, our feature vector, as well as LUO, are rather sensitive to θ because they *count* the mismatched pixels instead of utilizing their *parity*. While this allows them to detect embedding operations other than LSB flipping, this generality lowers their robustness.

The overall detection performance of any JPEG-compatibility detector will necessarily strongly depend on the accuracy of the estimator of θ as well as the prior distribution of θ in the testing set. Despite some encouraging work, such as [11], we consider the problem of estimating θ as an open and quite difficult problem for the following reasons. Most JPEG images today originate in digital cameras, which, unfortunately, almost exclusively use quantization tables customized for the image content, the imaging sensor, the manufacturer's color space, and the image size [17].[3] For color images, one may have to estimate up

[3] http://www.hackerfactor.com/blog/index.php?/archives/244-Image-Ballistics-and-Photo-Fingerprinting.html
http://www.impulseadventure.com/photo/jpeg-quantization.html

to three quantization tables, one for the luminance and one for each chrominance component, as well as the chrominance subsampling. The quantization tables may even be different between different cameras of the same model as manufacturers continue to upgrade the firmware. Multiple JPEG compressions further complicate the matter. Thus, the search space may be quite large even when one considers estimating only the quantization tables themselves. Methods that estimate the individual quantization steps, such as [6,11,10], may fail for high compression ratios as there may be little or no data in the JPEG file to estimate the quantization steps for sparsely populated medium–high frequency DCT modes.

The only meaningful evaluation of the robustness requires the steganalyzer to be tested as a whole system, which includes the compression estimator, and testing on non-standard quantization tables as well as multiply compressed images. The authors feel that the problem of robust compression parameter estimation is a separate issue that is beyond the scope of this paper.

7 Conclusions

This paper describes a new implementation of JPEG-compatibility steganalysis capable of detecting a wide range of embedding operations at very low change rates. As proposed previously, the image under investigation is first recompressed with a JPEG compressor estimated from the test image. The recompression artifacts are described using a 65-dimensional feature vector formed as the histogram of blocks with a certain number of mismatched pixels. This feature vector can better distinguish between recompression artifacts and embedding changes than the scalar proposed by Luo *et al.* [12]. In particular, it allows accurate detection of fewer than ten embedding changes for quality factors up to 92. For higher quality factors, the detection error sharply increases due to the onset of quantization steps equal to one. Nevertheless, very reliable detection of change rates as low as 0.005 remains possible for quality factors up to 100 (in 512×512 grayscale images).

Three types of detectors are constructed for a fixed quality factor – a family of clairvoyant detectors trained for a specific change rate, a constant false-alarm rate detector for unknown change rate for practical applications, and a quantitative detector.

The proposed method, as well as all JPEG-compatibility detectors, need to be supplied with an estimator of the JPEG compressor parameters (quantization table(s), DCT implementation, etc.). Future research will focus on tests with real-life datasets, including images compressed with non-standard quantization tables and multiply-compressed images, and on extension of this work to color images. The latter would require estimation of chrominance quantization table(s) as well as chrominance subsampling.

Acknowledgements. The work on this paper was partially supported by Air Force Office of Scientific Research under the research grants number FA9550-08-1-0084 and FA9950-12-1-0124. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of AFOSR or the U.S. Government. The authors would like to thank Vojtěch Holub and Miroslav Goljan for useful discussions and Rainer Böhme for help with correctly implementing the WS attack.

References

1. Böhme, R.: Weighted Stego-Image Steganalysis for JPEG Covers. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 178–194. Springer, Heidelberg (2008)
2. Böhme, R.: Advanced Statistical Steganalysis. Springer, Heidelberg (2010)
3. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Filler, T., Pevný, T., Bas, P.: BOSS (Break Our Steganography System) (July 2010), http://www.agents.cz/boss/
5. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI. Proceedings SPIE, San Jose, CA, January 19–22, vol. 5306, pp. 23–34 (2004)
6. Fridrich, J., Goljan, M., Du, R.: Steganalysis based on JPEG compatibility. In: Tescher, A.G. (ed.) Special Session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV, Denver, CO, August 20–24, vol. 4518, pp. 275–280 (2001)
7. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA. Proceedings SPIE, January 27–31, vol. 6819, pp. 5:1–5:17 (2008)
8. Kodovský, J., Fridrich, J.: Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In: Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics of Multimedia XIII, San Francisco, CA, January 23–26, vol. 7880, pp. OL 1–OL 13 (2011)
9. Kodovský, J., Fridrich, J., Holub, V.: Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security 7(2), 432–444 (2012)
10. Lewis, A.B., Kuhn, M.G.: Exact JPEG recompression. In: Memon, N.D., Delp, E.J., Wong, P.W., Dittmann, J. (eds.) Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII, San Jose, CA, January 17–21, 2010, vol. 7543, p. 75430V (2010)
11. Luo, W., Huang, F., Huang, J.: JPEG error analysis and its applications to digital image forensics. IEEE Transactions on Information Forensics and Security 5(3), 480–491 (2010)
12. Luo, W., Wang, Y., Huang, J.: Security analysis on spatial ±1 steganography for JPEG decompressed images. IEEE Signal Processing Letters 18(1), 39–42 (2011)
13. Pevný, T.: Detecting messages of unknown length. In: Memon, N.D., Delp, E.J., Wong, P.W., Dittmann, J. (eds.) Electronic Imaging, Media Watermarking, Security and Forensics of Multimedia XIII, San Francisco, CA. Proceedings SPIE, January 23–26, vol. 7880, pp. OT 1–OT 12 (2011)

14. Pevný, T., Filler, T., Bas, P.: Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 161–177. Springer, Heidelberg (2010)
15. Pevný, T., Fridrich, J., Ker, A.D.: From blind to quantitative steganalysis. IEEE Transactions on Information Forensics and Security 7(2), 445–454 (2012)
16. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). The MIT Press (2001)
17. Taro, Y.: Image coding apparatus and method, US Patent 6968090 (2005)

Hiding a Second Appearance in a Physical Relief Surface

Yi-Liu Chao and Daniel G. Aliaga

Purdue University

Abstract. We present a novel information hiding process that couples geometrical modeling with automated 3D fabrication for creating hidden-appearance reliefs. Our relief surface produces a first grayscale appearance visible by simple direct illumination and a second grayscale appearance ensured to be visible when the relief is lit by a digital projector with a specifically designed pattern and from a particular direction. The two appearances/images can be different yet embedded in the same physical relief. Since the second appearance appears only on demand, it could be used to hide a second image, a company logo, or a watermark image, for instance. Our novel method calculates a relief surface that maintains the properties needed for producing a second (hidden) appearance while also ensuring the first appearance is visible under normal direct illumination. Our experiments show that our method robustly produces reliefs with two arbitrary desired appearances.

Keywords: information hiding, images, reliefs, surfaces, watermarks, 3D manufacturing.

1 Introduction

In this paper, we present a novel application of information hiding whereby two visual appearances (or "images") are encoded into a single physical relief surface. Our work exploits advances in digital manufacturing but focuses on a computational modeling component. We wish to design a physical *relief* surface to have a first appearance visible to the naked eye under normal directional illumination and defined by a provided arbitrary image (Figure 1, shaded image A). In addition, we wish the same relief to have a second appearance, defined by an arbitrary second image, which is made visible only when the relief is lit by a carefully designed illumination pattern (e.g., by using a digital projector) (Figure 2, shaded image B). Since the second appearance can be made to appear only on demand, it could be used to hide a second image, a company logo, or a watermark image, for instance. To our knowledge, there is no previous information hiding approach as ours. Some previous works do incorporate multiple appearances into a relief/object, however our novel process ensures the second appearance is always possible despite potential self-shadows and the implicit finite projector light radiance. In the absence of our method, the two appearances/images are not always possible (Figure 1, bottom row). We anticipate our methodology will lead to significant more work in this exciting novel application.

Previously, papers have addressed generating surfaces with purposefully encoded data and/or purposefully crafted visual behaviors. In the synthetic world, a relevant set of works are algorithms which robustly or fragilely encode watermarks into the

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 94–109, 2013.

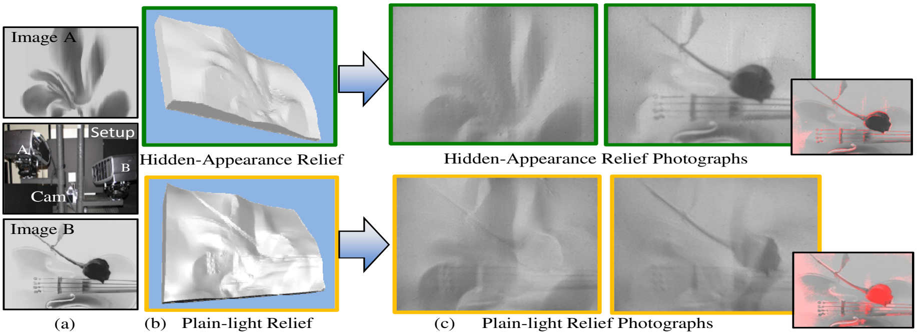


Fig. 1. Hidden-Appearance Reliefs. a) We create a relief that produces image A under normal direct illumination and image B only under projector illumination. b) We show our hidden relief geometry (top) and a plain-light relief geometry (bottom) – an implementation of [1]. c) Left pair are photographs of the digitally manufactured physical relief surface under direct illumination and yielding image A; right top is image B with our approach and right bottom is image B with plain-light relief geometry. The small insets are a visualization of the per pixel intensity error between the photographs and the desired image B (red = large error). As compared to previous works, our method is able to produce both images A and B, despite them being very different.

digital representation of the mesh (e.g., [2]). In the world of digitally manufactured physical objects, methods have encoded fragile marks (e.g., for genuinity detection) into physical surfaces [3] or into paper [4], and have manufactured surfaces yielding a pre-specified shading, or appearance, behavior (e.g., [5, 6]).

Our work is inspired by an observation in Chen et al. [7] which states that given any two appearances for a single diffuse surface there is, in theory, always a combination of surface geometry, albedo patterns, and light sources that can produce the appearance pair. In practice, the limited amount of light, manufacturing restrictions on heightfield sharpness, and self-shadows imposes practical restrictions on the images. However, we have found that a wide range of imagery is possible with our method. More concisely, our methodology for generating a hidden appearance in a relief is based on the following three key observations:

- *there are multiple relief geometries that yield the same shaded image when viewed from above the surface*; we exploit the multiplicity of solutions to find a combination of surface heights that produces both shaded image A and shaded image B;
- *if the relief were not designed to explicitly support/hide the second appearance, then the second image cannot in general be produced;* this is true even with the help of a projector emitting any desired illumination pattern; and
- *the use of an illumination pattern for generating the second appearance enables using a constant albedo to produce any two grayscale shaded images A and B*; image A is visible to the naked eye; however, since the albedo is constant (e.g., no paint or material change is visible to the naked eye), image B is only visible to the naked eye by using the proper illumination pattern; the pattern itself could, for example, be encoded, or generated, by a key-based procedure and our overall methodology ensures an arbitrary chosen second appearance is possible.

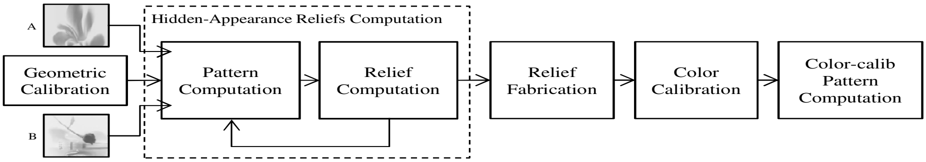


Fig. 2. System Pipeline. A summary of the major components of our method and the computational flow.

Our approach uses an optimization process to create a diffuse surface heightfield that will be subsequently manufactured. Creating a relief surface that produces the appearance of a single shaded image A under normal direct illumination is simple: using optimization a set of normals producing image A is computed and surface integration yields a suitable heightfield. From the many possible surface geometry configurations (as will be described in Section 3.2 and using Figure 4), we also wish to ensure an arbitrary shaded image B is visible when illuminated by a digital projector shining a particular illumination pattern. When creating a hidden-relief surface, a suitable combination of surface geometry and illumination pattern are initially unknown and must be determined. Further, since the projector illumination range is limited (i.e., light is additive and a given projector has a limited maximum illumination intensity), the pattern must also be constrained to lie within possible illumination values. A straightforward formulation results in a large set of constrained inequality equations that is nonlinear in the unknowns – solving this system is both impractical and highly non-robust. Instead, we perform several simplifications that result in an efficient solution using equations that are linear with respect to the unknowns and include linear smoothing equations and linear constraints. A suitable relief can be computed, tested in simulation, and then fabricated.

We have implemented a complete prototype system to produce *hidden reliefs* (Figure 2). The relief is automatically manufactured from our computed model using a 3D printer. The relief is placed on a stage in front of a digital camera and two digital projectors: one projector is used as a simple point light source to directly illuminate the relief so as to produce image A, and the other projector emits the illumination pattern for image B. The camera and projectors have been geometrically and radiometrically calibrated beforehand. Our results include the design and fabrication of several two appearance reliefs, and theoretical and empirical comparisons to previous related works (e.g., Alexa and Matusik [1] -- we call them *plain-light reliefs*) and to reliefs created for a single appearance but using projector patterns to obtain the second appearance (we call these *single-appearance reliefs*). Our experiments consistently demonstrate that using our approach yields an improved ability to encode both image A and image B into a single relief.

2 Related Work

Information hiding can be viewed as an exploitation of flexibility and, in some cases, redundancy. With this in mind, the concept of watermarking has been extended to the digital domain. Abundant literature investigates watermarks in digital images [8, 9] and in digital audio files [10]. It has been used to seamlessly hide watermarks in 3D meshes (e.g., [2, 11, 12]). Approaches are concerned with robustness, security, or both (e.g., [13]). These methods have been designed for digital data which can be created, read, and replicated with zero error. Some efforts have brought the watermark concept to physical surfaces. For instance, the work of Aliaga et al. [3] fabricates relief surfaces encoding watermarks such that physically copying the watermark is hard. While, similar to these works, we wish to embed information, we seek to do so in the visual appearance of a physical relief surface.

Although, we could hide a second appearance by secretly using relief materials or unbeknownst reflective paint patterns, we seek an "open method", closer to Kerckhoff's principle [14]. Our methodology assumes a public method and a single albedo (i.e., a single material and color) yet is still enable to hide an arbitrary second appearance (e.g., one that can be procedurally generated based on a key), and is visible only when appropriately illuminated. Additional surface hiding methodologies can be viewed as complementary.

Some previous surface and relief work has attempted to encode multiple appearances. For example, Oliva et al. [15] design a colored pattern (i.e., single flat colored image) which gives the illusion of a different appearance at different viewing distances. Alexa and Matusik [1] use constant albedo and alter the surface geometry so as to produce a different image when directly illuminated from one of two different directions. However, in both works the two involved images cannot be arbitrary -- they must be designed to work well together.

In contrast, our method yields two novel abilities. First, shaded image A and shaded image B can be arbitrary, very different, gray-shaded images. In fact, since we are using a projector to produce image B, it can even include the physically impossible shaded images described by Horn et al. [16], without affecting image A. Second, shaded image B is only visible when appropriately illuminated. While a geometrically and chromatically calibrated projector illumination system can impart a new appearance on physical surfaces (e.g., state-of-the-art report [17]), it is not sufficient to yield an arbitrary image A and image B. In particular, even with perfect calibration an arbitrary image B cannot be produced. Rather, the surface geometry must be altered so as to ensure image B can be produced by an illumination pattern while subject to the constraint of ensuring image A is what is visible under normal direct illumination of the relief – such a surface geometry is precisely what our method computes. Although, we do not explicitly maximize the imperceptibility of image B under normal illumination, its existence is not evident to the naked eye; in our results section, we do analyze the impact of the contrast and sharpness of image A and B on the hiding of image B. Collectively, these abilities lead to novel applications; for example, embedding a watermark into a physical relief or a desired alternative appearance suitable for other image processing (e.g., object tracking).

3 Hidden-Appearance Reliefs

The construction process for our reliefs iteratively finds a single surface that supports the two desired appearances. First, we describe the physical setup and present an appearance formation process for hidden-appearance reliefs. Then, we simplify the formulation and describe an iterative optimization process and smoothing equations.

3.1 Relief Setup

Our setup consists of the relief surface observed from above (i.e., viewing direction of $[0,0,-1]$) and two projectors (Figure 3a). A first projector is used to generate the directional light for producing image A. It is positioned along a direction l_A at a small angle to the viewing direction. A second projector is along the direction l_B at a larger angle to the viewing direction and is used to shine the illumination patterns for generating image B.

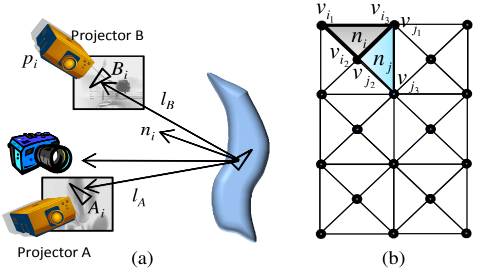


Fig. 3. Setup and Relief Mesh. a) A diagrammatic view of the relief, projectors, and camera. b) The symmetric triangulation of the relief mesh. Both diagrams are labeled with the variables used in our formulation.

Since the projector, camera, and relief mesh are usually of different resolutions, we choose to define computations in terms of each triangle $i \in [1, N]$ of the relief mesh M and project the triangle to the calibrated camera and projector image planes in order to calculate desired image intensity values and projector pattern values. This also enables us to control computational cost by altering the resolution of mesh M. In order to produce a symmetric mesh (i.e., one that is equivalent upon a rotation of 90, 180 or 270 degrees), we add an additional vertex in the middle of the quadrilaterals of a standard rectilinear meshing of vertices (Figure 3b). Furthermore, we assume M to be a heightfield over the XY plane, thus only the z-coordinates of the mesh vertices are free to move.

3.2 Appearance Formation

Our formulation of the appearance formation process uses a diffuse reflectance model to express the behavior of the relief mesh M under the desired two illumination scenarios. Figure 4 contains an intuitive and synthetic 2D example. Figure 4a shows a challenging pair of 1D image A and B -- image B is chosen as the "opposite image" of A, coincidentally an impossible shaded image as per Horn et al. [16]. The direct illumination for image A is from directly above and the pattern for producing image B is illuminated slightly from the right side. Figure 4b shows four surfaces (i-iv) that all yield image A; e.g., in the leftmost surface (i) the amount of reflected light is maximal in the middle and falls off to the sides, as in image A. The non-uniqueness of

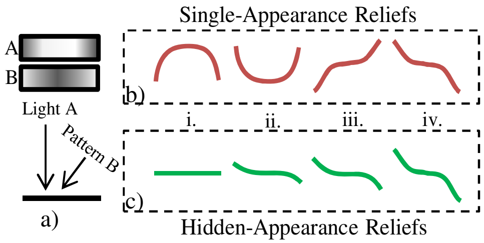


Fig. 4. Image Formation Example. a) Desired images A and B. b) Four possible reliefs (i-iv) that produce image A but not necessarily image B even when using a projector. c) Our optimization iteratively finds a relief surface able to produce both A and B (steps i-iv).

the solution for A is precisely what we exploit. However, these four meshes are not necessarily able to produce image B. Again, consider the leftmost surface in 4b: to yield a bright left-side of the surface, the projector must shine a large amount of light that might exceed the maximum illumination ability of the projector (even when disregarding self-occlusions). Figure 4c shows our approach which begins with a flat surface and iterative finds the relief surface (i-iv) that produces image A and also is able to produce image B with the help of a projector. In a full-fledged example, our iterative process concurrently finds over the entire surface a heightfield configuration able to produce both image A and B.

For producing appearance A, the image formation process for triangle i can be expressed by the well-known equation

$$\alpha(\mathrm{n_i} \cdot \mathrm{l_A}) = \mathrm{A_i} \tag{1}$$

where α is the constant surface albedo of M, $\mathrm{n_i}$ is the desired normal of relief triangle i, and $\mathrm{A_i}$ is the mean intensity of the pixels in A onto which triangle i projects.

To express the second appearance, we use the inequality

$$\alpha(\mathrm{n_i} \cdot \mathrm{l_B}) \geq \mathrm{B_i} \tag{2}$$

where $\mathrm{B_i}$ is the mean intensity of the pixels in B onto which triangle i projects. Equation (2) ensures that for surface normal $\mathrm{n_i}$ at least the intensity needed to yield $\mathrm{B_i}$ is possible. An inequality is appropriate because ultimately an illumination pattern is used and the pattern values can reduce the amount of incident light (but not increase it).

By explicitly factoring in the illumination patterns, the inequality in (2) can be converted to the following equality

$$\alpha \mathrm{p_i}(\mathrm{n_i} \cdot \mathrm{l_B}) = \mathrm{B_i} \tag{3}$$

where $\mathrm{p_i}$ is the mean intensity value of the projector pixels that cover relief triangle i.

Our approach seeks relief triangle normals $\mathrm{n_i}$ and bounded projector pattern values $\mathrm{p_i} \in [0,1]$ that simultaneously satisfy equations (1) and (3) for all triangles $\mathrm{i} \in [1, \mathrm{N}]$. We denote vertices of triangle i as $\mathrm{v_{i_1}}, \mathrm{v_{i_2}}, \mathrm{v_{i_3}}$ and write $\mathrm{n_i}$ in terms of the normalized cross product of the vertices:

$$\alpha \left(\frac{(\mathrm{v_{i_2}} - \mathrm{v_{i_1}}) \times (\mathrm{v_{i_3}} - \mathrm{v_{i_2}})}{\|(\mathrm{v_{i_2}} - \mathrm{v_{i_1}}) \times (\mathrm{v_{i_3}} - \mathrm{v_{i_2}})\|} \cdot \mathrm{l_A} \right) = \mathrm{A_i} \tag{4}$$

$$\alpha\, \mathrm{p_i} \left(\frac{(\mathrm{v_{i_2}} - \mathrm{v_{i_1}}) \times (\mathrm{v_{i_3}} - \mathrm{v_{i_2}})}{\|(\mathrm{v_{i_2}} - \mathrm{v_{i_1}}) \times (\mathrm{v_{i_3}} - \mathrm{v_{i_2}})\|} \cdot \mathrm{l_B} \right) = \mathrm{B_i}$$

which is a nonlinear expression with respect to the unknowns (i.e., the z coordinates of the triangle vertices and the pattern values $\mathrm{p_i}$). The complete equation set defined by (4) constitutes a large (though sparse) nonlinear irrational equation system that is difficult to solve.

3.3 Simplification

In order to efficiently and robustly solve for the relief mesh height values and projector pattern values, we use simplifying heuristics. Using the heuristics results is two sets of linear equations solved in an alternating fashion in order to incrementally find pattern values and relief heights, and to collectively approximate equations (4).

Our first heuristic is to assume the length of the triangle normals n_i are constant during an iteration and thus remove the square root in the denominator of equation (4). This rewrites equation (4) as

$$\frac{\alpha}{\|\hat{n}_i\|}\begin{bmatrix} y_{i_{21}}z_{i_{32}} - y_{i_{32}}z_{i_{21}} \\ z_{i_{21}}x_{i_{32}} - z_{i_{32}}x_{i_{21}} \\ x_{i_{21}}y_{i_{32}} - x_{i_{32}}y_{i_{21}} \end{bmatrix} \cdot l_A = A_i \tag{5}$$

$$\frac{\alpha}{\|\hat{n}_i\|}\, p_i \begin{bmatrix} y_{i_{21}}z_{i_{32}} - y_{i_{32}}z_{i_{21}} \\ z_{i_{21}}x_{i_{32}} - z_{i_{32}}x_{i_{21}} \\ x_{i_{21}}y_{i_{32}} - x_{i_{32}}y_{i_{21}} \end{bmatrix} \cdot l_B = B_i \tag{6}$$

where $\|\hat{n}_i\| = \left\|(v_{i_2} - v_{i_1}) \times \left(v_{i_3} - v_{i_2}\right)\right\|$. We write $v_{i_2} - v_{i_1}$as $[x_{i_{21}} \quad y_{i_{21}} \quad z_{i_{21}}]^T$ and $v_{i_3} - v_{i_2}$ similarly. Since $\|\hat{n}_i\|$ is considered a constant during an iteration and the x and y coordinates are fixed, equation (5) is a linear equation of the unknowns z_{i_j}'s. Equation (6) is not yet linear because of the multiplication by the unknown p_i. We use the following heuristic to further simplify the problem.

The second heuristic is to assume a current estimate either for mesh geometry M or for the pattern values p_i. This produces two formulations of equations (5) and (6).

i. The first formulation solves for p_i's using linear equations by assuming a known geometric mesh (i.e., all z's are constant). Equation (6) is used because no p_i term appears in equation (5). p_i values are restricted to the range $[0,1]$ by using a constrained linear optimization.
ii. The second formulation solves for the mesh heights by assuming constant values for p_i. This formulation uses both equations (5) and (6) which are now both linear in the unknowns (i.e., the z values of the vertices) and can be solved using linear optimization.

The full equations set are in general over-constrained but relatively sparse. Since a vertex is used in up to only eight adjacent triangles, the system of equations is always fairly sparse. Hence, a sparse (constrained) linear least squares optimization can solve formulation (i) or (ii) relatively quickly, even for a large number of mesh triangles.

3.4 Iterative Optimization

To compute the relief mesh, we iterate between solving for pattern values and for geometry mesh heights until converging to a final surface. The validity of our equations only holds for small height value changes. In particular, as the vertex heights change, the constant length of n_i's, the values of the p_i's, the triangle to projector correspondence used to calculate p_i's position on the projector image plane, and the triangle to camera correspondence used to compute the A_i's and B_i's pixel intensity become increasingly inaccurate. Hence, our optimization starts with a planar relief mesh and computes an initial set of pattern values p_i using formulation (i). Then, pattern values p_i are used to obtain new vertex heights using formulation (ii). The new mesh updates correspondence of relief, camera, and projectors, and triggers recomputing the pattern values and triangle normal lengths for the next iteration.

From a theoretical standpoint, our iterative optimization process and equation sets do not guarantee that a solution will be found nor that it is unique. Rather, we find a surface that satisfies the specified shading behavior in a least squares sense. In practice however, we found approximate solutions to always exist.

3.5 Smoothness

To provide support for ensuring incremental changes to the surface, for increased robustness, and for the creation of an approximately smooth surface (beneficial to physical manufacturing), we include additional equations. These equations ensure that the height changes of neighboring vertices are similar in one iteration. We define such an equation set for all edges in mesh M.

We rewrite the equations in terms of height changes in one computation in order to ensure similar variations of neighbors during an iteration. We denote with Δz_{i_j} the height change of the jth vertex of triangle i in one computation and we use Δz_u and Δz_v to represent height changes of two mesh vertices where the edge (u, v) is in relief mesh M. We rewrite equations (5) and (6) in terms of height changes and we incorporate a smoothness requirement. Altogether the per-iteration task is to minimize the following expression:

$$\sum\nolimits_{i\in[1,N]}\left[E_{Ai}\cdot\begin{bmatrix}\Delta z_{i_1}\\ \Delta z_{i_2}\\ \Delta z_{i_3}\end{bmatrix}-\left(\frac{\|\hat{n}_i^{(k)}\|}{\alpha}A_i-\hat{n}_i^{(k)}\cdot l_A\right)\right]^2+$$

$$\sum\nolimits_{i\in[1,N]}\left[E_{Bi}\cdot\begin{bmatrix}\Delta z_{i_1}\\ \Delta z_{i_2}\\ \Delta z_{i_3}\end{bmatrix}-\left(\frac{1}{p_i}\frac{\|\hat{n}_i^{(k)}\|}{\alpha}B_i-\hat{n}_i^{(k)}\cdot l_B\right)\right]^2+ \qquad (7)$$

$$1/\beta\sum\nolimits_{(u,v)\in M}[\Delta z_u-\Delta z_v]^2$$

where

$$E_{Ai}=\begin{bmatrix}l_{A_x}(y_{i_3}-y_{i_2})+l_{A_y}(x_{i_2}-x_{i_3})\\ l_{A_x}(y_{i_1}-y_{i_3})+l_{A_y}(x_{i_3}-x_{i_1})\\ l_{A_x}(y_{i_2}-y_{i_1})+l_{A_y}(x_{i_1}-x_{i_2})\end{bmatrix},$$

and $l_A=\begin{bmatrix}l_{A_x} & l_{A_y} & l_{A_z}\end{bmatrix}^T$ represents the normalized direction vector of light source A. E_{Bi} and l_B are defined similarly for light source B (i.e., that from the projector). The term $\hat{n}_i^{(k)}=(v_{i_2}^{(k)}-v_{i_1}^{(k)})\times\left(v_{i_3}^{(k)}-v_{i_2}^{(k)}\right)$ is the normal vector computed from the height values of the mesh during iteration k -- they are constant during an iteration.

The first part of equation (7) defines the relief appearance formation objective for all mesh triangles and the second part the smoothing desire. The $1/\beta$ term is a normalizing factor used to balance the relative importance of the image formation equations and the smoothing equations. In practice, we compute β so as to provide equal importance to image formation and to smoothing. Since the number of equations used in image formation is about half of those used for smoothing, we usually set $\beta = 2$.

4 Implementation Details

Our prototype system includes geometric and color calibration. Geometric calibration of the 1400x1050 resolution Optoma DLP projectors and the 10MP Canon Rebel XTi camera is done once. Color calibration is recomputed for each fabricated object [18]. Furthermore, since the resolution of the camera, projector, and relief mesh is not necessarily the same, we use splatting to project relief mesh triangles onto the camera and projector image planes. Given a relief mesh, the fabrication process is automated using our Alaris30 3D printer. After fabrication, we place the object in front of our camera and projectors on a platform that can be mechanically repositioned using knobs. To place the object accurately at the origin of the calibrated camera-projector coordinate frame, the projector illuminates a contour light pattern which is then used to manually align the object with the contour lit by the projectors.

5 Analysis of Intensity Coverage

We have analyzed the theoretical intensity ranges achievable for any given image pair (A, B). Our method can obtain a large range of intensity differences between A and B images, in fact more than the plain-reliefs of Alexa and Matusik's [1] (Figure 5). In particular, our method supports all lower intensity values for image B.

For the analysis, we focus on measuring the intensity of a plane since our mesh consists of triangles. We assume the simple light source direction to be l_A and the specifically designed digital projector light direction to be l_B. We are looking for a triangle i which has a normal n_i that satisfies the following two equations:

$$n_i \cdot l_A = A_i \quad \text{and} \quad p_i n_i \cdot l_B = B_i \tag{8}$$

where $p_i \in [0, 1]$ is the intensity of the incident projector from l_B, A_i is the desired intensity of the triangle when lit by the directional light from l_A, and B_i is the desired intensity of the triangle when lit by the designed light pattern from l_B. Hence, A_i and B_i is achievable if

$$n_i \cdot l_A = A_i \quad \text{and} \quad n_i \cdot l_B \geq B_i. \tag{9}$$

Geometrically, these equations define two cones shown in Figure 5(a). A_i is achievable when n_i falls exactly on the surface of a cone defined by the first equation. B_i is achievable when n_i falls inside the cone defined by the second inequality. The second equation is always achievable when $B_i = 0$. As shown in

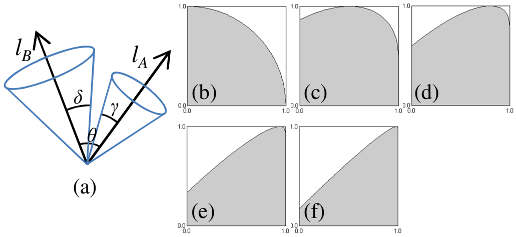


Fig. 5. Intensity Coverage. a) Setup used in b-f. b-f) Supported intensity coverage for image A & B using different light directions. From b to f, $h = \{0,1,2,4,8\}$, $l_A = (1+h^2)^{-0.5}\,[1\ 0\ h]$ and $l_B = (1+h^2\,)^{-0.5}\,[0\ 1\ h]$. x-axis represents image A intensity when lit by a simple light from direction l_A and y-axis represents image B intensity when lit by a specifically projector light pattern from direction l_B. Axes x and y $\in [0,1]$; gray pixels indicate the possible intensity pairs.

Figure 5(a), the light directions l_A and l_B define the centerlines of the cones. Let θ be the angle between l_A and l_B. Intensity values A_i and B_i define the angles γ and δ, the angles between the cone surfaces to cone centerlines l_A and l_B. n_i has a solution as long as any part of the cone centered around l_A falls inside the other cone; i.e., when $\delta \geq \theta - \gamma$. Hence, given light directions l_A and l_B and a particular value for γ (or A_i), there is a range of δ (or B_i) which produces at least one solution for n_i.

We show in Figures 5b-f the pairs of values for A_i and B_i that are possible for several light directions. In each of figures 5b-f, the x-axis is a value for $A_i \in [0,1]$ and y-axis is a value for $B_i \in [0,1]$. The angle θ between light directions varies from 90 to 0 degrees from (b) to (f). For a specific A_i value a, we draw a vertical gray segment along line $x = a$ to show what range of B_i values are achievable as per equation 9; in other words, a point (a, b) falling in the gray area implies that the intensity pair $A_i = a$ and $B_i = b$ is achievable. As observed, our method supports a larger set of intensity than previous work (i.e., [1]) since we cover all lower intensities in image B due to the simultaneous optimization of surface shape and projector light.

6 Results and Discussion

We have used our approach to design several relief surfaces both in simulation and in real-life. We used tessellated meshes of resolution 100x80 cells which require a compute time of about 30 minutes (about half of that time is spent in actual optimization computations and the rest in file I/O). Our typical 3D print time is 5-10 hours for 10x8 centimeter reliefs.

In Figure 1, we show photographs of an example hidden relief mesh and a plain light relief mesh produced by [1]. The latter relief mesh uses only simple lighting and is designed to yield image A when illuminated from one direction and image B when illuminated from a different direction. We use the same light directions for both reliefs. As seen, the relief mesh of [1] is not able to produce both appearances – this is mostly because of the significant intensity differences between images A and B (see Section 5). In contrast, our approach can produce both appearances quite well. The visualization inset on the right shows a color-coded image of the errors of both reliefs. Note that even though we take image B into account when computing the hidden-appearance relief, image B is not perceivable in the relief under normal illumination.

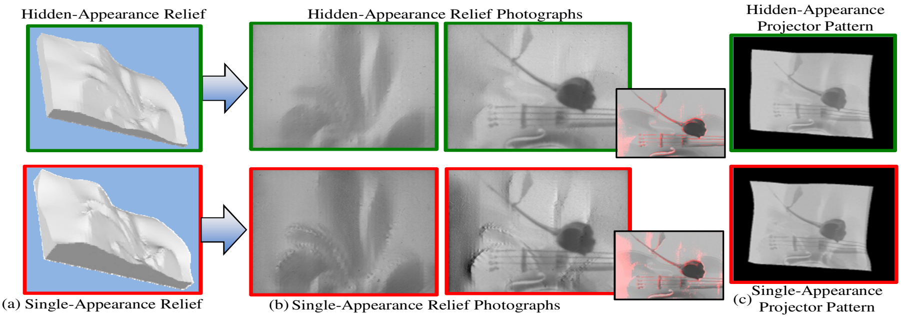

Hidden-Appearance Relief
Hidden-Appearance Relief Photographs
Hidden-Appearance Projector Pattern
(a) Single-Appearance Relief
(b) Single-Appearance Relief Photographs
(c) Single-Appearance Projector Pattern

Fig. 6. Hidden- vs Single-Appearance Reliefs. a) We show our hidden-appearance relief geometry (top) and a single-appearance relief geometry (bottom) which produces appearances of image A and B as shown in Figure 1. b) Photographs of our hidden-appearance relief under direct illumination yielding image A (top middle) and under projector light yielding image B (top right). A single-appearance relief is able to produce image A (bottom middle) but the second appearance cannot necessarily be produced even with the help of a digital projector (bottom right). The small insets are a visualization of the per pixel intensity error between the photographs and image B (red = large error). c) The projector patterns that shine on the two reliefs when producing image B.

In Figure 6, we compare our hidden-appearance relief to a single-appearance relief using the same image content as in Figure 1. The geometry of a single-appearance relief is computed for only one appearance (image A). Then, we compute the projector pattern that best achieves the second appearance (image B). Our approach is able to faithfully recreate both appearances despite both relief types using projectors. As seen in Figures 6c, the projector patterns for both relief types are similar. This means the need for the projector pattern is roughly equal in both cases. While one naïve option to produce appearance B would be to shift all the content to the projector pattern, it would require simplifying the relief geometry to nearly a plane. This would violate the desire to have appearance A be produced by a simple directional light. Instead, our optimization process finds a geometry able to produce image B, with the aid of a projector, while leaving the appearance of image A intact.

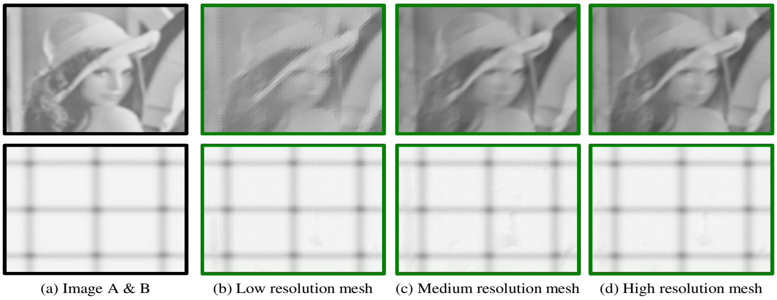


Fig. 7. Hidden-Appearance Reliefs with Different Mesh Resolutions. We experiment with altering mesh resolution. a) Shows image A (top) and image B (bottom). Appearances resulting from using b) low resolution mesh with 50x40 cells, c) medium resolution mesh with 100x80 cells, and d) high resolution mesh with 150x120 cells.

Figure 7 contains several instances of a hidden-appearance relief each created in simulation with a different mesh resolution. The lowest resolution mesh (50x40 cells, 10 minutes total compute time) shows noticeable visual artifacts and blurriness as compared to the highest resolution mesh (150x120 cells, 50 minutes). We found the mesh resolution of 100x80 cells (and 30 minutes total compute time) to be a reasonable balance of visual quality and computation time.

In order to better understand what type of images we can hide, we experimented in simulation with the effect of varying contrast levels and sharpness in image A and/or B. When A has a small contrast, the resulting relief only needs low frequency height changes and thus tends to be flat. It is easy for the projector to shine the patterns needed to produce image B. In short, low contrast in A makes the hidden-appearance relief problem easy. A similar effect occurs with a low contrast B image as well. When B has low contrast, even though the relief is not optimized for the image, the projector can do a lot to compensate for the undulations of the relief surface. In Figure 8 we show that when both A and B have low contrast, both hidden-appearance relief and single-appearance relief surfaces do a good job of producing an image B -- the problem itself is fairly easy. However, when A has high contrast, the relief surface needs significant height changes to produce image A under normal illumination. Hence, it is easy to unwillingly obtain a surface for which it is hard for the projector to alter the appearance to produce image B -- even self-shadows occur more readily. Moreover, if B has high contrast, it makes the inequality equation (2) even harder to achieve. As long as the inequality is not satisfied then B is not achievable given limited projector power. In Figure 8, we show the results of single-appearance reliefs and hidden-appearance reliefs with high contrast A and B images. As seen, the single-appearance relief must do a significant effort to achieve A, which generates a bumpy surface geometry and easily breaks the generation of image B. However, our

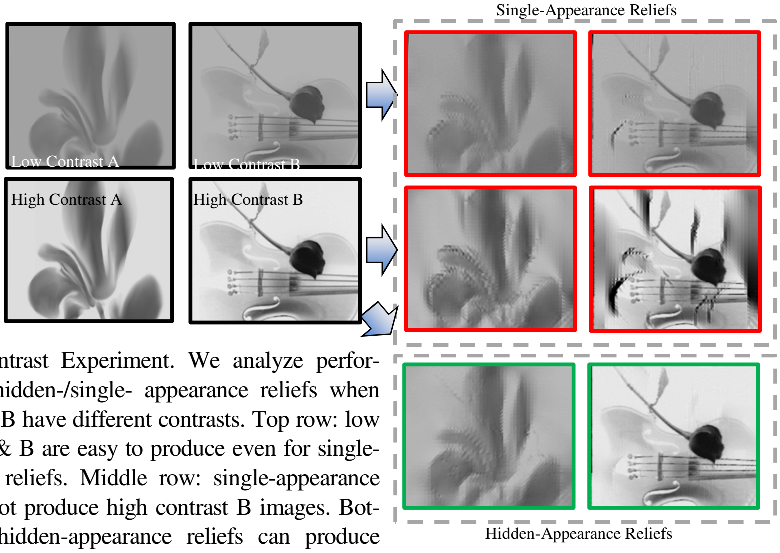


Fig. 8. Contrast Experiment. We analyze performance of hidden-/single- appearance reliefs when image A & B have different contrasts. Top row: low contrast A & B are easy to produce even for single-appearance reliefs. Middle row: single-appearance reliefs cannot produce high contrast B images. Bottom row: hidden-appearance reliefs can produce high contrast A and B.

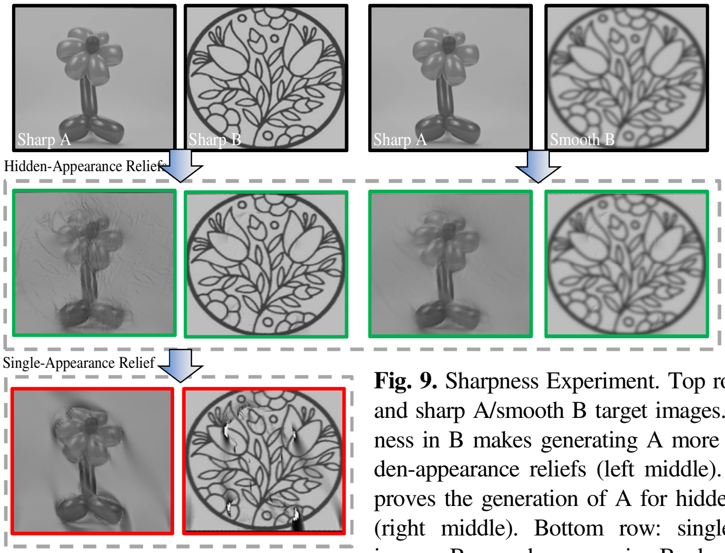


Fig. 9. Sharpness Experiment. Top row: sharp A/sharp B and sharp A/smooth B target images. Middle row: sharpness in B makes generating A more challenging for hidden-appearance reliefs (left middle). A smoothed B improves the generation of A for hidden-appearance reliefs (right middle). Bottom row: single-appearance reliefs ignore B so sharpness in B does not alter single-appearance reliefs (but B is not always possible).

hidden-appearance reliefs produce image B while maintaining the appearance of image A under normal illumination (and also not showing image B). Hence, although any second appearance can be encoded, using images A and B of relatively high contrast increases the benefit of using our method to hide the second appearance – a naively generated surface for the same images would not reproduce image B as well.

In Figure 9, we analyzed in simulation the effect of sharpness in the input image A and/or B. When A is sharp, the height values needs to change a lot to achieve the sharp image. This makes it difficult for both single- and hidden-appearance reliefs. Nevertheless, sharpness in B does not have any effect on single-appearance reliefs because it does not consider B at all. Generation of A in single-appearance relief is not affected by sharpness in B (bottom row of Figure 9). However, sharpness in B causes noise in hidden-appearance relief geometries. This effect results from the simplification we made about correspondences: we assume that the correspondences between mesh vertices and camera pixels do not change when the geometry change is small. However, when B is sharp, small changes in correspondences may cause large changes in corresponded intensity. It could be that one relief triangle is asked to have a white appearance in one iteration and a black appearance in the next iteration. This causes the noisy artifacts in left of middle row in Figure 9. Hence, to obtain a geometry that does not show remnants of image B under normal illumination, better results are achieved with a smoothed B, (right of middle row in Figure 9).

Figure 10 shows several real world experiments and photographs. We show hidden-appearance reliefs and single-appearance reliefs for various A and B image pairs. For each, we compute the hidden- and single-appearance relief, fabricate them, color calibrate them, compute the color calibrated pattern, and capture photographs of the physical object. Figures 10b-d show photographs of hidden-appearance reliefs producing images A and B better than single-appearance reliefs. In particular, Figure 10c shows the

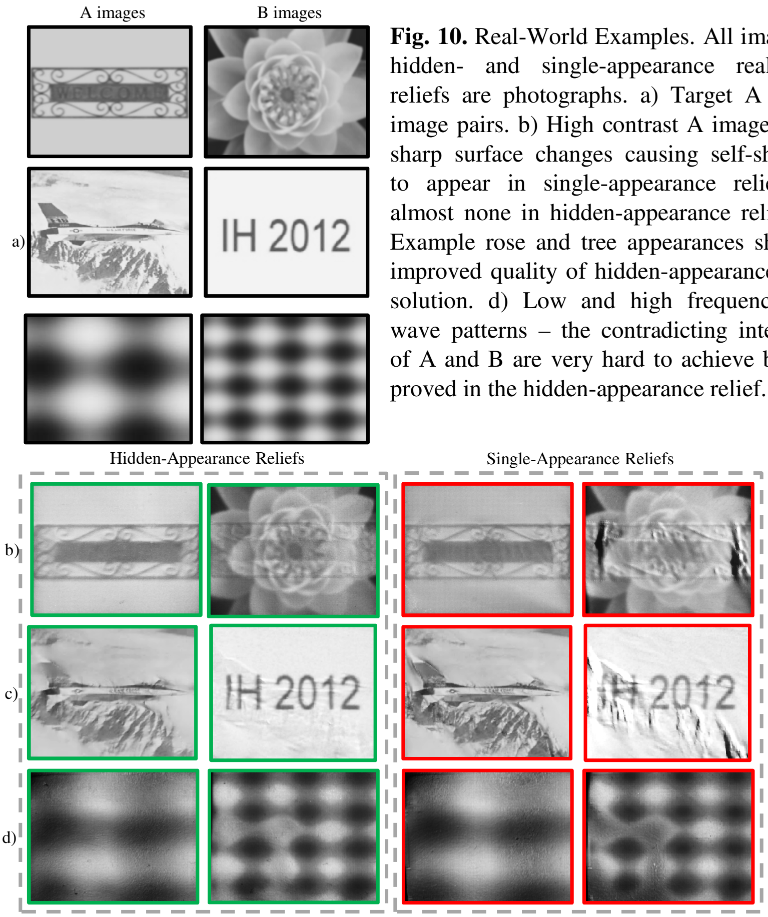


Fig. 10. Real-World Examples. All images of hidden- and single-appearance real-world reliefs are photographs. a) Target A and B image pairs. b) High contrast A image yields sharp surface changes causing self-shadows to appear in single-appearance relief and almost none in hidden-appearance reliefs. c) Example rose and tree appearances showing improved quality of hidden-appearance relief solution. d) Low and high frequency sine wave patterns – the contradicting intensities of A and B are very hard to achieve but improved in the hidden-appearance relief.

challenging case of sharp images A and B (see discussion for Figure 9) and Figure 10d shows a particularly hard example where A and B systematically contradict each other in their visual objectives. Nevertheless, our method shows notable improvement.

7 Conclusion and Future Work

We present hidden-appearance reliefs which enable a chosen appearance A to be observed under direct lighting while also enabling a hidden arbitrary appearance B which is only visible under a particular lighting setup. By doing this, we effectively hide a second piece of information into one single relief. We present a computational method which designs hidden-appearance reliefs and a full implementation. Our experiments show that our method is robust for various A and B image pairs both in simulation and in real-life.

Our future works includes the following. 1) We would like to extend our method to include colored images. 2) Although the second appearance is not recognizable under normal illumination in our current approach, we do not explicitly guarantee, or maximize, that it is unrecognizable when observed under normal illumination or when illuminated by a pattern other than the indicated one. Thus, we seek an extension that models the perceptibility of the second appearance and purposefully attempts to keep it small. 3) Another interesting extension is to incorporate multiple B appearances produced by different projector pattern illuminations and to quantify the "amount of information" that can be hidden. 4) We would also like to explicitly consider self-occlusion, self-shadowing, and inter-reflection within relief computation.

References

1. Alexa, M., Matusik, W.: Reliefs as Images. ACM Trans. on Graphics 29(4) (2010)
2. Wang, K., Lavoué, G., Denis, F., Baskurt, A.: Three-Dimensional Meshes Watermarking: Review and Attack-Centric Investigation. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 50–64. Springer, Heidelberg (2008)
3. Aliaga, D., Atallah, M.: Genuinity Signatures: Designing Signatures for Verifying 3D Object Genuinity. Computer Graphics Forum (Eurographics) 28(2), 437–446 (2009)
4. Volpe, H.R.: Printing method and copy-evident secure document. US Patent 5487567 (January 1996)
5. Weyrich, T., Peers, P., Matusik, W., Rusinkiewicz, S.: Fabricating Microgeometry for Custom Surface Reflectance. ACM Transactions on Graphics 28(3) (2009)
6. Weyrich, T., Deng, J., Barnes, C., Rusinkiewicz, S., Finkelstein, A.: Digital Bas-Relief from 3D Scenes. ACM Trans. on Graphics 26(3) (2007)
7. Chen, H., Belhumeur, P., Jacobs, D.: In search of illumination invariants. In: IEEE CVPR, pp. 254–261 (2000)
8. Ryu, S.-J., Lee, M.-J., Lee, H.-K.: Detection of Copy-Rotate-Move Forgery Using Zernike Moments. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 51–65. Springer, Heidelberg (2010)
9. Schwamberger, V., Le, P.H.D., Schölkopf, B., Franz, M.O.: The Influence of the Image Basis on Modeling and Steganalysis Performance. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 133–144. Springer, Heidelberg (2010)
10. Arnold, M., Chen, X.-M., Baum, P.G., Doërr, G.: Improving Tonality Measures for Audio Watermarking. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 223–237. Springer, Heidelberg (2011)
11. Kim, K., Barni, M., Tan, H.Z.: Roughness-Adaptive 3D Watermarking of Polygonal Meshes. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 191–205. Springer, Heidelberg (2009)
12. Cao, Y., Zhao, X., Feng, D., Sheng, R.: Video Steganography with Perturbed Motion Estimation. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 193–207. Springer, Heidelberg (2011)
13. Kerckhoffs, A.: La Cryptographie Militaire (Military Cryptography). le Journal Des Sciences Militaires (Journal of Military Science) IX, 5–38 (1883)
14. Oliva, A., Torralba, A., Schyns, P.: Hybrid Images. ACM Trans. on Graphics 25(3) (2006)

15. Horn, B., Szeliski, R., Yuille, A.: Impossible shaded images. IEEE Transactions on PAMI 15(2), 166–170 (1993)
16. Bimber O., Iwai D., Wetzstein G., Grundhofer A.: The Visual Computing of Projector-Camera Systems. In Eurographics. STAR, pp. 23–46 (2007); also CGF 27(8), 2219–2245 (2008)
17. Grossberg, M., Peri, H., Nayar, S., Belhumeur, P.: Making One Object Look Like Another: Controlling Appearance using a Projector-Camera System. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 452–459 (2004)

Blind Robust Watermarking Mechanism Based on Maxima Curvature of 3D Motion Data

Ling Du[1,2], Xiaochun Cao[1,⋆], Muhua Zhang[1], and Huazhu Fu[1]

[1] School of Computer Science and Technology,
Tianjin University, Tianjin, 300072, China
[2] School of Computer, Shenyang Aerospace University, Shenyang, 110136, China
{duling,xcao}@tju.edu.cn

Abstract. This paper presents a blind robust watermarking mechanism for copyright protection of 3D motion data. The mechanism segments motion data based on stable anchor-points captured by the maxima in spatio-temporal curvature and filtered by posterior attack model. For each segment, we make a randomized cluster division of 3D points based on a secret key. A watermark is then embedded within these clusters by Triangle Orthocenter based encoding approach. Experimental results show that the proposed watermarking scheme is robust against many possible attacks such as uniform affine transforms (scaling, rotation and translation), noise addition, reordering and cropping.

1 Introduction

With the rapid progress in motion capture (mocap) technology, 3D motion data are being widely used in animations, video games, movies, human motion analysis and other fields. 3D motion data has high scientific and commercial value, which makes its copyright protection becoming an important issue. Digital watermark technology [6] provides an effective method for digital copyright protection. So far, digital watermark techniques mainly consist of image watermarking [7][9][11][23], video watermarking [17][20][24], audio watermarking [3][13][21], mesh watermarking [4][5][14][15][19] etc. Although different watermarking methods have been developed for other kinds of media, they cannot be directly applied to 3D data. The most important reasons are dimensionality. Since other kinds of media are not generalized to handle problems related to higher dimensional data, developing watermarking methods for 3D motion data is more challenging.

3D human motion data consist of motion information related to human joints, and can be represented by a set of trajectories. Recently, for 3D motion trajectory data watermarking, Kim et al [10] present a algorithm based on multiresolution representation and spread spectrum. The algorithm not only can resist against random noises, but also has merits of spread spectrum such as the resilience to common signal processing as well as the robustness to time warping. In [22], original data is firstly transformed into the frequency domain by discrete cosine transformation, and then chose the most significant components to insert the

[⋆] Corresponding author.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 110–124, 2013.

watermark according to the amplitudes of the signal. Both methods in [10] and [22] implement the watermarking of motion data based on the spread spectrum method. However, they use principal component analysis (PCA) approach for segmentation, with only a rough semantic segmentation capability. Moreover, the time duration of motion is limited in their methods, and it only robust against similarity transformation attack, not robust against affine transformation.

Considering the download and transmission of motion data, Li and Okuda [12] propose a method based the progressive representation including the base frames and enhancement ones. The motion is sent frame by frame from the base frames to the enhancements until all the frames are restored. The progressive encoding method gives the frames of the motion an order which ranks frames by their importance. With such an order, watermark can be embedded into a characteristic of each frame. However, in terms of robustness, this method only tolerates random noises to some extent. In addition, the extraction algorithm needs information about the original frame, so it belongs to the non-blind watermarking algorithm as [10] and [22]. Motwani et al [16] propose a fragile watermarking algorithm for 3D motion curves. Their approach implements a prototype in spread spectrum domain by using a Haar wavelet transform on the 3D data and alters the wavelet coefficients. However, the algorithm also belong to non-blind watermarking algorithm.

In order to enhance the capacity of watermark robustness to affine transformation attacks, Agarwal and Prabhakaran [1][2] provide blind robust watermark algorithms of 3D motion data. They segment motion trajectory and identify clusters of 3D points per segment. The watermark can be embedded and extracted within these clusters by the proposed extension of 3D quantization index modulation. The watermarking schemes are robust against many types of attacks and works well. However, the motion segmentation method and ordering criteria of encoding points need more investigation to identify robustness against noise addition attacks. And for its Euclidian Distance based encoding method, the shifting direction of encoding points do not take the movement direction of joint into account. It may decrease the imperceptibility of the watermark scheme.

This paper presents a blind robust watermarking mechanism for the trajectory of human joints based on maxima curvature of 3D motion data. The technical contributions are identified as follows.

-Robust segmentation method based on maxima curvature and posterior attack model. Compared to current segmentation method, it is more robust against possible attacks due to the stability of anchor-points used for segmentation.

-Blind robust watermark encoding method. For bit encoding process inside the cluster, we propose a novel Triangle Orthocenter based encoding method. Compared to current encoding approach, it has better imperceptibility and robustness to noise addition attack.

The remainder of the paper is organized as follows. In section 2, we describe the scheme design for our watermarking method. In section 3, some typical motions with different parameter settings are employed to demonstrate the advantage of our approach with respect to other methods. Finally, we draw a conclusion for our work in section 4.

2 Scheme Design

In our work, the motion capture data is represented by a set of trajectories consisted of $M(1 \leq M \leq 19)$ joint trajectories of human body. The watermark is represented as multiple bits (series 0s and 1s). Fig.1 shows the watermark scheme proposed. Firstly, we extract the spatio-temporal curvature extremes for each trajectory as candidate anchor points, and then filter the stable anchor-points by posterior attack model. Secondly, motion trajectories are segmented based on the remaining stable anchor points. Randomized cluster division is done based on a secret key for each segment. Finally, the watermark is embedded within these clusters by Triangle Orthocenter based encoding approach. The watermark is extracted accordingly as shown in the right side of Fig.1.

2.1 Candidate Anchor Points

In order to ensure the embedded watermark can be extracted when a part of the marked motion trajectory encountering attacks, the original 3D motion trajectory is temporally divided into several parts before watermark embedding. Since segmentation can help pinpoint the presence of the watermark during the extraction, the point used for segmentation must be robust against motion editing operations such as noise additions and 3D transforms.

Motion trajectory of 3D motion captured data can be expressed by the position vector composed of positions for each time instants, as

$$\mathbf{r}(t) = [x(t), y(t), z(t)]^T, \tag{1}$$

where $x(t)$, $y(t)$, $z(t)$ represent the 3D coordinates of the joint at the time t. The quantitative measure of motion can be acquired by its velocity $\mathbf{v}(t)$, and acceleration $\mathbf{a}(t)$, which are given by the first and second derivatives of position.

$$\mathbf{v}(t) = \mathbf{r}^{'}(t) = [x^{'}(t), y^{'}(t), z^{'}(t)]^T, \tag{2}$$

$$\mathbf{a}(t) = \mathbf{r}^{''}(t) = [x^{''}(t), y^{''}(t), z^{''}(t)]^T. \tag{3}$$

Human observers are able to perceive dynamic instants that stem from discontinuities in velocity or acceleration and can be captured by the maxima of spatio-temporal curvature [18]. The curvature $k(t)$ at time t is given by

$$k(t) = \frac{\|\mathbf{v}(t) \times \mathbf{a}(t)\|}{\|\mathbf{v}(t)\|^3}, \tag{4}$$

where $'\times'$ represents the cross product and $\|\mathbf{v}(t)\|$ represents speed. Finally, the maxima in spatio-temporal curvature is captured as

$$\mathbf{P}(t) = \left\{ \mathbf{r}(t) \mid k^{'}(t) = 0 \right\}. \tag{5}$$

Because the maxima in spatio-temporal curvature of a trajectory is invariant to 3D affine transforms, we use these instants as candidate embedding anchor-points during the watermarking. Each dynamic instant represents an important

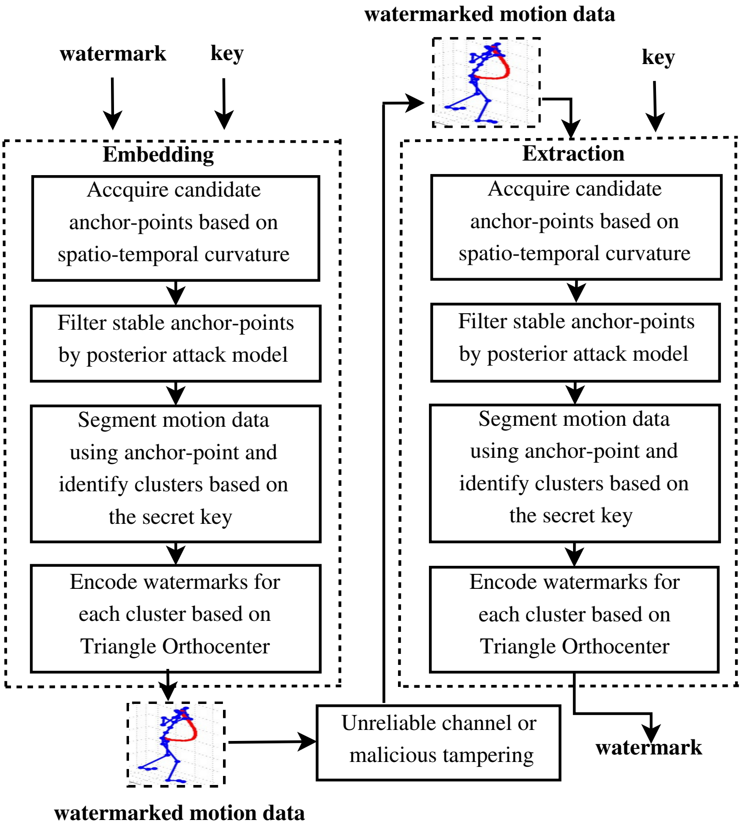

watermarked motion data
watermark
key
key
Embedding
Accquire candidate anchor-points based on spatio-temporal curvature
Filter stable anchor-points by posterior attack model
Segment motion data using anchor-point and identify clusters based on the secret key
Encode watermarks for each cluster based on Triangle Orthocenter
Extraction
Accquire candidate anchor-points based on spatio-temporal curvature
Filter stable anchor-points by posterior attack model
Segment motion data using anchor-point and identify clusters based on the secret key
Encode watermarks for each cluster based on Triangle Orthocenter
Unreliable channel or malicious tampering
watermark
watermarked motion data

Fig. 1. Watermark scheme design for 3D motion data

change in the motion characteristics. But for robust watermarking of 3D motion data, not all of these dynamic instants used for segmentation are robust against possible attacks such as noise addition.

2.2 Stable Anchor Points

In order to filter out those anchor points sensitive to noise, smoothing and other attacks, we introduce posterior attack model to obtain the stable anchor-points. Virtual attacks are done by adding Gaussian noise, smoothing and other attacks:

$$\widetilde{\mathbf{P}}_i(t) = f(\mathbf{P}(t), \mathbf{G}(\delta)), \tag{6}$$

where $\mathbf{P}(t)$ represents the candidate anchor-point captured by curvature maxima, $\mathbf{G}(\delta)$ represents combination of several attacks, $\widetilde{\mathbf{P}}_i(t)$ represents the candidate anchor-points after the i-th attack. We apply N times virtual attacks to candidate anchor-points with random attack parameters and combinations. The set of candidate anchor-points after virtual attacks is

$$\mathrm{P} = \left\{ \widetilde{\mathbf{P}}_i(t) \mid i = 1...N \right\}. \tag{7}$$

In our implementation, we conduct four groups of attacks with different intensities and ratio including Gaussian noise 10dB and 1dB, smoothing, on various ratios (1% to 50%) of attacked anchor points. By comparing the position of original candidate anchor-point $\mathbf{P}(t)$ to the candidate anchor-point $\mathbf{P}_i(t)$, we compute the posterior probability of attacks as

$$p(\mathbf{P}(t) \mid \mathrm{P}) = \frac{1}{N} \sum_{i=1}^{N} x_i(t), \tag{8}$$

where $x_i(t)$ is given by

$$x_i(t) = \begin{cases} 1, & \text{if } \mathbf{P}(t) = \widetilde{\mathbf{P}}_i(t) \\ 0, & \text{if } \mathbf{P}(t) \neq \widetilde{\mathbf{P}}_i(t) \end{cases}. \tag{9}$$

And then we select the points which have the larger posterior probability than the median of the posterior probability values, *Median*, for each candidate anchor-point as stable embedded anchor-points, which is given by

$$\mathbf{P}_{feature} = \{\mathbf{P}(t) \mid p(\mathbf{P}(t) \mid \mathrm{P}) > Median\}. \tag{10}$$

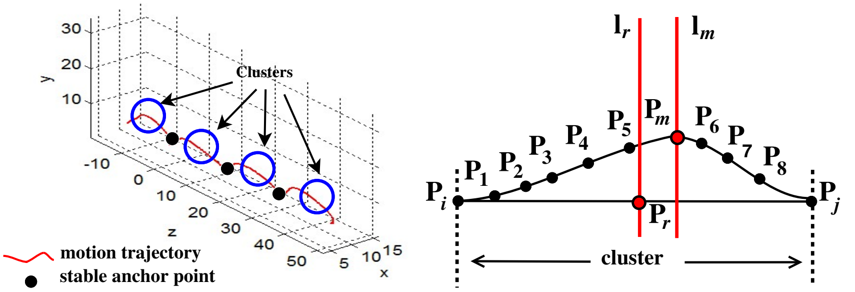
Finally, the motion trajectory data is segmented based on the remaining stable anchor points.

2.3 Watermark Embedding

In order to ensure the best robustness for our watermarking method, random choice for embedding position is an important and pivotal solution. Therefore, we divide each segment into clusters on the basis of a key *sk*. The clusters are chosen in a random fashion separated by a random number of points called embedding distance. Fig.2(a) shows the clusters based on segments of motion trajectory. Then for certain number of watermark copies, we randomly select the clusters to embed for each watermark copy.

For the watermark embedding inside each cluster, a new Triangle Orthocenter based approach is proposed to encode for every cluster in the trajectory. Firstly, we identify invariant point set and encoding point set which are used as reference and watermark bit hiding respectively. The choice of the invariant points is based on a scalar quantity which is invariant against uniform affine transformations, such as maximum Euclidian distance between the set of points. The encoding point set is the difference set between cluster set and invariant point set.

Secondly, we identify the order in which the watermark bits encoded. For a certain cluster, the order should not be dependent on the 3D information of



(a) (b)

Fig. 2. Identification of clusters (a) Clusters based on segments of a joint trajectory in 3D space. (b) Cluster in joint trajectory for dominant direction determination. $\mathbf{P}_i$ and $\mathbf{P}_j$ are the invariant points in the cluster. $\mathbf{P}_r$ is the midpoint of the line $\mathbf{P}_i\mathbf{P}_j$. $\mathbf{P}_m$ is the point within the cluster which is the most distant to the line $\mathbf{P}_i\mathbf{P}_j$.

the points, as it is vulnerable to noise adding attack. In addition, the order also should not be dependent on the occurrence of 3D point in time, as it is vulnerable to reordering attack. Therefore, we identify ordering criteria based on the occurrence order of 3D points and the dominant direction determined by statistics information of the points belonging to the cluster. The determination of dominant direction make our encoding method robust against global reorder attacks, which means completely inverting the point in a continuous manner in the trajectory. As shown in Fig.2(b), the statistics information are introduced in the following.

Point Numbers. The number of points separated by the line $\mathbf{l}_r$. It is described by N_l and N_r, which represents the number of points separated by line $\mathbf{l}_r$ respectively. In the example, $N_l = 5$ and $N_r = 3$.

Average Gradient. The average gradient of encoding points separated by the perpendicular line $\mathbf{l}_m$. It is described by Ag_l and Ag_r, which represents the module of average gradient separated by line $\mathbf{l}_m$ respectively. The *Average Gradient Ag* is described as

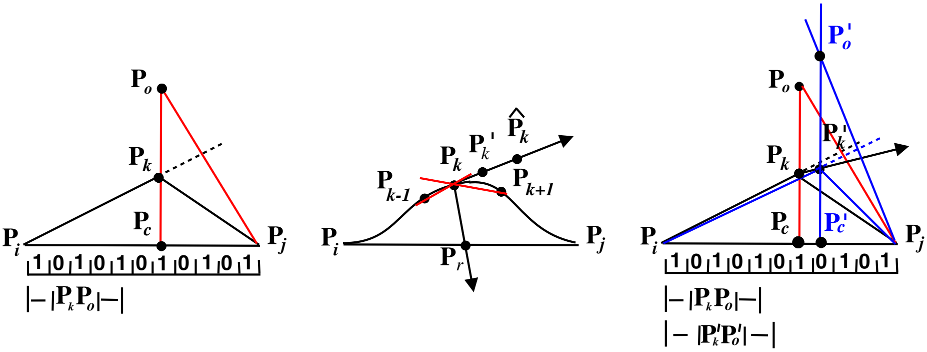
$$Ag = \frac{1}{n}\sum_{i=1}^{n} |gradient(\mathbf{P}_i)|\,, \tag{11}$$

where $gradient(\mathbf{P}_i)$ represents the gradient of point $\mathbf{P}_i$, n is the number of points in the left N_l or right N_r side of line $\mathbf{l}_m$.

Average Height. The average height of encoding points separated by the point $\mathbf{l}_m$. It is described by H_l and H_r, which represents average height separated by line $\mathbf{l}_m$ respectively. The *Average Height H* is described as

$$H = \frac{1}{n}\sum_{i=1}^{n} h_i. \tag{12}$$

where h_i represents the Euclidean distance described by the perpendicular distance from point $\mathbf{P}_i$ to the line joining $\mathbf{P}_i\mathbf{P}_j$.



(a)Watermark bit embedding (b) Shifting direction (c) Embedding after shifting

Fig. 3. Triangle Orthocenter based encoding scheme. $\mathbf{P}_o$ and $\mathbf{P}_o^{'}$ are the orthocenters of $\triangle\mathbf{P}_k\mathbf{P}_i\mathbf{P}_j$ and $\triangle\mathbf{P}_k^{'}\mathbf{P}_i\mathbf{P}_j$. $\mathbf{P}_c$ and $\mathbf{P}_c^{'}$ are corresponding pedals in the edge $\mathbf{P}_i\mathbf{P}_j$. $\mathbf{P}_{k-1}$, $\mathbf{P}_k$ and $\mathbf{P}_{k+1}$ are three continuous encoding points, $\hat{\mathbf{P}}_k$ is the predicted position of $\mathbf{P}_k$, $\mathbf{P}_k^{'}$ is the new shifted positions of $\mathbf{P}_k$ in our proposed.

For identifying dominant direction, because *Point Numbers* is robust against most attacks, we identify dominant direction by comparing the *Point Numbers* firstly. If $|N_r - N_l|$ is greater than a certain threshold, it is identified according to the value of N_r and N_l from high to low or vice versa. For example, if $N_r > N_l$, we choose $\mathbf{P}_i \rightarrow \mathbf{P}_j$ as dominant direction. Otherwise, if $|N_r - N_l|$ is less than a certain threshold, it means the point numbers N_r and N_l are at the same level. For this situation, we use $|Ag_l - Ag_r|$ and $|H_l - H_r|$ for further judgment. Once the dominant direction of points in the cluster has been identified, we hash a sequence of encoding points based on the key sk. Due to this randomness, for n encoding points in one cluster, the search space for the adversary is $n!$.

Finally, we encode the watermarking bit in each encoding point using Triangle Orthocenter based encoding approach. For the encoding progress, as shown in Fig.3(a), the Euclidean distance between $\mathbf{P}_k$ and $\mathbf{P}_o$ is used as the scalar quantity to encode a watermark bit information. Here we divide the line $|\mathbf{P}_i\mathbf{P}_j|$ into equal intervals and labeled as 0s and 1s, then measure the scalar quantity $|\mathbf{P}_k\mathbf{P}_o|$ against this scale and locate the interval, e.g. the 1 bit for the $\mathbf{P}_k$ point. If the bit (0 or 1) represented by the encoding point $\mathbf{P}_k$ is the same with the corresponding watermark bit, we move to the next encoding point. Otherwise, $\mathbf{P}_k$ is shifted along the motion direction predicted based upon DR (Dead Reckoning) technique [8], which can compute the predicted position of a joint based on the recent position, velocity and acceleration. For the encoding point $\mathbf{P}_k$, the corresponding predicted position $\hat{\mathbf{P}}_k$ at time t can be calculated as

$$\hat{\mathbf{P}}_k(t) = \mathbf{P}_{k-1}(t-1) + \mathbf{v}_{k-1}(t-1)\tau + 0.5\mathbf{a}_{k-1}(t-1)\tau^2, \qquad (13)$$

where $\mathbf{P}_{k-1}(t-1)$, $\mathbf{v}_{k-1}(t-1)$ and $\mathbf{a}_{k-1}(t-1)$ are the position, velocity and acceleration at the time instant $t-1$ for the last frame. In our example, we make a shift of $\mathbf{P}_k$ towards line $\mathbf{P}_k\hat{\mathbf{P}}_k$ to $\mathbf{P}_k^{'}$, as shown in Fig.3(b). Finally, it can be observed that since the encoding point $\mathbf{P}_k$ is displaced by $\mathbf{P}_k^{'}$ along the line $\mathbf{P}_k\hat{\mathbf{P}}_k$, the bit represented by it is changed from 1 to 0, as shown in Fig.3 (c). Because 3D trajectory is a curvilinear motion, and the $\triangle\mathbf{P}_k^{'}\mathbf{P}_i\mathbf{P}_j$ is obtuse triangle in most case, the Triangle Orthocenter based encoding approach with shifting towards predicted direction $\mathbf{P}_k\hat{\mathbf{P}}_k$ has better imperceptibility than the Euclidean Distance based encoding, which make the shift along line $\mathbf{P}_k\mathbf{P}_r$.

2.4 Watermark Extraction

Given the watermarked 3D motion trajectory data, we can extract a possible existing watermark by firstly identifying the segments using the maxima in spatio-temporal curvature and posterior attack model. Secondly, for each segment, we identify the cluster based on the secret key and extract all the embedded watermark bits in an inverse procedure of watermark embedding. Finally, we compare the detected watermark bit for all watermark copies K $(K \geq 3)$. Due to the randomness and unpredictability for the attack event, we vote for the majority bit value as the final detected data.

3 Experimental Results

In this section, we discuss experimental scheme and performance analysis for our proposed watermarking scheme from three aspects including imperceptibility, hiding capacity and robustness. For each aspect, we analyze the performance for different parameters settings. The watermarking scheme is implemented in MATLAB, and the source data are from CMU-MOCAP database. For the experiment analysis, the frames for motion data are from 448 to 1050 with frame rate being set as 120 fps. We set the watermark length as 32 bits with copy number of 3 for voting, and they are embedded into one cluster for each copy. Moreover, we conduct a comparison experiment between our method and Euclidean Distance based method for performance analysis.

3.1 Performance Analysis Metrics

In this section, we illustrate the metrics from the aspects of imperceptibility, hiding capacity and robustness used to evaluate our watermarking scheme.

For imperceptibility analysis, we use the following metrics:

$$DistortionforDistance = \frac{\sum_{i=1}^{m}\sum_{k=1}^{n} Euclidean(\mathbf{P}_{i,k}, \mathbf{P}_{i,k}^{'})}{m*n}, \tag{14}$$

where $DistortionforDistance$ represents the error for the distance between original position $\mathbf{P}_{i,k}$ and modified position $\mathbf{P}_{i,k}^{'}$ in the k-th frame of i-th joint. m is frame number and n is joint number. Besides the distortion for distance, considering the motion direction of joint, define the angle metric as:

$$AverageShiftAngle = \frac{\sum_{i=1}^{m}\sum_{k=2}^{n-1}(\pi - \angle\mathbf{P}_{i,k-1}\mathbf{P}_{i,k}^{'}\mathbf{P}_{i,k+1})}{m*n}, \tag{15}$$

where $\angle\mathbf{P}_{i,k-1}\mathbf{P}_{i,k}^{'}\mathbf{P}_{i,k+1}$ represents the trend of the movement direction.

Hiding capacity is measured by embedding rate, which is the ratio of the number of embedding bits to the total number of sampling points, as shown by:

$$ER = \frac{\sum_{i=1}^{s} Bits_i}{m} = \frac{\sum_{i=1}^{s} \lfloor \frac{S_i}{CSize} \rfloor (CSize - 2)}{m}, \tag{16}$$

where s is segment numbers, $Bits_i$ is the bit number that can be embedded in *i-th* segment, S_i is the length of *i-th* segment, and $CSize$ is cluster size.

The robustness can be evaluated by the following metrics:

Watermark Detection Rate (WDR). This is measured as the ratio of the number of watermark bits correctly detected to the original number of watermark bits encoded, which is defined as:

$$WDR = \frac{\sum_{i=1}^{wsize} \neg(w(i) \oplus w^{'}(i))}{wsize}, \tag{17}$$

where $w(i)$ and $w^{'}(i)$ represent *i-th* bit of original and detected watermark respectively, $wsize$ represents watermark length. Due to proper watermark replication technology, $w^{'}(i)$ can be acquired by voting from exacted bit copies.

Bit Error Rate (BER). This is measured as the ratio of the number of error bits extracted to the total number of watermark bits encoded in the given data set, which is defined as:

$$BER = \frac{\sum_{i=1}^{r*wsize} errorBits}{r * wsize}, \tag{18}$$

where $errorBits$ represents the number of error bits extracted, r represents the number of watermark copies, $wsize$ represents the length of watermark in bits.

3.2 Imperceptibility Analysis

As the distortion induced by encoding in our method is dependent on the scale (number of intervals), the imperceptibility analysis with varying number of intervals is shown in Fig.4. For both the distance and angle distortions, as we increase the number of intervals, the distortion is reduced. For the same scale, as shown in Fig.4(a), since we make a shifting of encoding points towards the predicted direction during our proposed encoding method, the $DistortionforDistance$ of Triangle Orthocenter based encoding (represented by TO) is consistently smaller than that of encoding based on Euclidean Distance (represented by ED). And for angle distortion, the difference of $AverageShiftAngle$ between our proposed and the original data without watermarking is also smaller, which is shown in Fig.4(b). Moreover, more intervals results in sensitivity to noises, and this will be discussed in section 3.4.

3.3 Hiding Capacity Analysis

Just as the cluster-based encoding scheme such as Euclidian Distance based strategy, we can encode a maximum of t-2 bits for a given cluster of size t.

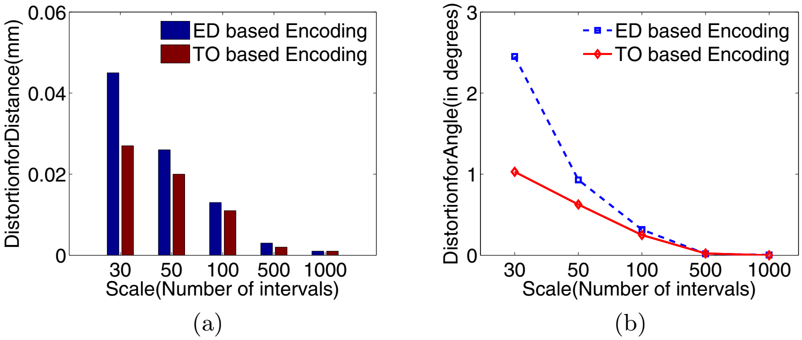


Fig. 4. Imperceptibility analysis for varying number of intervals("dance")

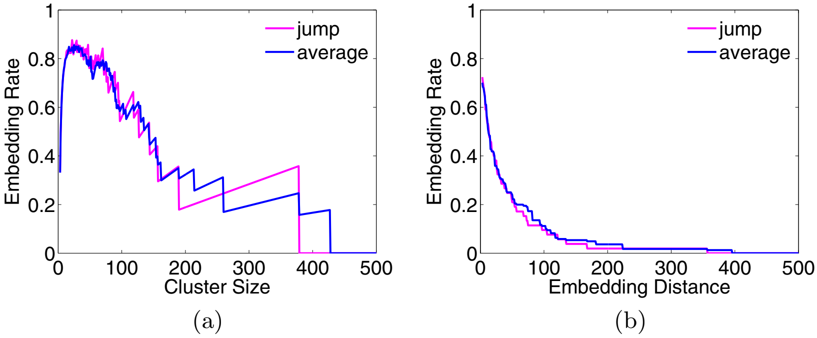


Fig. 5. Embedding rate analysis with respect to cluster size (a) or embedding distance (b)("average" means the value for different motion data)

As the hiding capacity is dependent on the cluster size and embedding distance, we test the influence of the two parameters on embedding rate. As shown in Fig.5(a), in the beginning, the gains from increased number of embedded bits per cluster dominate. However, this dominance fades quickly when the cluster is sizable. The zigzag effects are due to the integer and floor operation as in Eq.(16). Finally, when the cluster size is larger than any segment, the embedding rate is zero. With the optimal cluster size which maximizes the embedding rate, we test the performance of embedding rate with respect to embedding distance. As shown in Fig.5(b), embedding rate reduces as the distance increases, although not smoothly.

3.4 Robustness Analysis

In this section, the robustness against possible attacks including uniform affine transforms, cropping, noises addition and reordering of our proposed watermarking scheme is analyzed. And the noise attacks are done by adding four groups of attack including white Gaussian noise (10db to 1db) and the smooth attacks

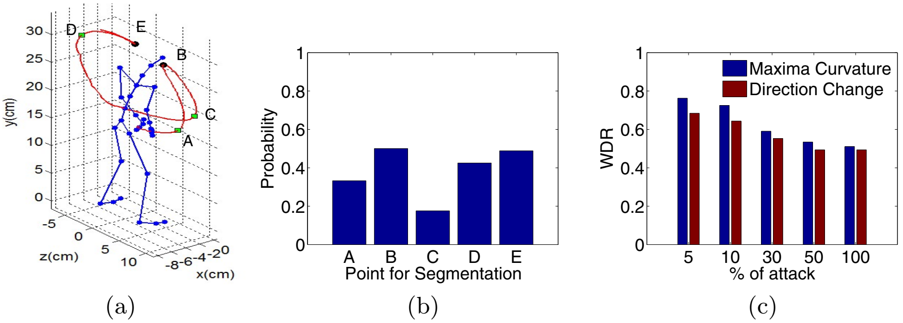


Fig. 6. Robustness analysis for segmentation

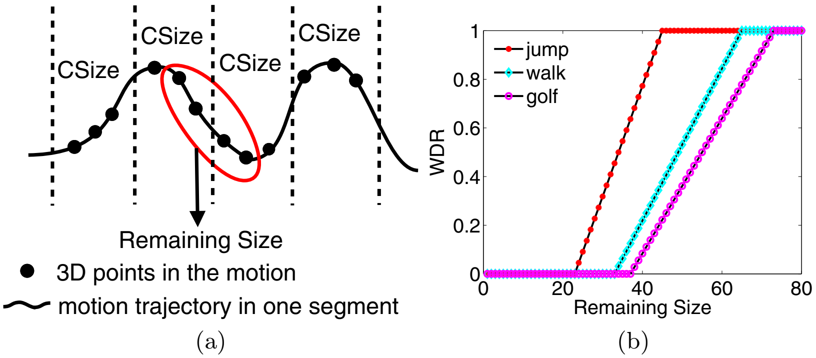


Fig. 7. Robustness analysis for cropping attacks (CSize for "jump"=22, CSize for "walk"=32, CSize for "golf"=36)

(represented by *smooth*1 and *smooth*2) on various ratios (5% to 100%). Here *smooth*1 and *smooth*2 are defined in [2]. The analysis for noise addition and reordering attacks take the golf motion (Scale=50) for example. We get the mean value for the four groups of attacks as the final result.

Segmentation Robustness. In order to identify robust segments, we make a segmentation of motion data with stable anchor-points captured by the maxima in spatio-temporal curvature and posterior attack model. While in [2], it records segments by identifying the change in angle direction of joint, which is shown in Fig.6(a). In order to measure its robustness, we conduct posterior attack and compute the probability for each point. As shown in Fig.6(b), the probabilities of points A, C and D are comparatively lower, and they may result in sensitivity to noises. However, in our segmentation method, the unstable points have been filtered by posterior attack model. The robustness analysis against noise attack with our "Maxima Curvature based segmentation" and "Direction Change based segmentation" is shown in Fig.6(c).

Uniform Affine Transforms. Since stable anchor point, and thus the segments and clusters are invariant to 3D affine transform action. In addition, the Triangle Orthocenter based encoding approach preserves collinearity and ratios of

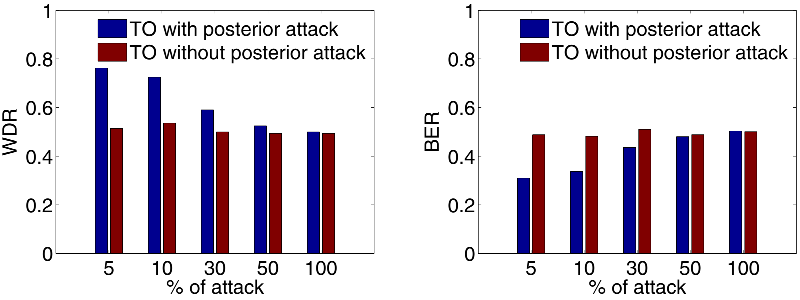


Fig. 8. Robustness analysis against noise addition attacks for Triangle Orthocenter based encoding with and without posterior attack model

distances between the points in clusters. The experiments verify that our scheme is completely robust against uniform affine transformation (translation, scaling and rotation).

Cropping Attacks. Since we embed watermarks into each joint trajectory independently, our scheme is robust against the joint cropping. That is, if one joint of the motion is preserved during the cropping, we can still detect the watermark. When the cropping occur inside one joint trajectory, the robustness against such attack depends on its presence in segments. Intuitively, in order to detect the watermark, the remaining part of a trajectory must be large enough to contain the watermarked points in one cluster. For analysis, assume watermarks are completely embedded in a cluster with the size represented by $CSize$, and the cluster is divided continuously in the segment, as shown in Fig.7(a). Fig.7(b) demonstrates the relation between watermark detection ratio (WDR) and the varying cropping size. Just as the Euclidean Distance based encoding method[2], the minimum remaining size $CropSize$ required to maximize the watermark detection ratio is limited by twice the cluster size $2 * CSize$.

Noise Addition Attacks. For robustness analysis for noise addition attacks, the comparison of our proposed watermarking method with and without posterior attack model is shown in Fig.8. The WDR increases from 0.50752 on average to 0.62064 by introducing the posterior attack model. Accordingly, the BER reduces from 0.49404 on average to 0.41344. Fig.9 illustrates the robustness analysis of our proposed Triangle Orthocenter based encoding (represented by TO) and Euclidean Distance based encoding (represented by ED) for both using the posterior attack model. As is shown, our scheme illustrates better robustness due to position independent ordering criteria of encoding points. The order based on position information in Euclidean Distance based encoding may be disturbed by noise addition. Moreover, the robustness against noise addition attacks can be improved by choosing smaller scales, which is shown in Fig.10. The robustness decreases for both method with larger scale, and our scheme is also better for a range of scales.

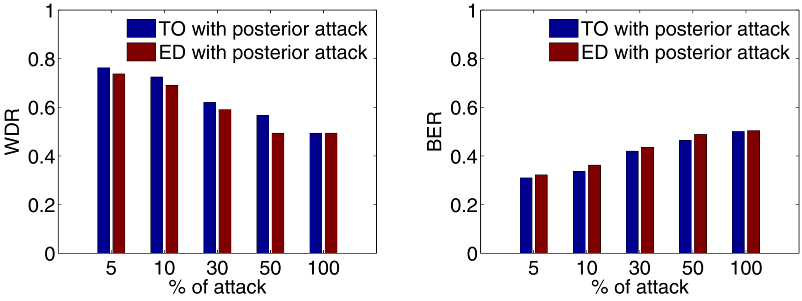


Fig. 9. Robustness analysis against noise addition attacks for Triangle Orthocenter based encoding and Euclidean Distance based encoding

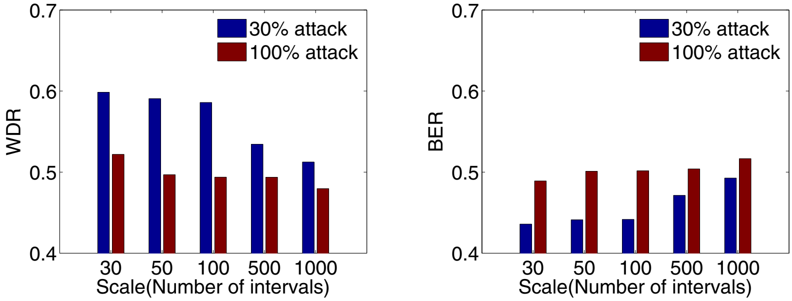


Fig. 10. Robustness analysis for noise addition attacks with varying number of intervals

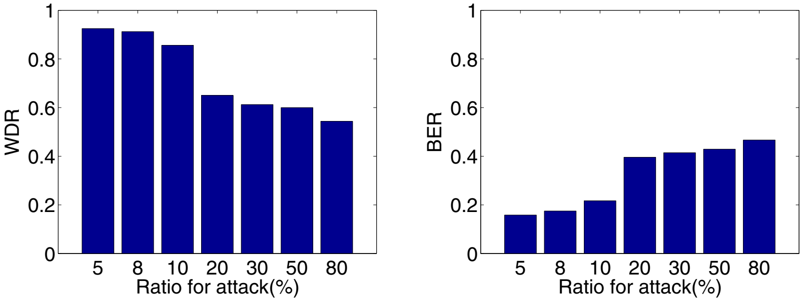


Fig. 11. Robustness analysis against reordering attacks with different attack ratios

Reordering Attacks. The reordering attacks are done either by completely inverting the point in the continuous section of the trajectory, or by randomly picking two neighboring points and swapping them. Because we determine the dominant direction of cluster based on the statistics information of encoding points, and the difference of these statistics information are almost robust against global reordering, so our proposed watermarking scheme is completely robust for global reordering in cluster level. For the random reorder attacks, Fig.11 illustrates robustness analysis for random reordering attacks to our watermarking scheme. Due to watermark replication technology adopted, we can see that with proper number of copies, our scheme is also highly robust against lower ratio partial random reordering attacks for the semantic preserved motion data.

4 Conclusions

This paper has presented a blind robust watermarking mechanism based on maxima curvature of 3D motion data. Analysis proves that the scheme has better imperceptibility by shifting the encoding point towards motion direction predicted based upon its recent position, velocity and acceleration. The hiding capacity has bounds based on cluster size. Moreover, experimental results show that our method is robust against many possible attacks such as uniform affine transforms, reordering and cropping. And particularly, it is more robust than the state of the art against noise addition attacks due to the stability of anchor-points used for segmentation and the position independent ordering criteria of encoding points in our proposed Triangle Orthocenter based encoding method.

Acknowledgments. This work was supported by National Natural Science Foundation of China (No.60905019), Tianjin Key Technologies R&D program (No.11ZCKFGX00800), Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, SKL of CG&CAD, and Strategic Priority Research Program of the Chinese Academy of Sciences(XDA06030601).

References

1. Agarwal, P., Adi, K., Prabhakaran, B.: Robust blind watermarking mechanism for motion data streams. In: Proceedings of the 8th Workshop on Multimedia and Security, pp. 230–235 (2006)
2. Agarwal, P., Prabhakaran, B.: Blind robust watermarking of 3d motion data. ACM Transactions on Multimedia Computing and Communication Applications 6(1), 1–32 (2010)
3. Arnold, M., Chen, X.-M., Baum, P.G., Doërr, G.: Improving Tonality Measures for Audio Watermarking. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 223–237. Springer, Heidelberg (2011)
4. Bors, A.: Watermarking mesh-based representations of 3-d objects using local moments. IEEE Transaction Image Process. 15(3), 687–701 (2006)
5. Cho, J., Prost, R., Jung, H.: An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms. IEEE Transaction Signal Process. 55(1), 142–155 (2007)
6. Cox, I.J., Miller, M.L.: The first 50 years of electronic watemrarking. Joumal of Applied Signal Proeessing 10(2), 126–132 (2002)
7. Dugelay, J.L., Roche, S., Rey, C.: Still-image watermarking robust to local geometric distortions. IEEE Transaction on Image Processing 15(9), 2831–2842 (2006)
8. Duncan, T., Gracanin, D.: Pre-reckoning algorithm for distributed virtual environments. In: Proceedings of the 2003 Winter Simulation Conference, pp. 1086–1093 (2003)

9. Gao, X., Deng, C., Li, X., Tao, D.: Geometric distortion insensitive image watermarking in affine covariant regions. IEEE Systems, Man, and Cybernetics Society 40(3), 278–286 (2010)
10. Kim, T., Lee, J., Shin, S.: Robust motion watermarking based on multiresolution analysis. Computer Graphics Forum 19(3), 189–198 (2000)
11. Lee, H., Kim, H., Lee, H.: Robust image watermarking using local invariant features. Optical Engineering 45(3), 1–11 (2006)
12. Li, S., Okuda, M.: Iterative frame decimation and watermarking for human motion animation. Vision and Image Processing, Special Issue on Watermarking 2010(7), 41–48 (2010)
13. Li, W., Xue, X., Lu, P.: Localized audio watermarking technique robust against time-scale modification. IEEE Transaction on Multimedia 8(1), 60–69 (2006)
14. Luo, M., Bors, A.G.: Principal component analysis of spectral coefficients for mesh watermarking. In: International Conference on Image Processing, pp. 441–444 (2008)
15. Luo, M., Wang, K., Bors, A., Lavoué, G.: Local Patch Blind Spectral Watermarking Method for 3D Graphics. In: Ho, A.T.S., Shi, Y.Q., Kim, H.J., Barni, M. (eds.) IWDW 2009. LNCS, vol. 5703, pp. 211–226. Springer, Heidelberg (2009)
16. Motwani, R., Bekris, K., Motwani, M., Harris, F.: Fragile watermarking of 3d motion data. In: International Conference on Computer Applications in Industry and Engineering, vol. (1), pp. 111–116 (2010)
17. Pankajakshan, V., Doerr, G., Bora, P.: Assessing motion coherency in video watermarking. In: ACM Multimedia and Security, pp. 114–119 (2006)
18. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. International Journal of Computer Vision 50(2), 203–226 (2002)
19. Rondao-alface, P., Macq, B.: Blind watermarking of 3d meshes using robust feature points detection. In: International Conference on Image Processing, vol. 1, pp. 693–696 (2005)
20. Su, K., Kundur, D., Hatzinakos, D.: Statistical invisibility for collusion-resistant digital video watermarking. IEEE Transaction on Multimedia 7(1), 43–51 (2005)
21. Wang, X., Zhao, H.: A novel synchronization invariant audio watermarking scheme based on dwt and dct. IEEE Transaction on Signal Processing 54(12), 4835–4840 (2006)
22. Yamazaki, S.: Watermarking motion data. In: Proceedings of Pacific Rim Workshop on Digital Steganography, pp. 177–185 (2004)
23. Yang, S., Song, Z., Fang, Z., Yang, J.: A novel affine attack robust blind watermarking algorithm. In: Symposium on Security Detection and Information Processing, vol. 7, pp. 239–246 (2010)
24. Zhang, J., Ho, A.T.S., Qiu, G., Marziliano, P.: Robust video watermarking of h. 264/avc. 264/avc. IEEE Transaction on Circuits and Systems 54(2), 205–209 (2007)

A Game-Theoretic Approach to Content-Adaptive Steganography

Pascal Schöttle and Rainer Böhme

Department of Information Systems, University of Münster, Germany
{pascal.schoettle,rainer.boehme}@uni-muenster.de

Abstract. Content-adaptive embedding is widely believed to improve steganographic security over uniform random embedding. However, such security claims are often based on empirical results using steganalysis methods not designed to detect adaptive embedding. We propose a framework for content-adaptive embedding in the case of imperfect steganography. It formally defines heterogeneity within the cover as a necessary condition for adaptive embedding. We devise a game-theoretic model for the whole process including cover generation, adaptive embedding, and a detector which anticipates the adaptivity. Our solution exhibits a unique equilibrium in mixed strategies. Its location depends on the level of heterogeneity of the cover source, but never coincides with naïve adaptive embedding. The model makes several simplifying assumptions, including independent cover symbols and the steganalyst's ability to recover the adaptivity criterion perfectly.

Keywords: Content-Adaptive Steganography, Game Theory, Security.

1 Introduction and Motivation

In the past couple of years, several so-called *content-adaptive* steganographic schemes have been proposed, e. g., [13,25,26,21,23]. They all have in common that they embed in the locations of the cover medium, which are most suitable for embedding, i. e., where changes are (supposed to be) harder to detect. To find these locations, the schemes specify an *adaptivity criterion*, e. g., the local variance. Most often the superiority of content-adaptive over random uniform embedding is claimed on the grounds of better resistance against selected steganalysis methods, not tailored to detect adaptive embedding. However, such arguments disregard Kerckhoffs' principle [20]: the warden knows the adaptivity criterion as well and may be able to reproduce or estimate its values. In other words, the adaptivity criterion leaks side information to the warden.

Furthermore, most of the adaptive schemes embed the m bits of the secret message M into the m most "secure" locations of the cover medium. From now on, we will call this kind of adaptive embedding *naïve adaptive steganography*. There is initial evidence that this is not optimal. For example, it is shown in [4] that the adaptive embedding function suggested in [9] is less secure than uniform random embedding, if the attacker recalculates the adaptivity criterion.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 125–141, 2013.

An implication of this finding is that restricting the steganographer to the most suitable embedding locations can lead to less secure steganography. Therefore, leaving the steganographer with more choice on where to embed may strengthen the resistance to steganalysis methods specifically designed to detect adaptive embedding. As the steganalyst, in turn, can anticipate this behavior, she has to be given choice, too. Game theory is the preferred method to model a situation with two (or more) opponents who can adjust their strategies according to assumptions about the behavior of the other(s). In general, they want to either maximize their gain or minimize their loss in a competitive environment. So-called Nash equilibria [22] are stable situations in this environment, where none of the players would benefit from unilaterally changing her strategy.

Game theory requires that all participants have a parameter of choice. In our case this choice is discrete for both players, steganographer and steganalyst. We model the choice of the steganographer as the decision to embed either in the better location or in the worse. A steganalyst who anticipates adaptive embedding can choose which of the symbols she pays more attention to, depending on their suitability for embedding.

This paper documents a first attempt to develop a rigorous approach to secure content-adaptive steganography. We formulate a game-theoretic model spanning the entire process from cover generation to embedding and detection. For now, we keep the model as simple as possible in order to be able to solve our game, and to calculate theoretical bounds of detectability for arbitrary embedding and detection functions. By this, we are able to prove that naïve adaptive steganography is never optimal and introduce the term of *optimal adaptive steganography* as an adaptive embedding function, which anticipates a steganalysis technique that is aware of content-adaptive embedding and may recover the adaptivity criterion. Depending on the level of heterogeneity, optimal adaptive embedding distributes the embedding changes between more secure and less secure locations.

This paper is organized as follows: Section 2 briefly reviews related work. Section 3 gives a formal definition of heterogeneity and develops our basic model including first conclusions about which strategies are possible at all. Section 4 deals with the game-theoretical payoff function and optimal strategies for both players. The results are discussed in Section 5. Finally, Section 6 draws a conclusion and prioritizes directions for future work.

2 Related Work

The idea of combining game theory with steganographic security was first mentioned by Ettinger in 1998 [7], who proposes zero-sum games to model the contest between a data-hider and a data-attacker. He studies active attackers who not only want to detect, but to suppress hidden communication. Consequently, this approach is less focussed on indistinguishability, but on the maximum capacity which can be hidden robust enough to prevent an attacker, who is bound by a distortion constraint, from suppressing the channel.

Ker [16] uses game theory to find best strategies for a steganographer who can spread her secret message over several homogeneous cover media (batch steganography), and a steganalyst who anticipates this and tries to detect the existence of at least one secret message (pooled steganalysis). He concludes that a (batch) steganographer should either spread her payload as thinly as possible or concentrate it as much as possible. The specific choice of the payoff function precludes to fully explore mixed strategy equilibria. So the author presents min-max and max-min solutions in pure strategies.

To our knowledge there are no other game-theoretic works in the area of steganographic security so far. However, in general, game theory is gaining popularity in the field of information security, e. g. [1,14].

In the context of syndrome coding, Fridrich [10] shows that for sufficiently large covers, it is never optimal to embed only into the symbols which cause the least amount of (additive) distortion. Her result, along with the definition of a *detectability profile*, which mirrors our notion of an adaptivity criterion, is relevant for adaptive steganography. However, her work does not specify a detector. Therefore, it solves an optimization problem and not a game.

3 Our Model

3.1 Definition of Heterogeneity

A precondition for adaptive steganography is heterogeneity within the cover. For example, in images, flat regions are less secure to embed, whereas edges and noisy areas are likely more secure. Until now, there is no formal definition of heterogeneity for the purpose of adaptive embedding. We try to close this gap.

Definition 1 (Cover). *A sequence of n k-bit symbols is called* cover*, if it is a realization of the (cover) distribution P_0. More specifically, every symbol of the cover can take values in $\{0, \ldots, 2^k - 1\}$.*

Cachin [5] defines information-theoretic security of a steganographic system. He assumes that the distribution of the covers P_0 and the distribution of the stego objects P_1 are known. Then he suggests to use the *Kullback–Leibler divergence* (KLD) as a measure of discrepancy between these two distributions. He derives bounds for the detectability of a steganographic embedding function. A lower KLD indicates more similar distributions and thus a more secure embedding function. Therefore, if the embedding function is fixed, it is convenient to base a definition of heterogeneity on KLD.

Definition 2 (Heterogeneity). *A cover is called* heterogeneous*, if it contains (well-defined) areas, where embedding changes result in a lower KLD. I. e., let P_0 be the probability distribution of the cover and $P_{(x_i)}$ be the altered probability distribution after making a specific embedding change at location x_i. Then, the cover is* heterogeneous *iff there exists $i, j \in [1, \ldots, n], i \neq j$ with $KLD(P_0, P_{(x_i)}) \neq KLD(P_0, P_{(x_j)})$. Otherwise the cover is* homogeneous.

So, the simplest model to study adaptive embedding consists of exactly two areas which differ in their detectability of embedding changes.

3.2 Game-Theoretical Setup

Let *Alice* be the steganographer and *Eve* be the steganalyst.

As mentioned in Section 1, Eve has access to the embedding function. This is a realistic assumption and in line with Kerckhoffs' principle. There are discussions on how to interpret this principle for steganography [6,2,8], but Eve's access to the embedding function should be undisputed. Alice does not know the cover distribution P_0, because with that knowledge she could choose her stego objects like realizations of P_0 and could thus perform perfect steganography [24]. Granting Eve access to both distributions P_0 and P_1 (which would be the case for a strict interpretation of Kerckhoffs' principle [8]) would enable her to detect at the information-theoretic bound. This is neither realistic nor interesting to examine. We follow [3,18] where it is argued that a more realistic setup is incomplete information on both sides. With this condition, neither perfect embedding nor best possible detection is practicable and thus, both players have to make choices. In particular, both players have to anticipate the choice of their opponent. By this we are in a classical game-theoretic situation.

As mentioned above, the simplest model to study adaptive embedding consists of exactly two areas. We further specify this to a model with exactly two 2-bit symbols $p_0^{(0)}, p_1^{(0)}$, one better suitable for embedding than the other, i. e., $n = k = 2$. Following the notation in [3], the superscript (0) in $p_i^{(0)}$ denotes a symbol before embedding and the superscript (1) in $p_i^{(1)}$ denotes a symbol after embedding. If symbols are independent (see Sect. 3.6 below), we can think of larger heterogeneous covers as sets of pairs of pixels $(p_0^{(0)}, p_1^{(0)})$ drawn from two equally sized areas of different detectability. The game is repeated for each pair.

Since steganographic security is defined by the indistinguishability between cover and stego objects, we start with the "game" introduced by Katzenbeisser and Petitcolas [15]. Despite the name, their setup is not a game in a game-theoretic sense, but inspired by cryptographic security proofs. We augment it with choice variables in adaptive embedding to make it a veritable game.

Figure 1 shows the extensive form of our game. The different entities in our game are: *Nature*, the steganographer *Alice*, the *Judge*, and the steganalyst *Eve*. Nature generates a cover with exactly two symbols $p_0^{(0)}, p_1^{(0)}$, according to a predefined probability mass function (PMF). Without loss of generality, among the two symbols, $p_0^{(0)}$ is always better or equally suitable for embedding than $p_1^{(0)}$. Upon receiving a heterogeneous cover from Nature, Alice embeds with probability $\bar{a}$ into $p_0^{(0)}$ and with probability $1 - \bar{a}$ into $p_1^{(0)}$. The Judge is fair and forwards with constant probability $\mu = 1/2$ either the cover or the stego object to Eve. In a game-theoretic sense, the Judge is a part of Nature. When Eve gets either the cover or the stego, she examines $p_0^{(1)}$ with probability $\bar{e}$ and $p_1^{(1)}$ with probability $1 - \bar{e}$. Then she has to make a decision about the type of object she received.

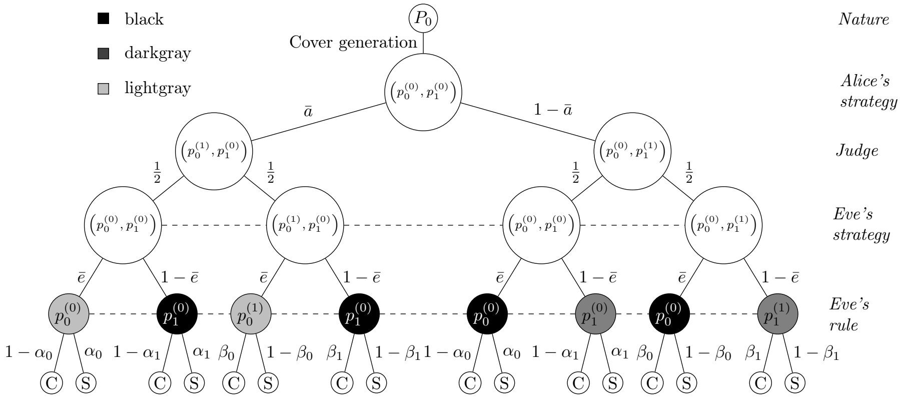


Fig. 1. Content-adaptive game in extensive form. The dashed line indicates Eve's information set, i. e., Eve does not know which of the connected nodes has been reached. α_i, β_i are the false positive, respectively false negative, rates for f_{m_i}, the PMF of $p_i^{(0)}$.

3.3 Embedding Function

We model LSB replacement as embedding function because it is best studied and well tractable. We will introduce one modification in that Alice always has to flip one bit instead of one on average. We justify this by the fact that in practice it is very unlikely that not a single bit has to be flipped. The corresponding probability is 2^{-m} for an m-bit message and thus negligible in m.

Note that changing exactly one symbol is incompatible with the popular simplifying assumption of independent embedding. It makes the symbols dependent in P_1 even if they were independent in P_0. Therefore P_1 cannot be decomposed into a product of the PMFs of its symbols. Other models are certainly conceivable, but not considered in this work.

3.4 Strategies

Alice's parameter of choice is $a \in \{0, 1\}$. A value of $a = 1$ means she embeds in $p_0^{(0)}$, i. e., the better suitable symbol, and $a = 0$ means she embeds in $p_1^{(0)}$. We assume that the order of suitability is perfectly preserved through embedding (not an unrealistic assumption for several so far proposed adaptivity criteria) and thus Eve can recover it. In future models we may relax the assumption of perfect recovery and replace it by a partial recovery.

Eve's parameter of choice is $e \in \{0, 1\}$. A value of $e = 1$ means she examines $p_0^{(1)}$, i. e., the better suitable symbol, and $e = 0$ means she examines $p_1^{(1)}$. We model Eve's decision in the way that she can either examine p_0 or p_1, but not both at the same time. We justify this by the fact that for real-world covers, it is intractable to use all information from the joint distribution of all symbols in the sequence. Although specific steganalysis methods can take all symbols into account, there has to be a weighting decision [19,17] and we model this decision in our model by an exclusive either p_0 or p_1. Note that whenever restricting the adversary, security claims may break down if she is more powerful than assumed.

Game theory differentiates between stable situations in *pure* and in *mixed* strategies. A pure strategy is a strategy where a player deterministically decides what to do, whereas a mixed strategy is a probability distribution over pure strategies. To be able to research the mixed strategies as well, we introduce the random variable A, of which Alice's choice a is a realization and the random variable E, of which Eve's choice e is a realization. Furthermore, let $\bar{a} = \text{prob}(A = 1)$ and $\bar{e} = \text{prob}(E = 1)$ be Alice's, respectively Eve's, parameter in mixed strategies. Now, a value of $\bar{a} = 1/2$ means that Alice embeds randomly without bias and a value of $\bar{e} = 1/2$ means that Eve examines both symbols with the same probability.

3.5 Exclusion of Pure Strategies

Lemma 1. *Under the assumption that $P_0 \neq P_1$ for LSB replacement, i. e., LSB replacement does not preserve the cover distribution perfectly, there is no equilibrium in pure strategies.*

Proof. There are exactly four pure strategies in the above described game.

1. Alice embeds always in $p_0^{(0)}$.
2. Alice embeds always in $p_1^{(0)}$.
3. Eve examines always p_0.
4. Eve examines always p_1.

If Alice follows strategy (1) (i. e., naïve adaptive embedding), Eve's best response would be strategy (3), because she would not gain from examining the other location. Hence, Alice would change her strategy to (2) so that Eve would not get any information from examining $p_0^{(0)}$. Now, Eve would switch to (4) because all information would be in $p_1^{(1)}$. Now, Alice's best response would be strategy (1) again, because Eve will not detect changes there. By this they are in an infinite loop.

So, in every situation in pure strategies, one of the players would benefit from changing her strategy. Therefore no equilibrium exists in pure strategies. □

3.6 Cover Generation Model

We need a model to represent some (simplified) conditions of heterogeneous cover sources. For this, our model should have one parameter m_i to adjust the level of heterogeneity. Now, the distribution P_0 according to which the two ordered symbols $p_0^{(0)}$ and $p_1^{(0)}$ are realised, is a discrete bivariate distribution of $f_{m_0}^{(0)}$ (the PMF of $p_0^{(0)}$) and $f_{m_1}^{(0)}$ (the PMF of $p_1^{(0)}$) with $m_0 \neq m_1$ (if $m_0 = m_1$, we model a homogeneous cover). Here, m_i measures the suitability for embedding. A value of $m_i = 0$ indicates a uniform distribution (i. e., maximal entropy) and allows perfect steganography. With increasing m_i, the entropy and the suitability for embedding decrease. As we assume that $p_0^{(0)}$ is more suitable for embedding, we define $m_0 \leq m_1$. In practice, the order of the symbols is established by the adaptivity criterion. Reordering the cover according to this criterion removes Markov-properties of the cover [8], but maintains some higher-order dependencies not regarded here, because most of them are incognizable or intractable in practice. Therefore, we may assume that the two ordered symbols are independent before embedding.

So the joint PMF of the cover generation $f^{(0)}(x_0, x_1)$ is given by

$$f^{(0)}(x_0, x_1) = f^{(0)}_{m_0}(x_0) \cdot f^{(0)}_{m_1}(x_1). \tag{1}$$

To fulfil the requirements from above, we model the family of probability mass functions depending on m_i as

$$f^{(0)}_{m_i}(x) = (2^k - x)m_i + \frac{1 - \left(\sum_{j=1}^{2^k} j\right) m_i}{2^k}, \quad x \in \{0, \ldots, 2^k - 1\}, \text{ with} \tag{2}$$

$$m_i \in \left[0; \left(\sum_{j=1}^{2^k-1} j\right)^{-1}\right), \text{ and therefore: } m_i \in \left[0; \frac{1}{6}\right) \text{ for } k = 2. \tag{3}$$

Equation (2) ensures that the sum of masses equals 1 and the masses for the different symbol values are strictly decreasing. The constraints in Equation (3) ensure that the PMF is never negative. Note that the interval has to be open. Otherwise the value $x = 2^k - 1$ would have zero mass. This would allow detection with certainty whenever this value occurs in a stego object after LSB flipping.

Figure 2 visualizes our cover generation model. For two fixed values of m_0, it shows the corresponding PMFs depending on m_1. A lower value of m_0 in the homogeneous case means a higher entropy. A bigger difference between m_0 and m_1 indicates a higher level of heterogeneity within the cover. As can be seen, by changing m_0 and m_1, the entropy as well as the level of heterogeneity change.

3.7 Embedding Impact

Let $f^{(1)}_{m_i}$ be the PMF resulting from always embedding in $p_i^{(0)}$. Then, for single symbol values x_j it holds, that:

$$f^{(0)}_{m_i}(x_j) = \text{prob}(x_j|\text{Cover}) \quad \text{and} \quad f^{(1)}_{m_i}(x_j) = \text{prob}(x_j|\text{Stego}). \tag{4}$$

As we are interested in the distribution after embedding P_1, we now proceed by examining the distribution after embedding in $p_0^{(0)}$ with probability $\bar{a}$ and embedding in $p_1^{(0)}$ with probability $1 - \bar{a}$.

The LSB replacement embedding operation emb(x) simply swaps the values $2j$ by $2j + 1$, and vice versa, for $j \in \{0, \ldots, 2^{k-1}\}$. This can be expressed by

$$\text{emb}(x) := x + (-1)^x \quad \Rightarrow \quad \text{emb}^{-1}(x) = \text{emb}(x). \tag{5}$$

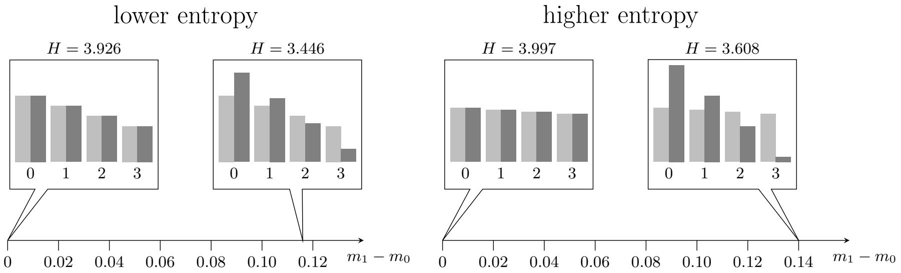


Fig. 2. Cover generation model with increasing levels of heterogeneity from left to right. $f_{m_0}^{(0)}$ is light gray, $f_{m_1}^{(0)}$ is dark gray. Left: $m_0 = 0.05, m_1 \in \{0.05, 0.165\}$. Right: $m_0 = 0.01, m_1 \in \{0.01, 0.15\}$.

Now in our model, where we always embed, it holds that

$$f_{m_i}^{(1)}(x_j) = f_{m_i}^{(0)}(emb^{-1}(x_j)), \quad j \in \{0, 1, \ldots, n\}. \tag{6}$$

This yields the following lemma about $f_{m_i}^{(1)}(x_j)$.

Lemma 2. *In our model, the PMF* $f_{m_i}^{(1)}(x_j)$ *is*

$$f_{m_i}^{(1)}(x_j) = \begin{cases} f_{m_i}^{(0)}(x_j + 1), & : x_j \equiv 0 \pmod 2 \\ f_{m_i}^{(0)}(x_j - 1), & : x_j \equiv 1 \pmod 2 \end{cases} \tag{7}$$

$$= \begin{cases} f_{m_i}^{(0)}(x_j) - m_i, & : x_j \equiv 0 \pmod 2 \\ f_{m_i}^{(0)}(x_j) + m_i, & : x_j \equiv 1 \pmod 2). \end{cases} \tag{8}$$

Proof. From Equation (6) we know that:

$$\begin{aligned} f_{m_i}^{(1)}(x_j) &= f_{m_i}^{(0)}(emb^{-1}(x_j)) \\ &= f_{m_i}^{(0)}(x_j + (-1)^{x_j}) \\ &= \begin{cases} f_{m_i}^{(0)}(x_j + 1), & : x_j \equiv 0 \pmod 2 \\ f_{m_i}^{(0)}(x_j - 1), & : x_j \equiv 1 \pmod 2). \end{cases} \end{aligned} \tag{9}$$

And with Equation (2):

$$\begin{aligned} f_{m_i}^{(1)}(x_j) &= \begin{cases} (2^k - (x_j + 1))m_i + \frac{1 - \left(\sum_{j=1}^{2^k} j\right) m_i}{2^k}, & : x_j \equiv 0 \pmod 2 \\ (2^k - (x_j - 1))m_i + \frac{1 - \left(\sum_{j=1}^{2^k} j\right) m_i}{2^k}, & : x_j \equiv 1 \pmod 2 \end{cases} \\ &= \begin{cases} f_{m_i}^{(0)}(x_j) - m_i, & : x_j \equiv 0 \pmod 2 \\ f_{m_i}^{(0)}(x_j) + m_i, & : x_j \equiv 1 \pmod 2). \end{cases} \end{aligned} \tag{10}$$

□

As Lemma 1 excludes both pure strategies, we get a mixed strategy and thus a mixture distribution of the kind,

$$f^{(1)}(x_0, x_1) = \bar{a}\left(f^{(1)}_{m_0}(x_0)\cdot f^{(0)}_{m_1}(x_1)\right) + (1-\bar{a})\left(f^{(0)}_{m_0}(x_0)\cdot f^{(1)}_{m_1}(x_1)\right). \qquad (11)$$

To quantify the overall information Eve can potentially gain from the embedding function, we can numerically calculate the KLD between $f^{(0)}$ and $f^{(1)}$ as benchmark for a numerical analysis. This is certainly precluded for real covers.

3.8 Eve's Decision

The parameter on which Eve's choice relies is $\bar{e}$. This indicates to which symbol she assigns a higher weight. This symbol will influence her decision and thus her false positive and false negative rates more. Conveniently, as will be shown in this paragraph, the false positive rate equals the false negative rate in our model. So we have only one variable of interest, the *equal error rate* (*EER*).

Recall that we have a strictly decreasing PMF and thus for P_0 it holds that,

$$f^{(0)}_{m_i}(0) > f^{(0)}_{m_i}(1) > f^{(0)}_{m_i}(2) > f^{(0)}_{m_i}(3). \qquad (12)$$

Therefore, we know from Lemma 2 that in pure strategies it holds that,

$$f^{(1)}_{m_i}(1) > f^{(1)}_{m_i}(0) > f^{(1)}_{m_i}(3) > f^{(1)}_{m_i}(2). \qquad (13)$$

This is sufficient to derive Eve's optimal decision rule $DR(x_j)$ between "Cover" and "Stego" for individual symbols.

Lemma 3. *Eve's best decision rule for individual symbol values x_j is:*

$$DR(x_j) = \begin{cases} \text{Cover}, & : x_j \equiv 0 \pmod 2 \\ \text{Stego}, & : x_j \equiv 1 \pmod 2. \end{cases} \qquad (14)$$

Proof. The decision rule implements the *maximum a posteriori* (MAP) estimation, which can be found, for example, in [11]. Here it is important to notice that the a priori probability of "Cover" prob(Cover) $= \mu = 1/2$ equals the probability of "Stego" prob(Stego) $= \mu = 1/2$ because the Judge is fair.

The MAP estimation minimizes the decision errors by calculating:

$$\hat{\theta} = \arg\max_{\theta} \text{prob}(\theta|x) = \arg\max_{\theta} \text{prob}(x|\theta)\cdot\text{prob}(\theta). \qquad (15)$$

With $\theta \in \{\text{Cover}, \text{Stego}\}$ and $x = x_j$, this results in

$$\begin{aligned} \hat{\theta} &= \arg\max_{\theta} \text{prob}(x_j|\theta)\cdot\mu \\ &\overset{Eq.\,(4)}{=} \max\left\{f^{(0)}_{m_i}(x_j), f^{(1)}_{m_i}(x_j)\right\} \\ &= \begin{cases} \text{Cover}, & : x_j \equiv 0 \pmod 2 \\ \text{Stego}, & : x_j \equiv 1 \pmod 2, \end{cases} \end{aligned} \qquad (16)$$

because of Equations (12) and (13). □

Thus, in our case with $n = k = 2$, Eve's decides for "Cover" whenever she sees a symbol with value 0 or 2, and "Stego" for values 1 and 3.

Let α_i and β_i be Eve's false positive, respectively, false negative rate for $f_{m_i}^{(0)}$ and $f_{m_i}^{(1)}$. By Lemma 3, her true positive rate $(1 - \alpha_i)$ (and consequently the false positive rate as well) is aggregated between the cases where her decision yields "Cover" and the same for the true negative rate $(1 - \beta_i)$ in all other cases.

Lemma 4. *In our model, Eve's false positive rate α_i equals her false negative rate β_i and thus is called* equal error rate EER_i*.*

$$EER_i = \alpha_i = \beta_i = \frac{1}{2} - m_i, \tag{17}$$

for $i \in \{0, 1\}$.

The proof can be found in Appendix A.1.

Equation (17) is intuitive, as values of $m_i = 0$ indicate an uniform distribution. In this case P_1 would equal P_0, i. e., the same distribution before and after embedding. Therefore the false positive and false negative rate would be 50%, i. e., random guessing. Furthermore, it follows our initial thoughts that a higher value of m_i implies a better detectability, which materializes in a lower EER.

Corollary 1. *The worst case for Eve would be Alice choosing $a \in \{0, 1\}$ and she herself choosing $e = 1 - a$ because by this, her decision would be merely guessing, i. e., $EER = 0.5$.*

The proof can be found in Appendix A.2.

This confirms Lemma 1 that there is no equilibrium in pure strategies, as with every pure strategy, one of the players would benefit from changing her strategy to the opposite. Now we are in the position to solve the game and to identify equilibria in mixed strategies.

4 Solving the Game

The EER described in Section 3.8 can be seen as the payoff function in our zero-sum game. As it is Alice's intention to perform least detectable steganography, her goal is to maximize the EER, whereas it is Eve's goal to maximize her detection rate and thus, to minimize the EER.

4.1 Payoff Function

From Figure 1 and the EER described in Section 3.8, the payoff function $\chi(\bar{a}, \bar{e})$ for mixed strategies can be derived and equals the overall EER. It is stated in the following corollary.

Corollary 2. *In our model, the payoff function in mixed strategies is*

$$\chi(\bar{a}, \bar{e}) = \frac{1}{2} - (\bar{a} \cdot \bar{e} \cdot m_0 + (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot m_1) \tag{18}$$

Proof. From Figure 1 it can be seen, that the nodes of Eve's decision (shaded nodes) can be partitioned into three different situations.

The first situation is that Alice embeds in $p_0^{(0)}$ and Eve anticipates this (lightgray nodes in Figure 1). This situation occurs with probability $\bar{a} \cdot \bar{e}$. When faced with a situation like this, we know from Equation (17) that Eve's EER equals α_0 $(= \beta_0)$. The second possible situation is that Alice embeds in $p_1^{(0)}$ and Eve, again, anticipates this (darkgray nodes in Figure 1). The occurrence probability of this situation is $(1 - \bar{a}) \cdot (1 - \bar{e})$. Again, we know the payoff from Equation (17), which is α_1 $(= \beta_1)$. The third and last situation is that Alice embeds in $p_i^{(0)}$, but Eve inspects the wrong location (black nodes in Figure 1). This situation occurs with probability $(1 - \bar{a}) \cdot \bar{e}$ (for Alice embedding in $p_0^{(0)}$, but Eve examining $p_1^{(1)}$) and $\bar{a} \cdot (1 - \bar{e})$ (for Alice embedding in $p_1^{(0)}$, but Eve examining $p_0^{(1)}$). Here, we know from Corollary 1 that Eve's decision rule is no better than random guessing and thus has an EER of $1/2$.

This leads to the following expression for $\chi(\bar{a}, \bar{e})$,

$$\begin{aligned}\chi(\bar{a}, \bar{e}) &= (\bar{a} \cdot \bar{e}) \cdot \alpha_0 + ((1 - \bar{a}) \cdot \bar{e} + \bar{a} \cdot (1 - \bar{e})) \cdot \frac{1}{2} + (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot \alpha_1 \\ &= (\bar{a} \cdot \bar{e}) \cdot \alpha_0 + \frac{\bar{a} + \bar{e} - 2\bar{a}\bar{e}}{2} + (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot \alpha_1 . \qquad (19)\end{aligned}$$

From Lemma 4 we know that $\alpha_i = 1/2 - m_i$ and thus:

$$\begin{aligned}\chi(\bar{a}, \bar{e}) &= (\bar{a} \cdot \bar{e}) \cdot (\frac{1}{2} - m_0) + \frac{\bar{a} + \bar{e} - 2\bar{a}\bar{e}}{2} + (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot (\frac{1}{2} - m_1) \\ &= \frac{\bar{a} \cdot \bar{e}}{2} - \bar{a} \cdot \bar{e} \cdot m_0 + \frac{\bar{a} + \bar{e} - 2\bar{a} \cdot \bar{e}}{2} \\ &+ \frac{1}{2} - \frac{\bar{a}}{2} - \frac{\bar{e}}{2} + \frac{\bar{a} \cdot \bar{e}}{2} - (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot m_1 \qquad (20) \\ &= \frac{1}{2} - (\bar{a} \cdot \bar{e} \cdot m_0 + (1 - \bar{a}) \cdot (1 - \bar{e}) \cdot m_1) \qquad (21)\end{aligned}$$

□

4.2 Best Strategies

As the payoff function is the same for both players but with contrary goals, i. e., Alice wants to maximize it, while Eve wants to minimize it, an equilibrium in mixed strategies can be found by looking at the partial derivatives of the payoff function and setting them to zero. With this method we find a unique equilibrium of our model, which happens to be symmetric.

Lemma 5. *In our model, there exists a unique symmetric Nash equilibrium in mixed strategies. In this equilibrium it holds that:*

$$\bar{a}^* = \bar{e}^* = \frac{m_1}{m_0 + m_1} \qquad (22)$$

Proof. The partial derivatives of the payoff function are,

$$\frac{\partial\chi(\bar{a},\bar{e})}{\partial\bar{a}} = -\,(m_0+m_1)\cdot\bar{e}+m_1 \tag{23}$$

$$\frac{\partial\chi(\bar{a},\bar{e})}{\partial\bar{e}} = -\,(m_0+m_1)\cdot\bar{a}+m_1. \tag{24}$$

Setting both derivatives to zero yields the values for the equilibrium,

$$-(m_0+m_1)\cdot\bar{e}+m_1 \overset{!}{=} 0 \Leftrightarrow \bar{e}^* = \frac{m_1}{m_0+m_1} \tag{25}$$

$$-(m_0+m_1)\cdot\bar{a}+m_1 \overset{!}{=} 0 \Leftrightarrow \bar{a}^* = \frac{m_1}{m_0+m_1}. \tag{26}$$

□

Inserting these optimal values into $\chi(\bar{a}^*,\bar{e}^*)$ yields the equilibrium EER .

Corollary 3. *In the equilibrium it holds that the EER is,*

$$EER^* = \chi\left(\frac{m_1}{m_0+m_1},\frac{m_1}{m_0+m_1}\right) = \frac{1}{2}-\frac{m_0\cdot m_1}{m_0+m_1}. \tag{27}$$

Proof. Equation (21) can be rearranged to

$$\chi(\bar{a},\bar{e}) = \frac{1}{2}-\left((m_0+m_1)\cdot(\bar{a}\cdot\bar{e})-m_1\cdot\bar{a}-\bar{e}\cdot m_1+m_1\right), \tag{28}$$

and using $\bar{e}=\bar{a}=\bar{a}^*=\frac{m_1}{m_0+m_1}$ from Lemma 5 we obtain,

$$\chi(\bar{a}^*,\bar{a}^*) = \frac{1}{2}-\left((m_0+m_1)\cdot(\bar{a}^*)^2-2\cdot m_1\cdot\bar{a}^*+m_1\right) \tag{29}$$

$$= \frac{1}{2}-\left((m_0+m_1)\cdot\left(\frac{m_1}{m_0+m_1}\right)^2-\frac{2\cdot {m_1}^2}{m_0+m_1}+m_1\right) \tag{30}$$

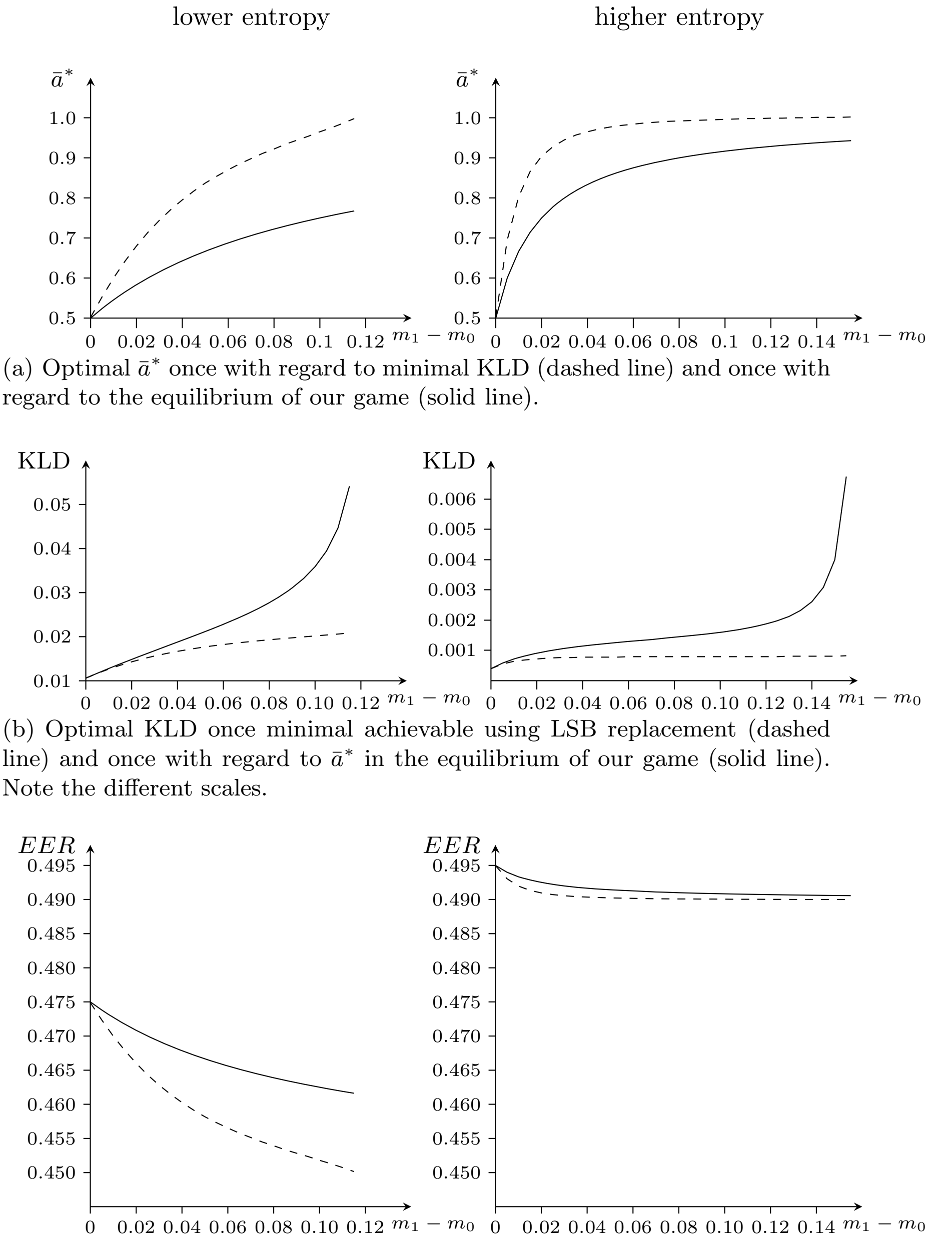
$$= \frac{1}{2}-\left(m_1-\frac{{m_1}^2}{m_0+m_1}\right) = \frac{1}{2}-\frac{m_0\cdot m_1}{m_0+m_1}. \tag{31}$$

□

With this unique value for $\bar{a}^*$, we say a steganographer performs *optimal adaptive steganography*. It is always less detectable than a steganographer who performs naïve adaptive steganography.

5 Discussion

One implication of our analysis is that the optimal distribution of embedding changes depends on the level of heterogeneity in the cover source. So, steganographer and steganalyst both have to adjust their strategy to the cover source.



(a) Optimal $\bar{a}^*$ once with regard to minimal KLD (dashed line) and once with regard to the equilibrium of our game (solid line).

(b) Optimal KLD once minimal achievable using LSB replacement (dashed line) and once with regard to $\bar{a}^*$ in the equilibrium of our game (solid line). Note the different scales.

(c) Optimal EER with optimal KLD and fixed detector (dashed line) and once in the equilibrium of our game (solid line).

Fig. 3. Comparsion of equilibrium parameters with numerical benchmarks based on KLD, as a function of the level of heterogeneity. Left figures: $m_0 = 0.05, m_1 \in [0.05, 0.165]$. Right figures: $m_0 = 0.01, m_1 \in [0.01, 0.165]$.

The discussion of our results is facilitated by looking at numerical examples in Figure 3. As one requirement for our model was simplicity, we are able to calculate the KLD as benchmark, which is infeasible for real-world cover sources.

Figure 3(a) shows the optimal value of $\bar{a}^*$, once by numerically minimizing KLD (dashed line) and once the value found in the equilibrium (solid line). Figure 3(b) shows the KLD created by the values for $\bar{a}^*$ from the figure above and Figure 3(c) shows the resulting EER. To recall how the corresponding PMFs look like, they are displayed in Figure 2. Figure 3(a) reveals that if Alice's goal was to minimize KLD, she would choose higher values for $\bar{a}^*$, i. e., embed with higher probability in the better suitable location. Furthermore, it can be seen in Figure 3(b) that the KLD generated by Alice's strategy in the equilibrium increases rapidly with an increasing level of heterogeneity. Nonetheless, Figure 3(c) shows that Alice's strategy in the equilibrium implicates a higher EER than in the situation with minimal KLD, and thus more secure steganography against the specific detector defined in our model. By this, both players could perform better, if the other would not follow the strategy in the equilibrium. So, it follows that if Alice tries to minimize the KLD and Eve anticipates this (still being bound to her specific detector), Eve's detection rate would increase and thus Alice would perform less secure steganography.

6 Conclusion and Outlook

The literature is full of content-adaptive embedding schemes, but most of them seem to be designed ad-hoc. Their security relies solely on the opinion of the respective developer that the adaptivity criterion of her choice is good at selecting secure embedding locations. To overcome such design methods in the medium term, we give a first definition of heterogeneity for content-adaptive steganography and specify a model of the entire process, covering the choices of Alice and Eve, and being simple enough to be tractable, both in terms of game-theoretic equilibria and information-theoretic benchmarks.

We show that *naïve adaptive steganography*, the strategy to embed only in the most suitable locations of a heterogeneous cover, is never optimal. We solve our model and find a unique equilibrium of our game, where none of the players would gain from changing her strategy. As a result, we define a new kind of adaptive embedding, the so-called *optimal adaptive steganography*, which takes into account the knowledge of an attacker who can recover (or estimate) the values of the adaptivity criterion as side information.

The way we model the level of adaptivity certainly needs further refinement and, in future works, we may be able to relax some of the restrictions we impose on our model. Furthermore, as mentioned in Section 3.6, changing m_0 and m_1 of the cover generation model influences the entropy and the level of heterogeneity. It would be more convenient if both quantities of interest could be adjusted independently. This is a goal for future models.

It is obvious that a cover model with exactly two locations is not realistic, so there is space for future work. Special attention in these future models has to be be paid to what happens if the parameters $\bar{a}, \bar{e}$, our players' parameters of choice, become divided into n instead of 2 parts. By this, we have to think about how to model the different weights and will most likely come to an optimization problem over the function field, similar to the batch steganography problem stated in [16].

Another open question is the relation between adaptive embedding and steganalysis based on machine learning. As a first remark on the combination of these two areas, [12] states that "[...] it does not appear that giving [Eve] probabilistic information about the selection channel is a weakness".

Another field for future research is the advantage the attacker gains from cover estimation in the case of heterogeneity *within* and *between* covers. By adding this to our model, we end up with a double-stochastic cover generation process. As can be seen by these examples, a rigorous understanding of content-adaptive steganography in theory and practice remains a relevant target for future investigations.

Acknowledgement. This research was funded by the German science foundation (DFG) under grant "Sichere adaptive Steganographie". The second author thanks Andrew Ker for useful discussions on game theory and steganography in general.

References

1. Bensoussan, A., Kantarcioglu, M., Hoe, S.: A Game-Theoretical Approach for Finding Optimal Strategies in a Botnet Defense Model. In: Alpcan, T., Buttyán, L., Baras, J. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 135–148. Springer, Heidelberg (2010)
2. Böhme, R.: An Epistemological Approach to Steganography. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 15–30. Springer, Heidelberg (2009)
3. Böhme, R.: Advanced Statistical Steganalysis. Springer, Berlin (2010)
4. Böhme, R., Westfeld, A.: Exploiting Preserved Statistics for Steganalysis. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 82–96. Springer, Heidelberg (2004)
5. Cachin, C.: An Information-Theoretic Model for Steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)
6. Cayre, F., Bas, P.: Kerckhoffs–based embedding security classes for WOA data hiding. IEEE Transactions on Information Forensics and Security 3(1), 1–15 (2008)
7. Ettinger, J.M.: Steganalysis and Game Equilibria. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 319–328. Springer, Heidelberg (1998)
8. Filler, T., Ker, A.D., Fridrich, J.: The square root law of steganographic capacity for Markov covers. SPIE, vol. 7254, p. 725408 (2009)
9. Franz, E.: Steganography Preserving Statistical Properties. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 278–294. Springer, Heidelberg (2003)
10. Fridrich, J.: Minimizing the embedding impact in steganography. In: Voloshynovskiy, S. (ed.) MM&Sec 2006: Proceedings of the 8th Workshop on Multimedia and Security, pp. 2–10. ACM (2006)

11. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications, 1st edn. Cambridge University Press, New York (2009)
12. Fridrich, J., Kodovský, J., Holub, V., Goljan, M.: Steganalysis of Content-Adaptive Steganography in Spatial Domain. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 102–117. Springer, Heidelberg (2011)
13. Fridrich, J., Du, R.: Secure Steganographic Methods for Palette Images. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 47–60. Springer, Heidelberg (2000)
14. Johnson, B., Grossklags, J., Christin, N., Chuang, J.: Are Security Experts Useful? Bayesian Nash Equilibria for Network Security Games with Limited Information. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 588–606. Springer, Heidelberg (2010)
15. Katzenbeisser, S., Petitcolas, F.: Defining security in steganographic systems. SPIE, vol. 4675, pp. 50–56 (2002)
16. Ker, A.: Batch steganography and the threshold game. SPIE, vol. 6505, p. 650504 (2007)
17. Ker, A.: Estimating Steganographic Fisher Information in Real Images. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 73–88. Springer, Heidelberg (2009)
18. Ker, A.: The Square Root Law in Stegosystems with Imperfect Information. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 145–160. Springer, Heidelberg (2010)
19. Ker, A., Böhme, R.: Revisiting weighted stego-image steganalysis. SPIE, vol. 6819, p. 681905 (2008)
20. Kerckhoffs, A.: La cryptographie militaire. Journal des Sciences Militaires IX(1), 5–38 (1883)
21. Luo, W., Huang, F., Huang, J.: Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security 5(2), 201–214 (2010)
22. Nash, J.: Non-cooperative games. The Annals of Mathematics 54(2), 286–295 (1951)
23. Pevný, T., Filler, T., Bas, P.: Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 161–177. Springer, Heidelberg (2010)
24. Wang, Y., Moulin, P.: Perfectly secure steganography: Capacity, error exponents, and code constructions. IEEE Transactions on Information Theory 54(6), 2706–2722 (2008)
25. Wu, H.C., Wu, N.I., Tsai, C.S., Hwang, M.S.: Image steganographic scheme based on pixel-value differencing and LSB replacement methods. In: IEE Proceedings – Vision, Image and Signal Processing, vol. 152(5), pp. 611–615 (2005)
26. Yang, C.H., Weng, C.Y., Wang, S.J., Sun, H.M.: Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Transactions on Information Forensics and Security 3(3), 488–497 (2008)

A Appendix

A.1 Proof of Lemma 4

As mentioned in Section 3.8, Eve's true positive and true negative rate can be calculated as follows:

True Positives $\mathrm{TP}(x_j)$:

$$x_j = 0 : \mathrm{TP}(0) = \frac{f_{m_i}^{(0)}(0)}{f_{m_i}^{(0)}(0) + f_{m_i}^{(1)}(0)} = \frac{f_{m_i}^{(0)}(0)}{f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(1)} \tag{32}$$

$$x_j = 2 : TP(2) = \frac{f_{m_i}^{(0)}(2)}{f_{m_i}^{(0)}(2) + f_{m_i}^{(1)}(2)} = \frac{f_{m_i}^{(0)}(2)}{f_{m_i}^{(0)}(2) + f_{m_i}^{(0)}(3)} \tag{33}$$

$$\Rightarrow (1-\alpha_i) = (f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(1)) \cdot TP(0) + (f_{m_i}^{(0)}(2) + f_{m_i}^{(0)}(3)) \cdot TP(2) \tag{34}$$

$$= f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(2) \tag{35}$$

True Negatives $\mathrm{TN}(x_j)$:

$$x_j = 1 : \mathrm{TN}(1) = \frac{f_{m_i}^{(1)}(1)}{f_{m_i}^{(0)}(1) + f_{m_i}^{(1)}(1)} = \frac{f_{m_i}^{(0)}(0)}{f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(1)} = TP(0) \tag{36}$$

$$x_j = 3 : \mathrm{TN}(3) = \frac{f_{m_i}^{(1)}(3)}{f_{m_i}^{(0)}(3) + f_{m_i}^{(1)}(3)} = \frac{f_{m_i}^{(0)}(2)}{f_{m_i}^{(0)}(2) + f_{m_i}^{(0)}(3)} = TP(2) \tag{37}$$

$$\Rightarrow (1-\beta_i) = (f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(1)) \cdot TN(1) + (f_{m_i}^{(0)}(2) + f_{m_i}^{(0)}(3)) \cdot TN(3) \tag{38}$$

$$= f_{m_i}^{(1)}(1) + f_{m_i}^{(1)}(3) = f_{m_i}^{(0)}(0) + f_{m_i}^{(0)}(2) = (1-\alpha_i) \tag{39}$$

$$\overset{Eq.(2)}{\Leftrightarrow} (1-\alpha_i) = (1-\beta_i) = 4 \cdot m_i + \frac{1-10m_i}{4} + 2 \cdot m_i + \frac{1-10m_i}{4} \tag{40}$$

$$= 6 \cdot m_i + 2 \cdot \frac{1-10m_i}{4} = \frac{2 \cdot m_i + 1}{2} = m_i + \frac{1}{2} \tag{41}$$

$$\Rightarrow EER_i = \alpha_i = \beta_i = \frac{1}{2} - m_i. \tag{42}$$

for $i \in \{0,1\}$. □

A.2 Proof of Corollary 1

If Eve chooses $e = 1 - a$ and $a \in \{0,1\}$, it holds that Alice always embeds in $p_a^{(0)}$ and by this never into $p_e^{(0)}$. From Eq. (6) it follows that $f_{m_a}^{(1)}(x_j) = f_{m_a}^{(0)}(emb^{-1}(x_j))$, but $f_{m_e}^{(1)}(x_j) = f_{m_e}^{(0)}(x_j)$, as there is no embedding in $p_e^{(0)}$. Therefore, it holds that:

$$\left.\begin{array}{l} x_j \in \{0,2\} : \mathrm{TP}(x_j) \\ x_j \in \{1,3\} : \mathrm{TN}(x_j) \end{array}\right\} = \frac{f_{m_e}^{(0)}(x_j)}{f_{m_e}^{(0)}(x_j) + f_{m_e}^{(1)}(x_j)}$$

$$= \frac{f_{m_e}^{(0)}(x_j)}{f_{m_e}^{(0)}(x_j) + f_{m_e}^{(0)}(x_j)} = \frac{f_{m_e}^{(0)}(x_j)}{2 \cdot f_{m_e}^{(0)}(x_j)} = \frac{1}{2}. \tag{43}$$

□

Key-Efficient Steganography

Aggelos Kiayias[1], Alexander Russell[1], and Narasimha Shashidhar[2]

[1] University of Connecticut, Storrs, CT 06269
{aggelos,acr}@cse.uconn.edu
[2] Sam Houston State University, Huntsville, TX 77341
karpoor@shsu.edu

Abstract. Steganographic protocols enable one to embed covert messages into inconspicuous data over a public communication channel in such a way that no one, aside from the sender and the intended receiver, can even detect the presence of the secret message. In this paper, we provide a new provably-secure, private-key steganographic encryption protocol secure in the framework of Hopper et al. [2]. We first present a "one-time stegosystem" that allows two parties to transmit messages of length at most that of the shared key with *information-theoretic* security guarantees; employing a pseudorandom generator (PRG) then permits secure transmission of longer messages in a striaghtforward manner.

The advantage of our construction in comparison with previous work is *key-length efficiency*: in the information-theoretic setting our protocol embeds a n bit message using a shared secret key of length $(1 + o(1))n$ while achieving security $2^{-n/\log^{O(1)} n}$: this gives a rate of key length over message length that converges to 1 as $n \to \infty$; the previous best result [5] achieved a constant rate > 1 regardless of the security offered. In this sense, our protocol is the first truly key-length efficient steganographic system. Furthermore, in our protocol, we can permit a portion of the shared secret key to be *public* while retaining precisely n private key bits. In this setting, by separating the public and the private randomness of the shared key, we achieve security of 2^{-n}. Our result comes as an effect of a novel application of randomness extractors to stegosystem design.

1 Introduction

The steganographic communication scenario can be described using Simmons' [15] formulation of the problem: Alice and Bob are prisoners who wish to communicate securely in the presence of an adversary, called the "Warden." The warden monitors the communication channel to detect whether they exchange "conspicuous" messages. In particular, Alice and Bob may exchange messages that adhere to certain channel distribution that represents "inconspicuous" communication. By controlling the messages transmitted over such a channel, a stegosystem permits Alice and Bob to exchange messages that cannot be detected by the Warden. There have been two approaches in formalizing this problem, one based on information theory [1, 17, 6] and one based on complexity theory [2, 5]. The

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 142–159, 2013.

latter approach is more concrete and has the potential of allowing more efficient constructions.

Most steganographic constructions supported by provable security guarantees are instantiations of the following basic procedure (often referred to as "rejection-sampling"). The problem specifies a family of message distributions (the "channel distributions") that provide a number of possible options for a so-called "covertext" to be transmitted. Additionally, the sender and the receiver possess some sort of private information (typically a keyed hash function, MAC, or other similar function) that maps channel messages to a single bit. In order to send a message bit m, the sender draws a covertext from the channel distribution, applies the function to the covertext and checks whether it happens to produce the "stegotext" m she originally wished to transmit. If this is the case, the covertext is transmitted. In case of failure, this procedure is repeated. We remark that this appears to be a minimal configuration for steganography to be feasible: (1) in the absence of a channel distribution there is no way to model what is allowed to be transmitted between the two communicating parties, (2) furthermore, if the channel distribution has no entropy, the cover communication between the two parties becomes deterministic and thus there is no way to communicate subliminally, (3) finally, the sender should be capable of sampling the channel distribution "in her mind" and introduce some appropriate biases in the distribution of covertexts that are communicated, otherwise, as before, no subliminal communication can occur.

The complexity-theoretic approach to secure steganography considers the following experiment for the warden-adversary: The adversary selects a message to be embedded and receives either covertexts that embed the message or covertexts simply drawn from the channel distribution (without any embedding): the adversary is then asked to distinguish between the two cases. If the probability of success is very close to 1/2, the stegosystem is declared secure against such (eavesdropping) adversarial activity. Formulation of stronger attacks (such as active attacks) is also possible.

Given the above framework, Kiayias et al. [4, 5] provided a provably secure stegosystem that pairs rejection sampling with a t-wise independent family of functions. They design a *one-time stegosystem*, a steganographic protocol that is meant to be used for a single message transmission and is proven secure in an information-theoretic sense, provided that the key that is shared between the sender and the receiver is of sufficient length. This system is a natural analogue of a one time-pad for steganography.

We work in this same information-theoretic framework, presenting a steganography protocol that embeds a message of length n using a shared secret key of length $(1 + o(1))n$ bits while achieving security $2^{-n/\log^{O(1)} n}$. In this sense, our protocol is **key-length efficient**: the rate of key over message approaches 1 for large values of n. In the best previous protocol [4], the length of the shared secret key is at least $(2 + o(1))n$ bits long regardless of the security achieved. Given our one-time stegosystem, it is straightforward to construct provably secure steganographic encryption for longer messages by using a pseudorandom generator (PRG) to stretch a random seed that is shared by the sender and the receiver to sufficient length as shown in [5]. The resulting stegosystem is provably secure in the complexity-theoretic model.

We are able to obtain an improvement in the key length by introducing two novelties: We perform a variant of rejection sampling which is more efficient in its use of the shared secret key and couple this sampling with judicious usage of randomness extractors. To the best of our knowledge, this is the first time extractors have been employed in the design of steganographic protocols. A further interesting feature of our protocol is that we can permit a portion of the shared secret key to be *public* while retaining precisely n private key bits. In this setting, by separating the public and the private randomness of the shared key, we can achieve security of 2^{-n}. We adopt the model of channel abstraction first defined by von Ahn [16] (and also used in [5]).

2 Preliminaries

We use the notation $x \leftarrow X$ to denote sampling an element x from a distribution X and the notation $x \in_R S$ to denote sampling an element x uniformly at random from a set S. For a function f and a distribution X on its domain, $f(X)$ denotes the distribution that results from sampling x from X and applying f to x. The uniform distribution on $\{0,1\}^d$ is denoted by U_d. We use the notation $|s|$ to stand for the number of symbols in a string s. For a probability distribution D with support X, the notation $\Pr_D[x]$ denotes the probability that D assigns to $x \in X$. We let $\mathbb{E}$ denote expectation. The concatenation of string s_1 and string s_2 is denoted by $s_1 \circ s_2$. All logarithms are taken base 2.

Definition 1 (Pointwise ϵ-biased functions). *Let P be a distribution with a finite support X. A function $f : X \to Y$ is said to be* pointwise ϵ-biased *with respect to P if* $\forall y \in Y \ |\Pr_{x \leftarrow P}[f(x) = y] - 1/|Y| \ | < \epsilon$.

In this paper, we refer to such functions as ϵ-biased and drop the "pointwise" qualification for simplicity.

Definition 2 (Min-entropy). *The* min-entropy *of a random variable X, taking values in a set V, is the quantity $H_\infty(X) \triangleq \min_{v \in V}(-\log \Pr[X = v])$. A random variable with min-entropy at least t is called a* t-source. *We apply this same terminology to distributions.*

Statistical Distance. We use statistical distance to measure the distance between two random variables. See Shoup [14] for a detailed discussion on statistical distance and its properties.

Definition 3. *Let X and Y be random variables which both take values in a finite set S with probability distributions P_X and P_Y. The statistical distance between X and Y is defined as $\Delta[X,Y] \triangleq (1/2)\sum_{s \in S} |P_X(s) - P_Y(s)|$.*

We say that X and Y are ϵ-close *if $\Delta[X,Y] \leq \epsilon$.*

We will use the following properties of statistical distance which follow directly from the definition.

Fact 1. *Let X, Y and Z be random variables taking values in a finite set S. We have (i.) $0 \leq \Delta[X,Y] \leq 1$ and (ii.) the triangle inequality: $\Delta[X,Z] \leq \Delta[X,Y] + \Delta[Y,Z]$.*

Fact 2. *If S and T are finite sets, X and Y are random variables taking values in the set S and $f : S \to T$ is a function, then $\Delta[f(X), f(Y)] \leq \Delta[X,Y]$.*

Lemma 1. *Consider two random variables (X,Y) and (X',Y'), both taking values in $\mathcal{X} \times \mathcal{Y}$. For a particular value $x \in \mathcal{X}$ in the support of X, we let Y_x denote the random variable Y conditioned on the event $X = x$ and define Y'_x likewise. Then $\Delta[(X,Y),(X',Y')] \leq \Delta[X,X'] + \mathbb{E}_X\left[\Delta[Y_X, Y'_X]\right]$.*

We include the proof in Appendix A for completeness.

2.1 Extractors

Extractors are deterministic functions that operate on arbitrary distributions with sufficient randomness and output "almost" uniformly distributed, independent random bits (see, e.g., [9]). Extractors require an additional input: a short seed of truly random bits as a catalyst to "extract" randomness from such distributions; thus the input of an extractor is two independent sources of randomness: the source of guaranteed min-entropy and a short uniformly random seed.

Definition 4. *A (t,ϵ)-*extractor *is a function* $\mathrm{Ext} : \{0,1\}^\nu \times \{0,1\}^d \to \{0,1\}^\mu$ *such that for every random variable X on $\{0,1\}^\nu$ with $H_\infty(X) \geq t$,* $\mathrm{Ext}(X, U_d)$ *is ϵ-close to U_μ.*

For our application, we require a stronger property from the extractor. We need the output of the extractor to remain essentially uniform even conditioned on the seed. A way of enforcing this condition is to demand that when the seed is concatenated to the output, the resulting distribution is still ϵ-close to uniform. Such an extractor is called a *strong* extractor to distinguish from the weaker notion of extractors defined above. As we shall require this stronger notion throughout, we shall use the term extractor to refer to a strong extractor.

Definition 5. *A (t,ϵ)-strong extractor is a function* $\mathrm{Ext} : \{0,1\}^\nu \times \{0,1\}^d \to \{0,1\}^\mu$ *such that for every random variable X on $\{0,1\}^\nu$ with $H_\infty(X) \geq t$, the random variable* $S \circ \mathrm{Ext}(X,S)$ *is ϵ-close to $U_{\mu+d}$ if S is distributed according to U_d.*

We refer to ν as the *length of the source*, t as the *min-entropy threshold*, ϵ as the *error* of the extractor, the ratio t/ν as the *entropy rate* of the source X and to the ratio μ/t as the *fraction of randomness* extracted by the extractor. The entropy loss of the extractor is defined as $t+d-\mu$. The two inputs of the extractor have a total min-entropy of at least $t+d$ and the entropy loss measures how much of this randomness was "lost" in the extraction process. Radhakrishnan and Ta-Shma [10] showed that no non-trivial (t, ϵ)-extractor can extract all the randomness present in its inputs and must suffer an entropy loss of $2\log(1/\epsilon) + O(1)$. For our application, we need efficient, explicit strong extractor constructions as defined below.

Definition 6 ([13]). *For functions* $t(\nu)$, $\epsilon(\nu)$, $d(\nu)$, $\mu(\nu)$ *a family* $\mathrm{Ext} = \{\mathrm{Ext}_\nu\}$ *of functions* $\mathrm{Ext}_\nu : \{0,1\}^\nu \times \{0,1\}^{d(\nu)} \to \{0,1\}^{\mu(\nu)}$ *is an explicit* (t, ϵ)*-strong extractor if* $\mathrm{Ext}(x, y)$ *can be computed in polynomial time in its input length and for every* ν, Ext_ν *is a* $(t(\nu), \epsilon(\nu))$*-extractor.*

An important property of strong extractors which makes them attractive for our application is that for any t-source, a $(1 - \epsilon)$ fraction of the seeds extract randomness from that source.

Remark ([12]). Let $\mathrm{Ext} : \{0,1\}^\nu \times \{0,1\}^d \to \{0,1\}^\mu$ be a (t, ϵ)-strong extractor. From the definition of a strong extractor, we know that $\mathbb{E}_s\left[\Delta\left[\mathrm{Ext}(X, s), U_\mu\right]\right] \le \epsilon$ where $s \in_R \{0,1\}^d$. By Markov's inequality, $\Pr_s[\Delta\left[\mathrm{Ext}(X, s), U_\mu\right] \ge \epsilon \cdot r] \le 1/r$. Later on in the paper, we will use this result for $r = \epsilon^{-2/3}$ and $r = \epsilon^{-1/2}$.

See the survey articles by Shaltiel [13], Nisan [7], and Nisan and Ta-Shma [8] for more details on extractors and their properties. In this paper, we use the explicit strong extractor construction by Raz, Reingold and Vadhan [11] which works on sources of any min-entropy. It extracts all the min-entropy using $O(\log^3 \nu)$ additional random seed bits while achieving an optimal entropy loss (up to an additive constant) of $\chi = 2\log(1/\epsilon) + O(1)$ bits.

Theorem 1 (RRV Extractor [11]). *For every* ν, $t \in \mathbb{N}$, *and* $\epsilon > 0$ *such that* $t \le \nu$, *there are explicit* (t, ϵ)*-strong extractors* $\mathrm{Ext} : \{0,1\}^\nu \times \{0,1\}^d \to \{0,1\}^{t-\chi}$ *with entropy loss* $\chi = 2\log(1/\epsilon) + O(1)$ *bits and requiring seeds of length*

$$d = O(\log^2 \nu \cdot \log(1/\epsilon) \cdot \log t)\,.$$

2.2 The Channel Model

The security of a steganography protocol is measured by the adversary's ability to distinguish between "normal" and "covert" message distributions over a communication channel. To characterize normal communication we define and formalize the *communication channel* following standard terminology used in the literature [2, 1, 16, 5, 3]. We let Σ denote the symbols of an alphabet and treat the *channel* as a family of distributions $\mathcal{C} = \{C_h\}_{h \in \Sigma^*}$; each C_h is supported on Σ. These channel distributions model a history-dependent notion of channel data.

We adopt the model of channel abstraction first defined by von Ahn and Hopper [16]. Here, Alice is provided with a means for sampling "deep into the channel." In particular, Alice and, consequently, the steganographic encoding protocol, has access to a channel oracle that can sample from the channel for *any* history. Formally, during the embedding process, Alice may sample from

$C_{h_1 \circ \ldots \circ h_\ell}$ for any history she wishes (though Alice is constrained to be efficient and so can make no more than polynomially many queries of polynomial length). This model allows Alice to transform a channel C with min-entropy δ into a channel C^π with min-entropy $\pi\delta$. Specifically, the channel C^π is defined over the alphabet Σ^π, whose elements we write as vectors $\mathbf{h} = (h_1, \ldots, h_\pi)$. The distribution $C^\pi_{\mathbf{h}^1,\ldots,\mathbf{h}^v}$ is determined by the channel C with history $(h_1^1 \circ \cdots \circ h_\pi^1) \circ \cdots \circ (h_\pi^v \cdots \circ h_\pi^v)$. This definition captures the adaptive nature of the channel by taking into account the dependence between symbols as is typical in real world communications. We say that a channel has *min-entropy* δ if $\forall h \in \Sigma^*$, $H_\infty(C_h) \geq \delta$. Observe that this implies that $H_\infty(C_h^\pi) \geq \delta\pi$ due to the additive nature of marginal min-entropy.

2.3 One-Time Stegosystem

Here, we give the definition of a *one-time stegosystem*, a steganographic system that enables the one-time steganographic transmission of a message provided that the two parties share a suitable key. We adopt the definitions used by Kiayias et al. [5].

Definition 7. *A* one-time stegosystem *consists of three probabilistic polynomial time algorithms* $S = (SK, SE, SD)$*, where:*

- SK *is the* key generation algorithm*; we write* $SK(1^k) = \kappa$*. It produces a key* κ *of length* k*.*
- SE *is the* embedding procedure *and has access to the channel;* $SE\,(\kappa, m; \mathcal{O}) = s \in \Sigma^*$*.*
- SD *is the* extraction procedure*;* $SD\,(\kappa, c) = m$*. It takes as input the key* κ *of length* k*, and some* $c \in \Sigma^*$*. The output is a message* m*.*

The embedding procedure SE takes into account the history h of communication that has taken place between Alice and Bob thus far and begins its operation corresponding to this history. It takes as input the key κ of length k, a message m of length $n = n(k)$ and accesses the channel through an (probabilistic) oracle $\mathcal{O}$. The oracle $\mathcal{O}$ accepts as input *any* polynomial length history $h' \in \Sigma^*$ and allows SE to draw independent samples repeatedly from $C_{h \circ h'}$. The output is the stegotext $s \in \Sigma^*$. Observe that in a one-time stegosystem, once a security parameter k is chosen, the length of the message n is a fixed function of k. In our model of channel abstraction, SE can access the channel for *any* history. We next define a notion of correctness for a one-time stegosystem.

Definition 8 (Correctness). *A one-time stegosystem* (SK, SE, SD) *is said to be* (ϵ, δ)-correct *provided that for all channels* $\mathcal{C}$ *of min-entropy* δ*, it holds that* $\forall h \in \Sigma^*$

$$\forall m \in \{0,1\}^{n(k)} \;\; \Pr[SD(\kappa, SE(\kappa, m; \mathcal{O})) \neq m \mid \kappa \leftarrow SK(1^k)] \leq \epsilon \; .$$

In general, we treat both $\epsilon = \epsilon(k)$ and $\delta = \delta(k)$ as functions of k, the security parameter and the oracle $\mathcal{O}$ as a function of the history h.

One-time stegosystem security is based on the indistinguishability between a transmission that contains a steganographically embedded message and a transmission that contains no embedded messages. The adversarial game discussed next is meant to model the behavior of a warden in Simmons' formulation of the problem discussed earlier.

An adversary $\mathcal{A}$ against a one-time stegosystem $S = (SK, SE, SD)$ is a pair of algorithms $\mathcal{A} = (SA_1, SA_2)$, that plays the following game, denoted $G^{\mathcal{A}}(1^k)$:

1. A key κ is generated by $SK(1^k)$.
2. Algorithm SA_1 receives as input the security parameter k and outputs a triple $(m^*, \text{aux}, h_\mathsf{c}) \in M_n \times \{0,1\}^* \times \Sigma^*$, where aux is some auxiliary information that will be passed to SA_2. SA_1 is provided access to the channel via the oracle $\mathcal{O}$, which takes the history h as input. The channel oracle $\mathcal{O}$, on input h, returns to SA_1 an element c selected according to C_h. Observe that the adversary can determine the channel distribution for any history.
3. A bit b is chosen uniformly at random.
   - If $b = 0$, let $c^* \leftarrow SE(\kappa, m^*; \mathcal{O}(h_c))$, so c^* is a stegotext.
   - If $b = 1$, let $c^* = c_1 \circ \cdots \circ c_\lambda$ where $\lambda = |SE(\kappa, m^*; \mathcal{O})|$ and $c_i \overset{r}{\leftarrow} C_{h_c \circ \mathsf{c}_1 \circ \cdots \circ \mathsf{c}_{i-1}}$. In this case, $c^* \leftarrow C_{h_c}^{\lambda}$.
4. The input for SA_2 is 1^k, h_c, c^* and aux. SA_2 outputs a bit b'. If $b' = b$ then we say that (SA_1, SA_2) *succeeded* and write $G^{\mathcal{A}}(1^k) = \text{success}$.

The *advantage* of the adversary $\mathcal{A}$ over a stegosystem S is defined as:

$$\mathbf{Adv}_S^{\mathcal{A}}(k) = \left| \Pr\left[G^{\mathcal{A}}(1^k) = \text{success}\right] - \frac{1}{2} \right| .$$

The probability includes the coin tosses of $\mathcal{A}$ and SE.

The (information-theoretic) insecurity of the stegosystem is defined as

$$\mathbf{InSec}_S(k) = \max_{\mathcal{A}} \{\mathbf{Adv}_S^{\mathcal{A}}(k)\} ,$$

this maximum taken over all (time unbounded) adversaries $\mathcal{A}$.

Definition 9 (Security). *We say that a stegosystem is* (ϵ, δ)-secure *if for all channels with min-entropy* δ *we have* $\mathbf{InSec}_S(k) \leq \epsilon$.

Overhead. The *overhead* of a one-time stegosystem is judged by the relation of the key length k and message length n. We adopt the ratio k/n as the measure of overhead (cf. [5]).

2.4 Rejection Sampling

As noted before, a common method used in steganography over an arbitrary channel distribution is that of *rejection sampling* (see, e.g., [1, 2, 4, 5]). Kiayias et al. [4, 5] provided a provably secure stegosystem that pairs rejection sampling with a t-wise independent family of functions and Hopper et al. [2] paired rejection sampling with a pseudorandom function family. These techniques do not provide a sufficiently low overhead. To obtain an overhead of $1 + o(1)$, we use a variant of rejection sampling to transmit bit vectors as opposed to a single bit. To transmit bit vectors, we amplify the entropy of the channel as discussed before and apply ρ-rejection sampling described below. More precisely, we transform a channel C with min-entropy δ into a channel C^π with min-entropy $\pi\delta$, defined over the alphabet Σ^π. We next perform ρ-rejection sampling over C^π as follows: Assuming that one wishes to transmit a bit vector $\boldsymbol{m} \in \{0,1\}^\eta$ and employs a random function $f : \Sigma^\pi \to \{0,1\}^\eta$ that is secret from the adversary, one performs the following “rejection sampling” process:

```
$\textsc{Rej}^f_h(\boldsymbol{m},\rho)$
let $j = 0$
    repeat:
        sample $\boldsymbol{c} \leftarrow C^\pi_h$ , increment $j$
    until $f(\boldsymbol{c}) = \boldsymbol{m}$ or $(j > \rho)$
output: $\boldsymbol{c}$
```

For a given history h, the procedure $\textsc{Rej}^f_h(\boldsymbol{m},\rho)$ draws independent samples from the channel distribution C^π_h in rounds until $f(\boldsymbol{c}) = \boldsymbol{m}$ or $j > \rho$. As there are at most a total of $\rho+1$ rounds, if none of the first ρ samples drawn map to the target bit vector, the sample drawn at round $\rho + 1$ is returned by the procedure. Here, as defined before, Σ^π denotes the output alphabet of the channel, h denotes the history of the channel at the start of the process, and C^π_h denotes the marginal distribution on sequences of π symbols given by the channel after history h. The receiver (also privy to the function f) applies the function to the received message $\boldsymbol{c} \in \Sigma^\pi$ and recovers $\boldsymbol{m}$ with some probability of success (related, ultimately, to the correctness of the protocol). Note that the above process performs $\rho + 1$ draws from the channel with the *same* history. These draws are assumed to be independent. One basic property of rejection sampling that we use is:

Lemma 2 ([16]). *If the function f is ϵ-biased on C^π_h for history h, then for any ρ and uniformly random $\boldsymbol{m} \in_R \{0,1\}^\eta$: $\Delta \left[\textsc{Rej}^f_h(\boldsymbol{m},\rho), C^\pi_h\right] \leq \epsilon$.*

Proof. Let us denote the samples drawn by the procedure $\textsc{Rej}^f_h(\boldsymbol{m},\rho)$ as $\boldsymbol{c_i}, i = 1, \cdots, \rho+1$. Suppose that the target bit vector $\boldsymbol{m}$ was chosen with the probability $P^{(\boldsymbol{m})}_f \triangleq \Pr[f(C^\pi_h) = \boldsymbol{m}]$ instead of being chosen uniformly at random, i.e, $\boldsymbol{m} \leftarrow P^{(\boldsymbol{m})}_f$. We first show that the output from $\textsc{Rej}^f_h(\boldsymbol{m},\rho)$ is distributed identically to C^π_h. For simplicity of notation, let us define $p_m \triangleq \Pr_{P^{(\boldsymbol{m})}_f}[\boldsymbol{m}]$. Let p_c denote the probability of drawing $\boldsymbol{c}$ from the channel distribution C^π_h, i.e., $p_c \triangleq \Pr_{C^\pi_h}[\boldsymbol{c}]$. For $\boldsymbol{c} \in C^\pi_h$, the probability of observing $\boldsymbol{c}$ under the $\textsc{Rej}^f_h(\boldsymbol{m},\rho)$ procedure is then given by $\Pr[\textsc{Rej}^f_h(\boldsymbol{m},\rho) = \boldsymbol{c}]$ which we may expand as

$$\begin{aligned}
&\Pr_{\boldsymbol{c}_1 \leftarrow C^\pi_h}[\boldsymbol{c}_1 = \boldsymbol{c}] \cdot \Pr[f(\boldsymbol{c}_1) = \boldsymbol{m}] + \Pr_{\boldsymbol{c}_2 \leftarrow C^\pi_h}[\boldsymbol{c}_2 = \boldsymbol{c}] \cdot \Pr[f(\boldsymbol{c}_2) = \boldsymbol{m}] \cdot \Pr[f(\boldsymbol{c}_1) \neq \boldsymbol{m}] \\
&\qquad + \Pr_{\boldsymbol{c}_3 \leftarrow C^t_h}[\boldsymbol{c}_3 = \boldsymbol{c}] \cdot \Pr[f(\boldsymbol{c}_3) = \boldsymbol{m}] \cdot \Pr[f(\boldsymbol{c}_1) \neq \boldsymbol{m} \wedge f(\boldsymbol{c}_2) \neq \boldsymbol{m}] + \cdots
\end{aligned}$$

equal to

$$p_c p_m + p_c p_m (1 - p_m) + \cdots + p_c p_m (1 - p_m)^{\rho - 1} + p_c (1 - p_m)^{\rho} = p_c \,.$$

From the above discussion, we can see that when the target bit vector $\boldsymbol{m}$ was chosen from the distribution $P_f^{(\boldsymbol{m})}$, the output from $\textsc{Rej}_h^f(\boldsymbol{m}, \rho)$ is distributed identically to C_h^π. Since f is ϵ-biased, $\Delta\left[U_\eta, P_f^{(\boldsymbol{m})}\right] \leq \epsilon$. Hence,

$$\Delta\left[\textsc{Rej}_h^f(\boldsymbol{m} \leftarrow U_\eta, \rho), \textsc{Rej}_h^f(\boldsymbol{m} \leftarrow P_f^{(\boldsymbol{m})}, \rho)\right] \leq \epsilon$$

by Fact 2 which gives us the statement of the lemma.

3 The Construction

In this section, we outline our construction of a one-time stegosystem as an interaction between Alice (the sender) and Bob (the receiver). Alice and Bob wish to communicate over a channel C_h^π with history h. We also assume that the support of $\mathcal{C}_h$ is $\{0,1\}^b$, i.e, $|\Sigma| = 2^b$.

3.1 A One-Time Stegosystem

Let $\boldsymbol{m} \in \{0,1\}^n$ be the message to be embedded. Our stegosystem uses the *RRV* strong-extractor construction as described in Theorem 1 which extracts randomness from the distribution C_h^π supported on $\{0,1\}^{\pi \cdot b}$ by rejection sampling as described in Section 2.4. Specifically, we will use the extractor with the seed s as the function f in the rejection sampling procedure.

Alice and Bob agree on the following:

Extractor Construction. Alice and Bob agree to use the explicit *RRV* strong-extractor construction as described in Theorem 1. They use a seed $s \in_R \{0,1\}^d$ for the extractor. The length of the seed d will be determined later as a function of δ, n, b and security ϵ. The notation E_s stands for the extractor used with the seed s i.e., $E(\cdot, s)$. Here, we treat the seed s as private and in Section 3.4 we show that the seed s may be *public* and discuss the implications of this choice.

One-Time Pad. Alice and Bob also use a shared one-time pad secret key $\kappa^{\text{otp}} \in_R \{0,1\}^n$ effectively transmitting $\boldsymbol{m'} = \kappa^{\text{otp}} \oplus \boldsymbol{m}$.

Shared Secret Key. The secret key that they now share is $\kappa = (\kappa^{\text{otp}}, s)$ of length $k = n + d$.

Key generation consists of generating the one-time pad secret key $\kappa^{\text{otp}} \in_R \{0,1\}^n$ and the random seed s of length d to be used with the extractor. The encoding procedure accepts an input message $\boldsymbol{m}$ of length n bits and outputs a stegotext of length λ. We will analyze the stegosystem below in terms of the parameters π, d, λ, ρ and some constant $c > 1$ relegating discussion of how these parameters determine the overall efficiency of the system to Section 3.4.

```
PROCEDURE $SE$:
Input: Key $\kappa = (\kappa^{\text{otp}}, s)$; $\boldsymbol{m} \in \{0,1\}^n$,
          history $h \in \Sigma^*$
let $\boldsymbol{m'} = \kappa^{\text{otp}} \oplus \boldsymbol{m}$
parse $\boldsymbol{m'}$ as $\boldsymbol{m'} = \boldsymbol{m'_1 m'_2 \dots m'_{\lceil n/c \log n \rceil}}$
for $i = 1$ to $\lceil n/c \log n \rceil$ {
        $\boldsymbol{c_i} \leftarrow \text{REJ}_h^{E_s}(\boldsymbol{m'_i}, \rho)$
        set $h \leftarrow h \circ \boldsymbol{c}_i$
}
Output: $\mathsf{c}_{\mathsf{stego}} = \boldsymbol{c_1 c_2 \dots c_{\lceil n/c \log n \rceil}} \in \Sigma^\lambda$
```

```
PROCEDURE $SD$:
Input: Key $\kappa = (\kappa^{\text{otp}}, s)$
          stegotext $\mathsf{c}_{\mathsf{stego}}$

parse: $\mathsf{c}_{\mathsf{stego}} = \boldsymbol{c_1 c_2 \dots c_{\lceil n/c \log n \rceil}}$
for $i = 1$ to $\lceil n/c \log n \rceil$ do {
        set $\boldsymbol{m_i}' = E_s(\boldsymbol{c}_i)$
}
set $\boldsymbol{m'} = \boldsymbol{m'_1 m'_2 \dots m'_{\lceil n/c \log n \rceil}}$
Output: $\boldsymbol{m'} \oplus \kappa^{\text{otp}}$
```

Fig. 1. Encryption and Decryption algorithms for the one-time stegosystem of 3.1

Alice and Bob communicate using the algorithm SE for steganographic embedding and SD for decoding as described in Figure 1. In SE, after applying the one-time pad to randomize her message $\boldsymbol{m}$, Alice obtains $\boldsymbol{m'} = \kappa^{\text{otp}} \oplus \boldsymbol{m}$. She then parses $\boldsymbol{m'}$ into $\lceil n/c \log n \rceil$ blocks, each block of length $c \log n$ for some constant $c > 1$, i.e., $\boldsymbol{m'} = \boldsymbol{m_1}'\boldsymbol{m_2}' \dots \boldsymbol{m}'_{\lceil n/c \log n \rceil}$. She then applies the procedure $\text{REJ}_h^{E_s}(\boldsymbol{m}'_i, \rho)$ to obtain an element $\boldsymbol{c_i} \in \Sigma^\pi$ for each block $\boldsymbol{m_i}', i = 1, \cdots, \lceil n/c \log n \rceil$ of the randomized message. Here, the history h represents the current history at the time of the rejection sampling procedure which is updated after the completion of the procedure. Recall that the notation E_s stands for the extractor used with the seed s i.e., $E(\cdot, s)$. The resulting stegotext, denoted by c_{stego} that is transmitted to Bob is $c_{\text{stego}} = \boldsymbol{c_1 c_2} \dots \boldsymbol{c}_{\lceil n/c \log n \rceil}$. In SD, the received stegotext is first parsed into $\lceil n/c \log n \rceil$ blocks as shown and then evaluated using the extractor with seed s for each block; this results in a message block. After performing this for each received block, a bit string of length n is obtained, which is subjected to the one-time pad decoding to obtain the original message. The detailed security and correctness analysis follow in the next two sections.

3.2 Security

In this section, we argue about the security of our one-time stegosystem. Specifically, we establish an upper bound on the statistical distance between the "normal" and "covert" message distributions over the communication channel. First, by Lemma 2, observe that if the function f is ϵ-biased on C_h^π for history h, then for any ρ, $\boldsymbol{m'} \in_R \{0,1\}^\eta$: $\Delta[\text{REJ}_h^f(\boldsymbol{m'}, \rho), C_h^\pi] \leq \epsilon$. Now, consider the strong extractor $\text{Ext} : \{0,1\}^\nu \times \{0,1\}^d \to \{0,1\}^\mu$ used in the rejection sampling procedure. Denote the error of extractor by ϵ_{ext}. Recall from the remark in Section 2.1 that, for a uniformly chosen seed $s \in_R \{0,1\}^d$, $\Pr_s[\Delta\,[\text{Ext}(X,s), U_\mu] \geq \sqrt{\epsilon_{ext}}] \leq \sqrt{\epsilon_{ext}}$. From this we can see that Ext fails to be a $\sqrt{\epsilon_{ext}}$-biased function with probability no more than $\sqrt{\epsilon_{ext}}$ in the choice of the seed s. Thus, for a random m' and s,

$$\Delta[\mathrm{REJ}_h^{E_s}(\boldsymbol{m'},\rho),C_h^{\pi}] \le 1\cdot\sqrt{\epsilon_{ext}}+\sqrt{\epsilon_{ext}}\cdot 1 \le 2\sqrt{\epsilon_{ext}}\,.$$

We obtain the above inequality by upper bounding the probability of the extractor being a $\sqrt{\epsilon_{ext}}$-biased function by 1 and observing that the statistical distance is also upper bounded by 1 by Fact 1. Suppose that in our stegosystem construction, we had used an independent and uniformly chosen seed $s_i \in_R \{0,1\}^d$ for each message block $i = 1,2,\cdots,\lceil n/c\log n\rceil$, the statistical distance between C_h^{λ} and the output of the procedure SE can be given by $\Delta\left[SE(\kappa,\boldsymbol{m};\mathcal{O}),C_h^{\lambda}\right] \le 2\sqrt{\epsilon_{ext}}\lceil n/c\log n\rceil$. However, employing an independent and uniformly chosen seed for each message block would require too much randomness. In our scheme, we employ a single seed s over all the message blocks and so we need to manage the dependencies between the various outputs, which is the major portion of the work done in the security proof. For a message $\boldsymbol{m} \in \{0,1\}^n$, we present an upper bound on $\Delta\left[SE(\kappa,\boldsymbol{m};\mathcal{O}),C_h^{\lambda}\right]$ when using a single seed $s \in_R \{0,1\}^d$ over all the message blocks.

Theorem 2. *For any $\epsilon,\delta > 0$ and message $\boldsymbol{m} \in \{0,1\}^n$, consider the stegosystem (SK,SE,SD) of Section 3.1 under the parameter constraint $\epsilon_{ext} \le \left(\frac{\epsilon}{3\ell}\right)^3$. Then it holds that the stegosystem is (ϵ,δ)-secure where ϵ_{ext} is the extractor error and $\ell = \lceil n/c\log n\rceil$ for some constant $c > 1$.*

Proof. We start the encoding procedure SE with history h which embeds message blocks into the channel using rejection sampling. We want to show that the statistical distance between the output of SE and C_h^{λ} is given by

$$\Delta\left[SE(\kappa,\boldsymbol{m};\mathcal{O}),C_h^{\lambda}\right] \le \epsilon$$

where λ is the length of the output by procedure SE.

First, we define some notation to capture the operation of the procedure SE. Let C_1 denote the distribution at depth 1 that results by sampling $\boldsymbol{c_1} \leftarrow C_h^{\pi}$; C_2 denotes the distribution at depth 2 that results by sampling $\boldsymbol{c_1} \leftarrow C_h^{\pi}$ and $\boldsymbol{c_2} \leftarrow C_{h\circ\boldsymbol{c_1}}^{\pi}$. We likewise define C_τ for $\tau \le \ell$. We define the random variables $R_1,\cdots,R_\tau$ obtained by rejection sampling in the same fashion. To be precise, for a message $\boldsymbol{m'} = \kappa^{\mathrm{otp}} \oplus \boldsymbol{m} = \boldsymbol{m_1}' \circ \boldsymbol{m_2}' \circ \cdots \circ \boldsymbol{m_\ell}'$ and $|\boldsymbol{m_\tau}'| = c\log n$ we define

$$C_1 \triangleq C_h^{\pi}\,, \quad C_\tau \triangleq C_{h\circ C_1\circ\cdots\circ C_{\tau-1}}^{\pi}\,,$$

for $\tau \in \{2,\ldots,\ell\}$. Likewise, we define the random variables R_τ:

$$R_1 \triangleq \mathrm{REJ}_h^{E_s(\cdot)}(\boldsymbol{m_1}',\rho)\,, \quad R_\tau \triangleq \mathrm{REJ}_{h\circ R_1\circ\cdots\circ R_{\tau-1}}^{E_s(\cdot)}(\boldsymbol{m_\tau}',\rho) \quad .$$

Finally, in anticipation of the proof below, we define a "hybrid" random variable

$$H_\tau = \mathrm{REJ}_{h\circ C_1\circ\cdots\circ C_{\tau-1}}^{E_s(\cdot)}(\boldsymbol{m_\tau}',\rho)$$

which corresponds to the distribution obtained by selecting $C_1,\ldots,C_{\tau-1}$ from the natural channel distribution, and then selecting the τth channel element via rejection sampling.

Now, let us analyze the implications of picking a uniformly random seed $s \in_R \{0,1\}^d$ for the extractor as we do in our construction. Recall that ϵ_{ext} denotes the error of the extractor. First, we show that for each depth τ, the probability mass of distributions for which the extractor coupled with the seed s yields a $\sqrt[3]{\epsilon_{ext}}$-biased function is large.

We say that a channel distribution C is $\left(s, \sqrt[3]{\epsilon_{ext}}\right)$-good if E_s is $\sqrt[3]{\epsilon_{ext}}$-biased on C. Otherwise we say that the distribution C is $\left(s, \sqrt[3]{\epsilon_{ext}}\right)$-bad. With this definition in place, recall that a strong extractor has the property that for any distribution C on the right domain with sufficient min-entropy,

$$\Pr_s[C \text{ is } (s, \sqrt[3]{\epsilon_{ext}}) \text{ -bad}] \leq \epsilon_{ext}^{2/3} . \tag{1}$$

Define now the following sets for $\tau \in \{0, \cdots, \ell - 1\}$:

$$G_s^\tau = \left\{(\boldsymbol{c_1}, \boldsymbol{c_2}, \cdots, \boldsymbol{c_\tau}) \mid C^\pi_{h \circ \boldsymbol{c_1} \circ \boldsymbol{c_2} \circ \cdots \circ \boldsymbol{c_\tau}} \text{ is } \left(s, \sqrt[3]{\epsilon_{ext}}\right) \text{-good}\right\}$$

and

$$B_s^\tau = \left\{(\boldsymbol{c_1}, \boldsymbol{c_2}, \cdots, \boldsymbol{c_\tau}) \mid C^\pi_{h \circ \boldsymbol{c_1} \circ \boldsymbol{c_2} \circ \cdots \circ \boldsymbol{c_\tau}} \text{ is } \left(s, \sqrt[3]{\epsilon_{ext}}\right) \text{-bad}\right\},$$

where $|\boldsymbol{c_i}| = \pi$. The two sets G_s^τ and B_s^τ denote the collection of $\left(s, \sqrt[3]{\epsilon_{ext}}\right)$-good and $\left(s, \sqrt[3]{\epsilon_{ext}}\right)$-bad distributions at depth τ, respectively. Let $\mu\left(B_s^\tau\right)$ denote $\Pr\left[C_h^{\tau\pi} \in B_s^\tau\right]$, the total probability mass of the set B_s^τ. Define $\mu\left(G_s^\tau\right)$ similarly. Observe that in light of Equation (1) above, the expected mass of B_s^τ over the choice of a uniform seed s is $\mathbb{E}_s\left[\mu\left(B_s^\tau\right)\right] \leq \epsilon_{ext}^{2/3}$. By Markov's inequality $\Pr_s\left[\mu\left(B_s^\tau\right) \geq \sqrt[3]{\epsilon_{ext}}\right] \leq \sqrt[3]{\epsilon_{ext}}$ and, then, by the union bound we conclude

$$\Pr_s\left[\exists \tau < \ell \mid \mu\left(B_s^\tau\right) \geq \sqrt[3]{\epsilon_{ext}}\right] \leq \ell \sqrt[3]{\epsilon_{ext}} \ .$$

where $\ell = \lceil n/c \log n \rceil$, the number of message blocks. We say that a seed s is *good* if $\forall \tau \in \{1, 2, \cdots, \ell\}$, $\mu\left(G_s^\tau\right) \geq 1 - \sqrt[3]{\epsilon_{ext}}$. To summarize the discussion above, for randomly chosen s, $\Pr_s[s \text{ is good}] \geq 1 - \ell \sqrt[3]{\epsilon_{ext}}$.

Now, fix a good seed s. We will now prove that for a good seed s,

$$\Delta\left[(C_1, C_2, \cdots, C_\ell), (R_1, R_2, \cdots, R_\ell)\right] \leq \ell \cdot (3 \sqrt[3]{\epsilon_{ext}}) \ . \tag{2}$$

We prove this by induction on τ, the number of message blocks. When $\tau = 1$, $\Delta\left[C_1, R_1\right] \leq 2\sqrt{\epsilon_{ext}} \leq 2\sqrt[3]{\epsilon_{ext}}$, as desired. In general, assuming

$$\Delta\left[(C_1, C_2, \cdots, C_\tau), (R_1, R_2, \cdots, R_\tau)\right] \leq \tau \cdot (2 \sqrt[3]{\epsilon_{ext}}) \ .$$

for a particular value τ, we wish to establish the inequality for $\tau + 1$. In light of Lemma 1, we conclude that $\Delta\Big[(C_1, C_2, \cdots, C_{\tau+1}), (R_1, R_2, \cdots, R_{\tau+1})\Big]$ is no more than

$$\begin{aligned} &\Delta\left[(C_1, C_2, \cdots, C_\tau), (R_1, R_2, \cdots, R_\tau)\right] + \mathop{\mathbb{E}}_{C_1, \ldots, C_\tau}\Big[\Delta\left[C_{\tau+1}, H_{\tau+1}\right]\Big] \\ &\qquad \leq \tau \cdot (2 \sqrt[3]{\epsilon_{ext}}) + \mathop{\mathbb{E}}_{C_1, \ldots, C_\tau}\Big[\Delta\left[C_{\tau+1}, H_{\tau+1}\right]\Big] \quad \text{(by induction.)} \end{aligned}$$

As for the expectation $\mathbb{E}_{C_1,\ldots,C_\tau}\left[\varDelta\left[C_{\tau+1}, H_{\tau+1}\right]\right]$, this may be expanded

$$\begin{aligned}
&\leq \Pr[(C_1,\ldots,C_\tau)\in G_s^\tau]\cdot\mathbb{E}\left[\varDelta\left[C_{\tau+1},H_{\tau+1}\right]\mid(C_1,\ldots,C_\tau)\in G_s^\tau\right]\\
&\quad+\Pr[(C_1,\ldots,C_\tau)\in B_s^\tau]\cdot\mathbb{E}[\varDelta\left[C_{\tau+1},H_{\tau+1}\right]\mid(C_1,\ldots,C_\tau)\in G_s^\tau]\\
&\leq \mathbb{E}[\varDelta\left[C_{\tau+1},H_{\tau+1}\right]\mid(C_1,\ldots,C_\tau)\in G_s^\tau]+\Pr[(C_1,\ldots,C_\tau)\in B_s^\tau]\\
&\leq \sqrt[3]{\epsilon_{ext}}+\sqrt[3]{\epsilon_{ext}}\,,
\end{aligned}$$

as s is good. We can conclude that for a good seed s,

$$\varDelta\left[\left(C_1,C_2,\cdots,C_\tau\right),\left(R_1,R_2,\cdots,R_\tau\right)\right]\leq\tau\cdot\left(2\sqrt[3]{\epsilon_{ext}}\right),$$

for any $\tau\leq\ell$. The total statistical distance is now given by

$$\begin{aligned}
&\varDelta\left[\left(C_1,C_2,\cdots,C_\ell\right),\left(R_1,R_2,\cdots,R_\ell\right)\right]\\
&\quad=\varDelta\left[\left(C_1,C_2,\cdots,C_\ell\right),\left(R_1,R_2,\cdots,R_\ell\right)\right]|_{s\ good}\cdot\Pr[s\ good]+\\
&\qquad\varDelta\left[\left(C_1,C_2,\cdots,C_\ell\right),\left(R_1,R_2,\cdots,R_\ell\right)\right]|_{s\ \mathrm{not}\ good}\cdot\Pr[s\ \mathrm{not}\ good]\\
&\quad\leq\ell\cdot\left(2\sqrt[3]{\epsilon_{ext}}\right)\cdot1+1\cdot\left(\ell\sqrt[3]{\epsilon_{ext}}\right)\leq3\ell\sqrt[3]{\epsilon_{ext}}\leq\epsilon\,.
\end{aligned}$$

The last inequality follows from the inequality $\epsilon_{ext}\leq\left(\epsilon/(3\ell)\right)^3$. We conclude that $\varDelta\left[SE(\kappa,\boldsymbol{m};\mathcal{O}),C_h^\lambda\right]\leq\epsilon$ and the theorem follows by the definition of insecurity.

3.3 Correctness

In this section we obtain an upper bound on the soundness of our stegosystem. We focus on the mapping between $\{0,1\}^n$ and $\varSigma^\lambda$ determined by the SE procedure of the one-time stegosystem. We would like to bound the probability of the stego decoding procedure's inability to faithfully recover the encoded message.

Theorem 3. *For any $\epsilon,\delta>0$, message $\boldsymbol{m}\in\{0,1\}^n$, consider the stegosystem of Section 3.1 under the parameter constraints $\epsilon_{ext}\leq\left(\frac{\epsilon}{6\ell^2}\right)^3$ and $\rho\geq2n^c\log(3\ell\epsilon^{-1})$ for some constant $c>1$. Then it holds that the stegosystem (SK,SE,SD) is (ϵ,δ)-correct where ϵ_{ext} is the extractor error and $\ell=\lceil n/c\log n\rceil$ for some constant $c>1$.*

Proof. Recall that the first step of the procedure SE is to randomize the message $\boldsymbol{m}$ to get $\boldsymbol{m'}=\boldsymbol{m}\oplus\kappa^{\mathrm{otp}}$. SE then proceeds to parse $\boldsymbol{m'}$ into blocks: $\boldsymbol{m'}=\boldsymbol{m'_1m'_2}\ldots\boldsymbol{m'_\ell}$, $\ell=\lceil n/c\log n\rceil$. Let F be the event that SD is unable to correctly decode the message encoded by SE. We seek to upper bound the probability of F. We proceed to first estimate the probability of failure for one message block $\boldsymbol{m_i}$. Let us denote this event by F'. We reuse the notations and definitions introduced in the security proof of the section above. Recall that we pick a seed $s\in_R\{0,1\}^d$ for the extractor we use in our construction and let ϵ_{ext} denote the error of the extractor. As discussed in the security proof, we say that a seed s is *good* if $\forall\tau,\ \mu\left(G_s^\tau\right)\geq1-\sqrt[3]{\epsilon_{ext}}$, $\tau=1,2,\cdots,\ell$. We showed in the security proof that the probability of seed s to be good is given by $\Pr_s\left[\forall\tau\mid\mu\left(G_s^\tau\right)\geq1-\sqrt[3]{\epsilon_{ext}}\right]\geq$

$1 - \ell\sqrt[3]{\epsilon_{ext}}$. This implies that the probability of seed s to be not good is no more than $\ell\sqrt[3]{\epsilon_{ext}}$. This yields

$$\begin{aligned}\Pr[F] &= \ell \cdot (\Pr[F' \mid s \text{ good}] \cdot \Pr[s \text{ good}] + \Pr[F' \mid s \text{ not good}] \cdot \Pr[s \text{ not good}]) \\ &\leq \ell \cdot (\Pr[F' \mid s \text{ good}] \cdot 1 + 1 \cdot (\ell\sqrt[3]{\epsilon_{ext}}))\,.\end{aligned}$$

We proceed to bound $\Pr[F' \mid s \text{ is good}]$. We know that when the seed s is good, for no more than $\sqrt[3]{\epsilon_{ext}}$ fraction of distributions in every level $\tau = 1, 2, \cdots, \ell$, the extractor coupled with the seed s is not a $\sqrt[3]{\epsilon_{ext}}$-biased function with probability no more than $\sqrt[3]{\epsilon_{ext}^2}$. So, we get

$$\Pr[F' \mid s \text{ good}] \leq 1 \cdot \left(1 - \left(\frac{1}{2^{|\boldsymbol{m}_i|}} - \sqrt[3]{\epsilon_{ext}}\right)\right)^{\rho} + \sqrt[3]{\epsilon_{ext}} \cdot 1 + \sqrt[3]{\epsilon_{ext}^2} \cdot 1$$

where ρ is the bound on the number of iterations performed by the rejection sampling procedure. Setting $\epsilon_{ext} \leq 1/(8 \cdot 2^{3|\boldsymbol{m}_i|}) = 1/(8 \cdot n^{3c})$ and $\rho = 2 \cdot 2^{|\boldsymbol{m}_i|} \cdot \log(3\ell\epsilon^{-1}) = 2n^c \log(3\ell\epsilon^{-1})$ (since in our construction $|\boldsymbol{m_i}| = c\log n$, and as ρ is exponential in the block length, we choose the message block length to be $c\log n$), we have $\Pr[F' \mid s \text{ good}] \leq \epsilon/(3\ell) + 2\sqrt[3]{\epsilon_{ext}}$. From the statement of the theorem we have that $\epsilon_{ext} \leq \left(\frac{\epsilon}{6\ell^2}\right)^3$, we can bound $\Pr[F]$ as

$$\Pr[F] \leq \ell \cdot (\Pr[F' \mid s \text{ good}] \cdot 1 + 1 \cdot (\ell\sqrt[3]{\epsilon_{ext}})) \leq \epsilon$$

and the statement of the theorem follows.

We record the security and correctness theorem below.

Theorem 4. *For any $\epsilon_{ext} \leq 1/8n^{3c}, \delta > 0$, $\rho \geq 2n^c \log(\epsilon_{ext}^{-1/3})$ and message $\boldsymbol{m} \in \{0,1\}^n$, the stegosystem (SK, SE, SD) of Section 3.1 is (ϵ_{cor}, δ)-correct and (ϵ_{sec}, δ)-secure, where $\epsilon_{cor} \leq 4\ell^2\sqrt[3]{\epsilon_{ext}}$ and $\epsilon_{sec} \leq 3\ell\sqrt[3]{\epsilon_{ext}}$. Here, ϵ_{ext} is the extractor error and $\ell = \lceil n/c\log n\rceil$ for some constant $c > 1$.*

3.4 Putting It All Together

The objective of this section is to integrate the results of the previous sections of the paper. We first show that our steganography protocol embeds a message of length n bits using a shared secret key of length $(1 + o(1))n$ bits while achieving security $2^{-n/\log^{O(1)} n}$. In this sense, our protocol is randomness efficient in the shared key. We next show that by permitting a portion of the shared secret key to be *public* while retaining n private key bits, we can achieve security of 2^{-n}. Let us first start our discussion by considering the parameters of the extractor construction we employ in our protocol.

Extractor Parameters. Recall that π is the parameter that dictates how many copies of the channel Alice decides to use in order to transform the channel C with min-entropy δ into a channel C^π with min-entropy $\pi\delta$.

If we let $\pi = \delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1))$ for some constant $c > 1$, the channel distribution C_h^{π} supported on $\{0,1\}^{\delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1)) \cdot b}$ has a min-entropy of at least $t = c \log n + 2 \log(1/\epsilon_{ext}) + O(1)$. To put this all together, the *RRV* strong-extractor is a function $\mathrm{Ext} : \{0,1\}^{\nu} \times \{0,1\}^{d} \to \{0,1\}^{t-\Delta}$ where

$$\begin{aligned}
\nu &= \delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1)) \cdot b \\
d &= O\left(\log^2\left(\delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1)) \cdot b\right) \cdot \log(1/\epsilon_{ext}) \cdot \log t\right) \\
t &= c \log n + 2 \log(1/\epsilon_{ext}) + O(1) \\
\Delta &= 2 \log(1/\epsilon_{ext}) + O(1) \text{ and} \\
t - \Delta &= c \log n
\end{aligned}$$

We can immediately see from the preceding discussion that our stegotext is of length

$$\frac{n}{c \log n} \cdot \delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1)) \cdot b = \frac{n}{\delta}\left(1 + \frac{2 \log(1/\epsilon_{ext})}{c \log n} + o(1)\right) \cdot b$$

bits to embed n bits of message.

Key-Length Efficiency. Recall that the shared secret key between Alice and Bob is comprised of the one-time pad $\kappa^{\mathrm{otp}} \in_R \{0,1\}^n$ of length n and the extractor seed $s \in_R \{0,1\}^d$ of length d bits, i.e., $\kappa = (\kappa^{\mathrm{otp}}, s)$. Also, the length of the seed from the above discussion is given by

$$d = O\left(\log^2\left(\delta^{-1} \cdot (c \log n + 2 \log(1/\epsilon_{ext}) + O(1)) \cdot b\right) \cdot \log(1/\epsilon_{ext}) \cdot \log t\right) .$$

Notice the relationship between the error of the extractor ϵ_{ext} and the desired security from our stegosystem ϵ is given by $\epsilon_{ext} \leq \left(\frac{\epsilon}{3\ell}\right)^3$ from Theorem 2. When we let $\epsilon = 2^{-n/\log^{O(1)} n}$, we can see that the length of the seed $d = o(n)$. Thus we can embed a message of length n bits using a shared secret key of length $(1 + o(1))n$ bits while achieving security $2^{-n/\log^{O(1)} n}$. Suppose, we were to let the extractor seed of length d be public, observe now that we can attain $\epsilon = 2^{-n}$ security in the length of the shared private key of length n. The seed length can now be given by $d = O(n \log n \log^2(\delta^{-1} bn))$. For small ϵ, the relationship between the seed length d and security ϵ can be given by $d = O\left(\log^3\left(\log\left(\epsilon^{-3}\right)\right) \log\left(\epsilon^{-3}\right)\right)$. We would like to note that our protocol offers a non-trivial improvement over the protocol offered by Kiayias et al. [5] as in their protocol, they need $O(n)$ secret bits regardless of the security achieved.

Also, when we elect to make use of the public randomness for the d bits for the extractor seed, we obtain constant overhead as well. In particular, the length of the shared secret key is equal to the length of the message, n bits while attaining 2^{-n} security.

In this context of making the seed of the extractor public, we would like to explain our model and clarify the implications of making the seed public. In our model for steganography, we assume that the communication channel is not adversarially controlled. In particular, the adversary is not allowed to reconfigure the channel distributions once the seed has been made public. In this sense, the channel is chosen and fixed first, then a seed s is chosen uniformly at random and made public. In other words, we require that the randomness in the seed s is independent of the channel. Indeed, in a stronger model where the adversary does have the ability to readapt the channel distributions, we would need to keep the seed private. From our above discussion, we can see that our stegosytem of Section 3.1 is still (ϵ, δ)-correct and (ϵ, δ)-secure when the seed s is public.

Theorem 5. *For any $\epsilon, \delta > 0$, message $\boldsymbol{m} \in \{0,1\}^n$ consider the stegosystem of Section 3.1 under the parameter constraints $\epsilon_{ext} \leq \left(\frac{\epsilon}{6\ell^2}\right)^3$ and $\rho \geq 2n^c \log(3\ell\epsilon^{-1})$ for some constant $c > 1$. Then for every channel, if the key $\kappa^{otp} \in_R \{0,1\}^n$ is private and the seed $s \in_R \{0,1\}^n$ is public, then it holds that the stegosystem is (ϵ, δ)-correct and (ϵ, δ)-secure. Here, ϵ_{ext} is the extractor error and $\ell = \lceil n/c \log n \rceil$ for some constant $c > 1$. The stegosystem exhibits $O(1)$ overhead, the length of the shared private key is equal to the length of the message.*

4 A Provably Secure Stegosystem for Longer Messages

In this section we show how to apply the "one-time" stegosystem of Section 3.1 together with a pseudorandom generator so that longer messages can be transmitted as shown by Kiayias et al. [5].

Definition 10. *Let U_k denote the uniform distribution over $\{0,1\}^k$. A polynomial time deterministic algorithm G is a pseudorandom generator (PRG) if the following conditions are satisfied:*

Variable output. *For all seeds $x \in \{0,1\}^*$ and $y \in \mathbb{N}$, $|G(x, 1^y)| = y$.*
Pseudorandomness. *For every polynomial p the set of random variables $\{G(U_k, 1^{p(k)})\}_{k \in \mathbb{N}}$ is computationally indistinguishable from the uniform distribution $\{U_{p(k)}\}_{k \in \mathbb{N}}$.*

For a PRG G and $0 < k < k'$, if A is some statistical test, we define the advantage of A over the PRG as follows:

$$\mathbf{Adv}_G^A(k, k') = \left| \Pr_{w \leftarrow G(U_k, 1^{k'})}[A(w) = 1] - \Pr_{w \leftarrow U_{k'}}[A(w) = 1] \right|.$$

The insecurity of the above PRG G against all statistical tests A computable by circuits of size $\leq P$ is then defined as $\mathbf{InSec}_G(k, k'; P) = \max_{A \in \mathcal{A}_P}\{\mathbf{Adv}_G^A(k, k')\}$ where $\mathcal{A}_P$ is the collection of statistical tests computable by circuits of size $\leq P$.

It is convenient for our application that typical PRGs have a procedure G' such that if $z = G(x, 1^y)$, it holds that $G(x, 1^{y+y'}) = G'(x, z, 1^{y'})$ (i.e., if one maintains z, one can extract the y' bits that follow the first y bits without starting from the beginning).

Consider now the following stegosystem $S' = (SK', SE', SD')$ that can be used for steganographic transmission of longer messages using the one-time stegosystem $S = (SK, SE, SD)$ as defined in Section 3.1. S' can handle messages of length polynomial in the security parameter k and employs a PRG G. The two players Alice and Bob, share a key of length k denoted by x. The function SE' is given input x and the message $m \in \{0,1\}^{\nu}$ to be transmitted of length $\nu = p(k)$ for some fixed polynomial p. SE' in turn employs the PRG G to extract k' bits (it computes $\kappa = G(x, 1^{k'})$, $|\kappa| = k'$). The length k' is selected to match the number of key bits that are required to transmit the message m using the one-time stegosystem of Section 3.1. Once the key κ of length k' is produced by the PRG, the procedure SE' invokes the one-time stegosystem on input κ, m, h. The function SD' is defined in a straightforward way based on SD.

The computational insecurity of the stegosystem S' is defined by adapting the definition of information theoretic stegosystem security from Section 2.3 for the computationally bounded adversary as follows: $\mathbf{InSec}_{S'}(k, k'; P) = \max_{\mathcal{A} \in \mathcal{A}_P} \{\mathbf{Adv}^{\mathcal{A}}_{S'}(k, k')\}$, this maximum taken over all adversaries $\mathcal{A}$, where SA_1 and SA_2 have circuit size $\leq P$ and the definition of advantage $\mathbf{Adv}^{\mathcal{A}}_{S'}(k, k')$ is obtained by suitably modifying the definition of $\mathbf{Adv}^{\mathcal{A}}_{S}(k)$ in Section 2.3. In particular, we define a new adversarial game $G^{\mathcal{A}}(1^k, 1^{k'})$ which proceeds as the previous game $G^{\mathcal{A}}(1^k)$ in Section 2.3 except that in this new game $G^{\mathcal{A}}(1^k, 1^{k'})$, algorithms SA_1 and SA_2 receive as input the security parameter k' and SE' invokes SE as $SE(\kappa, m^*; \mathcal{O})$ where $\kappa = G(x, 1^{k'})$.

Theorem 6. *The stegosystem $S' = (SK', SE', SD')$ is provably secure in the model of [2] (steganographically secret against chosen hiddentext attacks); in particular employing a PRG G to transmit a message m we get*

$$\mathbf{InSec}_{S'}(k, k'; P) \leq \mathbf{InSec}_{G}(k, k'; P) + \mathbf{InSec}_{S'}(k')$$

where $\mathbf{InSec}_{S'}(k')$ is the information theoretic insecurity defined in Section 2.3 and $|m| = \ell(k')$.

References

[1] Cachin, C.: An Information-Theoretic Model for Steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)

[2] Hopper, N.J., Langford, J., von Ahn, L.: Provably Secure Steganography. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 77–92. Springer, Heidelberg (2002)

[3] Hopper, N.J., von Ahn, L., Langford, J.: Provably secure steganography. IEEE Trans. Computers 58(5), 662–676 (2009)

[4] Kiayias, A., Raekow, Y., Russell, A.: Efficient Steganography with Provable Security Guarantees. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 118–130. Springer, Heidelberg (2005)

[5] Kiayias, A., Raekow, Y., Russell, A., Shashidhar, N.: Efficient steganography with provable security guarantees. Preprint, arXiv:0909.3658 (September 2009)

[6] Mittelholzer, T.: An Information-theoretic Approach to Steganography and Watermarking. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 1–16. Springer, Heidelberg (2000)

[7] Nisan, N.: Extracting randomness: How and why: A survey. In: Proceedings of the 11th Annual IEEE Conference on Computational Complexity, pp. 44–58. Citeseer (1996)

[8] Nisan, N., Ta-Shma, A.: Extracting randomness: A survey and new constructions. Journal of Computer and System Sciences 58(1), 148–173 (1999)

[9] Nisan, N., Zuckerman, D.: Randomness is linear in space. Journal of Computer and System Sciences 58, 43–52 (1993)

[10] Radhakrishnan, J., Ta-Shma, A.: Bounds for dispersers, extractors, and depth-two. SIAM Journal on Discrete Mathematics 13 (2000)

[11] Raz, R., Reingold, O., Vadhan, S.: Extracting all the randomness and reducing the error in trevisan's extractors. In: Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pp. 149–158 (1999)

[12] Reingold, O., Shaltiel, R., Wigderson, A.: Extracting randomness via repeated condensing. In: Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, pp. 22–31 (2000)

[13] Shaltiel, R.: Recent developments in explicit constructions of extractors. Bulletin of the EATCS 77, 67–95 (2002)

[14] Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press, New York (2005) ISBN 0-5218-5154-8

[15] Simmons, G.J.: The prisoners' problem and the subliminal channel. In: CRYPTO, pp. 51–67 (1983)

[16] von Ahn, L., Hopper, N.J.: Public-Key Steganography. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 323–341. Springer, Heidelberg (2004)

[17] Zöllner, J., Federrath, H., Klimant, H., Pfitzmann, A., Piotraschke, R., Westfeld, A., Wicke, G., Wolf, G.: Modeling the Security of Steganographic Systems. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 344–354. Springer, Heidelberg (1998)

A Omitted Proofs

Proof (Lemma 1). For $x \in \mathcal{X}$ denote $\Pr[X = x]$ by P_x and $\Pr[Y_x = y]$ by $P_{y|x}$. Define P'_x and $P'_{y|x}$ similarly. Then we may expand $\varDelta\left[(X,Y),(X',Y')\right]$ as

$$
\begin{aligned}
&\frac{1}{2}\sum_{x\in\mathcal{X},y\in\mathcal{Y}}\left|P_x\cdot P_{y|x}-P'_x\cdot P'_{y|x}\right|\\
\le&\frac{1}{2}\sum_{x,y}\left|P_x\cdot P_{y|x}-P_x\cdot P'_{y|x}\right|+\frac{1}{2}\sum_{x,y}\left|P_x\cdot P'_{y|x}-P'_x\cdot P'_{y|x}\right|\\
=&\frac{1}{2}\sum_{x,y}P_x\cdot\left|P_{y|x}-P'_{y|x}\right|+\frac{1}{2}\sum_{x,y}P'_{y|x}\cdot\left|P_x-P'_x\right|\le\mathop{\mathbb{E}}_X\left[\varDelta\left[Y_X,Y'_X\right]\right]+\varDelta\left[X,X'\right].
\end{aligned}
$$

Secret Agent Radio: Covert Communication through Dirty Constellations

Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker

University of Colorado Boulder
Boulder, CO 80309-0430 USA
{Aveek.Dutta,Dola Saha,
Dirk.Grunwald,Douglas.Sicker}@colorado.edu

Abstract. In this paper we propose a novel approach to implement high capacity, covert channel by encoding covert information in the physical layer of common wireless communication protocols. We call our technique Dirty Constellation because we hide the covert messages within a “dirty” constellation that mimics noise commonly imposed by hardware imperfections and channel conditions. The cover traffic in this method is the baseband modulation constellation. We leverage the variability in the wireless channel and hardware conditions to encode the covert channel. Packet sharing techniques and pre-distortion of the modulated symbols of a decoy packet allows the transmission of a secondary covert message while making it statistically undetectable to an adversary. We demonstrate the technique by implementing it in hardware, on top of an 802.11a/g PHY layer, using a software defined radio and analyze the undetectability of the scheme through a variety of common radio measurements and statistical tests.

1 Introduction

There are many times when communication needs to be secure. Common and obvious examples include providing security for electronic commerce or privacy for personal matters. At other times, communication must also be *covert*, or undetectable which has a *low probability of intercept* (LPI) or a *low probability of detection* (LPD). LPD communication mechanisms are useful when the very act of communication can raise concerns, such as communication during war-time or during surveillance. Usually it is difficult to detect the receiver of communication mechanisms that exploit the characteristics of radio propagation.

In this paper, we explore methods that provide LPD and LPI for high-bandwidth networks. Our method provides a high-bandwidth covert side-channel between multiple radios using a common wireless network, as indicated in Figure 1. The method is covert because the devices (laptops or smartphones) function as normal devices. Again, the devices “hide in plain sight”. Rather than raising suspicions by exchanging encrypted messages with each other or some centralized server, they appear to be conducting normal network communication (browsing web pages, sending mail, streaming multimedia) when in reality, they are able to communicate undetected. The adversary will face great challenge in discovering the side channel because the covert channel is being transmitted by mobile nodes. Monitoring to locate such nodes would require

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 160–175, 2013.

significant investment or infrastructure, such as monitoring in every coffee shop, bus or public venue where people may be near each other.

The technique uses a common, physical-layer protocol to mask the communication that takes advantage of the hardware imperfections present in commodity hardware, intrinsically noisy channel of wireless communication and receiver diversity. We have implemented this mechanism using software-defined radios, operating in 2.4GHz ISM band, but can also be easily extended to TV whitespaces. Our prototype uses an OFDM waveform. Most consumer electronic devices use OFDM waveforms for high-bandwidth networks (including DVB, DAB, WiFi, WiMAX and LTE), and there are some benefits in "hiding" in such a ubiquitous waveform.

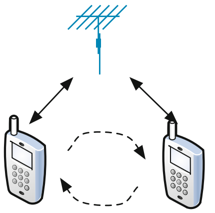
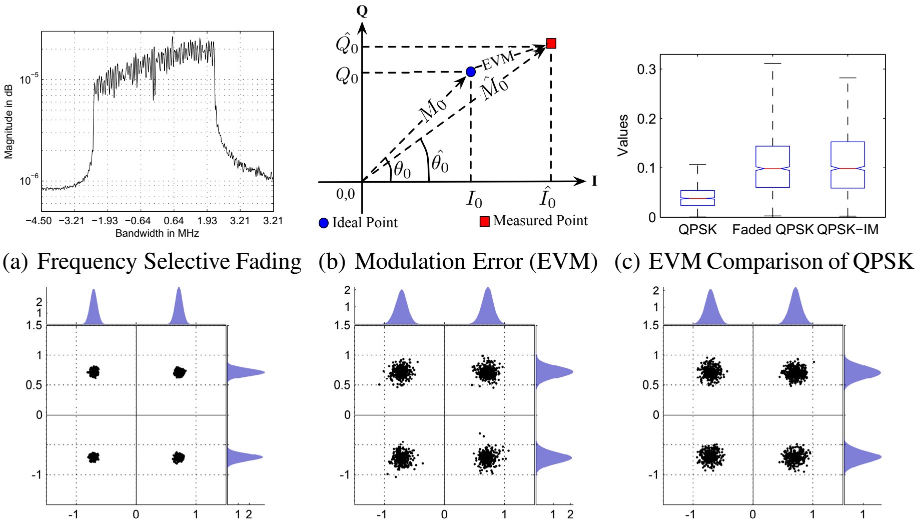


Fig. 1. Undetected Side-Channel Communication

Imperfections in off-the-shelf Network Interface Cards (NICs) [4], coupled with an additive random wireless channel cause the signal to degrade over time and distance. To mask our communication, we "pre-distort" the signal to mimic the normal imperfection of the hardware and Gaussian distortion arising from the channel. This distortion appears as noise to the unobservant receiver, be it the Wi-Fi access point or an adversary. However, a receiver aware of the presence of the signal and its encoding technique can decode the "noise" to reveal the hidden message.

Our motivation for hiding the data in physical layer (analog waveform domain) of common wired and wireless protocols are the following:

- *Hide in Plain Sight* - Using the physical properties of the transmission medium will allow the covert channel to resemble a common waveform, only distorted by channel noise, or transmitted by a NIC with imperfections.
- *Access to Covert Channel* - Since the covert channel uses the signal waveform, an adversary is easily abstracted from the covert channel, as opposed to other packet level techniques using higher layers [11]. In our method, the bits of the cover packet are not altered and hence the presence of the covert message is not detected at higher layers, or more specifically in digital domain.
- *Sample Collection* - The ubiquitous nature of wireless devices and their localized transmission make it difficult to detect the presence of a covert channel. As opposed to digital contents on the Internet (music, picture, video), which can be accessed from one physical location, acquiring signal waveforms requires hauling expensive, bulky equipment (signal analyzers) to every possible hotspot.
- *Search Complexity* - A $500 byte$ packet, modulated with QPSK-$1/2$ rate coding, results in $\approx 19KB$ (calculation omitted due to space constraints) of I/Q information. This increases the search space by $\approx$ 38 times, compared to packet level analysis of a covert channel.
- *Statistically Undetectable* - In higher layer techniques, an adversary can search the header fields (known as unused fields) of a packet stream and find the covert channel [3], whereas in physical layer, the adversary needs to perform several statistical tests on the I/Q samples, which are already tainted by time varying channel noise.
- *Capacity* - Compared to conventional techniques using higher layers, where only a few unused bits of any header field of a packet is used, our technique can easily utilize 10% of the cover signal to transmit covert messages.



(a) Frequency Selective Fading (b) Modulation Error (EVM) (c) EVM Comparison of QPSK

(d) Ideal QPSK (e) Faded QPSK (f) QPSK-IM

Fig. 2. Characterizing channel and hardware impairments with three waveforms: ideal QPSK, faded QPSK and "impaired," QPSK-IM. The QPSK-IM signal is indistinguishable from QPSK-faded signal using statistical measures.

These advantages coupled with relative ease of implementation using now popularized software defined radio, makes this technique extremely useful in providing high capacity covert channels.

2 Characterizing OFDM Signals

Signal quality in wireless channel depends primarily on two factors: channel impairments and hardware impairments. Channel impairments typically range from additive white noise to frequency selective fading and/or hidden terminal and Doppler shifts, which degrade signal properties in time and frequency domain. Figure 2(a) plots the spectrum for an OFDM waveform from a bench measurement that is skewed because of a frequency selective fading in the left-most subcarriers. Similarly, impairments due to various non-linearities in the transceiver pipeline are often reflected in the signal characteristics as well. Since these types of impairments are hardly deterministic, estimating the errors and compensating for them is a non-trivial task.

Signal-to-Noise Ratio (SNR) is a widely used metric, often measured in the time or frequency domain using averaged power measurements. A simple interpretation of the SNR is *"the higher the SNR, the higher the probability that the information can be extracted with acceptable error performance"*. However, high spatial-decorrelation of the wireless channel may render portions of the OFDM signal undecodable even though a high "average" SNR indicates otherwise. Figure 2(a) is an example of an OFDM spectrum of an ongoing communication that has an average SNR of 21dB but degraded in the frequency domain.

The Error Vector Magnitude (EVM), shown in Figure 2(b) is another metric that measures the deviation of the complex modulation vectors in the I/Q-plane from the ideal position. A bad channel leads to higher dispersion of these vectors and hence higher EVM, which affects the error performance as well. Modulation errors can also be introduced as imperfections in the transceiver hardware itself, which can cause the intended I/Q sample to be transmitted (or received) at a slight offset. In the IEEE 802.11a/g standard [9], this modulation error at the transmitter for a QPSK modulation is mandated to be no more than $10dB$ from an "ideal" I/Q mapping.

Figure 2(c) shows the distribution of EVM (in a boxplot) for three bench measurements of an OFDM waveform using QPSK modulation *where each of the transmissions have the same SNR*. The first measurement is based on an "ideal" transmission with low noise resulting in a low EVM with minimal variance, called ideal QPSK. The second measurement, faded QPSK, from a bench measurement with slightly different antenna orientation, has higher average EVM and wider variance. The difference between ideal QPSK and faded QPSK are due to multipath effects. The last measurement, termed the "impaired QPSK" or QPSK-IM signal, was recorded from a transmitter that predistorted the signal such that the average EVM is 10dB worse than the ideal. On the surface, the QPSK-IM signal appears to have similar properties to faded QPSK – both have higher average EVM and wider variance. Figures 2(d)-2(f) show the three constellations corresponding to the measurements described above. It is indeterminable whether the deterioration in the EVM is due to intentionally introduced noise at the transmitter, or due to imperfections in the hardware that is operating within tolerable limits, or is the result of poor channel quality.

From these examples, it is evident that impairments, whether in the channel or in the hardware, will cause statistical variation in the perceived value of the metrics and that the bounds on these metrics are only loosely defined and can only be formalized by various descriptive statistics and statistical tests.

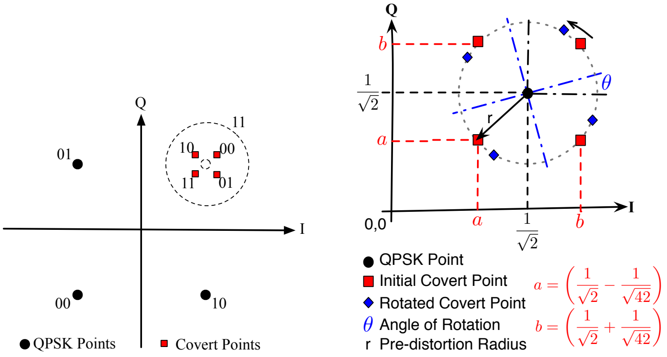
3 Dirty Constellation

Our method relies on being able to embed one message in another in the wireless channel, but goes well beyond that to then insure that the covert message is undetectable. There are several ways to embed messages by encoding the constellation symbols using bits of two distinct messages [13,7] but we use a simpler technique that uses existing modulation methods of OFDM.

Using a combination of adaptive modulation and efficient packet sharing using joint constellations we encode the covert channel. If a receiver is aware of our irregular mapping of bits, and it has sufficient SNR for that subcarrier, it is able to decode the covert message while to an uninformed user, the covert constellation points will be treated as random dispersed sample of a low-rate modulation, that reveals an innocuous message.

The key to such covert communication using the physical layer of an OFDM based wireless protocol are four fold: **1)** packets containing covert data must be indistinguishable from non-covert packets to all uninformed observers; **2)** the presence of any irregularity in the covert packets has to be kept hidden under rigorous *statistical tests* on the signal; **3)** the covert channel should be non-trivial to replicate, making it secure from spoofing and impersonation; and finally, **4)** it should have high capacity. In this paper we satisfy each of these requirements through a set of techniques.

Requirement 1: Identifying a Covert Channel: Our technique relies on encoding "cover packets" that are transmitted at a low rate (BPSK or QPSK) with supplemental information that can be decoded as an additional QPSK signal by an informed receiver. In the examples below, we use QPSK for both the cover and covert channel.



(a) Cover QPSK constellation (b) Covert constellation

Fig. 3. Encoding Dirty Constellation

In a QPSK encoding, the constellation points encode two bits of information as shown in Figure 3(a). To encode the covert channel, we deflect the placement of the QPSK points. This is similar to having a "covert QPSK" encoding with an origin around the ideal QPSK constellation points of the cover traffic. Figure 3(b) corresponds to the upper right quadrant of the cover QPSK constellation shown in Figure 3(a). To modulate a subcarrier carrying both the cover and covert message, first the cover constellation point (QPSK) is chosen (as per the cover message stream), specifying the quadrant, followed by re-mapping that point to one of the four "covert-QPSK" points around the "cover QPSK" point.

Clearly, the goal is to leave the cover message decodable by standard receivers. Only the covert receiver aware of the joint constellation will decode the subcarriers properly and extract the *two* covert bits to form the hidden packet. An adversary will decode at the base rate or the rate for cover message, as specified in the *signal symbol* of the packet; while the covert points will be treated as noisy points. The cover message could be intended for an access point (as part of a web browsing session) while the covert message can be overheard and decoded by a nearby radio. In this way we implement a covert channel while making it appear as completely innocuous to other users receiving the same transmission.

Requirement 2: Low Probability of Detection: How would an adversary detect such communication? As long as the packet can be decoded, a legacy receiver has no way of knowing how signals are being encoded at the core of the physical layer, because conventional packet decoding is performed by identifying the data rates embedded at the beginning of the packet which will always contain the base rate (QPSK) information. However, adversaries using measurement equipment like vector-signal analyzers or software defined radios can extract the digital samples from the radio pipeline at different stages of the signal processing. Therefore, our ultimate goal is to provide very low probability of detection not only at the packet level but also at the signal level.

One simple form of analysis is to look at the equalized I/Q vectors of the jointly encoded packet. The presence of the covert constellation at regular interval will appear as distinct point clouds that will set themselves apart from the cover QPSK point cloud and will reveal the presence of the covert channel, as shown in Figure 4.

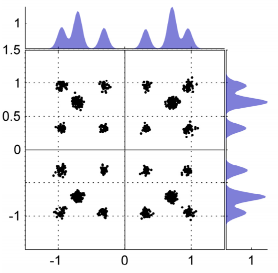


Fig. 4. Constellation without random pre-distortion of the QPSK points and using existing 16QAM points to map the joint covert constellations

We solve this problem by changing the I/Q vectors of the covert transmitter in three steps:

Step1: We bring the covert constellation points closer to the ideal QPSK point and re-map the covert constellation points symmetrically around the QPSK points, with a mutual separation of $\frac{2}{\sqrt{42}}$, a distance equal to that of a 64QAM constellation, so that a covert receiver can operate within the operating range of a WiFi receiver.

Step2: We randomize the I/Q vectors of the covert QPSK points with a Gaussian distribution but limit their dispersion to a radius of $\sqrt{\frac{2}{42}}$ as shown in figure 3(b). We call this as the *pre-distortion circle*; pre-distortion of the QPSK signal at the transmitter ensures that the covert constellations are hidden in the cloud of a dispersed (noisy) QPSK point cloud. We introduce imperfections to the transmitted signal in such a way that the average EVM error is equal to or less than 10dB compared to the ideal QPSK constellation points, which is within the limits of hardware anomaly allowed in the IEEE 802.11 standard [9]. Thus, it cannot be ascertained with certainty if the EVM error is due to hardware impairments, channel impairments or intentionally injected distortion.

Step3: To accommodate a higher rate covert channel, *e.g.*, when 50% of the OFDM subcarriers are covert, then at high SNR there is always a finite probability that the covert constellations are visible. To have the covert symbols blend with the pre-distorted QPSK point cloud, the covert symbols are rotated along the circumference of the pre-distortion circle for every subcarrier that is mapped to a covert constellation as shown in Figure 3(b). The rotation is performed using a monotonically increasing angle θ; the transmitter and receiver both start with $\theta = 0°$ at the start of the packet and increment θ for each covert subcarrier. In our implementation we use a $15°$ counter-clockwise rotation for the covert points.

These 3 steps allow us to hide the covert channel, even when an adversary has access to the I/Q samples of the packet. The adversary will interpret the point cloud as a noisy version of a valid (albeit noisy) QPSK constellation and would not suspect the presence of a covert communication. This compound constellation involving a covert channel hidden within a cover constellation is termed a *"Dirty Constellation"*. However, in order to avoid raising suspicion by any RF fingerprinting algorithms [4], a QPSK-IM waveform should *always* be used for non-covert transmissions, to avoid sudden changes in the modulation characteristics.

Requirement 3&4: Security and Higher Efficiency: These requirements are considered as an enhancement to the basic scheme of Dirty Constellation. We have implemented 10%, 30% and 50% encoding of subcarriers, as shown in Figure 7, yielding up to $9Mbps$ datarate with QPSK modulation and $3/4$ encoding rate. Using higher modulation constellation, *e.g.*, 256-QAM, we can further increase the capacity of the covert channel by encoding more bits per subcarrier. Due to space constraints we leave this as future work. Finally, we discuss the security aspect in §7.

4 Dirty Constellation on SDR

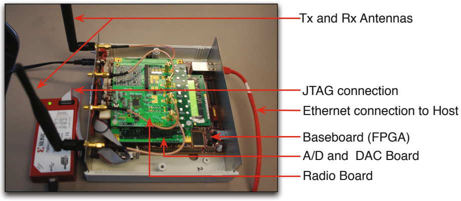
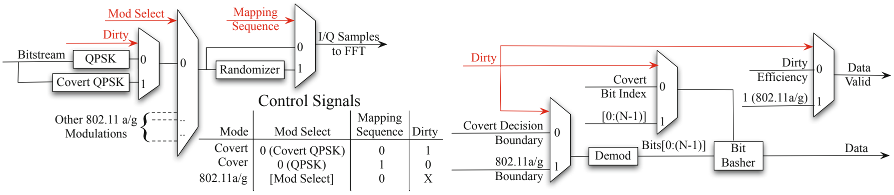


Fig. 5. SDR prototype using Virtex-V FPGA

Hiding a message in a randomized modulation constellation requires a programmable modulator and demodulator. Conventional radio 802.11 PHYs modulate all the subcarriers with one type of pre-defined modulation, For this scheme to work, we used a FPGA-based software defined radio platform based on our previous work [6,5], as shown in figure 5, and modified the modulator and demodulator to program each subcarrier with different modulations, adding either noise or covert constellations. Figure 6(a) shows the functional diagram of the programmable modulator. The notable parameters in the design are the *dirty* bit and the *mapping sequence* bit which are used to select the appropriate mapping for covert joint constellations and randomize (Gaussian) the cover symbols to engulf the higher order modulation points. The cover and the covert bits are independently packetized as per the 802.11a/g specification and the covert joint symbols are formed by merging the bits of the two packets prior to sending it to the modulator. The merging of packets is performed in software and then fed to the hardware along with the control information to create the Dirty Constellation. The QPSK-IM constellation is generated by using the randomizer unit that emulates an overall modulation error of 10dB, by setting the *dirty* bit to '0' and *mapping sequence* to '1' for all subcarriers.

The decoder employs maximum likelihood decoding and uses pre-defined thresholds to decode the constellation. Figure 6(b) shows the functional diagram of the demodulator. First the covert receiver demodulates the signal using the covert decision boundaries, 64QAM in this case and then extracts the covert bits. Since all subcarriers do not contain the hidden message, the receiver then uses the pre-assigned mapping sequence and its rotation information to filter out the covert subcarriers' information to form the covert packet.



Control Signals

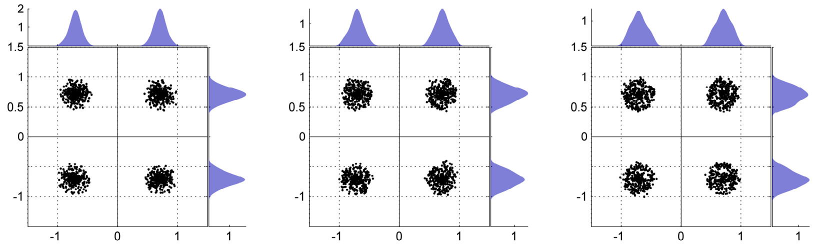
Mode	Mod Select	Mapping Sequence	Dirty
Covert	0 (Covert QPSK)	0	1
Cover	0 (QPSK)	1	0
802.11a/g	[Mod Select]	0	X

(a) Transmitter Pipeline

(b) Receiver Pipeline

Fig. 6. Mod/Demodulator for Dirty Constellation

Figure 7 shows an example of Dirty Constellation with varying frequency of the covert channel that has been transmitted by the SDR prototype and captured using a VSA. The I and Q histograms alongside the constellation shows the similarity of the distributions and that they are from the family of normal distributions.



(a) Frequency – 10% (b) Frequency – 30% (c) Frequency – 50%

Fig. 7. Examples of over-the-air transmission of Dirty Constellations with varying embedding frequency using the SDR prototype

5 Experiments and Measurements

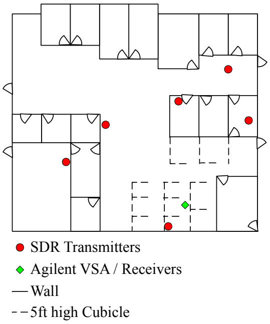


Fig. 8. Node placement

Using the hardware described in §4 as the transmitter, signal samples are collected in a lab/office environment. The transmitter nodes were placed as shown in the Figure 8. The signals were captured using a high-end Agilent vector signal analyzer (VSA) that provides the raw I/Q vectors of the packets transmitted by the SDR nodes. We record data from 6 locations, for ideal QPSK, QPSK-IM and Dirty Constellation with 10% covert channel efficiency. Each dataset contains measurement of 500 data packets of each type per transmit power level. The transmit power is varied in steps of 2.5dB such that the measured SNR at the VSA has a range of 7dB to 20dB. We have chosen this range because 7dB is the minimum SNR required to decode a QPSK packet with 98% packet reception rate. This has been empirically validated using bench measurements using our SDR transceivers. Likewise, 20dB was selected as the upper limit because the EVM doesn't decrease appreciably with higher SNR. After filtering out the required data range we find the average sample size is $10,000$ packets per type. We bin the packets by SNR in bins of size 1dB; each bin contains $500 - 800$ packets per SNR value. We perform all the statistical testing using this dataset which captures a wide range of SNR and channel conditions for all the type of modulations. In these measurements, the VSA is treated as both the covert receiver *and* a very aggressive adversary. As a covert receiver, the messages sent by the different transmitters can be received by the VSA receiver and the covert data can be extracted. As an adversary, the receiver has a high quality measurement device and also acts as the "most aggressive adversary" because it shares the same channel state as the receiver.

6 Analyzing Dirty Constellation

The core idea of testing a sample for adherence to a particular family of signals is performed by comparing test results with a known set of statistics for the same class. Therefore, the first step of the analysis process is to formalize the database of these statistics that characterizes an entire family of signals. In this paper, we intend to compare a Dirty Constellation with a QPSK waveform. We formulate the problem as a hypothesis test, with the null hypothesis:

> $\mathcal{H}_0$: Given a random sample from a Dirty Constellation packet, it is statistically same as any other QPSK packet.

Whereas the alternative hypothesis is:

> $\mathcal{H}_1$: Given a random sample from a Dirty Constellation packet, it can be statistically identified that it is not a QPSK packet.

In this section, we analyze whether the packets containing covert data can be distinguished from normal packets at the packet level or at the waveform level in the time and frequency domain. The test statistics of standard QPSK signals is lower bounded by the statistics of an “ideal QPSK” packet and upper bounded by a “QPSK-IM” packet. We used “QPSK-IM” packets to mimic a radio with hardware imperfections, but operating within the limits of IEEE 802.11 standard requirements. Each of these bounds have been empirically derived from the measurements collected as described in §5. If the Dirty Constellation sample is within the bounds set for that test then the null hypothesis is “not rejected”, meaning that the Dirty Constellation packet is statically indistinguishable from any other QPSK transmission within the expanse of 802.11a/g transmissions using that test.

6.1 Packet Based Analysis

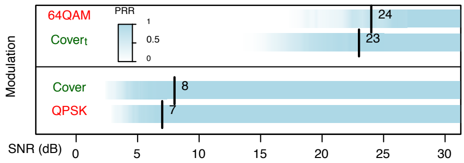


Fig. 9. Packet Reception Rate

Packet based analysis involves looking at parameters that can be extracted at the packet level, or in the digital domain, where there is no trace of the covert packet. To measure if the pre-distortion of the constellation effects the packet reception rate (for both the covert and the cover packet) we performed measurements over a one hop link between two SDR nodes over a wide range of SNR. Figure 9 shows the packet reception rate for the standard modulations used in 802.11a/g and also the SNR required by the intended receiver of the covert packet and the cover packet. The minimum SNR levels required for 98% packet reception rate is marked. For the cover packets, our mechanism is within 1dB of that required by standard 802.11a/g modulation. Given the stochastic nature of the wireless channel and high spatial de-correlation of the nodes, this difference is indistinguishable to an end user (the user would experience greater variance simply by moving their receiver a few inches). The covert receiver requires an SNR of 24dB, similar to the SNR needed to decode a 64QAM packet.

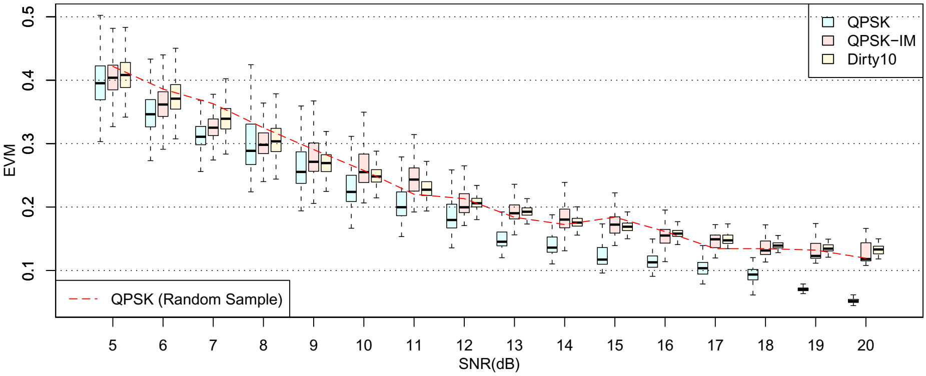


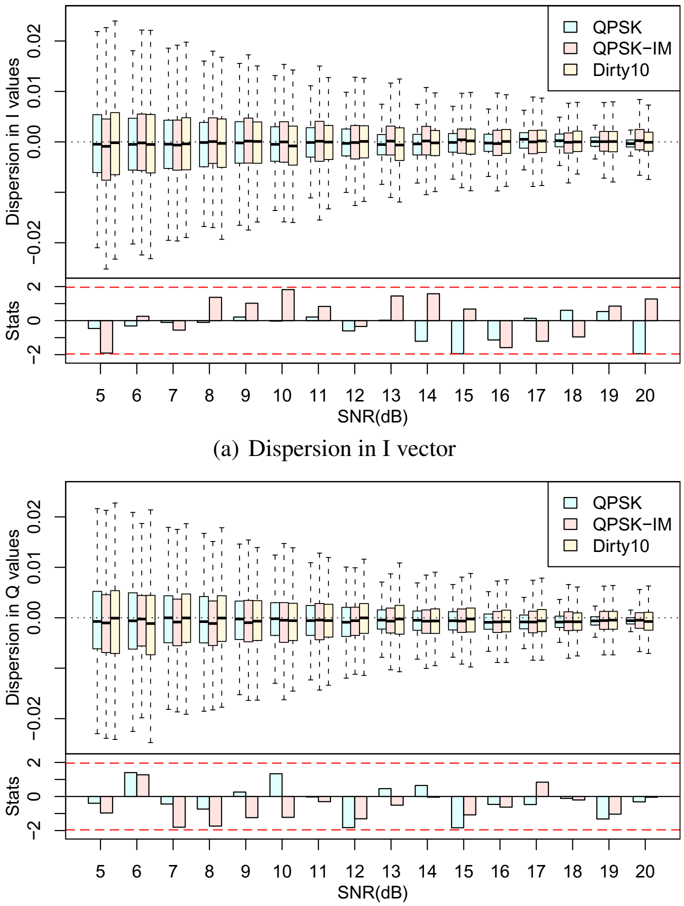
Fig. 10. Distribution of EVM. A faded ideal QPSK sample is also shown.

6.2 Signal Domain Analysis

A time varying signal is often characterized either by time-domain measurements (power envelope and peak to average power ratio) or by performing spectral measurements such as power spectral density, phase and magnitude distributions. Since OFDM encodes data in the frequency domain as coefficients of an inverse Fourier Transformation, a frequency domain analysis is of utmost importance and hence we conduct a set of frequency domain analysis, followed by tests in the time domain.

Frequency Domain Tests –
Test 1: EVM of Constellations: The real and imaginary vectors (I & Q) are available at the output of the Fourier transform unit. EVM is the absolute value of the dispersion of the I/Q-vector averaged over all OFDM symbols in a packet. Figure 10 shows EVM with varying SNR for the QPSK and QPSK-IM bounds and for the Dirty Constellation as well. The inter-quartile distances represents the spread of the I/Q vectors as they are degraded by channel noise. The EVM of the Dirty Constellation is distributed within the bounds set for QPSK making it statistically undetectable when compared with the empirical benchmarks. The plot also shows the average of EVM of a frequency faded random QPSK measurement, which emphasizes the non-deterministic effects of the channel that can push the envelope of the set bounds in either direction. That sample has the same parameters and configuration as the “ideal QPSK”, but with the antenna moved by 2 inches. We expect the test statistic to be correlated with the variation in the bounds.

Test 2: Measure of I/Q Dispersion: The relative dispersion of the I/Q vectors result in a change in the position of the constellation point. Although all receivers employ channel equalization to compensate for the channel distortion, there are always residual errors that cause the points to violate their respective decision threshold leading to bit errors. Figures 11(a) and 11(b) show how the *deviation* from an ideal QPSK constellation is distributed within the dataset. Deviations in the the Dirty Constellation packets are within the bounds for most of the SNR values. To ascertain that the distributions are indeed similar and highly correlated, and that they are normally distributed about the



(a) Dispersion in I vector

(b) Dispersion in Q vector

Fig. 11. Dispersion of I and Q vectors from ideal QPSK mapping. The distribution of the I/Q dispersion is verified with that of ideal QPSK and QPSK-IM using a two sample t-test.

ideal QPSK constellation, we perform a two sample *t*-test with the ideal QPSK packet and the QPSK-IM packet. The test statistics for all the SNR are found to be less than the critical value at the 0.05 significance level, as shown in the bottom part of figure 11. This also satisfies the test that the I/Q dispersion for all the three types are distributed in similar fashion and are from the family of normal distribution with statistically similar means.

Test 3: Phase and Magnitude Distribution: Often it is important to know how the phase and magnitude vary with the subcarrier index. Figure 13 shows a histogram of the subcarrier phases of all packets in the collected dataset at two SNR levels, low SNR (7dB) and high SNR (18dB). At low SNR the subcarriers undergo distortion over a wider range and so the phases have a wider distribution, while at high SNR the signal is closer to the ideal QPSK signal. However, in both the SNR levels, the phases

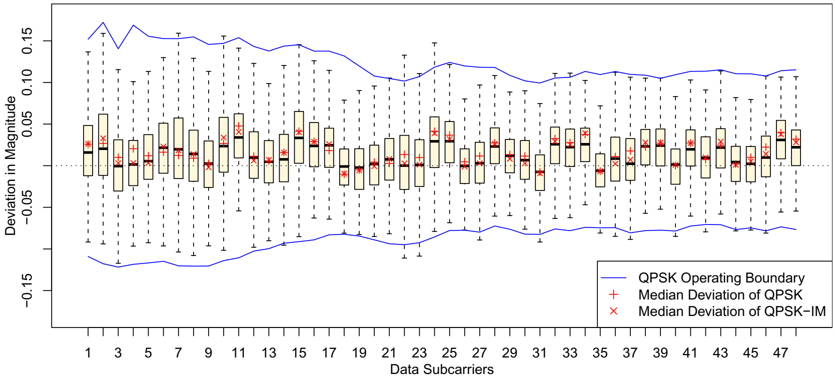


Fig. 12. Magnitude Dispersion per Subcarrier

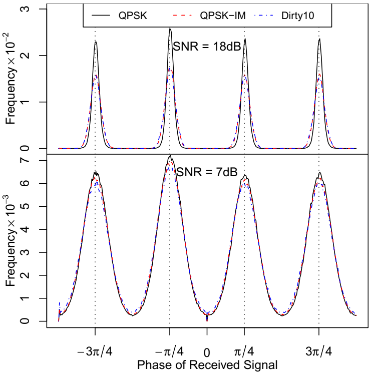
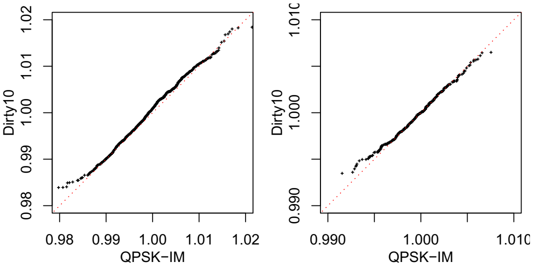


Fig. 13. Phase Distribution



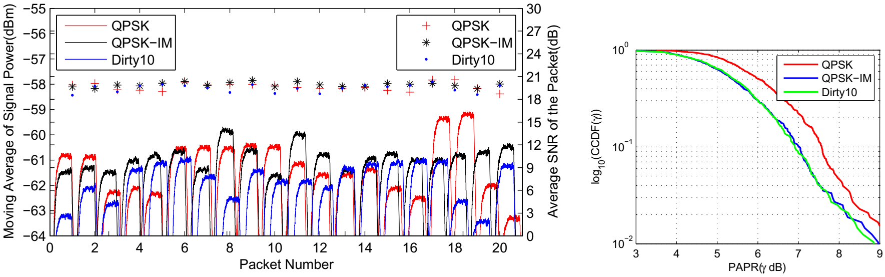
(a) At SNR = 7dB (b) At SNR = 18dB

Fig. 14. QQ-Plot of Magnitude

from the Dirty Constellation packets are distributed similarly to the ideal QPSK and QPSK-IM. The four distinct peaks at multiples of $45°$ ascertain that Dirty Constellation preserves the phase properties of the QPSK constellation. Similarly, the magnitude distribution across the subcarriers show that the magnitude of the subcarriers in a packet encoded with Dirty Constellation are distributed within the bounds of QPSK waveforms, as shown in figure 12. It is also seen that there is a high degree of correlation among the subcarrier from the three types of packets: the same multipath affects all three transmissions. To show that the distributions are correlated we also show the quantile-quantile (QQ) plot for subcarrier magnitudes of the QPSK-IM and the Dirty Constellation packets, as shown in figure 14. The linearity of the QQ plot indicates the signals have similar distributions.

Time Domain Tests –

Test 1: Temporal Variation of Average Signal Power: To test if the Dirty Constellation affects the signal power, we compare the temporal variation with that of a QPSK packet. In an experiment, 20 packets were captured using the VSA for all three types of packet at intervals of $\approx 500ms$. The average power is shown in Figure 15(a). The power envelope for the packets are randomly distributed even though the packets all have similar signal to noise ratios. Therefore, from this test we conclude that our method does not change the average signal power that is different from that of other QPSK packets.



(a) Moving average of signal power (b) CCDF of PAPR

Fig. 15. Time Domain Analysis

Test 2: Peak to Average Power Ratio (PAPR): OFDM can produce spurious increase in the peak power when the packet contains different types of modulations. PAPR is the measure of the spurious increase in power in the time domain. Figure 15(b) shows the complementary CDF (CCDF) of the PAPR for the three packet types. Research [2,10] shows that the PAPR in 802.11a/g can vary over a wide range with various PAPR optimization techniques. The PAPR for Dirty Constellation falls within that range and follows closely with that of QPSK-IM. Hence it cannot be distinguished as an anomaly compared to the ideal QPSK transmission.

In this section we conducted tests that fail to reject the null hypothesis leading us to conclude that our method is statistically undetectable when compared to known waveforms that spans over a wide range of SNR. The analysis in frequency as well as time domain ensures the completeness of the testing. Thus, we conclude that our method can be successfully used as a covert channel that has very low probability of detection.

6.3 Exceptions

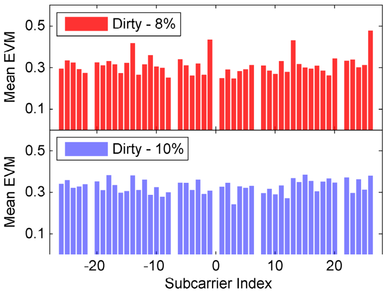


Fig. 16. Average EVM per subcarrier

In this section we provide examples of Dirty Constellation that *are* easily detectable, indicating that the methods and bit-mapping of the covert channel is non-trivial and requires careful analysis before adopting. One would guess that a lower embedding rate is better even though that results in a lower covert data rate. To see that this is not the case, we changed the embedding frequency to $\approx 8\%$ (1 in 12 subcarriers). Figure 16 shows the mean EVM of each data subcarrier of Dirty-8% compared to that of Dirty-10%.

Since the Dirty-8% affects 1 in 12 subcarriers, a regular pattern is emphasized in the EVM of certain subcarriers. The mean EVM for Dirty-8% clearly shows that four out of 48 subcarriers has significantly higher EVM. On the contrary, Dirty-10% has a more even distribution of mean EVM in all of its subcarriers because 48 is not evenly divisible by 10.

7 Security

In §6.3 we discussed that mapping of covert channel is a non-trivial problem. This mapping sequence could be generated using a pseudo random number (PRN) sequence generator. Dirty Constellation employs two forms of sequence or pattern: the covert carrier mapping sequence and the angle of rotation for the covert constellation along the pre-distortion circle. While one PR sequence controls the embedding frequency, another specifies the rotation parameters, such as the angle of rotation "θ" for the covert constellation and the direction of rotation. The receiver needs to know which packets contain covert communication as well as the PRN's used to mix the covert message into the cover message. The frequency of covert messages can also be randomly varied without the need for additional coordination. The PRN used to intermix the covert message is synchronized with the receiver at the beginning of a transmission and can vary over time using an agreed-upon PRN based on *e.g.* the time of day. Any existing encryption method (like AES, DES) can be used in each packet as an added measure to increase the security of the proposed method. However, due to space constraints, we do not analyze the details of the security aspects of this technique in this paper.

8 Related Work

Hiding information has been prevalent since ancient times; however hiding data in digital format is more a recent developments with the popularization of Computer Science. Much of the early work [12] in data hiding with low probability of detection and interception has been done by altering a few bits of the digital representation of an image [15], a sound [8] or video [16] files.

A relatively recent field of study called *network stenography* exploits the redundant fields present in various network protocols headers, like HTTP and TCP. Zander et. al. [17] provides a comprehensive survey of covert channels in computer network protocols. All of the methods detailed in the paper are confined to identifying anomalies or using the protocol properties at the application, transport or the data link layer. Also [11] proposes another scheme to hide data based on utilizing redundant fields in IPv4 header while [3] presents a practical analysis of covert channels in wireless LAN protocols at the transport layer. Information hiding at the application layer of a mobile telephony network has been discussed in [1]. These protocols depend on altering the data itself, which is susceptible to higher probability of interception, when the altered data is tested. Our procedure is significantly different from previous work in the sense that we modify the way of data transmission without altering the bits of any digitally transmitted data. In other words, higher layer stenography operates in the *digital* domain while our method operates in the *analog* domain.

Examples of covert channel implementation utilizing the physical layer are few and far between. A PHY layer based security scheme has been proposed in [14]. However, this method works only when more than one user is available to transmit stenographic packets to a common node. Also it relies on very tight synchronization between multiple transmitter and single receiver entity, which is not a practical assumption in real networks and will lead to erroneous formation of the joint constellations leading to degraded performance. Therefore, comparing to prior work, our method presents a more practical solution to implement covert channels at the PHY layer, while making it secure, high capacity, easily implementable and backward compatible.

9 Conclusion

In this paper, we proposed a technique to implement a covert channel at the physical layer of 802.11a/g wireless protocol. By hiding the covert channel within the perceived noise at the receiver, we can ensure high degree of undetectability. We have implemented the covert communication method using a SDR prototype and present results of a wide variety of statistical tests that confirms the low probability of detection of Dirty Constellation. Higher datarate, very low probability of detection coupled with easy implementation within existing protocol stacks make Dirty Constellation a very successful method to implement covert channels in wireless communication.

References

1. Agaian, S.S., Akopian, D., D'Souza, S.: Wireless steganography, No. 1, p. 60740G. SPIE (2006), http://link.aip.org/link/?PSI/6074/60740G/1
2. Aggarwal, A., Meng, T.: Minimizing the peak-to-average power ratio of ofdm signals using convex optimization, vol. 54, pp. 3099–3110 (2006)
3. Ahsan, K., Kundur, D.: Practical data hiding in TCP/IP. In: Proc. Workshop on Multimedia Security at ACM Multimedia 2002, French Riviera (December 2002)
4. Brik, V., Banerjee, S., Gruteser, M., Oh, S.: Wireless device identification with radiometric signatures. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, MobiCom 2008, pp. 116–127. ACM, New York (2008), http://doi.acm.org/10.1145/1409944.1409959
5. Dutta, A., Fifield, J., Schelle, G., Grunwald, D., Sicker, D.: An intelligent physical layer for cognitive radio networks. In: WICON 2008: Proceedings of the 4th International Conference on Wireless Internet (2008)
6. Fifield, J., Kasemir, P., Grunwald, D., Sicker, D.: Experiences with a platform for frequency agile techniques. In: DYSPAN (2007)
7. Ganti, R., Gong, Z., Haenggi, M., Lee, C., Srinivasa, S., Tisza, D., Vanka, S., Vizi, P.: Implementation and experimental results of superposition coding on software radio. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–5 (May 2010)
8. Gruhl, D., Bender, W., Lu, A.: Echo Hiding. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 295–315. Springer, Heidelberg (1996)
9. IEEE Computer Society: LAN/MAN Standards Committee: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, http://standards.ieee.org/getieee802/download/802.11-2007.pdf

10. Jayalath, A., Tellambura, C.: Peak-to-average power ratio of ieee 802.11 a phy layer signals. In: Wysocki, T.A., Darnell, M., Honary, B. (eds.) Advanced Signal Processing for Communication Systems. The International Series in Engineering and Computer Science, vol. 703, pp. 83–96. Springer US (2002), http://dx.doi.org/10.1007/0-306-47791-2_7
11. Krätzer, C., Dittmann, J., Lang, A., Kühne, T.: Wlan steganography: a first practical review. In: MM&S;Sec 2006: Proceedings of the 8th Workshop on Multimedia and Security, pp. 17–22. ACM, New York (2006)
12. Petitcolas, F., Anderson, R., Kuhn, M.: Information hiding-a survey. Proceedings of the IEEE 87(7), 1062–1078 (1999)
13. Shacham, N.: Multipoint communication by hierarchically encoded data. In: INFOCOM 1992. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3. IEEE (May 1992)
14. Tsouri, G.R., Wulich, D.: Securing ofdm over wireless time-varying channels using subcarrier overloading with joint signal constellations. EURASIP J. Wirel. Commun. Netw. 2009, 2–2 (2009)
15. Wu, M., Tang, E., Lin, B.: Data hiding in digital binary image. In: 2000 IEEE International Conference on Multimedia and Expo, ICME 2000, vol. 1, pp. 393–396 (2000)
16. Xu, C., Ping, X., Zhang, T.: Steganography in compressed video stream. In: First International Conference on Innovative Computing, Information and Control, ICICIC 2006, vol. 1, pp. 269–272 (2006)
17. Zander, S., Armitage, G., Branch, P.: A survey of covert channels and countermeasures in computer network protocols. IEEE Communications Surveys Tutorials 9(3), 44–57 (2007)

Lower Bounds for Private Broadcast Encryption

Aggelos Kiayias and Katerina Samari

Department of Informatics and Telecommunications, University of Athens
{aggelos,ksamari}@di.uoa.gr

Abstract. Broadcast encryption is a type of encryption where the sender can choose a subset from a set of designated receivers on the fly and enable them to decrypt a ciphertext while simultaneously preventing any other party from doing so. The notion of *private* broadcast encryption extends the primitive to a setting where one wishes to thwart an attacker that additionally attempts to extract information about what is the set of enabled users (rather than the contents of the ciphertext).

In this work we provide the first lower bounds for the ciphertext size of private broadcast encryption. We first formulate various notions of privacy for broadcast encryption, (priv-eq, priv-st and priv-full) and classify them in terms of strength. We then show that any private broadcast encryption scheme in the sense of priv-eq (our weakest notion) that satisfies a simple structural condition we formalize and refer to as "atomic" is restricted to have ciphertexts of size $\Omega(s \cdot k)$ where s is the cardinality of the set of the enabled users and k is the security parameter. We then present an atomic private broadcast encryption scheme with ciphertext size $\Theta(s \cdot k)$ hence matching our lower bound that relies on key privacy of the underlying encryption. Our results translate to the setting priv-full privacy for a ciphertext size of $\Theta(n \cdot k)$ where n is the total number of users while relying only on KEM security. We finally consider arbitrary private broadcast encryption schemes and we show that in the priv-full privacy setting a lower-bound of $\Omega(n+k)$ *for every ciphertext* is imposed. This highlights the costs of privacy in the setting of broadcast encryption where much shorter ciphertexts have been previously attained with various constructions in the non-privacy setting.

1 Introduction

Consider the setting of an encrypted file system. Each file is encrypted so that only a designated subset of the set of users of the system can retrieve it. An attacker, who may be controlling a set of system users should be incapable of recovering the contents of the file provided that none of the controlled users belong to the enabled set for the file.

This setting is one of the application domains for broadcast encryption, a cryptographic primitive introduced by Fiat and Naor [9]. Broadcast encryption is also suitable for application to the setting of content distribution and is indeed widely used as the encryption system of DVDs (for example in the form of the AACS [1]) and other media content carrying mechanisms. A variety of schemes

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 176–190, 2013.

have been developed over the years with the main objective of reducing the ciphertext length. Currently in the private key setting (see e.g. [14]) there are schemes that achieve a ciphertext length of $\Theta(r \cdot k)$ where r is the number of revoked users and k is the security parameter; in the public-key setting, using bilinear maps the scheme of [4] achieves a ciphertext length of $O(k)$ with public key of $O(n \cdot k)$ for any set of enabled users and the scheme of Delerablée [6] achieves a ciphertext length $O(k)$ while the public-key is of size $\Theta(s \cdot k)$ assuming that sets of enabled users never exceed cardinality s.

Barth, Boneh and Waters [3] put forth the notion of private broadcast encryption. Their objective is to consider another class of attacks for broadcast encryption where the goal of the attacker is to discover information about the set of enabled users rather than decrypting a ciphertext for which it is not enabled. Protecting the privacy of the users in the enabled set can be an equally and some times perhaps an even more important goal than the privacy of the message. Indeed, hiding the information that one is a recipient of a message, from other users and even from other recipients of the same message, is a critical security feature in any setting where the fact of receiving a message at a certain time or frequency reveals sensitive personal characteristics of the recipient. For example, in a file system, an encrypted system file under a certain account may reveal that the said account has a certain level of system privileges and this fact can assist an attacker in a more complex attack vector.

To address this important problem, Barth et al. [3] introduced a security model for private broadcast encryption and provided a first solution. The scheme of [3] applies to the public-key setting and has the characteristic of being linear in the number of users, i.e., has a ciphertext of length $\Theta(s \cdot k)$ where s is the number of enabled users. Given that, as shown above, previously known (non-private) schemes achieve much better ciphertext lengths, it is an important open question to improve this efficiency characteristic for private broadcast encryption schemes or demonstrate that no further improvement is possible.

In this work, motivated by the above, we provide various results suggesting the latter state of affairs by proving tight lower bounds for the ciphertext length of private broadcast encryption schemes. We outline our results below.

First, we study the formalization of the notion of privacy in the context of private broadcast encryption. We introduce three security formulations. The first notion we consider is inspired by that in [3] : it allows the adversary to interact with the broadcast encryption system by obtaining encryption and decryption queries as well as corrupting recipients. Upon completion of a first stage the adversary provides two target sets of users to be revoked $\mathsf{R}_0, \mathsf{R}_1$.Then, provided that $|\mathsf{R}_0| = |\mathsf{R}_1|$, the adversary receives as a challenge a message M and an encryption of M with the set of users R_b revoked where b is a random bit. The adversary has to guess the bit b under the constraint that it does not submit the challenge ciphertext to a decryption oracle and does not control any user in the symmetric difference $\mathsf{R}_0 \triangle \mathsf{R}_1$. We call this level of privacy $\mathsf{priv\text{-}eq}$.

We observe $\mathsf{priv\text{-}eq}$ is quite insufficient for many reasonable attack settings. Specifically, for a certain ciphertext the adversary may be absolutely certain that the set of users R is revoked and only wishes to test whether an additional target user i is also revoked or not. Clearly this attack objective is not captured by the above definition since in this case it holds that $\mathsf{R}_0 = \mathsf{R}$ and $\mathsf{R}_1 = \mathsf{R} \cup \{i\}$, two sets of different cardinality. We formalize this notion of privacy as priv-st. It is very easy to see that there exist schemes that satisfy priv-eq and fail priv-st; in particular, any scheme that leaks the cardinality of the set of revoked users is such a candidate and in fact the scheme of [3] is one such scheme.

Taking this one step further we introduce *full privacy* to be the property where the adversary cannot distinguish any two sets $\mathsf{R}_0, \mathsf{R}_1$; we term this notion as priv-full. We then prove that in fact priv-st and priv-full are equivalent.

Armed with this definitional basis we proceed to our lower bounds. We first consider the case of *atomic* broadcast encryption schemes. Atomic schemes have the characteristic that the ciphertext can be broken to a number of discrete components and each recipient when it is decrypting it applies a decryption function to one or more of those components. The private schemes of [3] satisfy this condition and it is also quite common in the wide class of combinatorial broadcast encryption schemes; a partial list of non-private atomic schemes is the following ([14],[12],[11],[16],[2]).

For such atomic schemes, we prove that any scheme that satisfies the priv-eq condition is susceptible to an attack against privacy in the case when the ciphertext drops below $s \cdot k$ where s is the cardinality of the set of enabled users. This means that a lower bound of $\Omega(s \cdot k)$ is in place. We then present an atomic private broadcast encryption scheme with this complexity hence showing the lower bound is tight. The scheme itself is a standard linear length construction; the scheme applies equally to the symmetric and public-key setting and abstracts the necessary properties needed for privacy to the existence of secure key-private encryption mechanism in the KEM sense [15]. We present a similar set of results for the priv-full level of privacy; in this case KEM security is sufficient and the corresponding tight bound is $\Theta(n \cdot k)$.

Having settled the case of atomic broadcast encryption, we switch our focus to the setting of general private broadcast encryption schemes (that are not necessarily atomic). We first show using an information theoretic argument that any broadcast encryption scheme should exhibit some ciphertexts of length $\Omega(n+k)$. Using this as a stepping stone we then prove that if a broadcast encryption scheme is assumed to be private in the sense of priv-st, priv-full, it will have to provide a ciphertext of length $\Omega(n + k)$ for any set of revoked users R hence a complexity bound sublinear in the number of users is impossible to be achieved if full privacy is desired.

Related Work. Independently of the present work, Libert, Paterson and Quaglia [13] have studied the problem of "anonymous broadcast encryption" where the main focus is to enable efficient decryption in the setting where the ciphertext is of length $\Theta(s \cdot k)$. In this case the known schemes that satisfy privacy require from the users to test sequentially until they find the proper element they can decrypt. In the public-key setting this can be an arduous task if the number of enabled users is large; by using some randomized tagging mechanism it is possible to improve the decryption time complexity. Our modeling is consistent with that of [13] and our lower bounds readily apply to their setting as well.

Fazio and Perera in [8], introduce a weaker notion of anonymity compared to the one considered here and in previous works, called *outsider-anonymity.* An *Outsider-anonymous broadcast encryption scheme* ensures that a user in the revoked set can gain no information about the enabled set while a member of the enabled set may extract information about some other users in it. Taking advantage of this relaxation to the anonymity definition, the authors employ an atomic scheme, i.e. the public key variant of Complete Subtree method [7], in order to achieve sublinear ciphertext size.

2 Privacy Notions for Broadcast Encryption

Broadcast encryption is a triple $\langle \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt} \rangle$ where KeyGen generates a set of n keys for any given n and $\mathsf{Encrypt}$ receives a set of revoked users $\mathsf{R} \subseteq [n]$ that should be barred from decrypting. We define privacy in broadcast encryption using an experiment between a challenger and an adversary. The adversary is given access to an Encryption Oracle which means that he is capable of obtaining ciphertext-message pairs that can be decrypted by an enabled set of his choice. Also, he is able to derive the secret keys of a selected set of users, by submitting a number of queries to a Corruption Oracle. We will distinguish three levels of privacy in our formalization. In the most general type (full privacy), $\mathsf{priv\text{-}full}$, the adversary should be unable to distinguish between any two sets of revoked users as long as the corrupted users do not cover the symmetric difference of the two sets. In the case of “single target” privacy, $\mathsf{priv\text{-}st}$ the adversary wishes to understand whether a single (target) user is included in an (otherwise) known revoked set. Finally, in privacy among equal sets, $\mathsf{priv\text{-}eq}$, is identical to the case of $\mathsf{priv\text{-}full}$ with the additional restriction that the adversary should challenge on two sets with equal cardinality. Formally, we have the following:

```
$\mathsf{EncryptionOracle}(\mathsf{R})$
    $retrieve\ ek$
    $m \xleftarrow{r} \mathsf{M}$
    $c \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$
    $return\ (c, m)$

$\mathsf{CorruptOracle}(u)$
    $\mathsf{T} \leftarrow \mathsf{T} \cup \{u\}$
    $return\ \mathsf{K}_u$

$\mathsf{DecryptionOracle}(u, c)$
    $\mathsf{D} \leftarrow \mathsf{D} \cup \{(u, c)\}$
    $retrieve\ \mathsf{K}_u$
    $return\ \mathsf{Decrypt}(\mathsf{K}_u, c)$
```

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}x}}(1^n, 1^\lambda)$
  $(ek, \mathsf{K}_1, \ldots, \mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n, 1^\lambda)$
  $\mathsf{T} \leftarrow \emptyset$
  $(state, \mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot), \mathsf{EncryptionOracle}(\cdot), \mathsf{DecryptionOracle}(\cdot)}(1^\lambda)$
  $b \xleftarrow{r} \{0, 1\}$
  $m \xleftarrow{r} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R}_b)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot), \mathsf{EncryptionOracle}(\cdot), \mathsf{DecryptionOracle}(\cdot)}(guess, (c^*, m), state)$
  if $\big(\exists i \in \mathsf{T}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)\big) \vee$
```

$\big(\exists(i,c) \in \mathsf{D}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)$ and $c = c^*\big)$
then output a random bit else if $b = b^*$ then return 1 else 0;

Definition 1 (Privacy). *Let Φ be a fully exclusive broadcast encryption scheme with n receivers. We say that Φ is private* $\mathsf{priv\text{-}x}$*, if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}x}}(1^n, 1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ and λ is the security parameter.

Based on the definition above, we provide three different definitions for privacy whose differences concern the form of the challenge $(\mathsf{R}_0, \mathsf{R}_1)$.

- We call $\mathsf{Exp}^{\mathsf{priv\text{-}full}}$ the experiment in which $\mathsf{R}_0, \mathsf{R}_1$ can be any set which is subset of $[n]$.
- With $\mathsf{Exp}^{\mathsf{priv\text{-}st}}$, we define the experiment where $\mathsf{R}_0, \mathsf{R}_1$ have to be of the form R and $\mathsf{R} \cup \{i\}$, accordingly.
- With $\mathsf{Exp}^{\mathsf{priv\text{-}eq}}$, we define the experiment where $\mathsf{R}_0, \mathsf{R}_1$ have to be of equal size. Consequently, it is necessary to add one more or-factor, $(|\mathsf{R}_0| \neq |\mathsf{R}_1|)$, in the condition of the last line of the experiment, to guarantee that the experiment outputs a random bit in case the adversary's challenge sets are of unequal size.

We then proceed to show relations between the three notions of privacy.

Theorem 1. *1. Privacy definitions* $\mathsf{priv\text{-}st}$ *and* $\mathsf{priv\text{-}full}$ *are equivalent.*
2. Privacy definition $\mathsf{priv\text{-}full}$ *implies the privacy definition* $\mathsf{priv\text{-}eq}$.
3. Privacy definition $\mathsf{priv\text{-}eq}$ *does not imply privacy definition* $\mathsf{priv\text{-}st}$.

Proof. 1. We need to prove two directions in order to show that these definitions are equivalent. The easy direction is the one which says that privacy definition $\mathsf{priv\text{-}full}$ implies privacy definition $\mathsf{priv\text{-}st}$. If we assume that there exists a PPT adversary $\mathcal{A}$ that breaks privacy definition $\mathsf{priv\text{-}st}$ challenging a pair $(\mathsf{R}, \mathsf{R} \cup \{i\})$ with non-negligible advantage α, this adversary also breaks privacy definition $\mathsf{priv\text{-}full}$ considering that $\mathsf{R}_0 = \mathsf{R}$ and $\mathsf{R}_1 = \mathsf{R} \cup \{i\}$. The opposite direction will be derived from lemma 1.

2. Assuming that there exists a PPT adversary that breaks privacy definition $\mathsf{priv\text{-}eq}$ having advantage α, then the same adversary does also break privacy definition $\mathsf{priv\text{-}full}$ with non-negligible advantage α.
3. It suffices to provide a broadcast encryption scheme which satisfies the definition $\mathsf{priv\text{-}eq}$ but not private according to the definition $\mathsf{priv\text{-}full}$. Let Φ be a broadcast encryption scheme which is $\mathsf{priv\text{-}eq}$. Now consider Φ' to be exactly like Φ but with the added feature that the encryption algorithm appends at the end of all ciphertexts the cardinality of the revoked set. It is obvious that this scheme is inherently incapable of satisfying privacy definition $\mathsf{priv\text{-}full}$ (while it remains $\mathsf{priv\text{-}eq}$). Such schemes exist under standard cryptographic assumptions as we will see in section 4.

■

Lemma 1. *Let Φ be a broadcast encryption scheme with n receivers. If there exists a PPT adversary that has advantage α in breaking privacy definition* $\mathsf{priv\text{-}full}$*, then there exists a PPT adversary that breaks privacy definition* $\mathsf{priv\text{-}st}$ *with probability at least $1/2 + \alpha/n$.*

Sketch of Proof: Let $\mathcal{A}$ be a PPT adversary that breaks $\mathsf{priv\text{-}full}$ definition with advantage α. Conditioning on the fact that $\mathcal{A}$ breaks privacy for a pair of sets $(\mathsf{R}_0, \mathsf{R}_1)$, we consider a sequence of sets $P_0, ..., P_{k-1}$, where $k = |\mathsf{R}_0 \triangle \mathsf{R}_1| + 1$, $P_0 = \mathsf{R}_0$ and $P_{k-1} = \mathsf{R}_1$. We set $m = |\mathsf{R}_0 \setminus \mathsf{R}_1|$ and we define P_i as follows: if $i \in \{0, \ldots, m\}$ $P_i = P_{i-1} \setminus \{j\}$, for some user $j \in \mathsf{R}_0 \setminus \mathsf{R}_1$, otherwise $P_i = P_{i-1} \cup \{j'\}$ for some user $j' \in \mathsf{R}_1 \setminus \mathsf{R}_0$. Namely, all the members of this sequence are supersets of $\mathsf{R}_0 \cap \mathsf{R}_1$ and every pair of consecutive sets are of the form $(\mathsf{R}, \mathsf{R} \cup \{i\})$ for some R. We denote as $\mathcal{A}_1$ the part of the algorithm $\mathcal{A}$ that corresponds to the training stage of the experiment, i.e. before the output of challenge, while with $\mathcal{A}_2$ we denote $\mathcal{A}$'s steps after the receipt of the response. Together with the challenge pair $(\mathsf{R}_0, \mathsf{R}_1)$, $\mathcal{A}_1$ outputs a random variable *state*.

We construct a PPT adversary $\mathcal{B}$ that breaks definition $\mathsf{priv\text{-}st}$ as follows: $\mathcal{B}$ runs $\mathcal{A}_1$ until he outputs the challenge pair $(\mathsf{R}_0, \mathsf{R}_1)$ together with *state*. Then $\mathcal{B}$ makes a guess $j \in \{0, \ldots, k-2\}$ and challenges the corresponding pair. Due to the structure of the sequence, if $j \in \{0, \ldots m-1\}$ $\mathcal{B}$ challenges (P_{j+1}, P_j), otherwise challenges (P_j, P_{j+1}). The received response is provided together with *state* to $\mathcal{A}_2$. Then, if $j \in \{0, \ldots, m-1\}$ $\mathcal{B}$ outputs the complement of $\mathcal{A}_2$'s output, otherwise outputs $\mathcal{A}_2$'s output. We conclude that $\mathcal{B}$ breaks definition $\mathsf{priv\text{-}st}$ with advantage $\alpha/(k-1)$ which is at least α/n. ■

3 Lower Bounds for Atomic Broadcast Encryption Schemes

Definition 2. *An atomic broadcast encryption scheme with n receivers is defined as a tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$:

- KeyGen: On input $1^n, 1^\lambda$, it generates the set of keys $(ek, \mathsf{SK}_1, ..., \mathsf{SK}_n)$, where ek is the encryption key and SK_i is the decryption key assigned to a user i. Each decryption key SK_i is a set which consists of elements $\{sk_{ij}\}_{j=1}^{\ell}$ (we call those atomic keys) for some value ℓ which is not necessarily the same for each user. It also produces the description of a language $\mathcal{L}$ which encodes all the possible subsets of users that may be provided as input to the encryption function.
- $\mathsf{Encrypt}$: On input a message m, the encryption key ek and a revocation instruction $\mathsf{R} \in \mathcal{L}$, it outputs a ciphertext C such that $C \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$ which among possibly other values, contains a number of components $c_1, ..., c_\rho$ (we call those the atomic ciphertexts of C).
- $\mathsf{Decrypt}$: On input a ciphertext C, such that $C \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R})$ and a decryption key SK_i: It outputs m if $i \notin \mathsf{R}$ and some value $x \neq m$ if $i \in \mathsf{R}$. Depending on the instantiation, x could be the symbol $\perp$, or some plaintext sampled independently of m.

For atomic broadcast encryption schemes we further assume the existence of a deterministic algorithm called Decryptmatching which matches the atomic ciphertexts of a ciphertext tuple C with the atomic keys under which they are decrypted. In all cases we know, this algorithm is in part of the Decryption algorithm.

Proposition 1. *The broadcast encryption schemes that rely on the subset cover framework [14] are atomic. The private schemes of [3] are atomic.*

Given that in this section we will provide lower bounds, we provide a weaker definition of privacy which departs from definition priv-eq in the existence of the CorruptOracle and DecryptionOracle in the security experiment. More precisely, the adversary is not given access to a Decryption Oracle and instead of being provided access to a Corruption Oracle, he is given access to an Atomic Decryption Oracle which operates as follows:

$$\mathsf{AtDecOr}(j,t,C) = \begin{cases} 0 & \text{if no atomic ciphertext in } C \text{ is supposed to be decrypted} \\ & \text{under the key } sk_{jt} \\ \perp & \text{if the number of keys in the set } \mathsf{SK}_j \text{ are less than } t \\ 1 & \text{if there exists an atomic ciphertext that can be decrypted} \\ & \text{under the key } sk_{jt} \end{cases}$$

```
EncryptionOracle(R)                      | AtDecOr($j,t,C$)
    retrieve $ek$                        |     $\mathsf{E} \leftarrow \mathsf{E} \cup \{(j,t)\}$
    $m \xleftarrow{r} \mathsf{M}$        |     return $x \in \{0,1,\perp\}$
    $c \leftarrow \mathsf{Encrypt}(ek,m,\mathsf{R})$
    return $(c,m)$
----------------------------------------------------------------
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}(1^n,1^\lambda)$
  $(ek,\mathsf{K}_1,\ldots,\mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n,1^\lambda)$
  $\mathsf{T} \leftarrow \emptyset$
  $(state,\mathsf{R}_0,\mathsf{R}_1) \leftarrow \mathcal{A}^{\mathsf{AtDecOr}(\cdot),\mathsf{EncryptionOracle}(\cdot)}(1^\lambda)$
  $b \xleftarrow{r} \{0,1\}$
  $m \xleftarrow{r} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek,m,\mathsf{R}_b)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{AtDecOr}(\cdot),\mathsf{EncryptionOracle}(\cdot)}(guess,(c^*,m),state)$
  if $\big(\exists(i,\cdot) \in \mathsf{E}$ such that $i \in (\mathsf{R}_0 \triangle \mathsf{R}_1)) \vee (|\mathsf{R}_0| \neq |\mathsf{R}_1|)$
  then output a random else if $b = b^*$ then return 1 else 0;
```

The experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}$ is defined identically to $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq}}$ with the oracle AtDecOr substituting the corruption and decryption oracles.

Definition 3. *Let Φ be a broadcast encryption scheme with n receivers. We say that Φ is private* priv-eq-at*, if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}eq\text{-}at}}(1^n,1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ and λ the security parameter.

The following proposition is easy:

Proposition 2. *Any broadcast encryption scheme Φ that satisfies privacy definition* $\mathsf{priv\text{-}eq}$*, does also satisfy privacy definition* $\mathsf{priv\text{-}eq\text{-}at}$.

Proof. It is easy to see that assuming the existence of a PPT adversary $\mathcal{A}$ that has non-negligible advantage in breaking privacy definition $\mathsf{priv\text{-}eq\text{-}at}$, there is a PPT adversary $\mathcal{B}$ that breaks privacy definition $\mathsf{priv\text{-}eq}$ with the same advantage as $\mathcal{A}$ executing $\mathcal{A}$ inside him. The proof relies on the fact that $\mathcal{B}$ can perfectly answer the queries submitted by $\mathcal{A}$ to the Atomic Decryption Oracle because of his access to a Corruption Oracle.

Theorem 2. *(Lower bound for atomic schemes) Let Φ be an atomic broadcast encryption scheme and suppose that there exists an enabled set $S \subseteq [n]$ such that the number of atomic ciphertexts included in the prepared ciphertext C_S are less than $|S|$. Then, the scheme is* not *private according to definition* $\mathsf{priv\text{-}eq\text{-}at}$.

Proof. We will assume that for every R the atomic ciphertexts produced by the algorithm $\mathsf{Encrypt}$ are always decrypted under the same set of atomic keys (in the other case, if the algorithm $\mathsf{Encrypt}$ flips a number of coins in order to decide the atomic keys that will be used, then the same argument we present below can take place with the only difference that in this case the adversary will have to run a number of times the algorithm $\mathsf{Encrypt}$ for the set R_0 to approximate the distribution). Let us assume that there exists such a set S_0 and let C_{S_0} be a ciphertext produced by the algorithm $\mathsf{Encrypt}$ on input ek, m, R_0 with $\mathsf{R}_0 = [n] \setminus S_0$. Then, according to the pigeonhole principle, there exists at least one atomic ciphertext c_k in the ciphertext C_{S_0} that can be decrypted by at least two users $i, j \in [n]$. As a result, the ciphertext c_k can be decrypted under an atomic key sk_m which is a member of both sets SK_i, SK_j, where SK_i, SK_j are the sets of atomic decryption keys of i and j accordingly. Given this, an adversary $\mathcal{A}$ that breaks privacy can be constructed following the logic presented below:

1. If $i, j \in [n]$ are two users which decrypt the same atomic ciphertext in a ciphertext tuple C_{S_0}, where $C_{S_0} \leftarrow \mathsf{Encrypt}(ek, m, \mathsf{R}_0)$, select a set R_1 such that $|\mathsf{R}_1| = |\mathsf{R}_0|$, $i \in \mathsf{R}_1$ and $j \notin \mathsf{R}_1$. Choose arbitrarily the other $|\mathsf{R}_1| - 1$ members of R_1 and challenge $\mathsf{R}_0, \mathsf{R}_1$.
2. When the response C^* is received, issue a query R_0 to the Encryption Oracle which is replied with a ciphertext C.
3. Submit a number of queries of the form (j, t, C) to the Atomic Decryption Oracle, for all the possible values of t, starting form $t = 1$, until $\mathsf{AtDecOr}$ returns $\perp$. If we ignore the symbol $\perp$, the output of this procedure is a bitstring x_1 of length s, where s is the number of atomic keys included in the decryption key of SK_j.
4. Repeat the same procedure submitting queries on inputs of the form (j, t, C^*) and obtain a bitstring x_2 of length k (note that this is allowed since j is enabled in both challenge ciphertexts). If $x_1 \neq x_2$, then answer 1 else 0. ■

Corollary 1. *Any atomic broadcast encryption scheme with n receivers and ciphertext length less than n cannot be private according to definition* priv-full*.*

Proof. If $\mathsf{R} = \emptyset$ and the atomic ciphertexts are less that n, the assumption of the Theorem 2 takes place for $S = [n]$. It is easily observed that the fact that the challenge sets $\mathsf{R}_0, \mathsf{R}_1$ were of equal length played no crucial role in the proof of Theorem 2. Thus, we can apply exactly the same arguments with $\mathsf{R} = \emptyset$ being the one set in the challenge.

Corollary 2. *For any atomic broadcast encryption scheme Φ with n receivers which is private according to* priv-eq *definition, it holds that for any enabled set $S \subseteq [n]$, the ciphertext length is $\Omega(k \cdot |S|)$ bits, where k is the maximum size of an atomic ciphertext. For any broadcast encryption scheme which is private according to* priv-full *definition, the ciphertext length is $\Omega(k \cdot n)$ for all the enabled sets $S \subseteq [n]$.*

4 Constructions of Atomic Private Broadcast Encryption Schemes

In this section, we present matching schemes for the lower bounds of the previous section. We focus on CCA-1 security for simplicity but our results can be easily extended to CCA-2 security. Due to lack of space most of our results are presented without proofs; full proofs are presented in the full version. We first consider security in the sense of key encapsulation mechanisms (KEM) defined with the aid of the following experiment:

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{KEM}(1^\lambda)$
  Select $k$ at random.
  $aux \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),\mathsf{Dec}_k(\cdot)}$
  $m_0, m_1 \overset{r}{\leftarrow} \mathsf{M}$;
  $b \overset{r}{\leftarrow} \{0,1\}; c \leftarrow \mathsf{Enc}_k(m_b)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(m_1, c)$
  if $b = b^*$ then return 1 else 0;
```

Definition 4. *We say that the symmetric encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is KEM-secure if for any probabilistic polynomial time adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{KEM}(1^\lambda)] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ.

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda)$
  $(ek, \mathsf{K}_1, \ldots, \mathsf{K}_n) \leftarrow \mathsf{KeyGen}(1^n, 1^\lambda)$
  $\mathsf{T} \leftarrow \emptyset$
  $\mathsf{R} \leftarrow \mathcal{A}^{\mathsf{CorruptOracle}(\cdot),\mathsf{EncryptionOracle}(\cdot),\mathsf{DecryptionOracle}(\cdot)}(\cdot)$
  $b \overset{r}{\leftarrow} \{0,1\}$
  $m_0, m_1 \overset{r}{\leftarrow} \mathsf{M}$
  $c^* \leftarrow \mathsf{Encrypt}(ek, m_b, \mathsf{R})$
  $b^* \leftarrow \mathcal{A}^{\mathsf{EncryptionOracle}(\cdot)}(c^*, m_1)$
  If $\mathsf{T} \nsubseteq \mathsf{R}$ then output a random bit
  else if $b = b^*$ then return 1 else 0;
```

Definition 5. *Let Φ be a broadcast encryption scheme with n receivers. We say that a broadcast encryption scheme Φ is KEM-secure if for any probabilistic polynomial time adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda) = 1] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ.

```
Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{key\text{-}priv}}(1^\lambda)$
  Select $k_0 \leftarrow \mathsf{Gen}(1^\lambda)$; $k_1 \leftarrow \mathsf{Gen}(1^\lambda)$
  $aux \leftarrow \mathcal{A}^{\mathsf{Enc}_{k_0}(\cdot), \mathsf{Enc}_{k_1}(\cdot), \mathsf{Dec}_{k_0}(\cdot), \mathsf{Dec}_{k_1}(\cdot)}$
  $m \overset{r}{\leftarrow} \mathsf{M}$
  $b \overset{r}{\leftarrow} \{0,1\}$;$c \leftarrow \mathsf{Enc}_{k_b}(m)$
  $b^* \leftarrow \mathcal{A}^{\mathsf{Enc}_{k_0}(\cdot), \mathsf{Enc}_{k_1}(\cdot)}(m, c)$
  if $b = b^*$ then return 1 else 0;
```

Definition 6. *We say that the symmetric encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is key private if for any probabilistic polynomial time adversary $\mathcal{A}$ it holds that*

$$\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{key\text{-}priv}}(1^\lambda)] \leq \frac{1}{2} + \varepsilon,$$

where ε is a negligible function of λ.

Scheme 1. This scheme is defined as a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ which are described below. A basic component of the scheme is the underlying symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

- KeyGen : On input $1^n, 1^\lambda$:
  - For any user $i \in [n]$ run the algorithm $\mathsf{Gen}(1^\lambda)$ which generates a key k_i. The encryption key is $ek = \{k_j\}_{j \in [n]}$.
- $\mathsf{Encrypt}$: On input a message m and a revoked set R:
  - By employing the scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ compute a ciphertext tuple c as follows: For each $i \in [n] \setminus \mathsf{R}$ compute $\mathsf{Enc}_{k_i}(m)$. Perform a random permutation f to the ciphertext components which results to a ciphertext tuple of length s, where s is the cardinality of the set $[n] \setminus \mathsf{R}$.
- $\mathsf{Decrypt}$: On input a ciphertext $c = \langle c_1, ..., c_s \rangle$ and a key k_u:
  - Starting from c_1, try to decrypt each ciphertext component under the key k_u. If there exists c_j that is supposed[1] to be decrypted by u, return $\mathsf{Dec}_{k_u}(c_j)$.

[1] In order to determine this *strong* correctness is required; this notion means that applying a wrong key to a ciphertext results to a special fail message to be returned. This can be achieved e.g., by appending a value $H(M)$ to all plaintexts M (here H is a hash function); we omit further details.

Scheme 2. This scheme is defined as a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ which we describe below. A basic component of the scheme is the underlying symmetric encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

– KeyGen : On input $1^n, 1^\lambda$:
  - For any user $i \in [n]$ run the algorithm $\mathsf{Gen}(1^\lambda)$ which generates a key k_i. The encryption key is $ek = \{k_j\}_{j\in[n]}$.

– $\mathsf{Encrypt}$: On input a message m and a revoked set R:
  - By employing a scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ compute a ciphertext tuple c of length n as follows: For any user $i \in [n]$, if $i \in \mathsf{R}$ choose randomly a message $m' \in \mathsf{M}$, compute $E_{k_i}(m')$ and place $E_{k_i}(m')$ at the i-th position. If $i \notin \mathsf{R}$, compute $\mathsf{Enc}_{k_i}(m)$ and place it to the i-th position.

– $\mathsf{Decrypt}$: On input a ciphertext $c = \langle c_1, ..., c_n\rangle$ and a key k_u of a user u:
  - Compute $\mathsf{Dec}_{k_u}(c_u)$.

Theorem 3. *If Scheme 1 satisfies that the underlying scheme* $(\mathsf{Gen}, \mathsf{Dec}, \mathsf{Enc})$ *key-private then Scheme 1 is private according to the definition* $\mathsf{priv\text{-}eq}$.

Theorem 4. *If Scheme 2 is a broadcast encryption scheme in which the underlying scheme* $(\mathsf{Gen}, \mathsf{Dec}, \mathsf{Enc})$ *is KEM-secure, then Scheme 2 is private according to definition* $\mathsf{priv\text{-}full}$.

It remains to show that the broadcast encryption schemes Scheme 1 and Scheme 2 are BE-KEM-secure, i.e. they are secure under the definition 5. The proofs of security are similar and we prove this only for Scheme 2.

Theorem 5. *If the underlying encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is KEM-secure then Scheme 2 is BE-KEM secure.*

Proof. Let $\mathcal{A}$ be a PPT adversary that breaks BE-KEM security such that $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}(1^n, 1^\lambda) = 1] \geq \frac{1}{2} + \alpha$, for α non-negligible. We define a sequence of experiments $\mathsf{Exp}_0^{\mathcal{A}}, ..., \mathsf{Exp}_n^{\mathcal{A}}$, where $\mathsf{Exp}_0^{\mathcal{A}}$ is the experiment $\mathsf{Exp}_{\mathcal{A}}^{BE-KEM}$. We define as $\mathsf{Exp}_v^{\mathcal{A}}$ the experiment which operates exactly as $\mathsf{Exp}_0^{\mathcal{A}}$ modified in a way that the first v enabled users will be given encryptions of randomly chosen plaintexts rather than the encryption of the appropriate plaintext. If s is the size of the enabled set, for $v = s, s+1, ..., n$ the experiments are the same.

Now, let $p_0 = \mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1]$ and $p_1 = \mathsf{Prob}[\mathsf{Exp}_1^{\mathcal{A}} = 1]$. Moreover, let $\mathcal{B}$ be an attacker against KEM-security of the scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. $\mathcal{B}$ guesses i to be the user he will play $\mathsf{Exp}_{\mathcal{B}}^{KEM}$ and then running $n-1$ times the algorithm $\mathsf{Gen}(1^\lambda)$ he generates the private keys for the other users. When $\mathcal{A}$ challenges R, $\mathcal{B}$ checks whether i is the first enabled user and returns 0 if this does not hold. Otherwise, when $\mathcal{B}$ receives $(m_1, \mathsf{Enc}_k(m_b))$, he places $\mathsf{Enc}_k(m_b)$ at the first position and then chooses randomly a message m' from the plaintext space and flips a perfect coin b'. $\mathcal{B}$ sets $m'_{b'} = m_1$ and $m'_{1-b'} = m'$ and encrypts the message $m'_{b'}$ for the enabled users except for i. $\mathcal{B}$ encrypts a message m'' for the revoked users which is randomly chosen from the plaintext space. $\mathcal{B}$ always sends to $\mathcal{A}$ the message m'_1 together with the prepared ciphertext tuple.

Due to the fact that for all $\mathcal{B}$, $\mathsf{Prob}[\mathsf{Exp}_{\mathcal{B}}^{KEM}(1^\lambda) = 1] \leq \dfrac{1}{2} + \varepsilon$, it can be proven that $p_0 - p_1 \leq 2n \cdot \varepsilon$. Similarly, we have that for all $i \in \{0, 1, .., n\}$, $p_i - p_{i+1} \leq 2n \cdot \varepsilon$. Summing these relations for both sides, we have that $p_0 - p_n \leq 2n^2 \cdot \varepsilon$. Because of $p_n = 1/2$, it holds that $\mathsf{Prob}[\mathsf{Exp}_0^{\mathcal{A}} = 1] - 1/2 \leq 2n^2 \cdot \varepsilon$, which contradicts the initial assumption. ■

5 Lower Bounds for General Broadcast Encryption Schemes

We now turn our attention to the setting of general, unrestricted broadcast encryption schemes. We will prove that any scheme that is private in the sense of priv-st, priv-full has ciphertext length that with reasonably high probability is linear. We denote as $|x|$, the number of bits of the value x.

Theorem 6. *For all the sets* $\mathsf{R} \subseteq [n]$, *we define the random variable*

$$S_{\mathsf{R}} : \mathsf{Encrypt}(ek, m, \mathsf{R}) \rightarrow |\mathsf{Encrypt}(ek, m, \mathsf{R})|,$$

where ek is an encryption key and m is a plaintext chosen from a message space M. *Suppose that Φ is a broadcast encryption scheme with n receivers, and let* R, R' *be two sets. If Φ is private according to* priv-full *definition, then for all* $\mathsf{R}, \mathsf{R}' \subseteq [n]$ *and for all the PPT statistical tests D, it holds that* $\Delta_D[S_{\mathsf{R}}, S_{\mathsf{R}'}] < \varepsilon$.

Proof. Suppose that there exists a pair of sets R, R' and a PPT statistical test D such that $\Delta_D[S_{\mathsf{R}}, S_{\mathsf{R}'}] \geq \alpha$, with α non-negligible. Then, a PPT adversary $\mathcal{A}$ breaks definition priv-full with advantage at least $\alpha/2$ following the steps below.

Phase 1: Challenge R, R'.
Phase 2: On input $\langle m, \mathsf{Encrypt}(ek, m, \mathsf{R}_b)\rangle$:

- Compute $|\mathsf{Encrypt}(ek, m, \mathsf{R}_b)|$.
- Run D on input $|\mathsf{Encrypt}(ek, m, \mathsf{R}_b)|$.
- Return the output of D.

The adversary can execute the algorithm D a number of times in order to understand whether it is biased to 1 on input S_{R} or vice versa. Without loss of generality we assume that D returns 1 with greater probability in case of input $|\mathsf{Encrypt}(ek, m, \mathsf{R}')|$. As a result, we have that

$$\mathsf{Prob}[D(S_{\mathsf{R}'}) = 1] - \mathsf{Prob}[D(S_{\mathsf{R}}) = 1] \geq \alpha.$$

We note that if D is biased to 1 on input S_{R} we can consider the adversary $\overline{\mathcal{A}}$ in order to obtain the same results.

$$\begin{aligned}
\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}}(1^\lambda) = 1] &= \frac{1}{2}\Big(\mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}} = 1 | b = 0] + \mathsf{Prob}[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{priv\text{-}full}} = 1 | b = 1]\Big) \\
&= \frac{1}{2}\Big(\mathsf{Prob}[D(S_{\mathsf{R}}) = 0] + \mathsf{Prob}[D(S'_{\mathsf{R}}) = 1]\Big) \\
&\geq \frac{1}{2} + \frac{\alpha}{2}.
\end{aligned}$$

■

Next, we will prove a lower bound on the ciphertext size that any private broadcast encryption scheme can achieve. Our proof is based on a standard information theoretic fact (cf. [5]), which is presented below:

Fact 1. *Suppose there is a randomized procedure* $Enc : \{0,1\}^n \times \{0,1\}^r \rightarrow \{0,1\}^m$ *and a decoding procedure* $Dec : \{0,1\}^m \times \{0,1\}^r \rightarrow \{0,1\}^n$ *such that*

$$\mathsf{Prob}_{r \in U_r}[Dec(Enc(x,r),r) = x] \geq \delta.$$

Then, $m \geq n - \log \frac{1}{\delta}$.

Theorem 7. *Let* Φ *be a broadcast encryption scheme with* n *receivers and let* $\varepsilon(\lambda)$ *be the upper bound of all the probabilities* $\mathsf{Prob}[E_{\mathsf{R},i}]$. *For any* $\mathsf{R} \subseteq [n]$ *and* $i \in [n]$, *we denote as* $E_{\mathsf{R},i}$ *the event*

$$(\mathsf{Decrypt}(\mathsf{SK}_i, c) \neq m \wedge i \notin \mathsf{R}) \vee (\mathsf{Decrypt}(\mathsf{SK}_i, c) = m \wedge i \in \mathsf{R}),$$

where $c = \mathsf{Encrypt}(ek, m, \mathsf{R})$. *If for any* λ *there exists some* β *for which* $\varepsilon(\lambda) < \frac{1}{2n} - \frac{\beta}{n}$, *then there exists a set* $\mathsf{R} \subseteq [n]$ *such that* $\mathsf{Prob}[S_{\mathsf{R}} \geq n] > \beta$.

Proof. Recall the definition of S_{R}:

$$S_{\mathsf{R}} : \mathsf{Encrypt}(ek, m, \mathsf{R}) \rightarrow |\mathsf{Encrypt}(ek, m, \mathsf{R})|.$$

We define a procedure f which is an encoding procedure of a set $\mathsf{R} \subseteq [n]$, while f^{-1} is a decoding procedure. The procedure f is a randomized procedure that takes as input two arguments $\rho \in \{0,1\}^r$ and $\mathsf{R} \subseteq [n]$ and outputs ψ. We note that ρ depends on the security parameter λ and represents all the coins needed in order for the system to setup and the encryption. The procedures f and f^{-1} are defined as follows:
$f(\rho, \mathsf{R})$:

1. Using ρ, compute a message m and the key ek which will be used by the encryption algorithm.
2. Compute $\mathsf{Encrypt}(ek, m, \mathsf{R})$.
3. If $|\mathsf{Encrypt}(ek, m, \mathsf{R})| \geq n$, output 0^{n-1} else $\mathsf{Encrypt}(ek, m, \mathsf{R})$.

$f^{-1}(\psi, \rho)$:

```
1. Use $\rho$ to compute $\mathsf{SK}_1, ..., \mathsf{SK}_n$.
2. Execute the following algorithm:
   $\mathsf{R} := \emptyset$.
   For $i = 1$ to $n$
           if $\mathsf{Decrypt}(\mathsf{SK}_i, \psi) \neq m$ then $\mathsf{R} := \mathsf{R} \cup \{i\}$ else $\mathsf{R}$.
```

Considering the definition of the decoding procedure, we say that f^{-1} fails when its result is $\mathsf{R}' \neq \mathsf{R}$, given that R is the encoded set. This happens either in case an event $E_{\mathsf{R},i}$ takes place or the output of f is 0^{n-1}. With δ we denote the probability that the procedure f^{-1} succeeds.

In order to prove the theorem, we assume that for any λ for which there exists a β such that $\varepsilon(\lambda) < \frac{1}{2n} - \frac{\beta}{n}$ it holds that $\mathsf{Prob}[S_{\mathsf{R}} \geq n] \leq \beta$ for all $\mathsf{R} \subseteq [n]$. Let us fix a value λ. From the above assumption, we have that $\mathsf{Prob}[f \text{ outputs } 0^{n-1}] \leq \beta$ which subsequently means that $\mathsf{Prob}[f^{-1} \text{ fails }] \leq n \cdot \varepsilon(\lambda) + \beta$. Consequently, we have that $\delta \geq 1 - n \cdot \varepsilon(\lambda) - \beta$.

Due to the fact that the length of the encoding produced by f^{-1} is always $n - 1$ bits at most, using the fact 1, we have that

$$n - 1 \geq n - \log \frac{1}{\delta} \Rightarrow \varepsilon(\lambda) \geq \frac{1}{2n} - \frac{\beta}{n}, \tag{1}$$

which is a contradiction. ■

Lemma 2. *Let Φ be a private broadcast encryption scheme with n receivers and λ a security parameter for which $\beta < 1/2$ and β non-negligible in λ. Then, for all $\mathsf{R} \subseteq [n]$, it holds that $\mathsf{Prob}[S_{\mathsf{R}} \geq n] \geq \alpha$, for α non-negligible.*

Proof. We assume that there exists a set R_0 such that $\mathsf{Prob}[S_{\mathsf{R}_0} \geq n] < \delta$, where δ is a negligible function of λ. We construct the following statistical test D:

D: On input S_{R}: If $S_{\mathsf{R}} \geq n$ return 1 else return 0.

According to the Theorem 7, we have that there exists a set R_1 for which $\mathsf{Prob}[S_{\mathsf{R}_1} \geq n] > \beta$. As a result, we have that

$$\mathsf{Prob}[D(S_{\mathsf{R}_1}) = 1] - \mathsf{Prob}[D(S_{\mathsf{R}_0}) = 1] > \beta - \delta,$$

which is non-negligible. This contradicts to Theorem 6. ■

Corollary 3. *For any broadcast encryption scheme Φ which is private in the sense of definition* priv-full,priv-st*, the ciphertext is of length $\Omega(n + k)$.*

The additive factor k stems from the fact that at least one ciphertext should be present in the encryption of a message m for any enabled set S.

6 Conclusion

The provided lower bounds highlight the high costs that privacy may incur for broadcast encryption schemes. The fact that privacy for atomic schemes requires a linear number of ciphertexts in the number of users, leaves essentially no room for improvement in terms of the ciphertext size. If the objective is to attain full privacy, this result suggests that our attention should be turned to non-atomic schemes. In the non-atomic case, our lower bound is much weaker. It is thus an interesting open problem to design a fully private scheme with sublinear ciphertext size.

References

1. AACS, http://www.aacsla.com/
2. Attrapadung, N., Imai, H.: Graph-Decomposition-Based Frameworks for Subset-Cover Broadcast Encryption and Efficient Instantiations. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 100–120. Springer, Heidelberg (2005)
3. Barth, A., Boneh, D., Waters, B.: Privacy in Encrypted Content Distribution Using Private Broadcast Encryption. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 52–64. Springer, Heidelberg (2006)
4. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
5. De, A., Trevisan, L., Tulsiani, M.: Time Space Tradeoffs for Attacks against One-Way Functions and PRGs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 649–665. Springer, Heidelberg (2010)
6. Delerablée, C.: Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 200–215. Springer, Heidelberg (2007)
7. Dodis, Y., Fazio, N.: Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
8. Fazio, N., Perera, I.M.: Outsider-anonymous broadcast encryption with sublinear ciphertexts. In: Fischlin, et al. [10], pp. 225–242
9. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
10. Fischlin, M., Buchmann, J., Manulis, M. (eds.): PKC 2012. LNCS, vol. 7293, pp. 2012–2015. Springer, Heidelberg (2012)
11. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient Tree-Based Revocation in Groups of Low-State Devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
12. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
13. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In: Fischlin, et al. [10], pp. 206–224
14. Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
15. Shoup, V.: A proposal for an iso standard for public key encryption. IACR Cryptology ePrint Archive 2001, 112 (2001)
16. Wang, P., Ning, P., Reeves, D.S.: Storage-Efficient Stateless Group Key Revocation. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 25–38. Springer, Heidelberg (2004)

The Dangers of Composing Anonymous Channels

George Danezis[1] and Emilia Käsper[2]

[1] Microsoft Research
[2] Google
gdane@microsoft.com, ekasper@google.com

Abstract. We present traffic analyses of two anonymous communications schemes that build on the classic Crowds/Hordes protocols. The AJSS10 [1] scheme combines multiple Crowds-like forward channels with a Hordes reply channel in an attempt to offer robustness in a mobile environment. We show that the resulting scheme fails to guarantee the claimed k-anonymity, and is in fact more vulnerable to malicious peers than Hordes, while suffering from higher latency. Similarly, the RWS11 [15] scheme invokes multiple instances of Crowds to provide receiver anonymity. We demonstrate that the sender anonymity of the scheme is susceptible to a variant of the predecessor attack [21], while receiver anonymity is fully compromised with an active attack. We conclude that the heuristic security claims of AJSS10 and RWS11 do not hold, and argue that composition of multiple anonymity channels can in fact weaken overall security. In contrast, we provide a rigorous security analysis of Hordes under the same threat model, and reflect on design principles for future anonymous channels to make them amenable to such security analysis.

1 Introduction

The design of anonymous communication channels is a well established field [4] with applications in election protocols, censorship resistance and support of free speech. Several proposed anonymous communications channels take advantage of specific networking layers, such as ISDN telephony [14], TCP [9], email [6] or ad-hoc networks [11]. Similarly, the AJSS10 [1] channel is crafted for use in hybrid mobile networks, where a local peer-to-peer network of mobile devices (using WiFi) is provided with wide area connectivity by a mobile (3G, GPRS) telephony provider.

The degree of security for an anonymity system comes down to the difficulty an adversary has in identifying the originators or intended receivers of messages. The security guarantees can be formalized by probabilities over senders or receivers [19], or by equivalence classes of possible actors (anonymity sets) [13]. A probability distribution over all possible actors yields more fine-grained guarantees than a division of the actors into equivalence classes, and probabilistic modelling is thus also the approach we take in this paper.

The capabilities of adversaries are represented by different threat models against which the security of an anonymity system must be evaluated, such

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 191–206, 2013.

as the global passive adversary [12], a fraction of malicious insiders [16], or specific corrupt entities. All schemes discussed in this paper aim to protect the identity of the sender from a malicious receiver as well as malicious insiders, while the RWS11 [15] scheme takes a step further by attempting to guarantee the anonymity of the receiver.

Performance metrics—communication overhead and latency as well as reliability—are important for anonymous communications as for any other networking primitive. Crucially, low performance of the communication channel leads to low usability of the system, which might in itself reduce anonymity [8]. Attacks on performance combined with retransmission mechanisms to ensure reliability may in fact lead to loss of anonymity [2].

Our contribution in this paper comprises three parts. We start by reviewing the classic Crowds [16] and Hordes [17] systems. We augment the security analysis of Hordes by proving new analytic bounds on sender anonymity in the case of a malicious receiver controlling a subset of nodes in the network. We proceed by studying the security and performance of two schemes that build on Crowds, AJSS10 [1] and RWS11 [15], and in both cases, demonstrate how the composition of multiple anonymous channels results in weaker anonymity as well as poorer performance. Finally, we reflect on our findings to provide a set of sanity checks for protocol designers. The key intuition behind our results is that multiple anonymous channels in general compose poorly: one cannot automatically argue that since they each separately provide some degree of anonymity, their composition will provide at least the same level of assurance. Further, while we are able to analytically bound the security of Crowds and Hordes, seemingly simple changes to those protocols make their modelling intractable. We argue that building anonymous communications channels whose security is easy to verify analytically should be a key design consideration for future proposals.

2 Crowds and Hordes

Crowds [16] uses a peer-to-peer network (a crowd) to pass messages anonymously from the sender to a receiver outside the crowd. A crowd member wishing to send a message first passes it to a random crowd node. Each subsequent node then flips a (biased) coin to decide whether to send the message to the destination (with probability p) or pass it to another crowd node. The latency of the channel, that is, the average number of hops a message travels in the crowd before being forwarded to the final destination is $1/p$.

The forward path of a Crowds message can also be used to anonymously receive replies. Crowds' bidirectional communication, however, is not robust in a mobile environment where nodes join and leave the crowd dynamically. Hordes [17] provides a solution by combining the forward path of Crowds with a multicast reply channel. In Hordes, the sender appends to the message a random session identifier as well as a *multicast group*—a subset of k crowd nodes that includes the sender herself. The outgoing message is then sent via a regular Crowds channel. The reply message, which must also include the session identifier, is sent directly to the multicast group acting as an anonymity set.

The original sender, who is part of this group, can use the session identifier to detect the reply, while the remaining members of the group will simply drop the message. The crowd forward latency of Hordes is still $1/p$, while the fast direct reply channel means that the latency of a round trip drops from $2/p$ to $1/p$.

2.1 Security Analysis

Throughout this paper, we measure sender anonymity in terms of probabilities linking the message to its sender. While it is generally prudent to consider the worst-case confidence an adversary has in the true sender, taken over all possible observations of messages, we shall shortly see that for the protocols scrutinized in this paper, the worst-case confidence can be 1. Thus, we resort to measuring the expected success rate of the adversary guessing the sender of a message.

We say that a system offers perfect anonymity if for any observation, the adversary's best strategy is to choose at random from the entire set of n possible senders: $\Pr[\mathsf{success}] = 1/n$. Crowds provides the sender with perfect anonymity with respect to an adversarial receiver, since the receiver is equally likely to receive the message from any crowd member. Collaborating dishonest crowd members, on the other hand, can infer some information about the sender. More specifically, in a crowd of size n, a fraction $0 < f < 1$ of dishonest nodes can correctly guess the sender of a captured message with expected success rate [16]

$$\mathbb{E}(\mathsf{success}) = 1-(1-f)(1-p)+(1-f)(1-p)\frac{1}{(1-f)n} = f+(1-f)p+\frac{1-p}{n}\,. \quad (1)$$

Hordes provides k-anonymity of the sender w.r.t. the receiver: the receiver will only learn that the sender is one of the k members in the multicast group, $\Pr[\mathsf{success}] = 1/k$. Assuming communication between the sender and the receiver is encrypted such that intermediate nodes do not learn the multicast group, Hordes provides the same sender anonymity w.r.t. malicious crowd nodes as original Crowds.

Finally, we consider sender anonymity in the case of malicious crowd nodes collaborating with a malicious receiver. A malicious Crowds receiver contributes no additional information, implying that for messages captured in the crowd, the success of this adversary is still bounded from above by (1). The average success rate over all observed messages, including those captured by the receiver, is

$$\mathbb{E}(\mathsf{success}) = f + (1-f)\frac{1}{(1-f)n} = f + \frac{1}{n}\,. \quad (2)$$

For Hordes, this adversary model is not considered in the original paper, however, the simplicity of the protocol allows us to now devise a strict bound for sender anonymity in this model (see Appendix A for the proof).

Theorem 1. *In a crowd of n nodes, a fraction $0 < f = \frac{c}{n} < 1$ dishonest crowd members collaborating with a dishonest receiver can identify the sender of a Hordes message with average success rate*

$$\mathbb{E}(\mathsf{success}) = f + \frac{1}{k}\left(1 - \frac{\binom{c}{k}}{\binom{n}{k}}\right) < f + \frac{1}{k}\,,$$

where k is the size of the multicast group chosen by the sender.

The effect of the security parameter k on the anonymity of Hordes is as expected: anonymity improves as k increases, and as k approaches n, the security of Hordes approaches that of Crowds (c.f. ([2])). While the latency of a single message does not increase with k, a large multicast set causes heavier total load on the network, implying a natural security-performance trade-off.

Curiously, the Crowds parameter p does not influence the average success rate of the adversary; it does, however, affect the worst-case confidence the adversary has in the sender of a particular message: as the exit probability p approaches 1, fewer messages get captured before reaching the receiver, but those that do can be attributed to the sender with higher confidence, and vice versa. Finally, we observe that for $k \leq 1 + fn$, Hordes provides no anonymity in the worst case as a particularly poor choice of the multicast set can result in the entire set colluding with the receiver, thus pinpointing the single honest member in the set as the sender (see the proof in Appendix A for details).

2.2 Advantages and Limitations

The security of Crowds and Hordes depends on the forwarding parameter p, as well as, in the case of Hordes, the size of the multicast group k. Both parameters have a negative impact on the latency of the protocol. In both cases, however, this relationship between latency and security is well understood and proven by rigorous security analysis. Indeed, the local randomized algorithm that determines the latency of a Crowds message provides optimal security [5].

On the other hand, both protocols are limited by only providing anonymity of the sender *with respect to the receiver*: neither can withstand attacks by even a passive network adversary that can eavesdrop communications in the crowd. Furthermore, the predecessor attack [21] can breach sender anonymity when forwarding paths are frequently resampled, and crowd paths should thus be fixed during a session between a sender and a receiver. We proceed to show that a failure to fully consider these limitations results in complex protocols that are harder to analyze, yet provide weaker anonymity.

3 The AJSS10 Scheme

We present here the features of the AJSS10 channel relevant to its study in terms of security and performance. Full details are provided in the original work [1].

AJSS10 is designed to provide anonymity in a hybrid networking environment, where local devices can communicate with one another using a local

Wifi network, but need to communicate with the wider area network through a mobile telephony operator – this is an appealing model as it matches the capabilities of smartphones which now outnumber PCs in sales[1].

The AJSS10 protocol aims to achieve sender k-anonymity: the adversary should not be able to reduce the number of possible senders of a message to less than k participants. To construct her anonymity set, a sender using AJSS10 splits her message into k parts.[2] She sends one part directly to the operator and $k - 1$ parts via other peers, using a variant of Crowds for each part. First, each message part is given to a random node in the local network; each receiving node flips a biased coin and with probability p sends the message to the operator *unless it has already sent a part of this message to the operator*; otherwise the message is sent to a random node in the network that repeats this process.

AJSS10 also attempts to conceal receiver anonymity from peers by using, for each message part, a different temporary identifier that only the operator can map to the recipient. Upon receiving a message part from a node, the operator thus looks up its destination address and forwards it to the receiving server. After all k parts have arrived, the server replies to the set of k peers (which, by design, includes the sender) and includes a server ID in that reply. The multicast reply channel of AJSS10 shares two properties with that of Hordes: the set of k peers aims to provide k-anonymity; while the inclusion of the original sender in this set guarantees robust reply message delivery in networks where peers join and leave dynamically.

The key difference between AJSS10 and Hordes routing is the use of multiple parallel paths on the forward channel, which has two effects. First, multiple paths are used to relay messages from the same sender to the same receiver—thus, we expect more opportunities for adversaries to perform traffic analysis. And second, the fact that the same node cannot send more than one part of the same message to the operator requires the inclusion of a message identifier visible to all peers, allowing them to link different parts of the same message together.

These changes also have repercussions on performance: all parts of the message need to be delivered for the message to be decoded, increasing protocol latency. Furthermore, some nodes will not be able to output a part as they have already delivered a previous part to the operator, forcing the message to continue its course within the network.

We note that the Hordes system assumptions must be satisfied to run the AJSS10 algorithm. Hordes relies on the sender directly choosing $k - 1$ other peers to build a reply anonymity set and sending this set to the server. AJSS10 uses a variant of Crowds which implies that a client knows all other local peers in the crowd and can create these anonymity sets as in Hordes. Conversely, Hordes can also be used when peers join and leave the network dynamically, as the reply channel relies on a multicast that can safely fail for some peers. The question we

[1] `http://www.pcmag.com/article2/0,2817,2379665,00.asp`

[2] An estimate of dishonest peers can be used to choose a higher parameter to achieve k-anonymity, taking into account some of the potential members of the anonymity set may be under the control of the adversary.

ask next is thus whether AJSS10 offers significant security benefits that outweigh the additional performance cost compared to Hordes.

3.1 Security Evaluation

We consider the security of the scheme against two threat models: a set of malicious peers in the wifi network, as well as a malicious operator that additionally controls a small fraction of peers.

The goal of the adversary in our analysis is to determine the most likely sender of a specific message for a set of observations of the anonymity network. The observations of malicious peers contain the identities of nodes that forwarded a message they wish to trace. In the second threat model, a malicious operator additionally records the identities of forwarding nodes for messages that do not get captured in the crowd. Our analysis relies on the following observations:

- Malicious nodes can link received message parts belonging to the same message together by their unique identifier. Similarly to the predecessor attack on dynamic Crowds paths [21], this results in an attack on sender anonymity, as the true sender will be observed on the paths of the message parts more often than any intermediate node. Even worse, while the predecessor attack relies on implicit identifiers such as user ID-s, cookies or other information at the application layer for linking messages together, these identifiers are explicitly included by design in AJSS10.
- A malicious operator cannot use this unique identifier as AJSS10 requires honest senders to strip the message parts of linkable information before giving them to the operator. Yet the server ID together with timing information may be sufficient in linking all parts back together in case of requests to relatively unpopular resources.
- Unlike Hordes, the receiver identity is initially concealed from peers; however, the server ID is also available in the reply message sent to the multicast group, allowing coalitions of malicious peers to restore the link between the message and its recipient. A successful attack on sender anonymity thus also results in sender-receiver linkability.

The state of the art in traffic analysis of anonymity protocols involves a full probabilistic modelling of the channel to extract posterior distributions over actions given the knowledge of an adversary [19]. Unlike Crowds, as well as Hordes which we modelled in Sect. 2.1, such an analysis is extremely complex for the AJSS10 protocol as modifications to the Crowds routing logic introduce temporal constraints. Instead, we provide an experimental upper bound on the security of the scheme by simulating the protocol and using heuristic analysis to decide upon the most likely sender of a message. The experimental bound is from above, as only partial information is used by our adversary to determine the sender (timing information is excluded), and even that partial information is not guaranteed to be used optimally.

Given a set of observed senders corresponding to parts of the same message, we simply pick as the most likely initial sender the peer that has sent the most

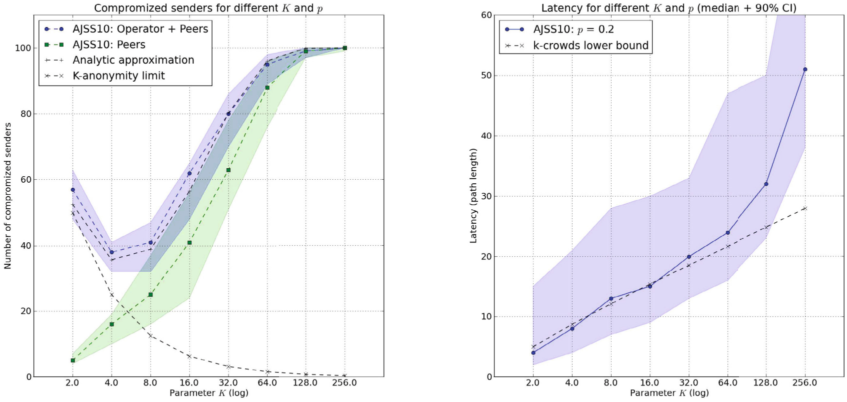


Fig. 1. The security and performance of AJSS10 versus the Hordes protocol

parts. If the operator colludes with malicious peers, we further restrict the sender to be within the set of honest peers that forwarded the message to the operator, since the true sender is guaranteed to be in the k-anonymity set observed by the operator. In general, the higher the fraction of malicious peers in the network, the higher the probability that the actual sender of the message is observed multiple times. Surprisingly, the same is true for larger values of the security parameter k.

We can reason about the success probability of our heuristic algorithm as follows. Assuming peers collaborate with the operator, capturing at least one of the $k-1$ message parts sent via peers during its first hop is likely to identify the correct sender (who is then linked with at least two message parts); conversely, if no parts land in the hands of malicious peers during the first hop then the adversary's guess will effectively be a random choice from the anonymity set. In Appendix B, we estimate the success rate of a fraction $f = c/n$ corrupt peers collaborating with the operator to be

$$\mathbb{E}(\mathsf{success}) \approx 1-(1-f)^{k-1} + \frac{(1-f)^{k-2}}{k}\left(1-\frac{\binom{c}{k}}{\binom{n}{k}}\right).$$

Figure 1 (left) compares the anonymity provided by the AJSS10 scheme in the two threat models with the anonymity provided by Hordes in the stricter model when both operator and peers are malicious, denoted as the k-anonymity limit. Specifically, we consider a network of 500 peers out of which 25 are malicious. We perform 100 simulations for $p \in [0.1, 0.9]$ and plot the number of those experiments in which the actual sender was correctly identified. The solid lines are the median number of successes and the shaded regions represent the minimum and maximum number of successes for different values of p. The analytic approximation for the security of AJSS10 is also plotted.

The complexity of the AJSS10 routing logic prevents us from accounting for the exact effect of the Crowds parameter p. We note that decreasing p increases latency, which results in more messages being captured before reaching the operator, thus on one hand increasing the number of different senders observed by peers, while on the other hand decreasing the size of the effective anonymity set of honest nodes observed by the operator. Our simulations confirm that varying p does not affect the security guarantees of AJSS10 significantly, while anonymity decreases rapidly as k increases, Even if only peers are malicious, even modest values of k lead to a greater probability of compromise than k-anonymity would suggest. In comparison, as k increases, the security of the Hordes channel increases together with the size of the anonymity set, as expected.

3.2 Performance Evaluation

The modifications introduced by the AJSS10 scheme also have a serious impact on the latency of messages. Traditional Crowds latency follows a geometric distribution [5], which has a high variance. Requiring the delivery of $k-1$ message parts through the crowd in effect involves sampling $k-1$ random variables for the latency of each part, leading to the *maximum* delivery time being the latency of the whole message. The expected maximum latency of $k-1$ independent random variables following the geometric distribution with parameter p is

$$\mathbb{E}(l^{\mathsf{Geom}}_{k-1,p}) = \sum_{i=1}^{k-1} \frac{\binom{k-1}{i}(-1)^{i-1}}{1-(1-p)^i} \geq -\frac{H_{k-1}}{\log(1-p)}\,, \qquad (3)$$

where H_{k-1} is the $(k-1)$th harmonic number, yielding a lower bound for the expected latency of an outbound message in the AJSS10 scheme. The fact that peers only deliver a single message further increases the delivery time and prohibits us from giving an *upper* bound to the latency.

Figure 1 (right) illustrates latency for AJSS10 with different parameters k and fixed $p = 0.2$ compared to Crowds/Hordes (median and 90% confidence intervals). It is clear for AJSS10 that as the parameter k increases, latency also increases to about an order of magnitude above Crowds. We also plot the theoretical lower bound (3) on the latency. As observed, assuming the AJSS10 system behaves like $k-1$ parallel Crowds is a good model for $k << n$ but path lengths increase significantly beyond this bound as k becomes larger, due to each peer being restricted to only sending out one message part.

4 The RWS11 Scheme

The RWS11 scheme [15] also uses parallel Crowds paths, this time to conceal the identity of the receiver from the crowd. To send a message m to a receiver R_r, the sender first constructs a path $R_1, R_2, \ldots, R_{r-1}, R_r$ of crowd nodes (in RWS11, the receiver is considered part of the crowd). She computes, for each node R_i on this path, a set of messages $s_i^{(0)}, s_i^{(1)}, \ldots, s_i^{(k-1)}$ and padding values such that

$$\begin{array}{lcl} s_1^{(0)} \quad\quad\quad\;\; \oplus s_1^{(1)} \;\;\, \oplus \cdots \oplus s_1^{(k-1)} & = & R_2 \mid\mid s_2^{(0)} \\ s_2^{(0)} \mid\mid \mathsf{pad}_1 \;\;\; \oplus s_2^{(1)} \;\;\, \oplus \cdots \oplus s_2^{(k-1)} & = & R_3 \mid\mid s_3^{(0)} \\ & \cdots & \\ s_{r-1}^{(0)} \mid\mid \mathsf{pad}_2 \;\;\; \oplus s_{r-1}^{(1)} \oplus \cdots \oplus s_{r-1}^{(k-1)} & = & R_r \mid\mid s_r^{(0)} \\ s_r^{(0)} \mid\mid \mathsf{pad}_{\mathsf{r-1}} \oplus s_r^{(1)} \;\;\, \oplus \cdots \oplus s_r^{(k-1)} & = & \mathsf{null} \mid\mid m\,. \end{array} \tag{4}$$

Padding is used to replace the removed address field, so that the size of the message remains constant throughout its course in the network. The padding values $\mathsf{pad}_1, \mathsf{pad}_2, \ldots, \mathsf{pad}_{r-1}$ are defined such that node R_i can compute $\mathsf{pad}_i = f(s_i^{(0)}, s_i^{(1)}, \ldots, s_i^{(k-1)})$ as a pseudorandom function of her shares.

The sender then forwards $s_1^{(0)}$ to R_1, as well as $s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(k-1)}$ for $i = 1 \ldots r$ to R_i, using Crowds for each share. The node R_i, upon receiving her k shares—one share from the previous node as well as $k-1$ shares from the sender—reconstructs $R_{i+1}||s_{i+1}^{(0)}$ and pad_i, and forwards $s_{i+1}^{(0)}||\mathsf{pad}_i$ to R_{i+1}, again using Crowds. The final receiver R_r, upon seeing null in the address field, thus knows that the message was intended for her.

The core idea behind the security of RWS11 is that all k shares $s_r^{(i)}$ intended for R_r are required to learn that R_r is the final receiver, and her identity thus remains hidden from malicious nodes who only intercept some of the shares. We proceed to show, however, that this construction significantly weakens sender anonymity, and further demonstrate active attacks on receiver anonymity.

4.1 Security Evaluation

Sender Anonymity. As before, we first consider the threat model where a malicious receiver R_r collaborates with a subset of malicious crowd nodes in order to learn the identity of the sender. Our observations on RWS11 are similar to those of AJSS10:

- Observing that $k-1$ of the k shares meant for R_r originate from the initial sender, we expect to see the true sender on the path to R_r more often than any other node.
- While RWS11 does not explicitly specify this, the k shares must include a common identifier that allows R_r to link the observed parts back together, making it easy for an attacker to distinguish shares belonging to the same message (*case 1*). In the absence of noise caused by other traffic, all kr shares can be linked together (*case 2*).

Again, if we assume that the adversary is successful with probability ≈ 1 in identifying the sender whenever she captures at least two message shares directly from the sender, we can predict the overall success rate to be

$$\mathbb{E}(\mathsf{success}) \approx 1 - (1-f)^{k-1} - (k-1)f(1-f)^{k-2} \text{ in case 1;} \tag{5}$$

$$\mathbb{E}(\mathsf{success}) \approx 1 - (1-f)^{r(k-1)+1} - (r(k-1)+1)f(1-f)^{r(k-1)} \text{ in case 2.} \tag{6}$$

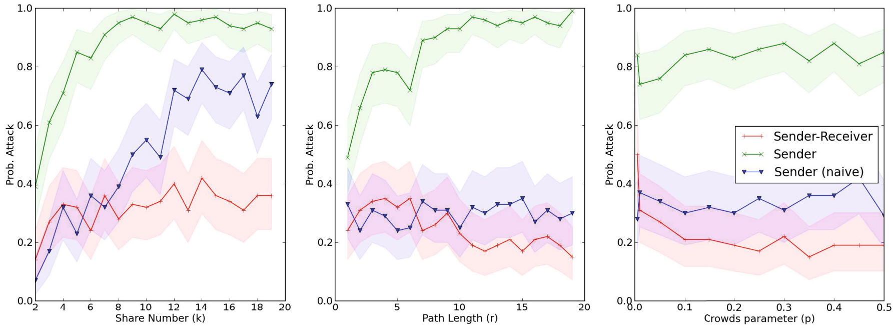


Fig. 2. Attack probabilities against the RWS11 protocols for different values of the security parameters. Green lines ("x" ticks) represent attacks against sender anonymity with a corrupt receiver; Blue lines (triangle ticks) represent attacks against a sender only using the shares sent to a corrupt receiver; Red lines ("+" ticks) represent attacks against an honest sender and receiver. (Shaded regions represent the 99% confidence intervals.

Sender-Receiver Anonymity. Second, we assume the receiver is honest, and consider the security of the RWS11 scheme against a delaying adversary that is otherwise passive. Strictly speaking, this adversary is outside the scope of RWS11, which assumes only a purely passive adversary [15]. But an adversary that can merely delay any observed traffic and is otherwise passive is a realistic extension, thus instructive to consider. As before, we assume that message parts are linkable, either through a common header, or simply through lack of noise.

We note that an adversary that observes shares from a specific sender is left with the task of identifying the ultimate receiver of the message out of potentially r choices. We use causality, namely that R_{i+1} cannot output a message before receiving her share from R_i, and that the receiver R_r is the last in this chain, to attack the scheme.

An adversary delays all shares received within the crowd for a specific time frame. These messages are considered to be within the same, first, epoch. The adversary then releases the shares and observes the sequence of captured shares as the message continues its course through the sequential chain of intermediate nodes to R_r. This allows the adversary to build an ordering over the observed potential receivers of the target message. Any receiving peers observed before the last observation can be discarded as potential receivers.[3]

As before, the adversary again selects as a candidate sender, the sender that sent most messages within the delay period. The candidate receiver is selected at random from the set of potential receivers observed, minus adversary peers and peers that were not last in the observed route.

[3] The original protocol does not specify whether the r nodes must be distinct; for simplicity, we assume they are. The attack also works if repetitions are allowed, since message part identifiers allow us to consider each appearance of a node separately.

Given the complexity of the attack, we simulate runs of the RWS11 protocol and perform the attack against them to evaluate its true success probability. This provides an *upper bound* on its security, as our heuristic adversary is, once again, not guaranteed to be optimal. The results are summarised in Figure 2 as the red line ("+" ticks). (Standard parameters used: 1000 peers including 10% corrupt; $k = 5$ shares; $r = 5$ path length; $p = 0.05$.) The figure illustrates the security of RWS11 as we vary its security parameters k, r and p.

Figure 2 also plots the success probability of the first attack with a malicious receiver. The green line ("x" ticks) assumes all message parts are linkable, while the blue line only uses the k shares destined for the corrupt receiver. As predicted by (5) and (6), the adversary is very likely to trace the sender for high k or r, while the average success rate is largely independent of p.

Broadly, increasing the security parameters introduced by RWS11 does not result in an significant increase in security. Surprisingly, we observe quite the opposite: as k and r increase, attacks become more successful. For no values of the security parameters does this probability becomes negligible.

Receiver Anonymity. Finally, we should not exclude the possibility that some crowd nodes may actively misbehave, intercepting as well as modifying and injecting messages in the network. State-of-the-art packet formats for anonymous communications offer provable security against active attacks [7], and conversely, heuristic security claims may not cover design flaws leading to replay and oracle attacks that completely foil the purported security of the system [3,20].

Lack of integrity protection in the RWS11 protocol opens up way to an oracle attack whereby any single corrupt node C on the path of any message part to R_i can determine whether R_i is the final receiver of the message. Namely, upon receiving a Crowds message $s_i^{(j)}$ meant for R_i, C sets $\hat{s}_i^{(j)} = s_i^{(j)} \oplus (C||\mathsf{null})$ and forwards the modified message $\hat{s}_i^{(j)}$ to R_i. By Eq. (4), upon receiving her k shares, R_i reconstructs

$$s_i^{(0)}||\mathsf{pad}_i \oplus s_i^{(1)} \oplus \cdots \oplus \hat{s}_i^{(j)} \oplus \cdots \oplus s_i^{(k-1)} = (R_{i+1} \oplus C)||(s_{i+1}^{(0)} \oplus \mathsf{null}),\ i+1 \neq r$$
$$s_i^{(0)}||\mathsf{pad}_i \oplus s_i^{(1)} \oplus \cdots \oplus \hat{s}_i^{(j)} \oplus \cdots \oplus s_i^{(k-1)} = (\mathsf{null} \oplus C)||(m \oplus \mathsf{null}),\ i+1 = r\,.$$

Thus, if R_i is the final receiver, the message m gets routed back to C, else the share $s_{i+1}^{(0)}$ gets routed to a random address $R_{i+1} \oplus C$. The node C, upon receiving m, will thus know that R_i was the true receiver[4]. C can easily distinguish the reply message m from an ordinary RWS11 message, for example by observing that it has no other matching parts.

Receiver anonymity is thus compromised whenever either R_{r-1} is corrupt, or at least one of the k Crowds messages $s_r^{(i)}$ meant for R_r gets captured by a corrupt node. From the analysis of Crowds, we know that the probability of a single message being captured before reaching the receiver is $f/(1-(1-f)(1-p))$, so the success rate of the attacker is

[4] As a bonus, C will learn the contents of the message m.

$$\mathbb{E}(\mathsf{success}) = 1 - \left(1 - \frac{f}{1-(1-f)(1-p)}\right)^k \cdot (1-f) = 1 - \frac{p^k(1-f)^{k+1}}{(p+f-pf)^k},$$

which approaches 1 quickly as k grows. In particular, the success probability of the active adversary is higher than the probability of a passive adversary learning the identity of the receiver in traditional Crowds by simple eavesdropping, so by attempting to protect receiver anonymity against the passive adversary, RWS11 in fact opens up an opportunity for a much more powerful active attack.

4.2 Performance Evaluation

Similarly to AJSS10, the RWS11 scheme requires the delivery of k message parts for each hop through parallel Crowds channels. By eq. (3), the expected latency for R_1 to receive all k parts is $\mathbb{E}(l_{k,p}^{\mathsf{Geom}}) \geq -H_k/\log(1-p)$, and the expected latency of the channel is thus bounded from below by

$$\mathbb{E}(l_{k,r,p}^{\mathsf{RWS11}}) \geq -\frac{H_k}{\log(1-p)} + (r-1)\frac{1}{p},$$

where $(r-1)/p$ is the expected latency of the $r-1$ sequential hops from R_1 to R_r. The bound is loose, as we ignore the delay effect of the remaining $k-1$ shares per hop that are delivered in parallel.

5 Design Principles for Anonymous Channels

We have seen that as the parameter k increases, both the quality of protection and performance of the AJSS10 and RWS11 schemes deteriorate. This is true for both threat models considered, while the simpler Crowds and Hordes schemes provide higher security even in the stringent threat model where operator and peers collaborate. Following our analysis we draw a few conclusions regarding design principles for robust and secure anonymity systems.

Composition. Composing secure anonymous channels does not guarantee that the resulting channel will be secure. We have seen how k-anonymous multicast and Crowds on their own are secure, but running multiple instances of Crowds in parallel is in itself fragile. Leaking further information through the k-anonymous reply channel is significantly weaker than any of the channels on their own. Failure to account for this lead the designers of RWS11 to assume there is no need to analyse sender anonymity at all. The literature on predecessor attacks [21] and disclosure attacks [10] provides a guide to understanding how parallel composition of channels leaks information.

Security Parameters. It is important to specify the security parameters and ensure that security increases as they increase. For instance, k-anonymous channels should provide better protection as k grows. Instead, we demonstrate that the security of AJSS10 and RWS11 decreases significantly as the security parameter k increases. This counter-intuitive result was first observed for Crowds itself a decade ago [18]. Thus any scheme, especially any scheme building on Crowds, should be designed mindful of the possibility. In comparison, the parameter p, which is the traditional Crowds security parameter, has little effect on the security of the new schemes.

Security Assumptions. It is crucial to distinguish the security-relevant state from incidental operational noise. Robust security analysis should assume that the adversary knows all security-irrelevant state. For example, the AJSS10 scheme does not disclose the message identifier to the operator, presumably in an attempt to keep separate message parts unlinkable. While this is prudent, it is a fragile security assumption, as the designers or users of the system have no way of ensuring that more than one message will be sent to a specific server. If only one message is sent to the server, then the identity of the server itself acts as an identifier that links the message parts. Similarly, RWS11 makes no explicit assumptions about delaying or mixing messages at the intermediate nodes, or the sender delaying messages. Thus, it is prudent to assume an adversary should be given information that links those messages together through timing when performing a security analysis. The assumption that the adversary is provided with all non-security-related information when attacking a system is common place in cryptology (through the use of artificial oracles), but not well established in the design of anonymity systems.

Threat Modelling. When modelling the adversary, we must always consider the possibility that all malicious parties collaborate. Crowds-like peer-to-peer systems that attempt to protect the identity of the sender from the receiver as well as the crowd must thus provide protection against a malicious receiver controlling a subset of crowd nodes. Further, adversarial behaviour is unpredictable, and designs whose security collapses completely in the presence of an active adversary are too fragile for general purpose applications.

Ease of Analysis. Schemes should be designed so that the security of the system can be analysed on the basis of a small amount of security state, assuming arbitrary values for the non-security relevant incidental operational noise. The AJSS10 mechanism makes such an analysis difficult: since a peer will never forward to the operator a second part of the same message, a race condition occurs. To analyse the performance and security of the system in an exact fashion one would need to introduce models of timing and network delays, and perform inference over different traces and timings of message transmissions. This is impractical, making the system difficult to analyse, without greatly improving its security.

Compare with Simple Designs. Finally, no anonymous channel is perfect, but some are better than others. For this reason it is important to compare new proposals with previous ones, making small modifications to existing protocols and comparing them all the time to ensure additional complexity in fact provides the advantages hoped for. For example the AJSS10 channel operational constraints, in terms of churn or knowledge of local peers, allow the application of the simple Hordes protocol, and as such Hordes can be used as a baseline for evaluating its security.

Acknowledgments. The authors would like to thank the anonymous reviewers for their comments, as well as the paper shepherd Paul Syverson for great advice on improving the work.

References

1. Ardagna, C.A., Jajodia, S., Samarati, P., Stavrou, A.: Providing Mobile Users' Anonymity in Hybrid Networks. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 540–557. Springer, Heidelberg (2010)
2. Borisov, N., Danezis, G., Mittal, P., Tabriz, P.: Denial of service or denial of security? In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 92–102 (2007)
3. Danezis, G.: Breaking Four Mix-Related Schemes Based on Universal Re-encryption. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 46–59. Springer, Heidelberg (2006)
4. Danezis, G., Diaz, C.: A survey of anonymous communication channels (2008)
5. Danezis, G., Diaz, C., Käsper, E., Troncoso, C.: The Wisdom of Crowds: Attacks and Optimal Constructions. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 406–423. Springer, Heidelberg (2009)
6. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a type iii anonymous remailer protocol. In: Proceedings 2003 Symposium on Security and Privacy, pp. 2–15 (2003)
7. Danezis, G., Goldberg, I.: Sphinx: A compact and provably secure mix format. In: Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P 2009), May 17-20, pp. 269–282. IEEE Computer Society Press (2009)
8. Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: Designing Security Systems That People Can Use. O Reilly Media (2005)
9. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th Conference on USENIX Security Symposium, vol. 13, p. 21 (2004)
10. Kesdogan, D., Agrawal, D., Penz, S.: Limits of Anonymity in Open Environments. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 53–69. Springer, Heidelberg (2003)
11. Kong, J., Hong, X.: ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing, pp. 291–302 (2003)
12. Murdoch, S., Zieliński, P.: Sampled traffic analysis by internet-exchange-level adversaries. In: Proceedings of the 7th International Conference on Privacy Enhancing Technologies, pp. 167–183 (2007)
13. Pfitzmann, A., Köhntopp, M.: Anonymity, Unobservability, and Pseudonymity - A Proposal for Terminology. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 1–9. Springer, Heidelberg (2001)
14. Pfitzmann, A., Pfitzmann, B., Waidner, M.: ISDN-MIXes: Untraceable communication with very small bandwidth overhead. In: Proceedings of the GI/ITG Conference on Communication in Distributed Systems (1991)

15. Rass, S., Wigoutschnigg, R., Schartner, P.: Doubly-anonymous crowds: Using secret-sharing to achieve sender- and receiver-anonymity. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 7(4), 25–39 (2011)
16. Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security (TISSEC) 1(1), 66–92 (1998)
17. Shields, C., Levine, B.N.: A protocol for anonymous communication over the internet. In: Gritzalis, D., Jajodia, S., Samarati, P. (eds.) CCS 2000, Proceedings of the 7th ACM Conference on Computer and Communications Security, November 1-4, pp. 33–42. ACM (2000)
18. Shmatikov, V.: Probabilistic analysis of anonymity. In: CSFW, pp. 119–128. IEEE Computer Society Press (2002)
19. Troncoso, C., Danezis, G.: The bayesian traffic analysis of mix networks. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 369–379 (2009)
20. Westermann, B., Kesdogan, D.: Malice versus AN.ON: Possible Risks of Missing Replay and Integrity Protection. In: Danezis, G. (ed.) FC 2011. LNCS, vol. 7035, pp. 62–76. Springer, Heidelberg (2012)
21. Wright, M.K., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: An analysis of a threat to anonymous communications systems. ACM Trans. Inf. Syst. Secur. 7(4), 489–522 (2004)

A Proof of Theorem 1

The intuition behind our analysis is this: if the sender forwards the message to a malicious peer during the first hop, the adversary can correctly attribute the message to the sender. In all other cases, the adversary's guess is random.

Adversary Strategy. Given an observed message obs and a probability distribution $\Pr[\mathsf{Sender} = s_i | \mathsf{obs}]$ over all possible senders s_i, the optimal adversarial strategy is to choose the most likely sender by $\max_i \Pr[\mathsf{Sender} = s_i | \mathsf{obs}]$. If multiple senders are equally likely, there exists no better strategy than to choose one of them at random.

Let $\mathcal{H}$ be the subset of honest peers in the multicast group. Since the receiver is malicious, $\mathcal{H}$ is known to the adversary, and her complete observation is $\mathsf{obs} = (s_j, \mathcal{H}, c)$, where s_j is the honest node that forwarded the message and $c \in \{\mathsf{crowd}, \mathsf{recv}\}$ indicates whether the message was captured by a crowd node, or only at the receiver. We group all possible observations into three groups.

Type 1. The message is captured in the crowd, and the forwarding node s_j is a member of the multicast group $\mathcal{H}$: $\mathcal{O}_1 = \{(s_j \in \mathcal{H}, \mathcal{H}, \mathsf{crowd})\}$. In this case, we know from the analysis of Crowds that the adversary should pick s_j as the most likely sender.

Type 2. The message is captured in the crowd, but the forwarding node s_j is not a member of the multicast group $\mathcal{H}$: $\mathcal{O}_2 = \{(s_j \notin \mathcal{H}, \mathcal{H}, \mathsf{crowd})\}$. In this case, we know that s_j is *not* the sender. Thus, all senders $s_i \in \mathcal{H}$ are equally likely, and the adversary's best strategy is to guess at random.

Type 3. The message is captured at the receiver: $\mathcal{O}_3 = \{(s_j, \mathcal{H}, \mathsf{recv})\}$. Similarly to Case 2, all nodes in $\mathcal{H}$ are equally likely senders.

Success Probability. With probability f, the sender forwards the message to a malicious peer during the first hop, thus generating an observation $\mathsf{obs} \in \mathcal{O}_1$ in the first set that leads to a correct guess. The remaining Type 1 observations, as well as all Type 2 and Type 3 observations lead to a guess that is correct with probability $1/|\mathcal{H}|$.

Noting that the multicast group contains the sender as well as $k-1$ nodes randomly chosen by the sender, and can thus include anywhere between 1 and k honest nodes, we can compute the success probability of the adversary as

$$\begin{aligned}\mathbb{E}(\mathsf{success}) &= f + (1-f) \cdot \sum_{k'=1}^{k} \Pr[|\mathcal{H}| = k'] \frac{1}{k'} = f + (1-f) \sum_{k'=1}^{k} \frac{\binom{n-c-1}{k'-1} \cdot \binom{c}{k-k'}}{\binom{n-1}{k-1} k'} \\ &= f + (1-f) \frac{n}{k(n-c)} \sum_{k'=1}^{k} \frac{\binom{n-c}{k'} \cdot \binom{c}{k-k'}}{\binom{n}{k}} = f + \frac{1}{k}\left(1 - \frac{\binom{c}{k}}{\binom{n}{k}}\right) \\ &< f + \frac{1}{k} .\end{aligned}$$

□

B Analysis of AJSS10

Assuming that the algorithm succeeds with probability ≈ 1 whenever peers capture a message on the first hop[5]; and makes a random guess from the subset of honest nodes $\mathcal{H}$ in the k-anonymity set otherwise, we can predict the success probability of our algorithm to be

$$\begin{aligned}\mathbb{E}(\mathsf{success}) &\approx 1 - (1-f)^{k-1} + (1-f)^{k-1} \sum_{k'=1}^{k} \Pr[|\mathcal{H}| = k'] \frac{1}{k'} \\ &= 1 - (1-f)^{k-1} + \frac{(1-f)^{k-2}}{k}\left(1 - \frac{\binom{c}{k}}{\binom{n}{k}}\right) ,\end{aligned}$$

where c is the number of corrupt peers, as before.

[5] The exact probability is analytically intractable, yet our simulations confirm that this assumption is reasonable.

A New Measure of Watermarking Security Applied on QIM

Teddy Furon[1] and Patrick Bas[2,⋆]

[1] INRIA Research Centre Rennes Bretagne Atlantique, France
teddy.furon@inria.fr
[2] CNRS-LAGIS, Ecole Centrale de Lille, France
patrick.bas@ec-lille.fr

Abstract. Whereas the embedding distortion, the payload and the robustness of digital watermarking schemes are well understood, the notion of security is still not completely well defined. The approach proposed in the last five years is too theoretical and solely considers the embedding process, which is half of the watermarking scheme. This paper proposes a new measurement of watermarking security, called the *effective key length*, which captures the difficulty for the adversary to get access to the watermarking channel. This new methodology is applied to the Distortion Compensated Dither Modulation Quantized Index Modulation (DC-DM QIM) watermarking scheme where the dither vector plays the role of the secret key. This paper presents theoretical and practical computations of the effective key length. It shows that this scheme is not secure as soon as the adversary gets observations in the Known Message Attack context.

Keywords: watermarking, security, Quantized Index Modulation.

1 Introduction

The Problem: This paper deals with the evaluation of the security level of a digital watermarking scheme. The problem is that the previous methodology on this topic [1], although applied on Spread Spectrum [2] and Dither Modulated Distortion Compensated Quantized Index Modulation (DM-DC QIM) [3] watermarking schemes, is not so successful. As detailed in Sect. 2, it does not fully capture the whole watermarking scheme as it only considers the embedding process. Its assessment is mostly theoretical and difficult to apply on real-life watermarking schemes. One has important difficulties in interpreting the quantity measuring the security level by relying only on information theory.

Example: Let us take the following scenario: consider a DC-DM QIM with a cubic lattice (a.k.a. SCS, Scalar Costa Scheme [4]) for embedding bits in a signal $\mathbf{x}$, at a given DWR (Document to Watermark power Ratio) and a given expected

[⋆] P. Bas' work was partly founded by the French National Research Agency program referenced ANR-10-CORD-019 under the Estampille project.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 207–223, 2013.

WNR (Watermark to Noise power Ratio). Denote Δ the quantization step and α the compensation parameter. Now, the security level when measured by the equivocation equals $\log((1-\alpha)\Delta)$ nats [3]. Suppose now that we watermark the scaled signal $2 * \mathbf{x}$ with the same technique and setup (DWR, WNR). Then, the quantization step is now 2Δ while α remains the same. The security level is now higher by 0.69 nats. It is counterintuitive that by doubling the amplitude of the host signal, we succeed to increase the security level. Moreover this amount is indeed hard to understand: Does 0.69 nats represent a big increase in term of security?

Our Contributions: This paper proposes a new way of defining the security level of a digital watermarking scheme in Sect. 3. Sect. 4 applies this methodology to QIM watermarking schemes from a theoretical point of view, while Sect. 5 presents an experimental framework to evaluate the security level. Our contributions are the following:

- A framework for security assessment in line with the cryptographic approach,
- A theoretical derivation of the security levels for watermarking schemes based on Quantized Index Modulation (QIM) with self-similar lattices,
- Theoretical bounds of the security levels when the lattices are not self-similar,
- An experimental setup for estimating the security levels for QIM.

After the talk at the Information Hiding conference, Prof. Jiwu Huang mentioned that his team showed similar results in a paper entitled "*Security Hole in QIM Watermarking*", at that time submitted to IEEE Trans. on Information Forensics and Security. Their work is independent, deals with attacks on QIM data hiding, but relies on a different approach than the concepts introduced in Sect. 3.

2 The Problem with Previous Security Measures

From the beginning, watermarking has been characterized by a trade-off between the embedding distortion and the capacity. The capacity is the theoretical amount of hidden data that can be reliably transmitted when facing an attack of a given strength. In practice, the operating point of a watermarking technique is defined by the embedding distortion (measured by a DWR for instance), a payload (measured in bits per host samples for instance) and the robustness (for instance, measured by a Symbol Error Rate SER after an attack - compression, rotation etc).

Security came as a fourth feature stemming from applications where there exist attackers willing to circumvent watermarking such as copy and/or copyright protection. The efforts of the pioneering works introducing this new concept first focused on stressing the distinction between security and robustness. An early definition was coined by Ton Kalker as *the inability by unauthorized users to have access to the raw watermarking channel* [5].

The problem we see lies in the fact that the methodology proposed so far poorly captures T. Kalker's definition. In a nutshell, the methodology of [1–3] is based on C. E. Shannon definition of security for crypto-systems. The security level is defined as the amount of uncertainty the attacker has about the secret key. This is measured by the equivocation which is the entropy of the key knowing some observations, which are for instance contents watermarked with the same technique and the same secret key. The equivocation, be it valued in nats or bits, can be negative (if the secret key is a continuous random variable), and as illustrated in the example of the introduction, the results of this approach are sometimes hard to understand.

The main pitfall is that watermarking and symmetric cryptography strongly disagree in the following point: In symmetric cryptography, the deciphering key is the secret key which is unique. Therefore, inferring this key from the observations (here, say some cipher texts) is the main task of the attacker. The disclosure of this key grants the adversary the access to the crypto-channel.

This is not the case in watermarking for the simple reason that there is no unique key to decode the hidden messages. In many watermarking schemes, the secret is a signal lying in the same space as the host vector: the carriers for Spread Spectrum, the dither for DC-DM QIM. They are generated by a Pseudo-Random Number Generator (PRNG) fed by a secret seed. Yet, the attacker may use another generator, or use some observations to estimate these signals. Therefore, the real secret granting access to the watermarking channel is less the seed of the PRNG than these signals. In the sequel, the secret signal is denoted by $\mathbf{k}$ and we show that a close enough signal $\mathbf{k}'$ may decode the hidden messages.

Consequently, inferring the secret key $\mathbf{k}$ from the observations (here, say some watermarked contents) is not the ultimate goal of the attacker. As T. Kalker stated, it is the access to the watermarking channel that matters. The estimation of the secret key is a possible path to this goal, but not the final destination. The limit of the past articles on watermarking security is that they focus on the estimation of the secret key, but very few works deal with the impact of the estimation accuracy on the access to the watermarking channel. It is quite symptomatic that almost none of them consider the decoding of the watermarking schemes. We strongly believe that this is the reason why the outcomes of this methodology are quite difficult to understand. C. E. Shannon was right, but those who translated his theory to watermarking only capture half the problem. The only exception we are aware of is [6] which intuitively sketched the idea that is formalized in this paper.

3 Our New Approach

3.1 The Idea

The keystone of our approach is the brute force attack. In cryptanalysis, the attacker randomly draws a test key and decrypts the ciphertexts. It is assumed that a genie tells the attacker when he succeeds, ie. when the test key equals the secret key. If the secret is a N-bit word, the probability of this event is $P = 1/2^N$, ie. one single secret key over 2^N possible keys. With some observations, the attacker might reduce the key space which increases the probability of success to $P = 2^{-L}$, with $L < N$. The security level is measured by $L = -\log_2(P)$ in bits.

We use the same approach for watermarking security. The *inability by unauthorized users to have access to the raw watermarking channel* is measured by $-\log_2(P)$, where P is the probability that the attacker finds a key granting the decoding of hidden messages embedded with the secret key. Contrary to symmetric cryptographic, there are a plurality of such a key ; and this is mainly due to the fact that the embedding has to be robust. We name them the *equivalent decoding keys*. Note that we could also consider *equivalent embedding keys*, ie. keys embedding messages in host content which are reliably decoded by the secret key. Our methodology aims at resolving the following questions:

- What is an equivalent decoding key?
- How many equivalent decoding keys do exist?
- What is the probability of picking an equivalent decoding key?
- How to improve the odds thanks to the observations?

3.2 The Setup

Before producing any watermarked content, the designer draws the secret key $\mathbf{k}$ in the key space $\mathcal{K}$ according to a given distribution $p_{\mathbf{K}}$. There is an extraction function that computes a vector $\mathbf{x} \in \mathcal{X}$ from a content. Usually, $\mathcal{X} = \mathbb{R}^{N_v}$. The embedding modifies this vector into $\mathbf{y}$ under a distortion constraint (here, given by a bound on the Euclidean distance $\|\mathbf{y} - \mathbf{x}\|^2 \leq N_v D$). There is an inverse extraction function which maps $\mathbf{y}$ back into the content. We assume that the extraction process is public, and that the secret key $\mathbf{k}$ is only used for shaping $\mathbf{x}$ into $\mathbf{y}$: The embedder creates a watermarked vector $\mathbf{y} \in \mathcal{X}$ with hidden message m using the embedding function $e(.)$: $\mathbf{y} = e(\mathbf{x}, m, \mathbf{k})$. At the decoding side, a vector is computed from the received content with the same extraction function. The message $\hat{m}$ is decoded from the watermarked vector by $\hat{m} = d(\mathbf{y}, \mathbf{k})$.

The adversary sees N_o independent observations $\mathbf{O}^{N_o} = (\mathbf{O}_1, \ldots, \mathbf{O}_{N_o})$. The nature of these observations defines the attack. In this paper, we restrict our attention to the Known Message Attack (KMA) where an observation is a pair of a watermarked content and the embedded message: $\mathbf{O}_i = \{\mathbf{y}_i, m_i\}$. The article [1] gives a list of other attacks.

We define by $\mathcal{D}_m(\mathbf{k}) \subset \mathcal{X}$ the decoding region associated to the message m and for the key $\mathbf{k}$ by:

$$\mathcal{D}_m(\mathbf{k}) \triangleq \{\mathbf{y} \in \mathcal{X} : d(\mathbf{y}, \mathbf{k}) = m\}. \quad (1)$$

The topology and location of this region in $\mathcal{X}$ depends of the watermarking scheme and of $\mathbf{k}$.

To hide message m, the encoder pushes the host vector $\mathbf{x}$ deep inside $\mathcal{D}_m(\mathbf{k})$, and this creates an embedding region $\mathcal{E}_m(\mathbf{k}) \subseteq \mathcal{X}$:

$$\mathcal{E}_m(\mathbf{k}) \triangleq \{\mathbf{y} \in \mathcal{X} : \exists \mathbf{x} \in \mathcal{X} \text{ s.t. } \mathbf{y} = e(\mathbf{x}, m, \mathbf{k})\}. \tag{2}$$

Watermarking provides robustness by pushing the watermarked vectors far away from the boundary of the decoding region. If the vector extracted from an attacked content $\mathbf{z} = \mathbf{y} + \mathbf{n}$ goes out of $\mathcal{E}_m(\mathbf{k})$, $\mathbf{z}$ might still be in $\mathcal{D}_m(\mathbf{k})$ and the correct message is decoded.

For QIM based watermarking schemes, we often have $\mathcal{E}_m(\mathbf{k}) \subseteq \mathcal{D}_m(\mathbf{k})$. Therefore, there might exist another key $\mathbf{k}'$ such that $\mathcal{E}_m(\mathbf{k}) \subseteq \mathcal{D}_m(\mathbf{k}')$, $\forall m$. A graphical illustration of this phenomenon is depicted on Fig. 1.

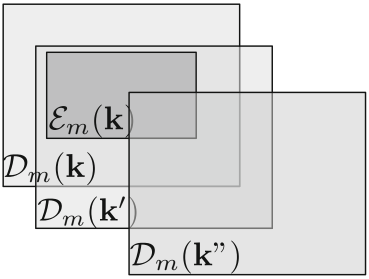

$\mathcal{E}_m(\mathbf{k})$
$\mathcal{D}_m(\mathbf{k})$
$\mathcal{D}_m(\mathbf{k}')$
$\mathcal{D}_m(\mathbf{k}")$

Fig. 1. Graphical representation in space $\mathcal{X}$ of three decoding regions $\mathcal{D}_m(\mathbf{k})$, $\mathcal{D}_m(\mathbf{k}')$ and $\mathcal{D}_m(\mathbf{k}'')$ and the embedding region $\mathcal{E}_m(\mathbf{k})$: $\mathbf{k}$ and $\mathbf{k}'$ belong to the equivalent decoding region $\mathcal{K}_{eq}^{(d)}(\mathbf{k}, 0)$, but $\mathbf{k}''$ does not

3.3 The Equivalent Keys

We now define the equivalent keys and the associated equivalent region. We should make the distinction between the equivalent decoding keys and the equivalent embedding keys. But we restrict our attention to the decoding problem in this paper, and we use the term equivalent keys.

The set of equivalent keys $\mathcal{K}_{eq}(\mathbf{k}, \epsilon) \subset \mathcal{K}$ with $0 \leq \epsilon$ is defined as the set of keys that allows a decoding of the hidden messages embedded with $\mathbf{k}$ with a probability bigger than $1 - \epsilon$:

$$\mathcal{K}_{eq}(\mathbf{k}, \epsilon) = \{\mathbf{k}' \in \mathcal{K} : \mathbb{P}\left[d(e(\mathbf{X}, M, \mathbf{k}), \mathbf{k}') \neq M\right] \leq \epsilon\}. \tag{3}$$

Due to a lack of space, this paper focuses on $\epsilon = 0$ giving birth to an equivalent definition:

$$\mathcal{K}_{eq}(\mathbf{k}, 0) = \{\mathbf{k}' \in \mathcal{K} : \mathcal{E}_m(\mathbf{k}) \subseteq \mathcal{D}_m(\mathbf{k}')\}. \tag{4}$$

This set is usually not empty for QIM: if $\mathcal{E}_m(\mathbf{k}) \subseteq \mathcal{D}_m(\mathbf{k})$, $\mathbf{k}$ is then an element of $\mathcal{K}_{eq}(\mathbf{k}, 0)$.

3.4 The Effective Key Length

We introduce the notion of *effective key length* as a way to measure security. The adversary picks a key $\mathbf{k}' \in \mathcal{K}$ taking into account the set of observations $\mathbf{O}^{N_o}$ with an estimator: $\mathbf{K}' = g(\mathbf{O}^{N_o})$. The estimator $g(\cdot)$ is either deterministic or stochastic such that $\mathbf{K}' \sim p(\mathbf{k}'|\mathbf{O}^{N_o})$ for instance. A graphical example of the key space $\mathcal{K}$ is depicted in Fig. 2.

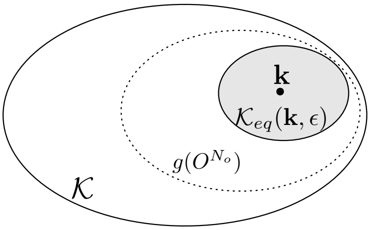


Fig. 2. Graphical representation of the key space $\mathcal{K}$ and the equivalent region $\mathcal{K}_{eq}(\mathbf{k},\epsilon)$. The dotted boundary represents the support of the estimator $g(O^{N_o})$ used to draw new test keys when the adversary has N_o observations.

The probability $P(\epsilon, N_o)$ that the adversary picks up a key belonging to the equivalent region is:

$$P^{(d)}(\epsilon, N_o) = \mathbb{E}_{\mathbf{K}}[\mathbb{E}_{\mathbf{O}^{N_o}}[\mathbb{E}_{\mathbf{K}'}[\mathbf{K}' \in \mathcal{K}_{eq}(\mathbf{K},\epsilon)|\mathbf{O}^{N_o}]]]. \tag{5}$$

Finally, to obtain an analogy with cryptography, the effective key length $\ell(\epsilon, N_o)$ translates this probability into bits as follows:

$$\ell(\epsilon, N_o) \triangleq -\log_2(P(\epsilon, N_o)) \quad \text{bits.} \tag{6}$$

The bigger the effective key length, the less likely is the attacker to find keys granting the access to the watermarking channel, and therefore, the more secure is the watermarking scheme. This measurement of the security is in line with Kalker's definition. It is easily interpretable. It doesn't rely on information theoretical element, and it takes into account the embedding and the decoding of the watermarking scheme.

4 Technical Details: Part I – Theoretical Analysis

This section applies the above methodology to DC-DM QIM watermarking. We give close form expressions for self-similar lattices and upper and lower bounds in the general case.

4.1 A Primer on DC-DM QIM Watermarking

Let us model the host signal by a vector $\mathbf{x} \in \mathbb{R}^{N_v}$. Consider a coarse Euclidean lattice $\Lambda_c \subset \mathbb{R}^{N_v}$. The origin $\mathbf{0} \in \mathbb{R}^{N_v}$ is an element of Λ_c and the Voronoi cell is defined as the set of vectors of $\mathbb{R}^{N_v}$ closer to $\mathbf{0}$ than to any other element of Λ_c: $\mathcal{V}(\Lambda_c) \triangleq \{\mathbf{v} \in \mathbb{R}^{N_v} | Q_{\Lambda_c}(\mathbf{v}) = \mathbf{0}\}$ where $Q_{\Lambda_c}(\cdot)$ is the Euclidean quantizer on Λ_c. The Voronoi cell of a lattice is a centrally symmetric, convex polytope.

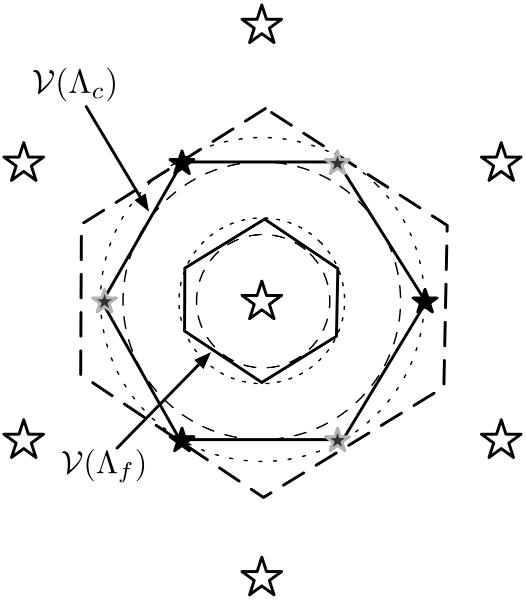


Fig. 3. 2D representation of the different elements used to compute the equivalent region. The large stars represent elements of the coarse lattice Λ_c, the small and large stars represents the fine lattice Λ_f, associated with the Voronoi cells $\mathcal{V}(\Lambda_c)$ and $\mathcal{V}(\Lambda_f)$. In this specific non-similar construction with the hexagonal lattice, $M = 3$. The dotted and dashed circles represent balls with radius of $R(\Lambda)$ and $r(\Lambda)$ respectively. The dashed hexagone is the scaled version of $\mathcal{V}(\Lambda_f)$ used to compute the lower bound in (20).

For each message $m \in \mathcal{M}$ with say $\mathcal{M} = \{1, 2, \dots, M\}$, a coset leader $\mathbf{d}_m \in \mathbb{R}^{N_v}$ is defined such that $\Lambda_f = \cup_{m=1}^{M}(\Lambda_c + \mathbf{d}_m)$ is a finer lattice. This induces the partition of Λ_f into M shifted versions of Λ_c, which implies that

$$|\mathcal{M}| = M = \mathrm{vol}(\mathcal{V}(\Lambda_c))/\mathrm{vol}(\mathcal{V}(\Lambda_f)), \tag{7}$$

with $\mathrm{vol}(\mathcal{A})$ the volume of subset $\mathcal{A} \subset \mathcal{X}$. Define $r(\Lambda)$ the packing radius of lattice Λ as the radius of the largest hyper-ball contained in $\mathcal{V}(\Lambda)$ and $R(\Lambda)$ the covering radius of Λ as the radius of the smallest hyper-ball containing $\mathcal{V}(\Lambda)$. Denote $\mathcal{B}(\mathbf{x}, r)$ the hyperball centered on $\mathbf{x}$ of radius r (see Fig. 3). Then,

$\mathcal{B}(\mathbf{0}, r(\Lambda)) \subset \mathcal{V}(\Lambda) \subset \mathcal{B}(\mathbf{0}, R(\Lambda))$. Finally, define $\rho(\Lambda)$ the effective radius of Λ such that $\mathrm{vol}(\mathcal{B}(\mathbf{0}, \rho(\Lambda))) = \mathrm{vol}(\mathcal{V}(\Lambda))$. Eq. (7) means that

$$M = (\rho(\Lambda_c)/\rho(\Lambda_f))^n. \tag{8}$$

Hiding message m in $\mathbf{x}$ with a DC-DM QIM technique yields watermarked vector $\mathbf{y}$:

$$\begin{aligned}\mathbf{y} = e(\mathbf{x}, m, \mathbf{k}) &= \mathbf{x} + \alpha(Q_{\Lambda_c}(\mathbf{x} - \mathbf{d}_m - \mathbf{k}) - \mathbf{x} + \mathbf{d}_m + \mathbf{k}) \\ &= Q_{\Lambda_c}(\mathbf{x} - \mathbf{d}_m - \mathbf{k}) + \mathbf{d}_m + \mathbf{k} + (1-\alpha)(\mathbf{x} - \mathbf{d}_m - \mathbf{k} - Q_{\Lambda_c}(\mathbf{x} - \mathbf{d}_m - \mathbf{k})) \end{aligned} \tag{9}$$

The key $\mathbf{k} \in \mathbb{R}^{N_v}$ is called the dither applying a secret shift of the quantizer. Due to the Λ_c-periodicity, the key ensemble $\mathcal{K}$ is the Voronoi cell $\mathcal{V}(\Lambda_c)$. We assume as in [3] that $\mathbf{k}$ has been uniformly drawn over $\mathcal{K} = \mathcal{V}(\Lambda_c)$. The last equation shows that the watermarked signal is an element of $\Lambda + \mathbf{d}_m + \mathbf{k}$ plus the self-inference noise $(1-\alpha)\tilde{\mathbf{x}}$, with $\tilde{\mathbf{x}} \triangleq [\mathbf{x} - \mathbf{d}_m - \mathbf{k} \mod \Lambda_c]$ and $[\mathbf{x} \mod \Lambda] \triangleq \mathbf{x} - Q_\Lambda(\mathbf{x})$. Parameter α with $0 < \alpha < 1$ is the distortion compensation factor. The two lattices are scaled by a factor Δ such that the Euclidean embedding distortion is below the distortion budget: $\alpha^2 \frac{\int_{\mathcal{V}(\Lambda_c)} \|\mathbf{x}\|^2 \partial \mathbf{x}}{\mathrm{vol}(\mathcal{V}(\Lambda_c))} \leq N_v D$ (under the flat host assumption, see [3]).

The message decoded from $\mathbf{y}$ with key $\mathbf{k}'$ is given by

$$\hat{m} = d(\mathbf{y}, \mathbf{k}') = \arg\min_{m \in \mathcal{M}} \|\mathbf{y} - \mathbf{d}_m - \mathbf{k}' - Q_{\Lambda_c}(\mathbf{y} - \mathbf{d}_m - \mathbf{k}')\|, \tag{10}$$

which is m for $\mathbf{y} = e(\mathbf{x}, m, \mathbf{k})$ if:

$$[(1-\alpha)\tilde{\mathbf{x}} + \mathbf{k} - \mathbf{k}' \mod \Lambda_c] \in \mathcal{V}(\Lambda_f). \tag{11}$$

We suppose that, in the noiseless case, the self-interference doesn't give birth to decoding errors when we decode with the secret key $\mathbf{k}' = \mathbf{k}$. It implies that $(1-\alpha)\mathcal{V}(\Lambda_c) \subset \mathcal{V}(\Lambda_f)$, or more simply $(1-\alpha)R(\Lambda_c) \leq r(\Lambda_f)$. If $\alpha \geq \alpha_{\min}$ with

$$\alpha_{\min} \triangleq 1 - r(\Lambda_f)/R(\Lambda_c), \tag{12}$$

the decoding is error free. But, there might be some values of $\alpha < \alpha_{\min}$ which yield error-free decoding. If $\alpha = \alpha_{\min}$, then $\mathbf{k}$ can decode without error: the set of equivalent keys at least comprises the singleton $\{\mathbf{k}\}$.

There are several constructions of the partition (Λ_c, Λ_f) provably good for data hiding. Their description is out of the scope of this paper (see [3]). However, we detail one in particular: We say that (Λ_c, Λ_f) are self-similar lattices if $\Lambda_f = \beta\Lambda_c$ with $0 < \beta < 1$ (ie. we exclude the case where Λ_f is a scaled rotation of Λ_c). Eq. (8) imposes that $M = \beta^{-N_v}$ which must be an integer bigger than 1. Decoding without error in the noiseless case implies $\beta \geq (1-\alpha)$ so that $\alpha \geq \alpha^{\mathrm{ss}}_{\min}$ (superscript ss means self-similar) with

$$\alpha^{\mathrm{ss}}_{\min} \triangleq 1 - \beta. \tag{13}$$

4.2 No Observation – $N_o = 0$

The attacker has no observation. He randomly picks a test key $\mathbf{k}'$ uniformly over $\mathcal{V}(\Lambda_c)$. What is the probability that $\mathbf{k}'$ is an equivalent key of $\mathbf{k}$?

Self-similar Lattices Construction. We are able to write a close form expression of this probability for this construction thanks to the following lemma. For two sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^{N_v}$, define $a\mathcal{A} = \{\mathbf{x}|\exists \mathbf{a} \in \mathcal{A} : \mathbf{x} = a\mathbf{a}\}$ and $\mathcal{A} \oplus \mathcal{B} = \{\mathbf{x}|\exists(\mathbf{a}, \mathbf{b}) \in \mathcal{A} \times \mathcal{B} : \mathbf{x} = \mathbf{a} + \mathbf{b}\}$.

Lemma 1. *For (a, b) two positive real numbers, $a\mathcal{V}(\Lambda) \oplus b\mathcal{V}(\Lambda) = (a + b)\mathcal{V}(\Lambda)$ for any Euclidean Lattice Λ.*

Proof. Take any $\mathbf{z} \in (a+b)\mathcal{V}(\Lambda)$, then $\mathbf{x} = a/(a+b)\mathbf{z}$ lies in $a\mathcal{V}(\Lambda)$, $\mathbf{y} = b/(a+b)\mathbf{z}$ lies in $b\mathcal{V}(\Lambda)$ while $\mathbf{z} = \mathbf{x} + \mathbf{y}$. Take now $\mathbf{x} \in a\mathcal{V}(\Lambda)$ and $\mathbf{y} \in b\mathcal{V}(\Lambda)$. Consider a codeword $\mathbf{c} \in \Lambda$ with $\mathbf{c} \neq \mathbf{0}$. Vector $\mathbf{x}$ is closer to codeword $\mathbf{0}$ than to any other codeword $a\mathbf{c}$ of $a\Lambda$. We have $\|\mathbf{x}\| \leq \|a\mathbf{c} - \mathbf{x}\|$ so that $a\|\mathbf{c}\|^2 - 2\mathbf{c}^\top\mathbf{x} \geq 0$. In the same way, $b\|\mathbf{c}\|^2 - 2\mathbf{c}^\top\mathbf{y} \geq 0$. Then $\|(a + b)\mathbf{c} - (\mathbf{x} + \mathbf{y})\|^2 = \|\mathbf{x} + \mathbf{y}\|^2 + (a + b)((a + b)\|\mathbf{c}\|^2 - 2\mathbf{c}^\top(\mathbf{x} + \mathbf{y})) \geq \|\mathbf{x} + \mathbf{y}\|^2$. This holds for any codeword $(a + b)\mathbf{c}$ of $(a + b)\Lambda$ so that $\mathbf{x} + \mathbf{y} \in \mathcal{V}((a + b)\Lambda) = (a + b)\mathcal{V}(\Lambda)$.

If $\mathbf{k}' \in [\mathbf{k} + (\beta - (1 - \alpha))\mathcal{V}(\Lambda_c) \mod \Lambda_c]$, then Eq. (11) is satisfied thanks to this lemma. Because there is no aliasing since $0 \leq \beta - (1 - \alpha) \leq 1$, the volume of $\mathcal{K}_{eq}(0, \mathbf{k})$ is the same for any $\mathbf{k}$. For the sake of simplicity, we can restrict our attention to the case $\mathbf{k} = \mathbf{0}$ which makes the modulo Λ_c useless. In the end, the probability of picking an equivalent key is the ratio:

$$P^{(d)}(0, 0) = \frac{\text{vol}(\mathcal{K}_{eq}(0, \mathbf{k}))}{\text{vol}(\mathcal{K})} = (\beta - (1 - \alpha))^{N_v} \tag{14}$$

$$= \frac{1}{M}\left(1 - \frac{1 - \alpha}{1 - \alpha^{\text{ss}}_{\text{min}}}\right)^{N_v}, \tag{15}$$

with $\alpha^{\text{ss}}_{\text{min}}$ given in (13). This expression does not depend on factor Δ.

Bounds for a General Construction. For $\alpha = 1$, (11) states that $\mathcal{K}_{eq}(0, \mathbf{k}) = \mathbf{k} + \mathcal{V}(\Lambda_f)$ and $P^{(d)}(0, 0) = 1/M$. For $\alpha < 1$, we cannot determine $\mathcal{K}_{eq}(0, \mathbf{k})$.

Upper Bound. We upper bound $\mathcal{K}_{eq}(0, \mathbf{k})$ with an hyperball. Since $\tilde{\mathbf{x}} \in \mathcal{V}(\Lambda_c)$, then $(1 - \alpha)\|\tilde{\mathbf{x}}\| \leq (1 - \alpha)R(\Lambda_c)$. If $\|\mathbf{k} - \mathbf{k}'\| \leq r(\Lambda_f) - (1 - \alpha)R(\Lambda_c)$, then $\|(1 - \alpha)\tilde{\mathbf{x}} + \mathbf{k} - \mathbf{k}'\| \leq r(\Lambda_f)$, which implies that (11) is satisfied. This means that $\mathcal{B}(\mathbf{k}, r(\Lambda_f) - (1 - \alpha)R(\Lambda_c)) \subset \mathcal{K}_{eq}(0, \mathbf{k})$. Therefore,

$$P^{(d)}(0, 0) \geq \frac{\text{vol}(\mathcal{B}(\mathbf{0}, r(\Lambda_f) - (1 - \alpha)R(\Lambda_c)))}{\text{vol}(\mathcal{V}(\Lambda_c))} \tag{16}$$

$$\geq \left(\frac{r(\Lambda_f) - (1 - \alpha)R(\Lambda_c)}{\rho(\Lambda_c)}\right)^{N_v} \tag{17}$$

$$\geq \frac{1}{M}\bar{r}(\Lambda_f)^{N_v}\left(1 - \frac{1 - \alpha}{1 - \alpha_{\text{min}}}\right)^{N_v}, \tag{18}$$

where $\bar{r}(\Lambda) \triangleq r(\Lambda)/\rho(\Lambda) \leq 1$ is the packing efficiency of the lattice Λ and $\alpha_{\min}$ is given in (12). Equality holds however if $\mathcal{V}(\Lambda_f)$ and $\mathcal{V}(\Lambda_c)$ are both spherical:

$$\bar{R}(\Lambda_f) = \bar{r}(\Lambda_f) = \bar{R}(\Lambda_c) = \bar{r}(\Lambda_c) = 1. \tag{19}$$

This is only the case for $N_v = 1$ where the Voronoi cell are intervals of $\mathbb{R}$, and we find back the expression for self similar lattices.

Lower Bound. We lower bound $\mathcal{K}_{eq}(0, \mathbf{k})$ with a scaled Voronoi cell of Λ_f (see Fig. 3). Suppose $\mathbf{k}' \in \mathcal{K}_{eq}(0, \mathbf{k})$, then $\mathbf{k}' = \mathbf{k} + \mathbf{x}_f + (1-\alpha)\mathbf{x}_c$ with $\mathbf{x}_f \in \mathcal{V}(\Lambda_f)$ and $\mathbf{x}_c$ belonging to:

$$\mathcal{V}(\Lambda_c) \subset \mathcal{B}(\mathbf{0}, R(\Lambda_c)) = \mathcal{B}\left(\mathbf{0}, \frac{R(\Lambda_c)}{r(\Lambda_f)} r(\Lambda_f)\right) \subset \frac{R(\Lambda_c)}{r(\Lambda_f)} \mathcal{V}(\Lambda_f).$$

Therefore, $\mathcal{K}_{eq}(0, \mathbf{k}) \subset \mathbf{k} + \left(1 + (1-\alpha)\frac{R(\Lambda_c)}{r(\Lambda_f)}\right) \mathcal{V}(\Lambda_f)$ and

$$P^{(d)}(0,0) \leq \frac{1}{M}\left(1 - \frac{1-\alpha}{1-\alpha_{\min}}\right)^{N_v}, \tag{20}$$

which is the same expression as for self-similar lattices, but with the $\alpha_{\min}$ of (12). Equality holds if the lattices are self-similar.

It may surprise the reader that no figure of merit about the coarse lattice Λ_c appears in these bounds. This is not true because $\alpha_{\min}$ indeed depends on its covering efficiency. These bounds depend on the distortion compensation factor α but not on the scale Δ of (Λ_c, Λ_f). These bounds may not be tight in general. For instance, for $\alpha = 1$, $P^{(d)}(0,0) = M^{-1}\ \forall(\Lambda_c, \Lambda_f)$, whereas the lower bound adds a scaling factor $\bar{r}(\Lambda_f)^{N_v}$. In the end, we obtain upper and lower bounds for the effective key length with a gap between the two of $N_v \log_2 \bar{r}(\Lambda_f)$ bits.

4.3 Some Observations – $N_o > 0$

In the KMA setup, the attacker observes N_o watermarked vectors together with their hidden message: $\mathbf{o}_i = \{\mathbf{y}_i, m_i\}$ with $1 \leq i \leq N_o$. We only detail the calculus for SCS: $N_v = 1$ and $\Lambda_c = \Delta\mathbb{Z}$, which can be used for self similar cubic lattices. We drop the boldface font since the host, the watermarked content and the key are now scalars. In other words, the embedding (9) simply gives:

$$y \in l\Delta + d_m + k + (1-\alpha)\mathcal{V}(\Delta\mathbb{Z}) \tag{21}$$

with $l \in \mathbb{Z}$, $d_m = (m-1)\Delta/M$ and $k \in \mathcal{V}(\Delta\mathbb{Z}) = \Delta/2.(-1,1]$. We also assume that $\alpha > 1/2$ and that the adversary knows d_m under KMA. The observations are:

$$o_i \triangleq y_i - d_{m_i} \in l_i\Delta + k + (1-\alpha)\Delta/2.(-1,1].$$

If we take these observations modulo Δ, the results may lie in a non convex set. However, there exist some r for which $[o_i - r \mod \Delta]$ are all in a convex interval of length $(1-\alpha)\Delta/2.(-1,1]$ (see [3, Prop. 2]). In other words, $\tilde{o}_i \triangleq [o_i - r \mod \Delta] + r = k + (1-\alpha)\tilde{x}_i$, and we get rid off the modulo operation. This implies in return that $k \in \tilde{o}_i + (1-\alpha)\Delta/2[-1,1)$. This holds for all the observations so that k must lie in the intersection of these intervals and we have:

$$k \in [\max \tilde{o}_i - (1-\alpha)\Delta/2, \min \tilde{o}_i + (1-\alpha)\Delta/2). \tag{22}$$

This interval is called the feasible set in [3] and we denote it by $\mathcal{K}(o^{N_o})$. In words, thanks to the observations, the attacker knows that the secret key lies into the feasible set. Therefore, he randomly picks a key k' in this set, and the probability that k' is an equivalent key is given by the ratio:

$$P^{(d)}(0, N_o) = \frac{\text{vol}(\mathcal{K}_{eq}^{(d)}(k,0) \cap \mathcal{K}(o^{N_o}))}{\text{vol}(\mathcal{K}(o^{N_o}))}. \tag{23}$$

Fig. 4 shows that $\mathcal{K}_{eq}^{(d)}(k,0)$ has a volume equalling $\Delta(1/M - (1-\alpha))$.

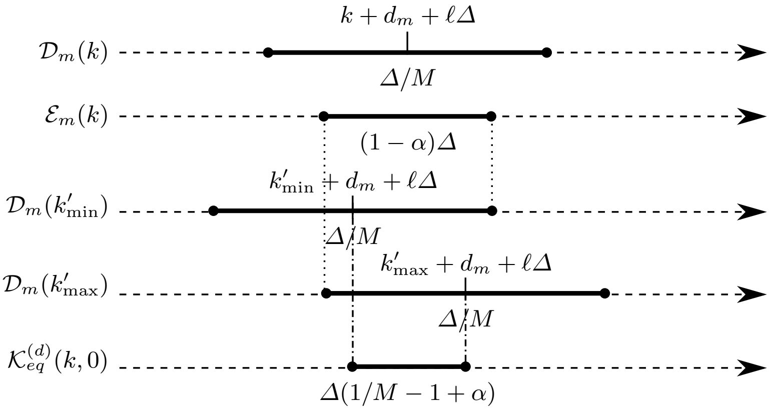

$k + d_m + \ell\Delta$
$\mathcal{D}_m(k)$
Δ/M
$\mathcal{E}_m(k)$
$(1-\alpha)\Delta$
$k'_{\min} + d_m + \ell\Delta$
$\mathcal{D}_m(k'_{\min})$
Δ/M
$k'_{\max} + d_m + \ell\Delta$
$\mathcal{D}_m(k'_{\max})$
Δ/M
$\mathcal{K}_{eq}^{(d)}(k,0)$
$\Delta(1/M - 1 + \alpha)$

Fig. 4. Computation of $\text{vol}(\mathcal{K}_{eq}^{(d)}(k,0))$ for DC-QIM

First Study: $N_o = 1$. Denote $l_{eq} = \text{vol}(\mathcal{K}_{eq}^{(d)}(k,0))/\Delta = 1/M - (1-\alpha)$ and $l_{fs} = \text{vol}(\mathcal{K}(O^1))/\Delta = (1-\alpha)$ (see (22) with $\max \tilde{o}_i = \min \tilde{o}_i$ for $N_o = 1$). There are three cases depending on the values of l_{eq} and l_{fs}.

1. For $1 - 1/M \leq \alpha \leq 1 - 1/2M$, we have $l_{eq} \leq l_{fs}$.
   The probability $P^{(d)}(0,1)$ is given by $\int \mathbb{P}[k' \in \mathcal{K}_{eq}(k,0)|\tilde{o}_1] f(\tilde{o}_1)\partial\tilde{o}_1$, with $f(\tilde{o}_1) = (\Delta l_{fs})^{-1}$ and $\mathbb{P}[k' \in \mathcal{K}_{eq}(k,0)|\tilde{o}_1]$ given in Fig. 5 (left). We find:

$$P^{(d)}(0,1) = \frac{l_{eq}}{l_{fs}}\left(1 - \frac{l_{eq}}{4l_{fs}}\right) = 1 - (1-d)^2, \tag{24}$$

   with $d \triangleq \frac{1}{2M(1-\alpha)} - \frac{1}{2} \leq 1$.

2. For $1-1/2M \leq \alpha \leq 1-1/3M$, we have $l_{fs} \leq l_{eq}$. Although $\mathbb{P}\left[k' \in \mathcal{K}_{eq}(k,0)|\tilde{o}_1\right]$ has a different expression as shown in Fig. 5 (right), after integration, we find the same expression as (24).
3. For $1-1/3M \leq \alpha \leq 1$, we have $l_{eq} \leq 2l_{fs}$ and $P^{(d)}(0,1)=1$.

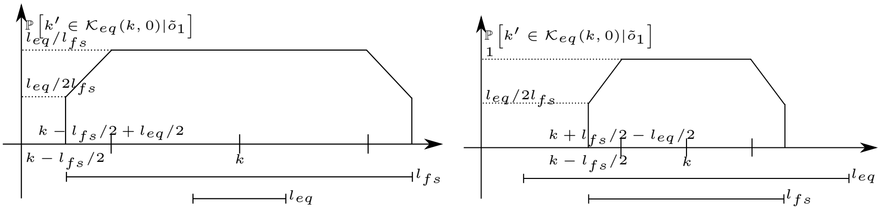

$\mathbb{P}\left[k' \in \mathcal{K}_{eq}(k,0)|\tilde{o}_1\right]$
l_{eq}/l_{fs}
$l_{eq}/2l_{fs}$
$k - l_{fs}/2 + l_{eq}/2$
$k - l_{fs}/2$
k
l_{fs}
l_{eq}
$\mathbb{P}\left[k' \in \mathcal{K}_{eq}(k,0)|\tilde{o}_1\right]$
1
$l_{eq}/2l_{fs}$
$k + l_{fs}/2 - l_{eq}/2$
$k - l_{fs}/2$
k
l_{eq}
l_{fs}

Fig. 5. SCS with $1-1/2M \leq \alpha \leq 1-1/3M$ (left) or $1-1/M \leq \alpha \leq 1-1/2M$ (right)

Second Study: $N_o > 1$: We introduce two random variables: $\underline{O} = \min \tilde{O}_i$ and $\bar{O} = \max \tilde{O}_i$ which are defined on the following interval: $-(1-\alpha)\Delta/2 \leq \underline{O} \leq (1-\alpha)\Delta/2$ and $\underline{O} \leq \bar{O} \leq (1-\alpha)\Delta/2$. The pdf of $(\underline{O}, \bar{O})$ is given by:

$$p_{\underline{O},\bar{O}}(\underline{o},\bar{o}) = \frac{N_o(N_o-1)}{((1-\alpha)\Delta)^{N_o}}(\bar{o}-\underline{o})^{N_o-2}. \tag{25}$$

For a given couple $(\underline{o},\bar{o})$, the probability of picking an equivalent key is as follows:

$$A(\underline{o},\bar{o}) = 1 - \frac{|\underline{o}+(1-\alpha-1/2M)\Delta|^+ + |(1-\alpha-1/2M)\Delta-\bar{o}|^+}{(1-\alpha)\Delta+\underline{o}-\bar{o}},$$

with $|x|^+ \triangleq \max(x,0)$. Note that if $\alpha \geq 1-1/3M$, then $A(\underline{o},\bar{o})=1$, $\forall(\underline{o},\bar{o})$ in the definition set, so that $P^{(d)}(0,N_o)=1$, which is consistent with the first study. Note also that if $\alpha = 1-1/M$, then $A(\underline{o},\bar{o})=0$ and the attacker never succeeds. Finally,

$$P^{(d)}(0,N_o) = \int_{-(1-\alpha)\Delta/2}^{(1-\alpha)\Delta/2} \int_{\underline{o}}^{(1-\alpha)\Delta/2} p_{\underline{O},\bar{O}}(\underline{o},\bar{o}) \cdot A(\underline{o},\bar{o}) \partial\underline{o}\partial\bar{o}. \tag{26}$$

After some cumbersome manipulations, we have for $1-1/M \leq \alpha \leq 1-1/3M$:

$$\begin{aligned} P^{(d)}(0,N_o) = {} & 1-(1-d)^{N_o} \\ & + dN_o(N_o-1)\left(d\ln(d)+1-d-\sum_{\ell=1}^{No-2}\frac{(1-d)^{\ell+1}}{\ell(\ell+1)}\right). \end{aligned} \tag{27}$$

This shows that when α increases from $1-1/M$ to $1-1/3M$, $P^{(d)}(0,N_o)$ goes from 0 to 1.

It is easy to extend these results to self similar cubic lattices: $\Lambda_c = \Delta\mathbb{Z}^{N_v}$. The probability to find an equivalent key over the block of size N_v is the product of the N_v probabilities per component. Therefore, one just has to take Eq. (24) and (27) to the power N_v, and the effective key length is N_v times the key length per component.

5 Technical Details: Part II – Experimental Setup

This section presents an experimental framework to numerically evaluate the effective key length. We assume that there exist efficient quantizers for the chosen lattices (Λ_c, Λ_f). This means that we know how to embed, decode and make modulo Λ_c operation. The subsections below explain how we overcome two difficulties.

5.1 Indicator Function of $\mathcal{K}_{eq}(\mathbf{0}, \mathbf{k})$

Consider the case $N_o = 0$. A naive experimental protocol based on a Monte Carlo simulations would be to generate one secret key $\mathbf{k}$, and then N test keys $\{\mathbf{k}'_i\}_{i=1}^N$ and to count the number of times $\mathbf{k}'_i$ is an equivalent decoding key of $\mathbf{k}$. The problem is that, if the partition is not based on self similar lattices, we do not know the shape of $\mathcal{K}_{eq}(0, \mathbf{k})$ and there is no indicator function of this set. The only thing we have is that Eq. (11) holds for any $\tilde{\mathbf{x}} \in \mathcal{V}(\Lambda_c)$ if $\mathbf{k}'_i \in \mathcal{K}_{eq}(0, \mathbf{k})$.

A first possibility is to generate N_t vectors $\{\tilde{\mathbf{x}}_i\}_{i=1}^{N_t}$ uniformly distributed over $\mathcal{V}(\Lambda_c)$. Thanks to the convexity of the Voronoi cells, we know that if Eq. (11) holds for the N_t elements, then it holds for any point in their convex hull of which is a subset of $\mathcal{V}(\Lambda_c)$. Therefore, this method is only an approximation of the indicator function, which becomes inaccurate if N_t is too small. This in turn raises a problem of complexity since we need to check (11) N_t times per test key.

A second possibility benefits from the convexity property. Since $\mathcal{V}(\Lambda_c)$ is convex, setting $\{\tilde{\mathbf{x}}_i\}_{i=1}^{N_t}$ as its vertices is sufficient. However, the dimension of the space strikes us again. For instance, there are 2^{N_v} such vertices for $\Lambda_c = \Delta\mathbb{Z}^{N_v}$ and $19,440$ for $\Lambda_c = E_8$. For the latter case, we only consider the $2,160$ deep holes of E_8, i.e. the most far away from $\mathbf{0}$ vertices [7].

5.2 Rare Event Probability Estimator

Since the probabilities to be estimated can be low, the complexity of Monte Carlo simulations is another difficulty. The number of test keys N must be in the order of $1/P^{(d)}(0, N_o)$ to achieve a reasonably low relative variance of estimation. This is the reason why we also use a rare event probability estimator[1]. Three ingredients are needed:

- A generator of test keys. The test keys are to be drawn uniformly over a convex set (e.g. $\mathcal{K} = \mathcal{V}(\Lambda_c)$ for $N_o = 0$). This is done by the rejection method: We randomly draw a vector $\mathbf{v}$ in the hypercube $R(\Lambda_c)[-1, 1]^{N_v}$ and we accept it as an occurence of $\mathbf{K}' \sim U(\mathcal{V}(\Lambda_c))$ if $Q_{\Lambda_c}(\mathbf{v}) = \mathbf{0}$ indicating that $\mathbf{v} \in \mathcal{V}(\Lambda_c)$. If not, we reject it and redraw a vector $\mathbf{v}$ until the condition is checked.
- A modification process. It randomly modifies a key $\mathbf{K}'$ into $\mathbf{K}''$ so that the latter is exactly distributed like the former. One says that the process is

[1] Available as a Matlab Toolbox at www.irisa.fr/texmex/people/furon/src.html

distribution invariant. Since the law is indeed the uniform distribution over a convex set, we use the "Hit and Run" algorithm [8]. In a nutshell, from a point $\mathbf{K}'$ in the set, one uniformly draws a direction Θ is the space. Then, one seeks the 2 points A and B of this line $(\mathbf{K}', \Theta)$ that intersect with the frontier of the set. At the end, one draws a point uniformly over $[A, B]$. The process is repeated several times and the output $\mathbf{K}''$ is the last point.

- A score function $s(\cdot) : \mathcal{K} \rightarrow \mathbb{R}$. It is designed such that $s(\mathbf{k}') = 1$ implies that $\mathbf{k}' \in \mathcal{K}_{eq}(0, \mathbf{k})$. However, it must be a soft function: $s(\mathbf{k}')$ graciously tends to 1 when $\mathbf{k}'$ gets closer to $\mathcal{K}_{eq}(0, \mathbf{k})$ in some sense. We propose the following trick: We compute the difference $d_i = \|(1-\alpha)\tilde{\mathbf{x}}_i + \mathbf{k} - \mathbf{k}' \mod \Lambda_c\| - r(\Lambda_f)$. Therefore, $d_i > 0$ for the vectors violating (11). We set $s(\mathbf{k}') = 1 - \max(\{|d_i|^+\})$. If (11) holds for the N_t vectors defined in Sub. 5.1, then $s(\mathbf{k}') = 1$.

With this setting, the algorithm described in [9] estimates $\mathbb{P}[s(\mathbf{K}') = 1]$ with $\mathbf{K}'$ uniformly distributed over a convex set $\mathcal{K}$. Its properties in term of bias, relative variance and confidence interval are given in [9]. Its complexity is in $O(\log(1/P^{(d)}(0, N_o)))$. In practice, if $P^{(d)}(0, N_o)$ is lower than 10^{-3}, this algorithm runs faster than the Monte Carlo simulations.

6 Discussions

6.1 Scalar Costa Scheme

We first analyze the security of the Scalar Costa Scheme where $N_v = 1$, $\Lambda_c = \Delta\mathbb{Z}$, $\Lambda_f = M^{-1}\Delta\mathbb{Z}$, and $\alpha^{\mathrm{ss}}_{\mathrm{min}} = 1 - M^{-1}$. This is the only case where we have a complete picture for any value of N_o. Fig. 6 shows the effective key length in bits per component .

The embedding distortion increases with Δ and with α, and so is the robustness. However, the effective key length decreases with α and does not depend on Δ. This stems in a trade-off between robustness and security. For a given Δ, α closer to 1 provides more robustness but less security.

There is a big discrepancy w.r.t. the value of N_o. When $N_o = 0$, the effective key length is always bigger $\log_2 M$ bits per component, which is the rate of the watermarking scheme. Hiding symbols at a higher rate does increase the security, but the robustness would be much smaller.

When $N_o > 0$, the effective key length vanishes to 0 bit as $\alpha \rightarrow 1 - 1/3M$. Fig. 6 (right) shows that the effective key length quickly vanishes as N_o increases. Note the big loss between $N_o = 0$ and $N_o = 1$.

6.2 Lattice Embedding

The only setup where we have a full analysis is the cubic self-similar lattices: the effective key length for a block of size N_v is the effective key length of SCS times N_v. Therefore, the effective key length per component remains the same.

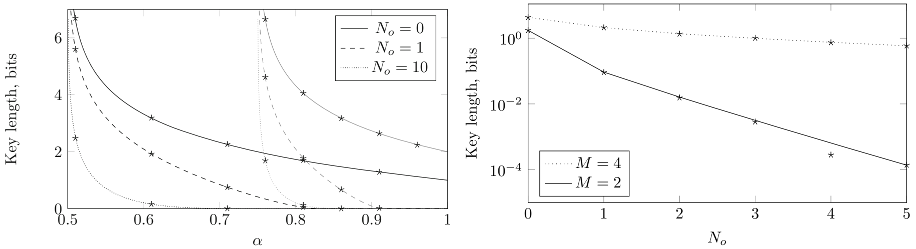


Fig. 6. Key length in bits for the SCS scheme, (left) vs. the distortion compensation factor α. (right) vs. the number of observations N_o for $\alpha = 0.8$. Stars mark experimental estimations as described in Sect. 5.1.

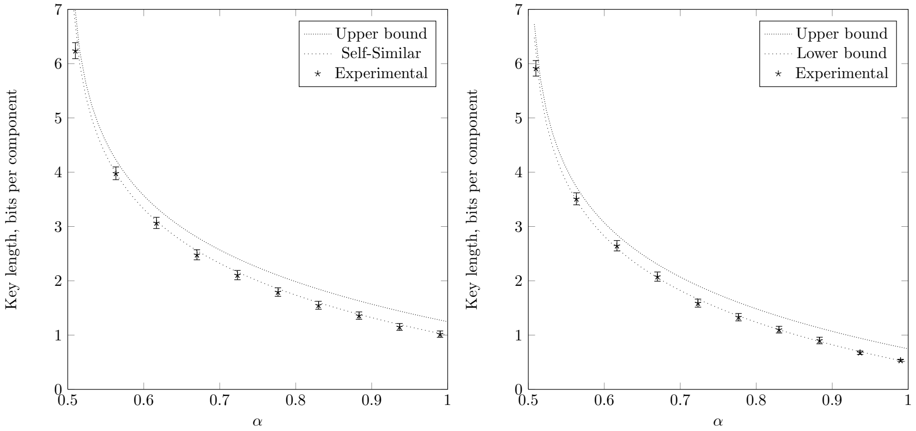


Fig. 7. Key length in bits for constructions 1 (left) and 2 (right) vs. the distortion compensation factor α. Stars mark experimental estimations as described in Sect. 5.2; the intervals are the 95% confidence intervals of these estimations.

For any other construction, we only have results for $N_o = 0$. As above, when $\alpha = 1$, the effective key length per component equals the rate of the watermarking scheme: $\log_2(M)/N_v$ bits. Surprisingly, two self-similar constructions operating with the same β and at the same rate, share the same effective key length per component. For instance, SCS with $M = 2$ and the construction 1 detailed below share the same plot for $N_o = 0$ (Fig. 6 (left) and Fig. 7 (left)). In the same way, two non-similar constructions operating with the same $\alpha_{\min}$ and at the same rate share the same lower bound on the effective key length per component. In general, $\alpha_{\min}$ has an impact on the decay rate of the effective key length, whereas the rate of message hiding shifts the plot.

We apply the experimental benchmark detailed in Sect. 5 to two constructions for $N_v = 8$ (RE_8 denotes a rotated version of lattice E_8 [7]):

1. Self similar: $\Lambda_c = E_8$, $\Lambda_f = \beta E_8$, $\beta = 0.5$, $M = 256$, $\alpha^{\mathrm{ss}}_{\min} = 0.5$.
2. Non similar: $\Lambda_c = RE_8$, $\Lambda_f = E_8$, $\bar{r}(\Lambda_f) = 0.842$, $M = 16$, $\alpha_{\min} = 0.5$.

Fig. 7 validates the experimental evaluation of the effective key length: for the self-similar lattices, the estimation is in line with the close form expression since it lies in the confidence interval except for the smallest value of α (see Fig. 7 (left)). This is due to the approximation of the equivalent region (see Sect. 5.1). For non similar lattices, the bounds are so close that the experimental evaluation does not bring much information. It seems that the key length is closer to the upper bound for weak α, and closer to the lower bound for strong α. The rare event estimator (see Sect. 5.2) is useful because the probabilities to be evaluated are as low as 10^{-16} for the smallest value of α. This algorithm succeeds to estimate such order of probability within two minutes on a regular computer.

7 Conclusion and Future Works

This paper introduces a new approach to gauge the security of watermarking schemes. The keystone is the notion of equivalent keys: there exist a plurality of keys granting access to the watermarking channel. The scheme is more secure if the attacker has greater difficulty in finding an equivalent key.

This approach is then applied to DC-DM QIM watermarking schemes. The lesson is that, as soon as the attacker observes some watermarked contents and their hidden message, the scheme is then broken if it is designed to be robust.

The paper lacks a part of the study: for lattice embedding, the computation of the effective key length is missing when the attacker has some observations. This will be done in a future work. The experimental evaluation should not be difficult: we will use Set Member Estimation technique to approximate the feasible set yielded by the observations by a bounding ellipsoid as done in [3]. Then, the attacker has to randomly pick a key inside this region. The theoretical part however seems much more difficult. Another point is that we work with $\epsilon = 0$ (perfect access to the watermarking channel), it is interesting to see how the effective key length evolves when we relax this strong constraint.

References

1. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Trans. Signal Processing 53(10), 3976–3987 (2005)
2. Pérez-Freire, L., Pérez-González, F.: Spread-spectrum watermarking security. IEEE Transactions on Information Forensics and Security 4(1), 2–24 (2009)
3. Pérez-Freire, L., Pérez-González, F., Furon, T., Comesaña, P.: Security of lattice-based data hiding against the Known Message Attack. IEEE Trans. on Information Forensics and Security 1(4), 421–439 (2006)

4. Eggers, J., Baüml, R., Tzschoppe, R., Girod, B.: Scalar Costa Scheme for information embedding. IEEE Trans. on Signal Processing 51(4), 1003–1019 (2003); Special Issue on Signal Processing for Data Hiding in Digital Media and Secure Content Delivery
5. Kalker, T.: Considerations on watermarking security. In: Dugelay, J.L., Rose, K. (eds.) Proc. of the Fourth Workshop on Multimedia Signal Processing (MMSP), Cannes, France, pp. 201–206. IEEE (2001)
6. Cox, I.J., Doërr, G., Furon, T.: Watermarking Is Not Cryptography. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 1–15. Springer, Heidelberg (2006)
7. Conway, J.H., Sloane, N.J.A.: Sphere packings, lattices, and groups, vol. 290. Springer, New York (1988)
8. Lovász, L., Vempala, S.: Hit-and-run is fast and fun. Technical Report MSR-TR-2003-05, Microsoft Research (2003)
9. Guyader, A., Hengartner, N., Matzner-Løber, E.: Simulation and estimation of extreme quantiles and extreme probabilities. Applied Mathematics & Optimization, 1–26 (2011)

Non-Integer Expansion Embedding for Prediction-Based Reversible Watermarking

Shijun Xiang

School of Information Science and Technology, Jinan University, Guangzhou, China
xiangshijun@gmail.com

Abstract. This paper aims at reducing the embedding distortion by improving predictor's performance for prediction-error expansion (PE) based reversible watermarking. In the existing PE embedding methods, the predicted values or their variety should be rounded to integer values. This will restrict predictor's performance since the prediction context is only with past pixels (image) or samples (audio). In this paper, we propose a non-integer PE (NIPE) embedding approach, which can proceed non-integer prediction errors for data embedding by only expanding integer element of a prediction error while keeping its fractional element unchanged. More importantly, the NIPE scheme allows the predictor to estimate the current pixel/sample not restricted only past pixels/samples. We also propose a novel noncausal prediction strategy by combining past and future pixels/samples as the context. Experimental results for some standard test clips show that the non-integer output of predictor provides higher prediction performance, and the proposed NIPE scheme with the new predicting strategy can reduce the embedding distortion for the same payload.

Keywords: Reversible Watermarking, Non-Integer Prediction Error, Expansion Embedding, Noncausal Prediction.

1 Introduction

Reversible watermarking (also known as lossless/distortion-free/invertible data hiding) is a technique to embed data in a host signal (for example, an image or audio clip) and allow for the original digital media to be exactly recovered. There are two important requirements for reversible watermarking techniques: 1) a larger embedding payload and 2) a lower embedding distortion. The two requirements conflict with each other since a higher embedding payload usually results in a higher degree of distortion. In recent 10 years, reversible watermarking has been an active research topic.

In the literature, reversible watermarking algorithms can be categorized as four main types:

1) Type-I algorithms use modulo-arithmetic-based in additive spread frequency techniques, which often cause salt-and-pepper artifacts due to many pixels wrapped [1]. In this direction, a different approach proposed by Vleeschouwer *et al.* [2] reduced the artifacts by using the circular interpolation of the bijective transform of image histogram.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 224–239, 2013.

2) Type-II algorithms compress a set of selected features from an image in a lossless way and embed the information into the space saved due to the compression [3–5]. In [6] Celik *et al.* proposed a generalized LSB (g-LSB) embedding algorithm, which is an extension of the work [3]. Compared with Type-I algorithms, Type-II ones have higher payload.
3) The third category of reversible watermarking algorithms can be classified as difference expansion (DE) embedding methods, in which a common feature is to use difference operators to create features with a small magnitude, and further expand these features in order to create vacancies for bits embedding. The DE embedding technique was originally developed by Tian [7] and have been extended in [8–10]. The DE technique involves grouping the pixels of the host image and transforming them into a low-pass image (including integer averages) and a high-pass image (the pixel differences). The technique can embed larger amounts of data than the earlier approaches. For example, it's capacity is close to 0.5 bpp for Tian's method in a single pass).
4) A new research direction, proposed by Thodi *et al.* [11], is prediction-error expansion (PE) embedding technique. Comparing with the DE-based methods, one of the advantages of the PE technique is that it significantly adds the number of the feature elements that expanded for data embedding. The other advantage is that a predictor generates feature elements that are often smaller in magnitude than the feature elements generated by a difference operator. Instead of embedding the entire expanded difference into the present pixel, Coltuc split the difference between the current pixel and its prediction context, and successfully reduced the embedding distortion for PE-based reversible watermarking [12]. With embedding into each pixel, the PE embedding techniques provided the maximal capacity up to 1 bpp in a single pass.

Reversible watermarking algorithms have also been proposed for digital audio [13–15]. In [13], Veen *et al.* proposed a novel reversible audio watermarking approach by first compressing the dynamic range of the original signal to render a number of unused bits. These unused bits are used to embed data including payload and information relevant to the bit-exact reconstruction of the original audio file. This method can achieve a satisfactory embedding capacity but suffer from a undesirable distortion due to quantization error and loudness change in the compression-expansion embedding phase. By introducing DE embedding technique [8] for audio, Bradley *et al* addressed two DE-based reversible watermarking methods: dyad-based (two samples as a group) and triad-based (three samples as a group) [14]. The dyad-based method can achieve at best 0.5 bits per sample (bps) while the triad-based one providing the maximal capacity of 2 bits in a group of three neighboring samples. The PE embedding technique [11] has also been introduced for digital audio in [15] in a way that the current sample is a linear combination of three past samples (in which each sample is corresponding to an integer weight coefficient).

This paper aims at reducing the embedding distortion by improving predictor's performance for PE-based reversible watermarking. It is worth noting that the predicted value of a pixel/sample (see [11, 15]) or a variety of the predicted value (see Equation (4) in [12]) must be rounded to integer value to add or subtract to the context for PE-based data embedding. These integer operations limit predictor's performance since only past pixels/samples can be defined as the prediction context. In this paper we propose a non-integer PE (NIPE) embedding strategy, which can proceed non-integer prediction values for data embedding by expanding the integer element of a prediction error and keeping the fractional element unchanged. A novel prediction model, using both past and future pixels/samples as the prediction context, is designed for the NIPE embedding technique. Experimental results for some standard test clips show that the predictor with non-integer output has higher prediction performance, and the proposed NIPE method with the predictor can significantly reduce the embedding distortion for the same payload.

The outline of this paper is as follows. In the next section, the proposed PE embedding technique is introduced. This is followed by a description of a new prediction strategy. We then address the proposed reversible watermarking scheme and test the scheme's performance by comparing with existing reversible audio watermarking works. Finally, we draw the conclusions.

2 Prediction-Error Expansion Embedding

Prediction-error expansion (PE) embedding is a technique to expand a prediction error to create a vacant position and insert a bit into the vacant position, generally at the least significant bit (LSB). The PE-based scheme was originally developed by Thodi [11], and later improved by Coltuc [12] by marking the present pixel and its context for reducing the embedding distortion.

In this section, we presents a non-integer prediction-error expansion (NIPE) embedding technique, which really brings a predictor into full play in comparison with the integer prediction-error expansion (IPE) methods in [11, 12]. In the NIPE-based method, the predicted value is not needed to be rounded to integer number. This is beneficial to apply not only past pixels/samples but also future pixels/pixels as the prediction context to reduce the embedding distortion for PE-based reversible watermarking. Before introducing the proposed scheme, the basic principle of the IPE method [11] and an improvement of its [12] is briefly reminded.

2.1 IPE Embedding

In the IPE embedding technique [11], the prediction error is the difference between a pixel intensity y and its predicted intensity $\hat{y}$, denoted by $e = y - \hat{y}$. After embedding a bit w, the watermarked prediction error is

$$e_w = 2 \times e + w. \tag{1}$$

The marked pixel intensity is $y_w = \hat{y} + e_w$. Since y_w should be integer number, $\hat{y}$ must be rounded to integer values. As a result, the prediction error e $(e = y - \hat{y})$ is also an integer value. This is why we denote Thodi's method as the IPE in this paper. It is worth noting that the condition that $\hat{y}$ take integer values makes an undesirable requirement, that is, the prediction context only contains *past pixels* because the fractional element of $\hat{y}$ is lost in the embedding.

The hidden bit, w, is extracted from the LSB of e_w and the original pixel intensity y is recovered by

$$w = mod(e_w, 2),\ y = \hat{y} + \lfloor \frac{e_w}{2} \rfloor, \tag{2}$$

where $mod(e_w, 2)$ is the remainder on division of e_w by 2.

2.2 Improved PE Embedding

In [12], Coltuc improved Thodi's PE embedding method by modifying the current pixel and its context for reducing the embedding distortion. The basic principle is described as follows.

In the PE embedding method, a pixel y is replaced by y_w, where

$$y_w = \hat{y} + 2 \times (y - \hat{y}) + w = y + (y - \hat{y}) + w. \tag{3}$$

Equation (3) indicates that the prediction error $(y - \hat{y})$ and the bit w to be embedded are added to the gray level of the current pixel y. At detection, if the prediction context is not altered, the embedded data is recovered from the LSB of e_w:

$$w = mod(e_w, 2) = y_w - \hat{y} - 2\lfloor \frac{y_w - \hat{y}}{2} \rfloor, \tag{4}$$

From (4), Coltuc observed that in order to extract w and restore the original sample y, not the exact of $\hat{y}$ is needed, but of the difference of $y_w - \hat{y}$. Thus, Y_w and $\hat{y}$ can be simultaneously modified for data embedding in a way that $\hat{y}$ is modified by adding or subtracting an integer value ξ $(\xi = \lfloor \alpha p_w + \frac{1}{2} \rfloor)$ to its context, where $0 \leq \alpha < 1$ and $p_w = y - \hat{y} + w$. As a result, the estimate of y is computed as $\hat{y_\xi}$, and the new value of y becomes $y_{w\xi} = y_w - \hat{y} - \hat{y_\xi}$. In the detector the bit w can be recovered from Equation (4) and the context can be recovered by computing ξ. Finally, the original pixel y is recovered.

We can observe from Equation (3) that in the improved PE embedding method [12], though the prediction value $\hat{y}$ can take non-integer value, the variety of $\hat{y}$ or $e(e = y - \hat{y})$, $\xi = \lfloor \alpha(y - \hat{y} + w) + \frac{1}{2} \rfloor$, is needed to be rounded to integer value. The integer value ξ is further added or subtracted to the context. This typical embedding process proposed in [12] does not allow the predictor to use *future pixels* of the current pixel for prediction. The basic reason is that a future pixel (y_{i+1}) of the current pixel (y_i) can be considered as a past pixel of another pixel (y_{i+2}). When past and future pixels of the current pixel can be defined as the context in the work [12], a pixel, denoted as past pixel of some pixels and future pixel of other pixels, may be repeatedly modified due to the operation that adds or subtracts ξ to the context.

2.3 NIPE Embedding

Sections 2.1 and 2.2 show that the existing two PE embedding approaches [11, 12] are suffering from an undesired requirement that the context of a pixel (or a sample) cannot involve future pixels in the predictor. This requirement can be considered as a weakness since the predictor's performance is restricted. In this section, we present a new NIPE embedding approach, one of the advantages of which is able to deal with non-integer prediction values. The other, more important, advantage of the approach is that the predictor allows to use past and future pixels/samples as the context to improve prediction performance.

From the expression $y_w = \hat{y} + 2 \times (y - \hat{y}) + w$ in Equation (3), we find that in order to recover the marked pixel y_w, not exact of an integer $\hat{y}$ is needed, but of the sum of $\hat{y} + 2 \times e$. Towards this direction, the basic idea of our approach is to allow $\hat{y}$ to take non-integer value but make sure that the combination of $\hat{y}$ and e takes integer value for hiding the bit w.

In the encoder, the prediction error e is a non-integer value when the predicted value $\hat{y}$ takes non-integer number. Split e into two parts: integer part ℓ ($\ell = fix(e)$) and fractional part δ ($\delta = e - \ell$). $fix(.)$ is a function to strip off the fractional part of its argument, and returns the integer part. The function does not perform any form of rounding or scaling, e.g., $fix(-3.4) = -3$ and $fix(3.4) = 3$. *For a given bit w, the NIPE method expands the integer element of a prediction error for data embedding while keeping the fractional element unchanged.* The watermarked prediction error is computed by

$$e_w = \begin{cases} 2 \times \ell + \delta + w = e + \ell + w, & \text{if } e \geq 0, \\ 2 \times \ell + \delta - w = e + \ell - w, & \text{Otherwise .} \end{cases} \tag{5}$$

Such a expansion way can guarantee that the fractional part of e_w is equal to that of e.

The resulting watermarked pixel/sample is

$$y_w = \hat{y} + e_w = \begin{cases} \hat{y} + e + \ell + w = y + \ell + w, & \text{if } e \geq 0, \\ \hat{y} + e + \ell - w = y + \ell - w, & \text{Otherwise .} \end{cases} \tag{6}$$

Equation (6) shows that though $\hat{y}$ and e take non-integer values, the watermarked pixel is an integer number.

In the decoder, the bit w is extracted from e_w and the original pixel/sample y is restored by

$$w = mod(\ell_w, 2), \text{and } y = \hat{y} + fix(\frac{\ell_w}{2}) + \delta_w, \tag{7}$$

where ℓ_w is the integer element of e_w and $\delta_w = e_w - \ell_w$.

Equations (5), (6) and (7) form the proposed NIPE embedding strategy, which can allow the predictor to output non-integer values. More importantly, this approach allows a predictor to apply both past and future pixels/samples for prediction because the fractional element of e keeps unchanged in the embedding. The detail is described in Section 3.

2.4 Example for NIPE Embedding

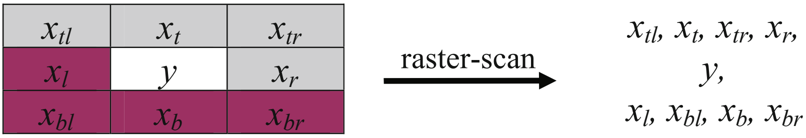
Let $y = 100$, $w = 1$ and $e = 100 - 100.4 = -0.4$ when $\hat{y} = 100.4$. The IPE and NIPE scheme are computed as follows:

1) *IPE scheme*: In the encoder, $e_w = 2 \times (-0.4) + 1 = 0.2$, $y_w = \hat{y} + e_w = 100.4 + 0.2 = 100.6$. In the decoder, $w = mod(e_w, 2) = 0.2$, and $y = \hat{y} + \lfloor \frac{e_w}{2} \rfloor = 100.4 + \lfloor \frac{0.2}{2} \rfloor = 100.4$.
2) *NIPE scheme*: In the encoder, $e_w = e + \ell - w = -0.4 + 0 - 1 = -1.4$. In the decoder, $\ell_w = fix(e_w) = -1$, $\delta_w = e_w - \ell_w = -0.4$, $w = mod(\ell_w, 2) = 1$, $y = \hat{y} + fix(\frac{\ell_w}{2}) + \delta_w = 100.4 + fix(\frac{-1}{2}) - 0.4 = 100$.

We can see from the above example that the NIPE scheme can deal with non-integer prediction value but the IPE can not.

3 Signal Prediction

Signal prediction is an important step in reversible watermarking. Usually, the prediction error is computed from the neighborhood of a pixel/sample. As shown in Fig. 1, the eight neighboring pixels of the present pixel (y) at the top-left (x_{tl}), top (x_t), top-right (x_{tr}), right (x_r), left (x_l), bottom-left (x_{tl}), bottom (x_b) and bottom-right (x_{br}) are defined as the context of y. After proceeding in a raster-scan order, we can observe that the context include four *past* pixels (x_{tl}, x_t, x_{tr}, x_r) and four *future* pixels (x_l, x_{bl}, x_b, x_{br}) as illustrated in Fig. 1 (b).



(a) Two-dimensional (2-D) image

(b) 1-D form after the scanning

Fig. 1. Context of a pixel

3.1 Prediction with Past Pixels/Samples

In the previous PE-based works [11, 12], the median edge detection (MED) predictor (already used in JPEG-LS Standard [16]) was used to report the performance of their algorithms. Also in [12], the gradient adaptive predictor (GAP) and the simplified GAP are applied for data embedding. No matter MED predictor or GAP, after proceeding in a raster-scan order, it is worth noting that only past pixels are combined as the context of the current pixel for reversible watermarking in [11, 12]. For example, the MED predictor applies three past pixels (x_t, x_{tr}, x_r) as the context of the current pixel (y) as illustrated in Fig. 1. The output of the MED predictor is

$$\hat{y} = \begin{cases} max(x_t, x_r), \text{ if } x_{tr} \leq min(x_t, x_r) \\ min(x_t, x_r), \text{ if } x_{tr} \geq max(x_t, x_r) \\ x_t + x_r - x_{tr}, \text{ Otherwise.} \end{cases} \tag{8}$$

The IPE-based reversible watermarking has also been introduced for audio [15], where three past samples are selected as the prediction context to output an integer value. This audio prediction strategy was originally designed for lossless compression coding of audio signals [17].

From the existing PE embedding algorithms (such as [11, 12, 15], and others), we can observe that the exploited predictors (such as MED, GAP and difference predictor) share a common property, that is, only past pixels/samples are defined as the context due to the fact that the prediction value or its variety is rounded to integer value in the embedding, as discussed in Sections 2.1 and 2.2.

3.2 Proposed Prediction Strategy

Obviously, only applying past pixels/samples as the context will reduce the performance of a predictor. In this section, we propose a noncausal prediction model for the NIPE embedding scheme. The new prediction model can predict the current pixel/sample not restricted to only past pixels/samples.

Noncausal Prediction Model. Assume we have a time-discrete signal Y of length N, $Y = \{y_1, y_2, \cdots, y_N\}$ with $y_i \in \{0, 1, \cdots, 2^m - 1\}^N$, and where m indicates the number of bits used to represent a sample/pixel[1]. The signal after the prediction is $\hat{Y}$. The residual signal is $E = Y - \hat{Y}$. Here, the predicted waveform is a linear combination of past and future pixels/samples:

$$\hat{y_i} = \sum_{t=1}^{p} a_{i-t} y_{i-t} + \sum_{t=1}^{p} a_{i+t} y_{i+t}, p < i < N - p + 1 \tag{9}$$

where $\sum_{t=1}^{p} a_{i-t} y_{i-t}$ is the linear combination of p past pixels/samples, $\sum_{t=1}^{p} a_{i+t} y_{i+t}$ that of p future pixels/samples. The prediction error is computed as

$$e_i = y_i - \hat{y_i} = y_i - \sum_{t=1}^{p} a_{i-t} y_{i-t} - \sum_{t=1}^{p} a_{i+t} y_{i+t}, p < i < N - p + 1. \tag{10}$$

The above equation can be rewritten as

$$y_{i+p} = \frac{y_i - e_i - \sum_{t=1}^{p} a_{i-t} y_{i-t} - \sum_{t=1}^{p-1} a_{i+t} y_{i+t}}{a_{i+p}}, p < i < N - p + 1. \tag{11}$$

[1] For a 2-D image, it can be proceeded as 1-D form by using a scanning operation (e.g., in a raster-scan or zigzag-scan order). For a signed audio, it can be mapped into the unsigned form by adding 2^{m-1}.

Equation (11) shows that in order to recover the original audio from the prediction errors, the information of the first $2p$ pixels/samples is also needed. The further information is referring to the following example predictor with $p = 1$. The prediction model shows that if there is no integer operation on the predicted value, those future pixels/samples can be defined as the prediction context of the current pixel/sample.

An Example Predictor. The section above has addressed the basic principle of the proposed prediction strategy. Here, an example predictor, used in this paper for audio files, is a simplified version. Beginning from the second sample, the sample y_i is predicted by averaging its two closest neighbors (y_{i-1}, y_{i+1}):

$$\hat{y_i} = \frac{y_{i-1} + y_{i+1}}{2}, 1 < i < N. \tag{12}$$

The difference is computed as

$$e_i = y_i - \hat{y_i} = y_i - \frac{y_{i-1} + y_{i+1}}{2}, 1 < i < N. \tag{13}$$

The original pixel/sample y_{i+1} is recovered by

$$y_{i+1} = 2y_i - y_{i-1} - 2e_i, 1 < i < N, \tag{14}$$

We can observe from Equation (14) that when the information of y_1 and y_2 is saved, the original signal can be recovered from the prediction errors[2]. Let $e_0 = y_1, e_1 = y_2 - y_1$. Overall, the output of the predictor is denoted as $E = \{e_0, e_1, \cdots, e_{N-1}\}$.

4 Proposed Watermarking Scheme

The proposed watermarking scheme is a combination of existing techniques (histogram shifting in [11]) and new techniques (NIPE embedding and data prediction not restricted only casual pixel/sample).

4.1 Prediction Expansion with Histogram Shift

The histogram shift method, introduced in [11], is an efficient reversible watermarking technique to enhance fidelity of the marked signal and avoid overlapping problems caused by expansion embedding. The combination of histogram shifting and IPE has been previously addressed in [11]. Here, we present how to combine the NIPE method with histogram shifting technique. We adopt a positive threshold value T to control the embedding distortion.

[2] When the first two pixels y_1 and y_2 are saved, the third pixel y_3 can be recovered by referring to the prediction error e_2 in Equation (14), then recovering y_4, y_5 and the other samples in sequential order. This explains the reconstruction process in Equation (11).

Specifically, only those prediction values in $[-T, T]$ are selected for NIPE embedding (denoted as the expanding set S_1), the prediction errors not in the range $[-T, T]$ are going to be shifted (denoted as the shiftable set S_2) to avoid overlapping problems. The reversible watermarking rules are formulated as follows.

$$e_{wi} = \begin{cases} 2 \times \ell_i + \delta_i + w_i & \text{if } e_i \in [0, T] \\ 2 \times \ell_i + \delta_i - w_i & \text{if } e_i \in [-T, 0) \\ e_i + T + 1 & \text{if } e_i > T \\ e_i - T, & \text{if } e_i < -T, \end{cases} \tag{15}$$

where ℓ_i is integer part of the i^{th} prediction error, e_i, satisfying $e_i = \ell_i + \delta_i$. The marked prediction error is denoted by e_{w_i} after the bit w_i is inserted.

The decoder recovers the original prediction error e_i and the bit w_i from e_{w_i} by:

$$e_i = \begin{cases} fix(\frac{\ell_{wi}}{2}) + \delta_{wi} & \text{if } e_{wi} \in [-2T, 2T + 1] \\ e_{wi} - T - 1 & \text{if } e_{wi} > 2T + 1 \\ e_{wi} + T, & \text{if } e_{wi} < -2T \end{cases} \tag{16}$$

and

$$w_i = \text{mod } (\ell_{wi}, 2), \text{ if } e_{wi} \in [-2T, 2T + 1], \tag{17}$$

where $\ell_{wi} = fix(e_{wi})$ is the integer element of e_{wi} and $\delta_{wi} = e_{wi} - \ell_{wi}$. It is worth noting that $\delta_{wi} = \delta_i$ since the embedding process only expands the integer part while keeping the fraction element unchanged. Finally, the original pixels are recovered from the prediction errors by performing inverse prediction operation.

The ratio between the sets S_1 and S_2 can be controlled by changing the embedding threshold T. The bigger the threshold value T, the higher the embedding payload, the more the embedding distortion is.

4.2 Overflow and Underflow

The marked signal may suffer from overflow and underflow problems due to NIPE and histogram shifting operations. Towards this direction, an embedding testing step is first performed to pick up those *bad pixels/samples*, as described in [10]. When a marked pixel/sample in intensity/magnitude is not in the interval $[0, 2^m - 1]$, the sample is labeled as a bad pixel/sample. All bad pixels/samples in position will be recorded as part of the payload and keep their value unchanged in the embedding.

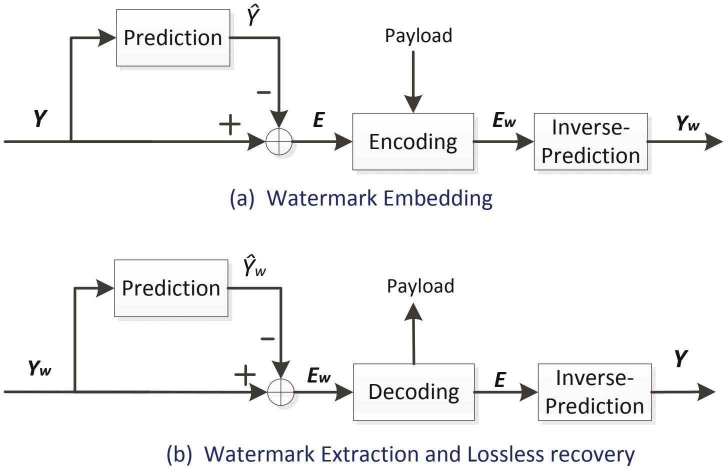
Take digital audio files as examples. Usually, the length of an audio file is not longer than 6 minutes. For the sampling rate of 44.1 kHz, the number of the samples is $6 \times 60 \times 44,100 = 15,876,000 < 2^{25}$. Therefore, 25 bits of information is required to indicate the position of a bad sample. In addition, 15 bits of information is required for conveying the parameter T to the decoder. The capacity of the proposed method can be computed as:

$$C = \frac{N_1 - 25 \times N_p - 15 - 25}{N}, \tag{18}$$

where N_1 is the number of the expandable set S_1, N_p the number of the bad samples and N the length of cover-signal. Without recursive embedding, the maximal capacity of the proposed NIPE scheme is close to 1 bps (or 1 bpp for images).

5 Encoder and Decoder

The proposed reversible watermarking scheme, as illustrated in Fig. 2, can be used for image or audio files. In this paper, we take audio clips for experimental testing. In the embedding, the maximal capacity (P_{max}) of an audio signal is first computed by using the proposed reversible watermarking strategy, $P_{max} <= N$. When an actual payload size $P (P <= P_{max})$ is given, the threshold T can be computed. For recovering the cover-signal, the information of P and T is needed to be sent to the decoder in a way that the LSB values of the first 40 prediction errors are kept (as part of the payload) and then replaced by the parameters P (25 bits) and T (15 bits).



(a) Watermark Embedding

(b) Watermark Extraction and Lossless recovery

Fig. 2. Proposed reversible watermarking scheme

Referring to Sections 2 and 3, the embedding process of the proposed scheme is described as follows:

1) Predict the cover-signal Y to get the prediction errors E;
2) Find the bad samples in position by using the embedding testing operation. Each bad sample consumes 25 bits of payload;
3) Embed the data (including P, T and the bad samples in position) into E to generate E_w;
4) Reconstruct the marked audio signal Y_w from E_w by using inverse-prediction operation.

In the decoder, the same prediction operation is performed on Y_w to get E_w. Then the information of P and T is extracted from the LSB values. Furthermore, the hidden data and the original prediction errors E are extracted from E_w. Finally, the original audio signal Y is recovered by using the inverse-prediction operation.

6 Experimental Testing and Analysis

We choose 9 standard audio signals downloaded from the web site (*http : //sound.media.mit.edu/resources.php*) as test data set to report: 1) performance of the proposed predictor, 2) effect of the embedding threshold T on the embedding payload and distortion and 3) the embedding payload and distortion comparison of the proposed NIPE algorithm against several existing state of the art reversible audio watermarking algorithms [13–15].

6.1 Prediction Accuracy

Several reversible audio watermarking algorithms [13–15] exist in the literature. The method in [13] compresses the dynamic range and then commands the range for data embedding. Two difference operators labeled as dyad-based and triad-based transforms in [14] are adopted to decorrelate digital audio for DE-based reversible watermarking. In [15], audio signal is predicted by using three past samples as the prediction context for IPE-based reversible watermarking. The predictor proposed in this paper has been described in the Section 3.2.

Consider a signed example clip titled by *'karaoke_tempo'* (the mean μ and the standard derivation σ of which are -0.4402 and 2855.8, respectively). Fig. 3 plots histograms of the example clip proceeded with four different methods: a) dyad-based transform (two neighboring samples as a vector to generate an integer average and a difference error, where $\mu = 0.2976$ and $\sigma = 2019.4$) [14], b) triad-based transform (three neighboring samples to generate an integer average and two difference errors, where $\mu = 0.0383$ and $\sigma = 1757.5$) [14], c) difference coding using past samples (where $\mu = 0$ and $\sigma = 585.1$) [15, 17] and d) the predictor proposed in this paper (where $\mu = 0$ and $\sigma = 325.6$).

We can observe that the proposed prediction method provides a smaller standard deviation ($\sigma = 325.6$) while the mean value (μ) is close to Zero. This indicates that the proposed predictor provides higher prediction accuracy than the existing several methods. Thus, the proposed predictor is beneficial to reduce the embedding distortion for reversible audio watermarking.

6.2 Effect of the Parameter *T*

As mentioned above, the parameter T plays a role to make a trade-off between the embedding capacity and distortion. In order to achieve better fidelity for the marked signal, it is necessary to look for an appropriate embedding threshold

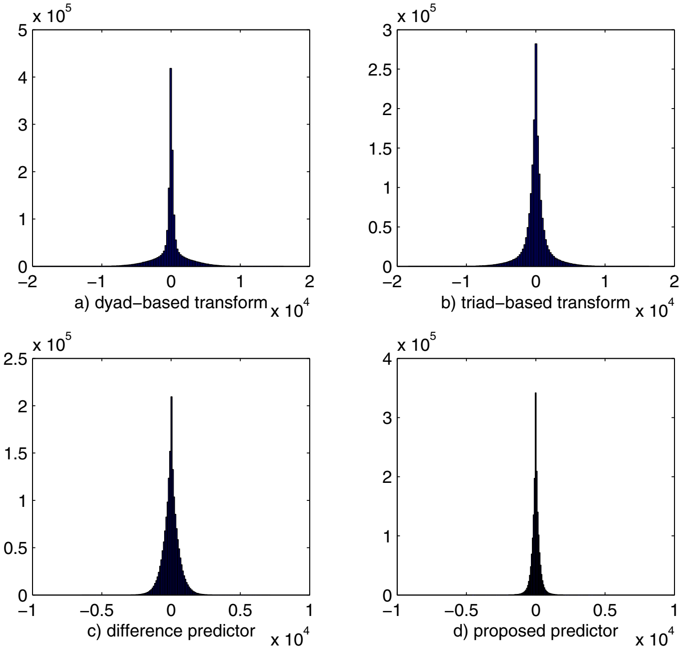


Fig. 3. Histograms of four residual signals resulted from the clip $'karaoke\_tempo'$

value for a given payload. Fig. 4 plots the effect of the different T ($T \in \{1 : 2 : 7, 10, 30, 50, 100, 200 : 200 : 1000, 1500, 2000\}$) values based on three example clips in length of 10 seconds (the former portion of the files $'karaoke\_tempo'$, $'track4'$ and $'track7'$) randomly chosen from 9 test files. We can see from the figure that the SNR values and the embedding capacity are different from three different clips. In practice, in order to satisfy some degree of the fidelity (such as the SNR values of over 20 dB [18]), the embedding threshold and the payload for a given audio file should be estimated in advance.

6.3 Performance Comparison

The embedding capacity and the fidelity (such as the SNR standard in [18]) are two important factors. The three chosen example clips above are adopted to report the performance of the proposed scheme in comparison with four existing reversible audio watermarking algorithms [13–15]. The simulation results are plotted in Figures 5, 6 and 7. We can observe from these three figures that

1) In [14], the embedding distortion of the dyad-based transform method is somewhat lower than that of the triad-based one. Their maximal capacity values are bounded to 0.5 bps and $\frac{2}{3}$ bps, respectively.

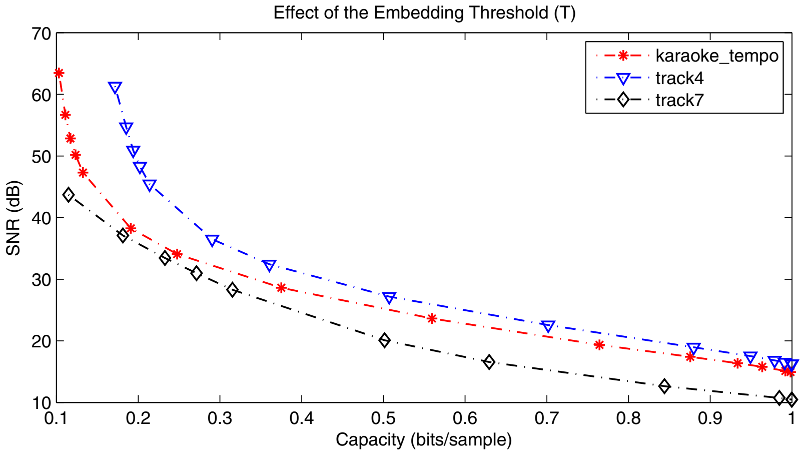


Fig. 4. Effect of the parameter T on the embedding capacity and distortion by using three example clips of 10 seconds

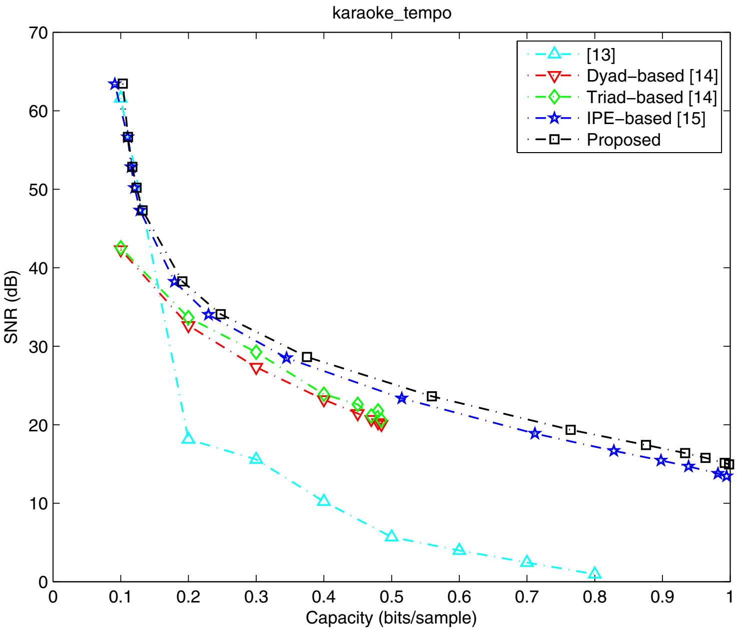


Fig. 5. Performance of five algorithms with the clip $'karaoke\_tempo'$

2) The algorithm in [13] achieves higher embedding payload but suffers from an undesirable distortion due to quantization error and loudness change in the compression-expansion embedding phase. Comparing with the expansion embedding methods (in [14, 15]), the proposed method in this paper provides the lowest SNR values for the same payload. Of course, it is unfair to measure the algorithm in [13] with the SNR standard because the loudness change often causes a low SNR value but can keep the audible quality well.
3) PE based methods (IPE in [15] and NIPE proposed in this paper) significantly improve the embedding capacity or reduce the embedding distortion in comparison with the previous algorithms. The maximal capacity is close to 1 and the SNR values are satisfactory.
4) Comparing with the IPE embedding method, the proposed NIPE scheme reduces the embedding distortion. It is owing to the fact that the NIPE method allows a predictor generating non-integer prediction values and predicting a sample with both past and future neighboring samples. In the IPE-based scheme, the predicted value must take integer value and the context cannot involve future samples for prediction of the current sample.

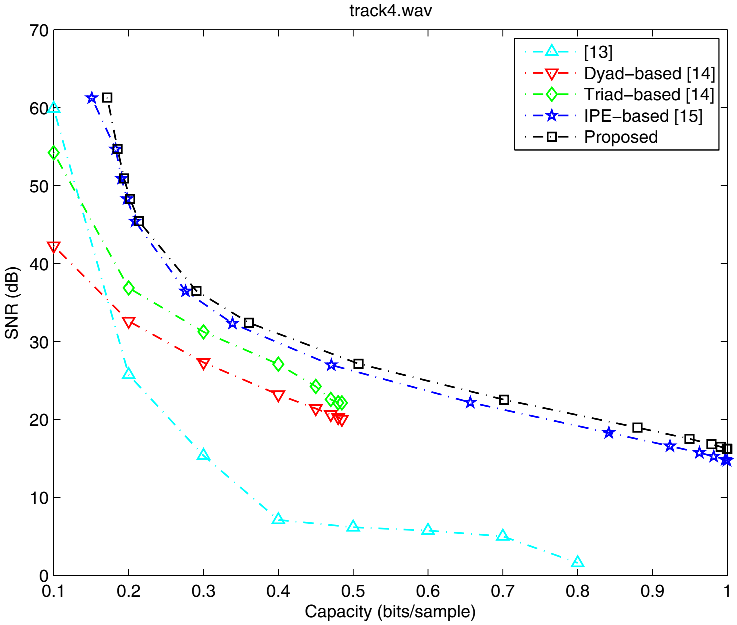


Fig. 6. Performance of five algorithms with the clip $'track4'$

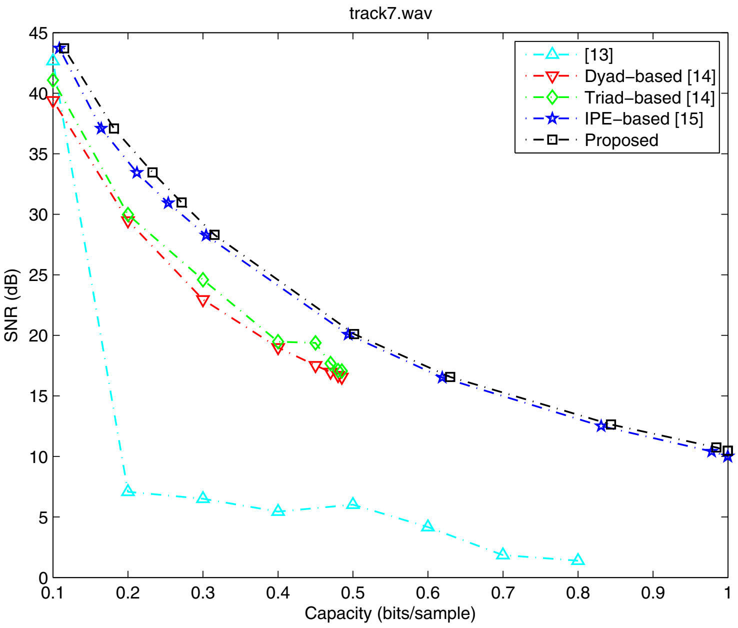


Fig. 7. Performance of five algorithms with the clip $'track7'$

7 Conclusions

This paper presents a NIPE embedding algorithm, which is more scalable than the IPE method developed by Thodi [11] since it can deal with non-integer prediction errors for reversible watermarking. We also show that the proposed method can allow a predictor to estimate the current sample with past and future pixels/samples but the existing PE embedding methods (such as [11, 12, 14, 15]) only agree with its predictor to predict a pixel/sample with past pixels/samples. Furthermore, we design a non-integer output of prediction model to show that a predictor with non-integer output can achieve higher prediction precision since more neighboring samples in an audio file can be applied as the prediction context. Experimental results have shown that the proposed NIPE method with the new audio predictor has better results compared to the original IPE method and the previous several reversible audio watermarking algorithms. In the future research, there is one room to design better image predictor for the NIPE embedding approach since this approach allows a predictor to output non-integer values by using noncausal prediction with past and future pixels.

Acknowledgments. This work was supported in part by NSFC (No. 60903177, 61272414), in part supported by the Science and Technology Project of Guangzhou of China (No. 2012J4100108). The author would like to thank the reviewer's professional suggestions for improving this paper.

References

1. Ni, Z., Shi, Y., Ansari, N., Wei, S.: Reversible data hiding. In: Proc. IEEE Int. Symp. Circuits and Systems, vol. 2, pp. 912–915 (May 2003)
2. De Vleeschouwer, C., Delaigle, J.E., Macq, B.: Circular interpretation of bijective transformations in lossless watermarking for media asset management. IEEE Trans. Multimedia 5(1), 97–105 (2003)
3. Fridrich, J., Goljan, M., Du, R.: Invertible authentication. In: Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents III, vol. 3971, pp. 197–208 (January 2001)
4. Fridrich, J., Goljan, M., Du, R.: Lossless data embedding for all image formats. In: Proc. SPIE Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, pp. 572–583 (January 2002)
5. Kalker, A.A.C.M., Willems, F.M.J.: Capacity bounds and constructions for reversible data-hiding. In: Proc. 14th Int. Conf. Digital Signal Processing, vol. 1, pp. 71–76 (July 2002)
6. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized-LSB data embedding. IEEE Trans. Image Process. 14(2), 253–266 (2005)
7. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. 13(8), 890–896 (2003)
8. Alattar, A.M.: Reversible watermark using the difference expansion of a generalized integer transform. IEEE Trans. Image Process. 13(8), 1147–1156 (2004)
9. Kamstra, L., Heijmans, H.: Reversible data embedding into images using wavelet techniques and sorting. IEEE Trans. Image Process. 14(12), 2082–2090 (2005)
10. Sachnev, V., Kim, H.J., Nam, J., Suresh, S., Shi, Y.: Reversible watermarking algorithm using sorting and prediction. IEEE Trans. Circuits Syst. Video Technol. 19(7), 989–999 (2009)
11. Thodi, D.M., Rodriguez, J.J.: Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process. 15, 721–729 (2007)
12. Coltuc, D.: Improved embedding for prediction-based reversible watermarking. IEEE Trans. Inf. Forensics Security 6(3), 873–882 (2011)
13. van der Veen, M., Bruekers, F., van Leest, A., Cavin, S.: High-capacity reversible watermarking for audio. In: Proc. SPIE Photonics West, Electronic Imaging 2003, Security and Watermarking of Multimedia Contents V, San Jose, California, vol. 5020, pp. 1–11 (January 2003)
14. Bradley, B., Alattar, A.M.: High-capacity, invertible, data-hiding algorithm for digital audio. In: Proc. SPIE Photonics West, Electronic Imaging 2005, Security and Watermarking of Multimedia Contents VII, San Jose, California, vol. 5681, pp. 789–800 (January 2005)
15. Yan, D., Wang, R.: Reversible data hiding for audio based on prediction error expansion. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 249–252 (2008)
16. Weinberger, M., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Trans. Image Process. 9(8), 1309–1324 (2000)
17. Robinson, T.: SHORTEN: Simple lossless and near-lossless waveform compression. Cambridge Univ. Eng. Dept., Cambridge, UK. Tech. Rep. 156 (1994)
18. Katzenbeisser, S., Petitcolas, F.A.P. (eds.): Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Inc. (2000)

Walsh-Hadamard Transform in the Homomorphic Encrypted Domain and Its Application in Image Watermarking

Peijia Zheng and Jiwu Huang

School of Information Science and Technology
Sun Yat-Sen University
Guangzhou, 510006, China
zhengpj@mail2.sysu.edu.cn, isshjw@mail.sysu.edu.cn

Abstract. How to embed and/or extract watermarks on encrypted images without being able to decrypt is a challenging problem. In this paper, we firstly discuss the implementation of Walsh-Hadamard transform (WHT) and its fast algorithm in the encrypted domain, which is particularly suitable for the applications in the encrypted domain for its transform matrix consists of only integers. Then by modifying the relations among the adjacent transform coefficients, we propose an WHT-based image watermarking algorithm in the encrypted domain. Due to the constrains of the encryption, extracting a watermark blindly from an encrypted image is not a easy task. However, our proposed algorithm possesses the characteristics of blind watermark extraction both in the decrypted domain and the encrypted domain. This means neither the plain image nor its encrypted version is required for the extraction. The experiments demonstrate the validity and the advantages of our proposed method.

Keywords: Secure signal processing, watermark, homomorphic encryption, signal processing in the encrypted domain, Walsh Hadamard transform.

1 Introduction

Watermarking is an method to protect the copyright of digital media by hiding proprietary information in media. The security of watermarking is a challenging problem in the watermarking community. Many efforts focusing on watermark security have been reported in literature [1] [2]. In fact, there are at least two problems on the security. The first one is the security of the original media under being watermarked. Almost all the existing watermark schemes accomplish the watermark embedding and extraction on the plain media. Hence, the watermark embedder must be the owner of the plain media or the trusted third party, in order to make sure the original media is not exposed to the untrusted party. The second one is the security of the watermark scheme itself. For example, how to prevent illegal watermark embedding, extracting, and removal.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 240–254, 2013.

Though there are some reports on integrating watermark embedding and encrypting [3] [4], it causes additional constraints to the watermarking algorithm, meanwhile. Some works [5] have been proposed to solve the first problem, however, the visual quality of the watermarked images are not so good as expected. Single processing in the encrypted domain, also referred to as secure signal processing (SSP), provides another way to solve the first problem. This new technology allows one to manipulate the encryption data by means of signal processing without decrypting.

There have been some related works on secure signal processing over the past few years. An interactive buyer-seller watermarking protocol for invisible watermarking was proposed in [6], where the seller does not get to know the exact watermarked copy that the buyer receives. Bianchi *et al.* [7] conducted an investigation on the implementation of the discrete Fourier transform (DFT) as well as the fast Fourier transform (FFT) on encrypted signals. A data encrypting method, which packs several samples as a single one, was proposed by Troncoso-Pastoriza *et al.* [8], and later generalized by Bianchi *et al.* [9]. In [10] [11], the authors proposed schemes for privacy-preserving face recognition by using the Paillier cryptosystem. Zheng *et al.* [12] presented a new technique to implement the discrete wavelet transform (DWT) and Multiresolution Analysis (MRA) in the encrypted domain. They also provided a new method to handle the data expansion without decrypting. Barni *et al.* gave a privacy-preserving fingercode authentication in [13]. In [14], they proposed a system for the secure classification of ECG (electrocardiogram) signals with branching programs and neural networks.

Due to the limitation of the encryption, it is very difficult, sometimes impossible, to transplant the existing mature watermark scheme to the encrypted domain. Thus it is meaningful to design a new image watermark scheme under the constraints of the homomorphic encrypted domain. Generally, the watermark algorithms based on transform domain are more robust than the others. Owing to the quantization error, DFT [7] and DCT [15] in the encrypted domain will bring a noise to the plain reconstructed image, which may decrease the visual effect of the watermarked image. Since the transform matrix of the Walsh-Hadamard transform (WHT) contains only $+1$ and -1, one can avoid the quantization error of its implementation in the encrypted domain. Therefore WHT is particularly suitable to be used as a transform method for image watermarking in the encrypted domain.

This paper addresses the issue of image watermarking in the encrypted domain. Firstly, we describe a framework for performing WHT in a homomorphic encrypted domain. Secondly, we develop a WHT-based image watermarking scheme and transplant it to the encrypted domain. The proposed scheme possesses the characteristics of blind watermark extraction both in the decrypted domain and the encrypted domain. Finally, we conduct several experiments to substantiate the proposed scheme. Our technique can be applied to other applications where a secure watermarking algorithm is required.

The remainder of this paper is organized as follows. In Section 2, we discuss the implementation of WHT in the encrypted domain. In Section 3, we propose the blind-extraction image watermarking algorithm in the encrypted domain. Section 4 gives some experiments on the image watermarking algorithm. We conclude the paper and provide suggestions for future work in Section 5.

2 Walsh-Hadamard Transform in the Encrypted Domain

WHT is used widely in the field of signal processing. The transform matrix of WHT contains only ±1, and no multiplications are required in the computation. Thus WHT is more efficient than other orthogonal transformations, such as DFT or DCT. Another advantage of WHT has is that WHT will not bring the quantization error in the encrypted domain. WHT can therefore be perfectly reconstructed in the encrypted domain, which is shown in Section 2.3. Hence, in contrast to DFT and DCT, WHT is particularly suitable for image watermarking in the encrypted domain. Since the implementation of WHT in the encrypted domain has not been reported yet, we present the implementation first.

2.1 Homomorphic Cryptosystem

The homomorphic cryptosystem [16] is an encryption function which allows one to operate the ciphertexts without decrypting. Specifically, suppose $\mathcal{D}[\cdot]$ and $[\![\cdot]\!]$ are the decrypting operator and encrypting operator, respectively. If m_1 and m_2 are any two plaintexts, we have

$$\mathcal{D}\left[[\![m_1]\!] \star [\![m_2]\!]\right] = m_1 * m_2 \tag{1}$$

where operator '$\star$' and '$*$' are the algebraic operations performed in the ciphertext space and the plaintext space, respectively.

For convenience, we use the Paillier cryptosystem as data encryption method in this paper. We refer to [17] for the detailed definition of the Paillier cryptosystem. Based on the definition, we have the additive homomorphic properties as

$$\mathcal{D}\left[[\![m_1]\!]\,[\![m_2]\!] \bmod N^2\right] = m_1 + m_2 \mod N, \tag{2}$$

$$\mathcal{D}\left[[\![m_1]\!]^{m_2} \bmod N^2\right] = m_1 m_2 \mod N. \tag{3}$$

The Paillier cryptosystem also has the self-blinding property, i.e.,

$$\mathcal{D}\left[[\![m_1]\!]\, r^N \bmod N^2\right] = m_1 \mod N \tag{4}$$

where r is a random element in $\mathbb{Z}_N^*$. $\mathbb{Z}_N^*$ consists of all the integers in $\mathbb{Z}$ which are relative prime with N. The self-blinding property means that every ciphertext can be publicly changed into another ciphertext which has the same plaintext.

These properties will be applied in the following sections to perform the implementation of WHT and image watermarking in the encrypted domain.

2.2 Integer Approximation and Evaluation

Let us consider the image $I(x,y)$, with the size of $M \times M$, where M is assumed to be the power of two. The 2D WHT of natural ordering is defined as

$$X(k,l) = \frac{1}{M} \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \mathbf{H}_\mu(k,x)\, I(x,y)\, \mathbf{H}_\mu(y,l), \quad k,l = 0,1,\ldots,M-1 \tag{5}$$

where $\mu = \log_2 M$ and $\mathbf{H}_\mu$ denotes the Hadamard transform matrices. $\mathbf{H}_\mu$ can be generated by the core matrix

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{6}$$

and the Kronecker product recursion

$$\mathbf{H}_\mu = \mathbf{H}_1 \otimes \mathbf{H}_{\mu-1} = \begin{pmatrix} \mathbf{H}_{\mu-1} & \mathbf{H}_{\mu-1} \\ \mathbf{H}_{\mu-1} & -\mathbf{H}_{\mu-1} \end{pmatrix} \tag{7}$$

where $\otimes$ is the Kronecker product operator. According to the method in [18], one can easily obtain WHT of sequency ordering and other orderings by rearranging the outputs (5). Therefore we will focus on WHT of natural ordering in the following.

Since all the plaintexts and the ciphertexts are represented by integers in the cryptosystem, the signal must also be represented by integers too. Obviously, all the elements of $I(x,y)$ are integers between 0 and 255, i.e., $I(x,y) \in \mathbb{Z}_{256}$. However, the transform coefficients of an image may be negative, and we still need to consider the problem of representing the negative integers in the cryptosystem. Suppose N is the modulus of the cryptosystem. We let $N \geq 2\sup\{|S(k)|\} + 1$, where $\sup\{\cdot\}$ denotes the least upper bound operator performed on a sequence, and $S(k)$ is the plain value of the processed result in the encrypted domain.

According to the above discussion, we give the definition of the integer approximation of the 2D WHT as

$$V(k,l) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \mathbf{H}_\mu(k,x)\, I(x,y)\, \mathbf{H}_\mu(y,l), \quad k,l = 0,1,\ldots,M-1. \tag{8}$$

Since all the operations are either integer additions or integer subtractions, (8) can be implemented in the encrypted domain by using the homomorphic properties. In the case that the input signal is encrypted with the Paillier cryptosystem, by means of the equations (2) and (3), the implementation of the 2D WHT in the encrypted domain is given as

$$[\![V(k,l)]\!] = \prod_{x=0}^{M-1} \prod_{y=0}^{M-1} [\![I(x,y)]\!]^{\mathbf{H}_\mu(k,x)\mathbf{H}_\mu(y,l)} \triangleq \tilde{V}(k,l), \quad k,l = 0,1,\ldots,M-1 \tag{9}$$

where all the multiplications and exponentiations are carried out under N^2.

The definition of the inverse WHT (IWHT) is identical to the forward WHT. If $X'(k,l)$ is the input transform coefficients, which may not be identical to $X(k,l)$, then the reconstructed image is given as

$$\hat{I}\left(x,y\right)=\frac{1}{M}\sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\mathbf{H}_{\mu}\left(x,k\right)X'\left(k,l\right)\mathbf{H}_{\mu}\left(l,y\right),\quad x,y=0,1,\ldots,M-1. \quad (10)$$

A similar approach leads to the definition of the integer IWHT. Assuming we have already obtained the integer 2D WHT coefficients $V'(k,l)$, the integer approximation of the 2D IWHT is defined as

$$I'\left(x,y\right)=\sum_{x=0}^{M-1}\sum_{y=0}^{M-1}\mathbf{H}_{\mu}\left(x,k\right)V'\left(k,l\right)\mathbf{H}_{\mu}\left(l,y\right),\quad x,y=0,1,\ldots,M-1, \quad (11)$$

where $V'(k,l)$ is corresponding to $X'(k,l)$. Since all the input arguments are integers, (11) can be computed in the encrypted domain as

$$[\![I'\left(x,y\right)]\!]=\prod_{k=0}^{M-1}\prod_{l=0}^{M-1}\tilde{V}'\left(k,l\right)^{\mathbf{H}_{\mu}\left(x,k\right)\mathbf{H}_{\mu}\left(l,y\right)}\triangleq\tilde{I}'\left(x,y\right),\quad x,y=0,1,\ldots,M-1. \quad (12)$$

For the sake of simplicity, we use WHT-ed and IWHT-ed to denote the implementation of WHT and IWHT in the encrypted domain, respectively.

2.3 Data Recovery and Upper Bound

In order to implement WHT and IWHT in the encrypted domain by using (9) and (12), we need to consider some issues. Since all the calculations of (9) and (12) are in the finite ring $\mathbb{Z}_N$, the plain value of the processed result S must not be larger than N. Thus we should find a upper bound on S. Let us consider the implementation of WHT in the encrypted domain first. It is obvious that

$$\begin{aligned}\mathcal{D}\left[\tilde{V}\left(k,l\right)\right]&=V\left(k,l\right)\bmod N=MX\left(k,l\right)\bmod N\\&\triangleq Z\left(k,l\right).\end{aligned} \quad (13)$$

However, $Z(k,l)$ may sometimes be negative. Taking the negative coefficients into account, the recovery condition is given as

$$2M\sup_{k,l}\left\{\left|X\left(k,l\right)\right|\right\}+1<N. \quad (14)$$

Moreover, we must find a method to recover every value from the decryption of the output. Actually, under the condition (14), $X(k,l)$ can be obtained directly from $\tilde{V}(k,l)$ as

$$X\left(k,l\right)=\begin{cases}\dfrac{\mathcal{D}\left[\tilde{V}\left(k,l\right)\right]}{M}, & \text{for } Z\left(k,l\right)<N/2\\ \dfrac{\mathcal{D}\left[\tilde{V}\left(k,l\right)\right]-N}{M}. & \text{for } Z\left(k,l\right)>N/2\end{cases} \quad (15)$$

As for the inverse WHT in the encrypted domain, a similar approach leads to the upper bound of the reconstructed image. By using the homomorphic property, we have

$$\begin{aligned}\mathcal{D}\left[\tilde{I}'(x,y)\right] &= \sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\mathbf{H}_\mu(x,k)\,\mathcal{D}\left[\tilde{V}'(k,l)\right]\mathbf{H}_\mu(l,y)\\ &= M\sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\mathbf{H}_\mu(x,k)\,X'(k,l)\,\mathbf{H}_\mu(l,y) \mod N\\ &= M^2\hat{I}(x,y) \mod N \triangleq Y(x,y)\,. \end{aligned}\tag{16}$$

Specifically, if $V'(k,l) = V(k,l)$, then $Y(x,y) = M^2 I(x,y)$. It implies that any image can be completely reconstructed in the encrypted domain, i.e. perfect reconstruction. The recovery condition of the reconstructed image is given as

$$2M^2\sup_{x,y}\left\{\hat{I}(x,y)\right\}+1<N. \tag{17}$$

When condition (17) is satisfied, we can obtain $\hat{I}(x,y)$ from the $\tilde{I}'(x,y)$ as

$$\hat{I}(x,y)=\begin{cases}\dfrac{\mathcal{D}\left[\tilde{I}'(x,y)\right]}{M^2}, & \text{for } Y(x,y)<N/2\\ \dfrac{\mathcal{D}\left[\tilde{I}'(x,y)\right]-N}{M^2}. & \text{for } Y(x,y)>N/2\end{cases}\tag{18}$$

Obviously, $\sup_{x,y}\{I(x,y)\} = 255$. The first element of matrix $X(k,l)$ is the sum of all the pixels in $I(x,y)$. Thus we have $\sup_{k,l}\{|X(k,l)|\} = 255M^2$. In the case of $V'(k,l) = V(k,l)$, by combining (14) and (17), the final recovery condition can be given as

$$N > \max\{510M^3, 510M^2\} = 510M^3. \tag{19}$$

According the above analysis, an interesting phenomenon may be obtained. In contrast to the implementation of WHT in the plain domain, the implementation in the encrypted domain will expand the plain value of the expected value. The expanding factor depends on two parameters, the dimension and the length of the input signal. More specifically, each implementation of 2D WHT-ed and 2D IWHT-ed will expand the plain value by a fixed factor M. Generally, the image size M is only tens of bits for real images, while N should be 1024 bits according to [17]. Therefore the expanding factor M is negligible compared with N, and the WHT-based applications can be well transplanted to the encrypted domain, without considering the data overflow.

2.4 Fast WHT in the Encrypted Domain

2D WHT is a separable transform, i.e., a 2D transform which can be decomposed into two 1D transforms. Specifically, performing 2D WHT on $I(x,y)$ is equivalent to performing 1D WHT on the each column of $I(x, y)$ first and then performing 1D WHT on the each row to the former result. Hence, we focus on the fast algorithm of 1D WHT-ed in this paper.

In fact, the computational complexity of WHT can be reduced from M^2 to $M \log M$ by a fast algorithm [18]. The fast algorithm follows the recursive definition of the Hadamard matrix (7). Similar to FFT, the fast WHT recursively breaks down a WHT of size M into two smaller WHTs of size $M/2$. Therefore there are totally $\log_2 M$ stages of breaking down by means of the fast algorithm. Since there are only M additions/subtractions at each stage, there are totally $M \log_2 M$ additions/subtractions for the fast WHT. More specifically, every two coefficients are obtained at one stage from another two coefficients at the previous stage by using only addition or substraction. That is, by omitting the scaling factor, the fast WHT at i-th stage can described as

$$v^i(k_0) = v^{i-1}(k_0) + v^{i-1}(k_1) \tag{20}$$

$$v^i(k_1) = v^{i-1}(k_0) - v^{i-1}(k_1) \tag{21}$$

where $v^i(k_0)$ and $v^i(k_1)$ are the two coefficients obtained at i-th stage, $i = 1, 2, \ldots, \log_2 M$. The indices k_0, k_1 are integers which vary between 0 and $M-1$.

By using the homomorphic properties, we implement the fast WHT at i stage in the encrypted domain as

$$[\![v^i(k_0)]\!] = [\![v^{i-1}(k_0)]\!]\,[\![v^{i-1}(k_1)]\!]\,, \tag{22}$$

$$[\![v^i(k_1)]\!] = [\![v^{i-1}(k_0)]\!]\,[\![v^{i-1}(k_1)]\!]^{-1}\,. \tag{23}$$

Suppose $\{[\![v^{\log_2 M}(k)]\!]\}$ are the encrypted coefficients obtained at the final stage. After a simple deduction, we get the relationship between the direct WHT-ed and the fast WHT-ed as

$$[\![v^{\log_2 M}(k)]\!] = \tilde{v}(k) \tag{24}$$

where $\tilde{v}(k)$ is the coefficient obtained by the direct WHT-ed. Since the definition of IWHT is identical to that of WHT, the method described above can also be used as a fast algorithm to implement IWHT in the encrypted domain.

3 Blind Image Watermarking in the Encrypted Domain

In order to embed a watermark on an encrypted image, we should tackle two challenging issues. The first one is how to achieve the goal of blind watermark exaction. Since the original image is protected by the encryption, it is not practical to involve the plain original image into the extraction. The second one is how to evaluate the visual quality of the watermarked image. Since the input image is in the encrypted form and the embedder don't have the decrypting key, it is difficult for him/her to determine whether the visual effect of the watermarked images is good or bad.

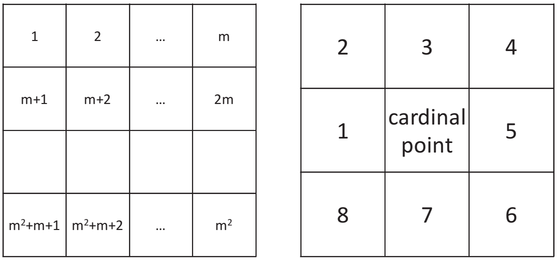


Fig. 1. The relationship between the reference positions and the values e_j and d_j

3.1 Watermark Embedding

The embedding domain, e.g. the spatial domain or the transform domain, plays a crucial role in robust performance and the visual quality of the watermarked image. In order to make the watermark scheme more robust, we choose to embed the watermark in the transform domain rather than the spatial domain. We describe the algorithms in the plain domain first and then give its implementation in the encrypted domain.

Watermarking in the Plain Domain. Suppose the embedding message is a binary signal $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$, where $w_j \in \{0, 1\}$. Our watermarking algorithm in the plain domain can be described as follows.

(1) To segment the original image $I(x, y)$ into non-overlapping blocks of $m \times m$. m is assumed to be an integral power of two. Thus there are totally $M_b = (M/m)^2$ blocks after the segmentation.
(2) To perform WHT of sequency ordering on each segmented block and obtain the transform coefficient blocks, denoted by $\{V_j\}_1^{M_b}$. In order to protect the watermarked images from illegal extraction, a random number sequence is introduced to control the embedding. Denote the random number sequence by $\mathbf{a} = \{a_1, a_2, \ldots, a_n\} \in \mathscr{P}(\{1, 2, \ldots, M_b\})$, where $\mathscr{P}(\cdot)$ denotes the power set of a set. Select n coefficient blocks from $\{X_j\}_1^{M_b}$ according to $\mathbf{a}$ in sequential scan order. The selected blocks are denoted by $\{X_1, X_2, \ldots, X_n\}$.
(3) To choose two random sequences $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$ and $\mathbf{d} = \{d_1, d_2, \ldots, d_n\}$, where $e_j \in \{2, 3, \ldots, m^2\}$ and $d_j \in \{1, 2, \ldots, 8\}$. e_j denotes one special point in block X_j, called the cardinal point of X_j. The value of e_j corresponds to the position in X_j in sequential scan order. Whereas d_j stands for the orientation which surrounds the cardinal point. The value of d_j increases as we revolve clockwise around the cardinal point. We show the corresponding relation between the values of e_j and d_j and the positions in block X_j in Fig. 1.
(4) In the selected block X_j, we choose the cardinal point according to the value of e_j. The cardinal point of X_j is $X_j(k_0, l_0) = V_j(\lfloor e_j/m \rfloor, e_j \bmod m)$. We use $X_j(k_1, l_1)$ to denote the adjacent point surrounding $X_j(k_0, l_0)$, with respect

to d_j. The watermark is embedded by modifying the transform coefficient $X_j(k_1, l_1)$. The detailed modification of the coefficient $X_j(k_1, l_1)$ is given as

$$X_j\left(k_1, l_1\right) = \begin{cases} X_j\left(k_0, l_0\right), & \text{if } w_j = 0 \\ X_j\left(k_0, l_0\right) + \alpha_j. & \text{if } w_j = 1 \end{cases} \tag{25}$$

where $\alpha_j \in \mathbb{N}^*$ is a locally adjustable amplitude factor. Since the other coefficients is quite small compared with the $V_j(1, 1)$, this modification is actually very slight. We use X_j^* to denote the coefficient block which has been modified.

(5) To perform IWHT on all the coefficient blocks, including the modified blocks and the unmodified ones, in order to output the watermarked image, denoted by $I_w(x, y)$. In order to keep format compliance, $I_w(x, y)$ will undergo the quantization process. The quantized watermarked image is denoted by $I_{w,256}(x, y)$.

The triple $(\mathbf{a}, \mathbf{e}, \mathbf{d})$ is the secret key of the watermark algorithm. It determines the positions where the watermark is embedded. It will be sent to the watermark extractor and take part in the process of watermark extraction.

Watermarking in the Encrypted Domain. By using the homomorphic properties of the cryptosystem, the watermark embedding algorithm can also be implemented in the encrypted domain. Suppose the input to the watermark embedder is an encrypted image $[\![I(x, y)]\!]$. The embedder knows nothing about the plain image while still try to embed $\mathbf{w}$ in the plain image. Actually the watermark embedding can be carried out in the encrypted domain without an interactive protocol. The detail of the implementation is given as follows.

We segment the encrypted image $[\![I(x, y)]\!]$ into $(M/m)^2$ blocks of $m \times m$. Then we apply WHT-ed to each block. According to the random integer sequence $\mathbf{a}$, n blocks are selected for watermark insertion. We denote those selected blocks by $\{\tilde{V}_1, \tilde{V}_2, \ldots, \tilde{V}_n\}$. In the block $\tilde{V}_j$, the cardinal point $\tilde{V}_j(k_0, l_0)$ is chosen according to the value of e_j, i.e., $\tilde{V}_j = \tilde{V}_j(\lfloor e_j/m \rfloor, e_j \bmod m)$. With respect to the value of d_j, we choose the adjacent point of $\tilde{V}_j(k_0, l_0)$, denoted by $\tilde{V}_j\left(k_1, l_1\right)$. Then the watermark embedding in the encrypted domain can be accomplished by modifying the encrypted coefficients. Specifically, the coefficient modification of j-th selected block can be given as

$$\tilde{V}_j\left(k_1, l_1\right) = \begin{cases} \tilde{V}_j\left(k_0, l_0\right) r^N \bmod N^2, & \text{if } w_j = 0 \\ \tilde{V}_j\left(k_0, l_0\right) [\![\alpha_j m]\!] \bmod N^2. & \text{if } w_j = 1 \end{cases} \tag{26}$$

where r is a random number chosen in $\mathbb{Z}_N$. We use $\tilde{V}_j^*$ to denote the encrypted coefficient block which has been modified. After modifying the coefficients, we perform IWHT-ed on all the coefficient blocks, including both the modified blocks and the unmodified ones. The processed encrypted image, i.e. the encrypted version of the watermarked image, is denoted by $\tilde{I}_w\left(x, y\right)$. The above manipulations only use the homomorphic properties of the encryption, and rely on no interactive protocol.

We now explain why we call $\tilde{I}_w(x,y)$ the encrypted version of $I_w(x,y)$. Since the homomorphic cryptosystem possesses the self-blinding property (4), by using the equation (13), we have

$$\mathcal{D}\left[\tilde{V}_j\left(k_0,l_0\right)r^N \mod N^2\right] = mX_j^*\left(k_0,l_0\right). \tag{27}$$

Similarly, by using the homomorphic properties (2) and equation (13) , we have

$$\mathcal{D}\left[\tilde{V}_j\left(k_0,l_0\right)[\![\alpha_j m]\!] \mod N^2\right] = mX_j^*\left(k_0,l_0\right) + m\alpha_j. \tag{28}$$

Hence, by combining the two equations (27) and (28), we obtain

$$\mathcal{D}\left[\tilde{V}_j^*\left(k_1,l_1\right)\right] = mX_j^*\left(k_1,l_1\right). \tag{29}$$

Since $\tilde{I}_w(x,y)$ is obtained by performing 2D IWHT-ed on all the encrypted coefficient blocks, we can get the relationship between $\tilde{I}_w(x,y)$ and $I_w(x,y)$ by using (16). Specifically, the relationship can be obtained as

$$\mathcal{D}\left[\tilde{I}_w\left(x,y\right)\right] = m^2 I_w\left(x,y\right) \mod N. \tag{30}$$

This means that the image $\mathcal{D}[\tilde{I}_w(x,y)]$ is the same as the image $I_w\left(x,y\right)$ in the finite ring $\mathbb{Z}_N$ if the scale factor m^2 is not considered. By using a method similar to (18), we are able to recover the desired watermarked image from the encrypted image $\tilde{I}_w(x,y)$.

3.2 Watermark Extraction

For our watermark scheme, the watermark extraction can be accomplished in either the plain domain or the encrypted domain. That is, we can extract the watermark either from the image $I_{w,256}\left(x,y\right)$ or from the encrypted image $\tilde{I}_w\left(x,y\right)$.

After the watermark has been extracted, it will be compared to the original watermark with some metrics. We use the bit error rate (BER) to measure the difference between the extracted watermark and the original one. If we denote the extracted watermark by w_j', then the BER of w_j' and w_j is given as

$$\mathrm{BER}\left(\mathbf{w}',\mathbf{w}\right) = \frac{1}{n}\sum_{j=0}^{n-1} w_j' \,\mathrm{XOR}\, w_j \tag{31}$$

where XOR is the *exclusive or* operator. If the BER is less than or equal to some threshold τ, it indicates the presence of watermark, otherwise it indicates the absence of watermark.

We shall show that our watermark scheme possesses the characteristics of blind extraction in two domains, i.e., the decrypted domain and the encrypted domain. More specifically, in the plain domain, the watermark can be extracted from the watermarked image I_w or $I_{w,256}$ without requiring the original image I. While in the encrypted domain, the watermark can be extracted from the encrypted data $\tilde{I}_w$ without requiring either $[\![I]\!]$ or I. We describe the extracting algorithm of our watermark scheme below.

Extraction in the Encrypted Domain. In order to extract the watermark from the encrypted image, we segment $\tilde{I}_w(x, y)$ into non-overlapping blocks of $m \times m$. According to the sequence $\mathbf{a}$, we select n blocks from total $(M/m)^2$ blocks. We then apply WHT-ed of size $m \times m$ to all the selected blocks to output the encrypted coefficients. Let us denote those encrypted coefficient blocks by $\{\tilde{V}_1^{\varepsilon}, \tilde{V}_2^{\varepsilon}, \ldots, \tilde{V}_n^{\varepsilon}\}$. According to the values of e_j and d_j, we choose the cardinal point $\tilde{V}_j^{\varepsilon}(k_0, l_0)$ and the adjacent point $\tilde{V}_j^{\varepsilon}(k_1, l_1)$ in the block $\tilde{V}_j^{\varepsilon}(k, l)$. If we use $\tilde{w}_j'$ to denote the extracted information from j-th selected block, then the watermark extraction in the encrypted domain can be given as

$$\tilde{w}_j' = \tilde{V}_j^{\varepsilon}(k_1, l_1)\,[\tilde{V}_j^{\varepsilon}(k_0, l_0)]^{-1}. \tag{32}$$

By using the homomorphic properties of the cryptosystem and (30), we have

$$\mathcal{D}\left[\tilde{w}_j'\right] = \begin{cases} 0, & \text{if } w_j = 0 \\ m^3\alpha_j. & \text{if } w_j = 1 \end{cases} \tag{33}$$

The scaling factor m^3 can be easily removed after decryption, or directly removed from $m^3\alpha_j$ in the encrypted domain by using the multiplicative inverse method [12]. If the scaling factor is not considered, there is no difference between $\mathcal{D}[\tilde{w}_j']$ and w_j. Assuming that $\frac{\mathcal{D}[\tilde{w}_j']}{m^3\alpha_j}$ is denoted by ϖ_j, then we have

$$\varpi_j = w_j. \tag{34}$$

Therefore we have proved the extracted encrypted watermark $\tilde{w}_j'$ is the encrypted version of the original watermark w_j. We also show an interesting property of the watermark extraction in the encrypted domain by using equation (34). It means that after performing a simple scaling, the extracted watermark is identical to the original watermark without any distortion.

Extraction in the Decrypted Domain. Let us consider the case of extracting the watermark from the decrypted watermarked image. Based on the analysis in Section 3.1, the implementation of watermarking in the encrypted domain will enlarge the plain value of the watermarked image. And small modification of the transform coefficients may result in large variation in the spatial domain. Thus the decrypted values are very likely to be greater than 255 or less than 0. Moreover, all the elements of I_w may not be integers. In order to keep the format compliance, the decrypted values should be mapped to the integers between 0 and 255. Suppose we have already recovered the correct value $m^2 I_w$ from the decryption of $\tilde{I}_w$. Generally, the process of mapping can be given as

$$I_{w,256} = \left\lfloor 255 \cdot \frac{m^2 I_w - \min\{m^2 I_w\}}{\max\{m^2 I_w\} - \min\{m^2 I_w\}} \right\rfloor \tag{35}$$

where $\lfloor\cdot\rfloor$ is the flooring function, while $\min\{\cdot\}$ and $\max\{\cdot\}$ are the minimum and maximum operators, respectively.

Both the quantized watermarked image $I_{w,256}$ and the watermarking key $(\mathbf{a}, \mathbf{e}, \mathbf{d})$ are sent to the extraction device for further processing. Specifically, we segment $I_{w,256}(x, y)$ into non-overlapping blocks of $m \times m$ first, then select n blocks among all the blocks according to the random integer sequence $\mathbf{a}$. WHT of size $m \times m$ is applied to all the selected blocks to output the encrypted coefficients. Let us denote the encrypted coefficients by $\{V_1^\varepsilon, V_2^\varepsilon, \ldots, V_n^\varepsilon\}$. According to the values of e_j and d_j, we choose the cardinal point $V_j^\varepsilon(k_0, l_0)$ and the adjacent point $V_j^\varepsilon(k_1, l_1)$ in the block V_j^ε. If we use w_j' to denote the extracted bit in the V_j^ε, then the process of watermark extraction can be given as

$$w_j' = V_j^\varepsilon\left(k_1, l_1\right) - V_j^\varepsilon\left(k_0, l_0\right). \tag{36}$$

$\{w_j'\}$ will be compared with the embedding message $\{w_j\}$ by using the BER metric to output the result that whether there is a watermark in $I_{w,256}(x, y)$ or not.

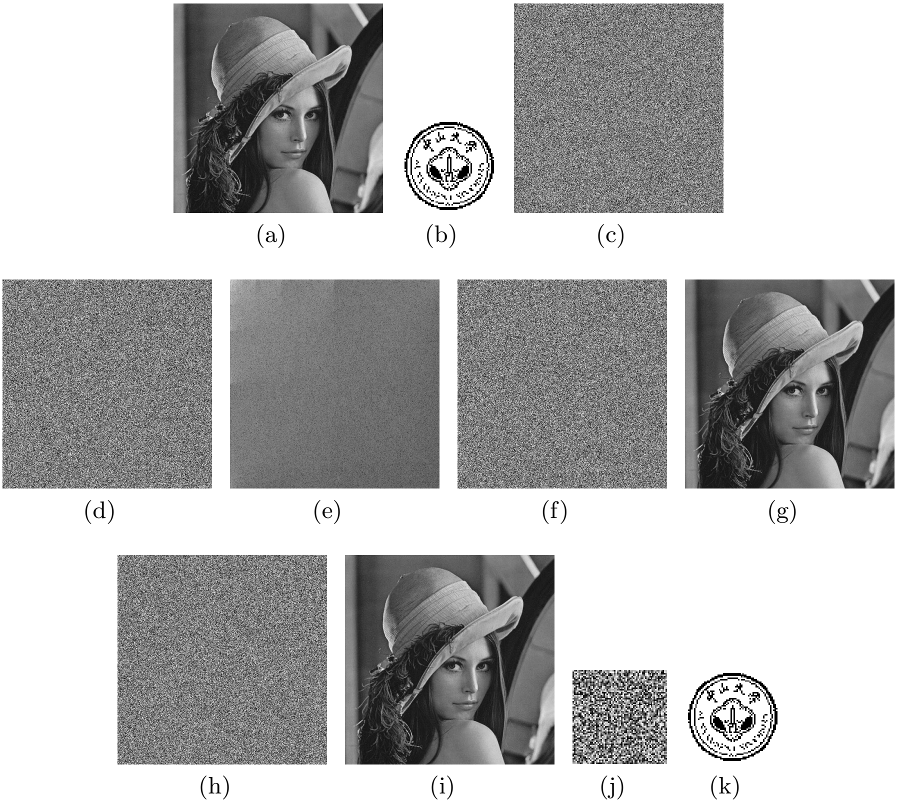
4 Experimental Results

We test the proposed algorithm on a few images. Due to the limitation of paper length, we only show the results on 'Lena' image of 512 × 512 × 8 bits. The original watermark message is chosen as a binary image of 64 × 64 × 1 bits. The original image and the watermark are shown in Fig. 2(d)-2(g). We exploit the 2D WHT in the experiments and choose two large prime numbers p and q for the cryptosystem. The product of p and q is longer than 1024 bits, so the encryption is secure in practice. We show the encrypted image in Fig. 2(c), which is sufficiently scrambled and secure enough to protect the image.

Firstly, we perform WHT-ed of size 512 × 512 to the whole image. The decryption of the result looks the same as the WHT of the plain image. We then perform IWHT-ed to reconstruct the image in the encrypted domain. After decrypting, we obtain an image which looks the same as the original one. The experimental result is shown in Fig. 2(h)-2(i)

Secondly, the encrypted image is segmented into non-overlapping blocks of 8 × 8. We perform WHT-ed of size 8 × 8 on each block. Since there are totally 4096(=64×64) bits in $\mathbf{w}$, we choose all the blocks for the watermark insertion. We adopt $e_j = 64$, $d_j = 1$ and $\alpha_j = 8$ for $j = 1, 2, \ldots, 4096$. According to the value of e_j and d_j, the cardinal point and its adjacent point are selected in $\tilde{V}_j$. By means of (26), we modify coefficients in all the selected blocks for watermark embedding. We then perform IWHT-ed to output the encrypted watermarked image. We show the encryption data and its decryption in Fig. 2(h)-2(i).

Thirdly, by using (32) we extract the encrypted watermark, which is embedded in the encrypted image. The extracted encrypted data and its decryption are shown in Fig. 2(j)-2(k). It can be seen that the decryption looks the same as the original watermark. Actually it is identical to the original watermark in Fig. 2(b) after removing the scaling factor.



(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Fig. 2. Experimental Results: (a) The Original "Lena" image; (b) The watermark; (c) The encrypted "Lena" image; (d) The WHT-ed coefficients; (e) Decryption of WHT-ed coefficients; (f) The encrypted reconstruction; (g) Decryption of reconstruction; (h) The encrypted watermarked image; (i) Decryption of the encrypted watermarked image; (j) The extracted watermark from the encrypted data; (k) Decryption of the extracted watermark

In order to evaluate the visual effect of the watermarked image, we compute the peak signal-to-noise ratio (PSNR) between the original and the watermarked images. The PSNR of the watermarked image in our experiment is 43.31 dB. We also apply our watermark algorithm to 100 grayscale images, each of which is of $512 \times 512 \times 8$ bits. The watermark we use is the one shown in Fig. 2(b). The average PSNRs of the rounding watermarked images $I_{w,256}$ and the no-rounding watermarked images I_w are 43.18 dB and 43.92 dB, respectively. However, all the BERs (error in detection) are 0 under these two situations. This means our algorithm can keep the watermarked image in a good visual quality.

The attackers may perform the attacks on the decrypted image or the encrypted image. Since the attack on the encrypted image may result in a random decrypted image, the attacker is more likely to attack the decrypted image.

Thus we consider the watermark detection performance against Gaussian noise. The Gaussian noise is added in the decrypted watermarked image $I_{w,256}$. For WNR (watermark to noise ratio) > -2 dB, the BER < 0.032 by using ([36]) in watermark retrieval. In practical applications, our watermark algorithm can be extended to the case of spread spectrum scheme, which will greatly improve the robust performance of our watermark.

5 Conclusions

This paper has investigated the implementation of WHT and its applications in image watermarking in a homomorphic encrypted domain. The main contributions are listed as follows:

1) We have described a method to perform WHT and the fast WHT in the encrypted domain, which is based on the homomorphic properties. By using our method, WHT can be implemented in the encrypted domain without any quantization error. We also deduce some elegant equations to show the relationship between WHT(IWHT) in the encrypted domain and WHT(IWHT) in the plain domain.
2) We have proposed an image watermarking scheme based on block WHT-ed. The watermark embedding is carried out in the encrypted domain. However, we can extract the watermark both in the plain domain and the encrypted domain. Both the extractions are blind processing, without involving either the plain original image or the encrypted one.

Our algorithm gives a possible solution to the security problem in the watermarking community. It is possible to use our watermarking scheme to design a secure media distribution system. However, due to the constraints of the homomorphic cryptosystems, the encryption of the original image results in a high store and computation overhead. It is our future work to address the issues regarding the limitation, and to extend our watermarking algorithms to other transforms, e.g., DWT in the encrypted domain.

Acknowledgements. This work was supported by 973 Program (2011CB302200) and NSFC (U1135001).

References

1. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing 53(10), 3976–3987 (2005)
2. Kalker, T.: Considerations on watermarking security. In: 4th IEEE Workshop on Multimedia Signal Processing–MMSP 2001, pp. 201–206. IEEE (2001)
3. Adelsbach, A., Huber, U., Sadeghi, A.-R.: Fingercasting—Joint Fingerprinting and Decryption of Broadcast Messages. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 136–147. Springer, Heidelberg (2006)

4. Celik, M., Lemma, A., Katzenbeisser, S., van der Veen, M.: Lookup-table-based secure client-side embedding for spread-spectrum watermarks. IEEE Transactions on Information Forensics and Security 3(3), 475–487 (2008)
5. Venkata, S., Emmanuel, S.K.M.: Robust watermarking of compressed and encrypted jpeg 2000 images. IEEE Transactions on Multimedia 99, 1 (2011)
6. Memon, N., Wong, P.: A buyer-seller watermarking protocol. IEEE Transactions on Image Processing 10(4), 643–649 (2001)
7. Bianchi, T., Piva, A., Barni, M.: On the implementation of the discrete fourier transform in the encrypted domain. IEEE Transactions on Information Forensics and Security 4(1), 86–97 (2009)
8. Troncoso-Pastoriza, J., Katzenbeisser, S., Celik, M., Lemma, A.: A secure multidimensional point inclusion protocol. In: 9th ACM Workshop on Multimedia and security–MM&Sec 2007, pp. 109–120. ACM (2007)
9. Bianchi, T., Piva, A., Barni, M.: Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Transactions on Information Forensics and Security 5(1), 180–187 (2010)
10. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-Preserving Face Recognition. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 235–253. Springer, Heidelberg (2009)
11. Sadeghi, A.-R., Schneider, T., Wehrenberg, I.: Efficient Privacy-Preserving Face Recognition. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 229–244. Springer, Heidelberg (2010)
12. Zheng, P., Huang, J.: Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. In: 19th ACM International Conference on Multimedia–MM 2011, pp. 413–422. ACM (2011)
13. Barni, M., Bianchi, T., Catalano, D., Di Raimondo, M., Donida Labati, R., Failla, P., Fiore, D., Lazzeretti, R., Piuri, V., Scotti, F., et al.: Privacy-preserving fingercode authentication. In: 12th ACM Workshop on Multimedia and Security–MM&Sec 2010, pp. 231–240. ACM (2010)
14. Barni, M., Failla, P., Lazzeretti, R., Sadeghi, A., Schneider, T.: Privacy-preserving ecg classification with branching programs and neural networks. IEEE Transactions on Information Forensics and Security, 452–468 (2011)
15. Bianchi, T., Piva, A., Barni, M.: Encrypted domain dct based on homomorphic cryptosystems. EURASIP Journal on Information Security 2009 1 (2009)
16. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. Foundations of Secure Computation, 169–178 (1978)
17. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–254. Springer, Heidelberg (1999)
18. Fino, B., Algazi, V.: Unified matrix treatment of the fast walsh-hadamard transform. IEEE Transactions on Computers 100(11), 1142–1146 (1976)

Asymptotic Fingerprinting Capacity in the Combined Digit Model

Dion Boesten and Boris Škorić

Eindhoven University of Technology

Abstract. We study the channel capacity of q-ary fingerprinting in the limit of large attacker coalitions. We extend known results by considering the Combined Digit Model, an attacker model that captures signal processing attacks such as averaging and noise addition. For $q = 2$ we give results for various attack parameter settings. For $q \geq 3$ we present the relevant equations without providing a solution. We show how the channel capacity in the Restricted Digit Model is obtained as a limiting case of the Combined Digit Model.

1 Introduction

1.1 Collusion Resistant Watermarking

Watermarking is a means of tracing the (re-)distribution of content. Before distribution, digital content is modified by applying an imperceptible watermark (WM), embedded using a watermarking algorithm. Once an unauthorized copy of the content is found, the WM helps to trace those users who participated in the creation of the copy. This is known as 'forensic watermarking'. Reliable tracing requires resilience against attacks that aim to remove the WM. Collusion attacks are a particular threat: multiple users cooperate, and differences between their versions of the content tell them where the WM is located. Coding theory has provided a number of collusion-resistant codes. The resulting system has two layers: The coding layer determines which message to embed, and protects against collusion attacks. The underlying watermarking layer hides symbols of the code in segments[1] of the content.

Many collusion resistant codes have been proposed in the literature. Most notable is the Tardos code [15], which achieves the asymptotically optimal proportionality $m \propto c^2$, with m the code length and c the size of the coalition of attackers. Tardos introduced a two-step stochastic procedure for generating binary codewords: (i) For each segment a bias is randomly drawn from some distribution. (ii) For each user independently, a 0 or 1 is randomly drawn for each segment using the bias for that segment. This construction was generalized to larger (q-ary) alphabets in [16].

The interface between the coding and watermarking layer is usually specified in terms of the *Marking Assumption* (MA), which states that the colluders

[1] The 'segments' are defined in a very broad sense.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 255–268, 2013.

are able to perform modifications only in those segments where they received different WM symbols. These segments are called detectable positions. Furthermore, within this class of attacks there is a classification of attacks according to the manipulations that can be performed *in the detectable positions*. In the *Restricted Digit Model* (RDM), the coalition is only allowed to pick one symbol that they received. In the *Unreadable Digit Model*, they are furthermore allowed to create an erasure. In the *Arbitrary Digit Model*, they can pick any symbol from the alphabet, even one that they did not receive (but not an erasure). The *General Digit Model* allows any symbol from the alphabet or an erasure.

For $q = 2$, all these MA attacks are equivalent. For larger alphabets, the general feeling is that realistic attacks are somewhere between the RDM and the Unreadable Digit Model. To come to an even more realistic attack model (also for $q = 2$) which additionally takes into account signal processing (e.g. averaging attacks and noise addition), one has to depart from the MA. Such attack models were proposed in [19] and [17] for general q, and for $q = 2$ in e.g. [7, 8].

1.2 Asymptotic Channel Capacity

In Tardos' scheme [15] and later improvements and generalisations (e.g. [3, 5, 9–11, 13, 14, 16–19]), users are found to be innocent or guilty via an 'accusation sum', a sum of weighted per-segment contributions, computed for each user separately. The analysis of achievable performance was greatly helped by the onset of an information-theoretic treatment of anti-collusion codes. The whole class of bias-based codes can be treated as a maximin game between the watermarker and the colluders [2, 6, 12], independently played for each segment, where the payoff function is the mutual information between the symbols $x_1, \ldots, x_c$ handed to the colluders and the symbol y produced by them. In each segment (i.e. for each bias) the colluders try to minimize the payoff function using an attack strategy that depends on the (frequencies of the) received symbols $x_1, \ldots, x_c$. The watermarker tries to maximize the average payoff over the segments by setting the bias distribution.

The rate of a fingerprinting code is defined as $(\log_q n)/m$, where n is the number of users and m the code length (the number of q-ary symbols). The *fingerprinting capacity* is the maximum achievable rate. For $q = 2$ it was conjectured [6] that the capacity is asymptotically $1/(c^2 2 \ln 2)$. The conjecture was proved in [1, 6]. Amiri and Tardos [1] developed a joint decoder accusation scheme (for the binary case) where candidate coalitions get a score related to the mutual information between their symbols and y. This scheme achieves capacity but is computationally very expensive. Huang and Moulin [6] proved for the large-c limit (in the binary case) that the interleaving attack and Tardos's arcsine distribution are optimal.

It was shown by Boesten and Škorić [4] that the asymptotic channel capacity for q-ary alphabets in the RDM is $(q-1)/(2c^2 \ln q)$. Their proof method revealed neither the optimal attack strategy nor the optimal bias distribution.

1.3 Contributions

In this paper we study the asymptotic channel capacity of q-ary fingerprinting in the Combined Digit Model (CDM) [17], following the approach of [4]. We choose for the CDM because this model is defined for general q and captures a large range of non-MA attacks.

The CDM allows the coalition to add noise and to do averaging attacks. Given the set of symbols used in the averaging, various model parameters describe the probability of these and other symbols being detected by the watermark detector (see Sections 2.3 and 2.4 for details).

We show that the asymptotic channel capacity in the CDM can be found by solving the following problem: Find a mapping γ from the hypersphere in q dimensions to the hypersphere in 2^q dimensions, such that γ minimizes the trace of the induced metric tensor in the latter space (see Section 3). The attack parameters of the CDM give rise to non-trivial constraints on the mapping, which have to be satisfied. One of the main differences between the RDM and CDM lies in the dimension of the target space of γ, which is $q-1$ in the RDM and 2^q-1 in the CDM. We show how the RDM capacity is re-obtained from the CDM setting (Section 4).

For $q \geq 3$ we have not solved the above mentioned minimization problem. For $q=2$ we present numerical results for various attack parameter choices. The numerics involve computations of constrained geodesics, a difficult problem in general. The resulting graphs show a nontrivial dependence of the capacity on the CDM attack parameters.

2 Preliminaries

2.1 Notation

We use capital letters to represent random variables, and lowercase letters to their realizations. Vectors are denoted in boldface and the components of a vector $\boldsymbol{x}$ are written as x_i. Vectors are considered to be column vectors. The expectation over a random variable X is denoted as $\mathbb{E}_X$. The mutual information between X and Y is denoted by $I(X;Y)$, and the mutual information conditioned on a third variable Z by $I(X;Y|Z)$. The base-q logarithm is written as $\log_q$ and the natural logarithm as ln. The standard Euclidean norm of a vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|$. The Kronecker delta of two variables α and β is denoted by $\delta_{\alpha\beta}$. A sum over all possible outcomes of a random variable X is written as $\sum_x$. In order not to clutter up the notation we will often omit the set to which x belongs when it is clear from the context. We use the notation $|\mathcal{Q}|$ for the size of a set $\mathcal{Q}$.

2.2 Fingerprinting with Per-segment Symbol Biases

Tardos [15] introduced the first fingerprinting scheme that achieves optimality in the sense of having the asymptotic behavior $m \propto c^2$. He introduced a two-step stochastic procedure for generating the codeword matrix X. Here we show the generalization to non-binary alphabets [16]. A Tardos code of length m for a number of users n over the alphabet $\mathcal{Q}$ of size q is a set of n length-m sequences of symbols from $\mathcal{Q}$ arranged in an $n \times m$ matrix X. The codeword for a user $i \in \{1, \ldots, n\}$ is the i-th row in X. The symbols in each column $j \in \{1, \ldots, m\}$ are generated in the following way. First an auxiliary bias vector $\boldsymbol{P}^{(j)} \in [0,1]^q$ with $\sum_\alpha P_\alpha^{(j)} = 1$ is generated independently for each column j, from a distribution F which is considered known to the attackers. (The $\boldsymbol{P}^{(j)}$ are sometimes referred to as 'time sharing' variables.) The result $\boldsymbol{p}^{(j)}$ is used to generate each entry X_{ij} of column j independently: $\text{Prob}\,[X_{ij} = \alpha] = p_\alpha^{(j)}$. The code generation has independence of all columns and rows.

2.3 The Collusion Attack in the Combined Digit Model

Let the random variable $\Sigma_\alpha^{(j)} \in \{0, 1, \ldots, c\}$ denote the number of colluders who receive the symbol α in segment j. It holds that $\sum_{\alpha \in \mathcal{Q}} \sigma_\alpha^{(j)} = c$ for all j. (We remind the reader that outcomes of random variables are written in lowercase.) From now on we will drop the segment index j, since all segments are independent. In the *Restricted Digit Model* the colluders produce a symbol $Y \in \mathcal{Q}$ that they have seen at least once. In the *Combined Digit Model* one also allows the attackers to output a mixture of symbols. Let

$$\Omega(\boldsymbol{\Sigma}) \triangleq \{\alpha \in \mathcal{Q} \mid \Sigma_\alpha \geq 1\} \tag{1}$$

be the set of symbols that the pirates have seen in a certain segment. Then the output of the pirates is a non-empty set $\Psi \subseteq \Omega(\boldsymbol{\Sigma})$. On the watermarking level this represents a content-averaging attack where all the symbols in Ψ are used. It has been shown [12] that it is sufficient to consider a probabilistic per-segment (column) attack which does not distinguish between the different colluders. Such an attack then only depends on $\boldsymbol{\Sigma}$, and the strategy can be completely described by a set of probabilities $\theta_{\psi|\boldsymbol{\sigma}} \in [0,1]$, which are defined as

$$\theta_{\psi|\boldsymbol{\sigma}} \triangleq \text{Prob}[\Psi = \psi \mid \boldsymbol{\Sigma} = \boldsymbol{\sigma}]. \tag{2}$$

For all $\boldsymbol{\sigma}$ conservation of probability gives $\sum_\psi \theta_{\psi|\boldsymbol{\sigma}} = 1$.

2.4 Detection Process in the Combined Digit Model

The Combined Digit Model also introduces a stochastic detection process. Let $|\Psi|$ be the cardinality of the output set Ψ. Then each symbol in Ψ is detected with probability $t_{|\Psi|}$. Each symbol not in the set Ψ is detected with error probability r. The set $W \subseteq \mathcal{Q}$ indicates which symbols are detected. Note that Ψ is forced to be non-empty but $W = \emptyset$ can occur.

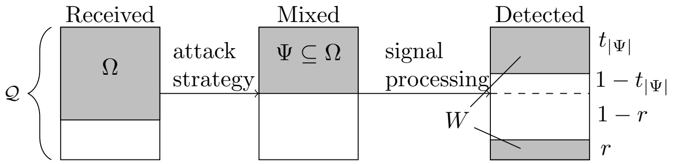


Fig. 1. Overview of the detection process in the Combined Digit Model. The detection probabilities are shown on the right.

The numbers t_i for $i = 1, 2, \ldots, q$ are decreasing, since mixing more symbols makes it more difficult to detect the individual symbols. The overall probability of detecting a set w, given ψ, depends on r, $t_{|\psi|}$ and the sizes of the fours sets shown under 'Detected' in Fig. 1. From top to bottom, these are (i) the number of used symbols that get detected, $|\psi \cap w|$; (ii) # used symbols that *do not* get detected, $|\psi \setminus w|$; (iii) # not used symbols that are not detected, $q - |\psi \cup w|$; and (iv) # not used symbols that *are* detected due to noise, $|w \setminus \psi|$. For a given ψ, the probability that the detector outputs a detected set w is

$$\begin{aligned} M_{w|\psi} &\triangleq \text{Prob}\left[W = w \mid \Psi = \psi\right] \\ &= t_{|\psi|}^{|\psi \cap w|} \left(1 - t_{|\psi|}\right)^{|\psi \setminus w|} (1-r)^{q - |\psi \cup w|}\, r^{|w \setminus \psi|}. \end{aligned} \tag{3}$$

These probabilities form a $2^q \times (2^q - 1)$ matrix M. In this way we can define

$$\tau_{w|\boldsymbol{\sigma}} \triangleq \text{Prob}\left[W = w \mid \boldsymbol{\Sigma} = \boldsymbol{\sigma}\right] = \sum_{\psi} M_{w|\psi} \theta_{\psi|\boldsymbol{\sigma}} = (M\theta)_{w|\boldsymbol{\sigma}}\,, \tag{4}$$

or, in matrix notation, $\tau = M\theta$. (The matrix notation for the relation (4) is novel.)

2.5 Collusion Channel and Fingerprinting Capacity

Similarly to the RDM [4] the attack can be interpreted as a noisy channel with input $\boldsymbol{\Sigma}$ and output W. A capacity for this channel can then be defined, which gives an upper bound on the achievable code rate of a reliable fingerprinting scheme. The first step of the code generation, drawing the biases $\boldsymbol{p}$, is not considered to be a part of the channel. The fingerprinting capacity C_q^{CDM} for a coalition of size c and alphabet size q in the CDM is equal to the optimal value of the following two-player game:

$$C_q^{\text{CDM}} = \max_F \min_\theta \frac{1}{c} I(W; \boldsymbol{\Sigma} \mid \boldsymbol{P}) = \max_F \min_\theta \frac{1}{c} \int F(\boldsymbol{p}) I(W; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) \mathrm{d}^q \boldsymbol{p}. \tag{5}$$

Here the information is measured in q-ary symbols. Our aim is to compute the fingerprinting capacity C_q^{CDM} in the limit $c \to \infty$.

The payoff function $I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P})$ is linear in F and convex in τ. Because $\tau = M\theta$ is linear in θ the game is also convex in θ and we can apply Sion's Theorem:

$$\begin{aligned}\max_F \min_\theta I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P}) &= \min_\theta \max_F \; I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P}) \\ &= \min_\theta \max_p \; I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}). \end{aligned} \tag{6}$$

In the second step we performed the maximization over F by choosing the optimum $F^*(\boldsymbol{p}) = \delta(\boldsymbol{p} - \boldsymbol{p}_{\max})$, where $\boldsymbol{p}_{\max}$ is one of the locations where $I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$ has its maximum.

3 Asymptotic Analysis for General Alphabet Size

We are interested in how the payoff function $I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$ of the alternative game (6) behaves as c goes to infinity. Following the same approach as in [4] our starting point is the observation that the random variable $\boldsymbol{\Sigma}/c$ tends to a continuum in $[0,1]^q$ with mean $\boldsymbol{p}$. Hence we introduce a continuous strategy $\boldsymbol{h}\left(\frac{\boldsymbol{\sigma}}{c}\right)$:

$$h_\psi\left(\frac{\boldsymbol{\sigma}}{c}\right) = \lim_{c\to\infty} \theta_{\psi|\boldsymbol{\sigma}}. \tag{7}$$

We also define

$$g_w\left(\frac{\boldsymbol{\sigma}}{c}\right) = \lim_{c\to\infty} \tau_{w|\boldsymbol{\sigma}} = \sum_\psi M_{w|\psi} h_\psi\left(\frac{\boldsymbol{\sigma}}{c}\right), \tag{8}$$

which in matrix notation can be written as $\boldsymbol{g} = M\boldsymbol{h}$. The next step is to do a second order Taylor expansion of $g_w\left(\frac{\boldsymbol{\sigma}}{c}\right)$ around the point $\frac{\boldsymbol{\sigma}}{c} = \boldsymbol{p}$. This allows us to expand I in powers of $1/c$, giving (see [4])

$$I(W;\boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) = \frac{T(\boldsymbol{p})}{2c\ln q} + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{9}$$

$$T(\boldsymbol{p}) \triangleq \sum_w \frac{1}{g_w(\boldsymbol{p})} \sum_{\alpha\beta} K_{\alpha\beta} \frac{\partial g_w(\boldsymbol{p})}{\partial p_\alpha} \frac{\partial g_w(\boldsymbol{p})}{\partial p_\beta}, \tag{10}$$

where $K_{\alpha\beta} = \delta_{\alpha\beta} p_\alpha - p_\alpha p_\beta$ is the scaled covariance matrix of $\boldsymbol{\Sigma}$. The asymptotic capacity $C^{\mathrm{CDM}}_{q,\infty}$ in the limit of $c \to \infty$ is then defined as the solution of the continuous version of the game (6):

$$C^{\mathrm{CDM}}_{q,\infty} \triangleq \frac{1}{2c^2 \ln q} \min_{\boldsymbol{h}} \max_{\boldsymbol{p}} T(\boldsymbol{p}). \tag{11}$$

At this point we introduce the variable transformations $u_\alpha \triangleq \sqrt{p}_\alpha, \gamma_w \triangleq \sqrt{g_w}$ and also the $2^q \times q$ Jacobian matrix $J_{w\alpha}(\boldsymbol{u}) \triangleq \frac{\partial \gamma_w(\boldsymbol{u})}{\partial u_\alpha}$. This transformation means we switch to hyperspheres ($\|\boldsymbol{u}\| = 1, \|\boldsymbol{\gamma}\| = 1$) instead of the hyperplanes ($\sum_\alpha p_\alpha = 1, \sum_w g_w = 1$) that we had before. The function $\boldsymbol{\gamma}(\boldsymbol{u})$ was originally defined only on the domain $\|\boldsymbol{u}\| = 1$, but the Taylor expansion forces us to define $\boldsymbol{\gamma}$ on a larger domain, i.e. slightly away from $\|\boldsymbol{u}\| = 1$. There are many consistent ways to do this domain extension. We choose to define $\boldsymbol{\gamma}$ such that it is independent of the radial coordinate $\|\boldsymbol{u}\|$. This choice yields $J\boldsymbol{u} = 0$, which allows us to simplify $T(\boldsymbol{u})$ to

$$T(\boldsymbol{u}) = \sum_{w,\alpha} \left(\frac{\partial \gamma_w}{\partial u_\alpha} \right)^2 = \mathrm{Tr}(J^T J) = \sum_{i=1}^{q-1} \lambda_i(\boldsymbol{u}), \tag{12}$$

where $\lambda_i(\boldsymbol{u})$ are the eigenvalues of $J^T J$. Because of our choice $J\boldsymbol{u} = 0$ we already know that one of the eigenvalues is 0 with eigenvector $\boldsymbol{u}$. Hence there are only $q - 1$ eigenvalues left. Note that (12) can be interpreted as the trace of a metric: if we define a metric $B_{\alpha\beta} = (\partial\boldsymbol{\gamma}/\partial u_\alpha) \cdot (\partial\boldsymbol{\gamma}/\partial u_\beta)$ in the usual way, then $T(\boldsymbol{u}) = \mathrm{Tr}\, B$.

We now wish to find

$$\min_{\gamma} \max_{\boldsymbol{u}} T(\boldsymbol{u}) \tag{13}$$

under the constraint

$$\gamma_w = \sqrt{g_w} = \sqrt{(M\boldsymbol{h})_w} \tag{14}$$

with M known and $\boldsymbol{h}$ satisfying

$$h_\psi \geq 0 \quad \forall\psi, \qquad\qquad \textstyle\sum_\psi h_\psi = 1. \tag{15}$$

The constraint (14) makes solving the min-max game (13) more difficult and we are unable to use the same machinery as for the RDM. The main problem is that it is no longer easy to characterize the allowed (sub)space that $\boldsymbol{\gamma}$ lives in.

For the binary alphabet we are however able to go further and compute the asymptotic capacity (see Section 5).

4 Limiting Case: Restricted Digit Model

We show how the known result for the Restricted Digit Model (RDM) follows as a limiting case of the CDM.

We set $r = 0$ and $t_i = 1$ for all $i \in \{1, \cdots, q\}$. This means that there is no noise, and any symbol that the attackers use will be detected with 100% certainty. Hence $W = \Psi$. In this situation there is no gain for the attackers to use fusion, as all the fused symbols are detected and provide the content owner with more information. Their best option is to use a single symbol; hence we are back at the RDM.

Mathematically it is slightly more involved to see how the reduction to the RDM channel capacity is obtained. The matrix M becomes $\begin{pmatrix} \mathbf{0} \\ I_{2^q-1} \end{pmatrix}$ where

I_{2^q-1} is the identity matrix of size 2^q-1. Since M has become trivial, (14) does not really represent a constraint on $\gamma_w(\boldsymbol{u})$ any more. The only difference with [4] is the dimension of the vector: $\boldsymbol{\gamma}$ has 2^q-1 components ($w=\emptyset$ is excluded), whereas in the RDM there were only q components. Consequently, the Jacobian J also has a larger dimension. However, the product J^TJ is still a $q\times q$ matrix, and the derivation in [4] can be applied in unchanged form to yield two results:

1. The solution of the min-max game satisfies $\max_{\boldsymbol{u}} T(\boldsymbol{u}) = \mathrm{Av}_{\boldsymbol{u}}[T(\boldsymbol{u})]$, i.e. the maximum is equal to the spatial average, and $T(\boldsymbol{u})$ is in fact a constant on the hypersphere $\|\boldsymbol{u}\|=1$, with
$$T(\boldsymbol{u}) \geq (q-1)\left(\frac{\int \mathrm{d}S_{\boldsymbol{\gamma}}}{\int \mathrm{d}S_{\boldsymbol{u}}}\right)^{2/(q-1)}. \qquad (16)$$
Here $\int \mathrm{d}S_{\boldsymbol{u}}$ is the $(q-1)$-dimensional 'volume' integral on the surface of the $\boldsymbol{u}$-hypersphere. The $\int \mathrm{d}S_{\boldsymbol{\gamma}}$ is the corresponding $(q-1)$-dimensional integral in the larger (2^q-2)-dimensional $\boldsymbol{\gamma}$-hypersphere, with $\boldsymbol{\gamma}=\boldsymbol{\gamma}(\boldsymbol{u})$. In [4] the γ-sphere had dimension $q-1$, and it was used that $\int \mathrm{d}S_{\boldsymbol{\gamma}} \geq \int \mathrm{d}S_{\boldsymbol{u}}$.
2. The interleaving attack yields $T(\boldsymbol{u}) = q-1$ on the hypersphere $\|u\|=1$.

We argue (without proof) that $\int \mathrm{d}S_{\boldsymbol{\gamma}} \geq \int \mathrm{d}S_{\boldsymbol{u}}$ still holds. This is because of the Marking Assumption, which fixes the values on the axes in $\boldsymbol{\gamma}$-space. Let e_α be the unit vector in the α-direction. Then $\boldsymbol{u}=e_\alpha \implies \boldsymbol{\gamma}=e_\alpha$. These 'corner' points live in a q-dimensional subspace. It is possible to step out of that subspace for general $\boldsymbol{u}$, but doing so increases the volume $\int \mathrm{d}S_{\boldsymbol{\gamma}}$.

Thus, result #1 gives the lower bound $\max_{\boldsymbol{u}} T(\boldsymbol{u}) \geq q-1$, while result #2 shows that there exists a strategy achieving the lower bound. The RDM channel capacity $C^{\mathrm{RDM}}_{q,\infty} = (q-1)/(2c^2\ln q)$ follows.

Remark: If M is perturbed away from the identity matrix, then the extreme points $\boldsymbol{u}=e_\alpha$ are no longer mapped to mutually orthogonal vectors $\boldsymbol{\gamma}$, but to vectors with smaller mutual angles; the reduction of the angles causes a reduction of $\int \mathrm{d}S_{\boldsymbol{\gamma}}$ and hence the channel capacity. The details are cumbersome and the general case $q\geq 3$ is left for future work.

5 Fingerprinting Capacity in the CDM for $q=2$

5.1 Solving the Max-Min Game

For the binary alphabet $q=2$ the expression (12) simplifies to
$$T(\boldsymbol{u}) = \mathrm{Tr}(J^TJ) = \lambda(\boldsymbol{u}) \qquad (17)$$

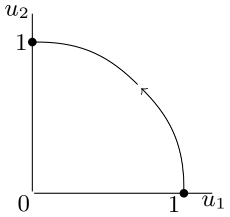


Fig. 2. The path for $\boldsymbol{u}$ is the positive quarter circle

since there is only one nonzero eigenvalue. Furthermore we have the relation $\mathrm{d}\boldsymbol{\gamma} = J\mathrm{d}\boldsymbol{u}$ and $\|\mathrm{d}\boldsymbol{\gamma}\| = \sqrt{\lambda}\|\mathrm{d}\boldsymbol{u}\|$ for an infinitesimal change $\mathrm{d}\boldsymbol{u}$. We proceed by rewriting

$$\begin{aligned}\max_{\boldsymbol{u}} T(\boldsymbol{u}) &= \max_{\boldsymbol{u}} \lambda(\boldsymbol{u}) = \left(\max_{\boldsymbol{u}} \sqrt{\lambda(\boldsymbol{u})}\right)^2 \\ &\geq \left(\frac{\int \sqrt{\lambda(\boldsymbol{u})}\|\mathrm{d}\boldsymbol{u}\|}{\int \|\mathrm{d}\boldsymbol{u}\|}\right)^2 = \left(\frac{\int \|\mathrm{d}\boldsymbol{\gamma}\|}{\int \|\mathrm{d}\boldsymbol{u}\|}\right)^2 \equiv \left(\frac{L_{\boldsymbol{\gamma}}}{L_{\boldsymbol{u}}}\right)^2, \end{aligned} \tag{18}$$

where the inequality results from replacing the maximum by a spatial average. The path in the integrals (see Fig. 2) is the quarter-circle $u_1^2 + u_2^2 = 1$ from $\boldsymbol{u} = (1,0)$ to $\boldsymbol{u} = (0,1)$ and hence $L_{\boldsymbol{u}} = \pi/2$.

The next step is to realize that for any curve $\gamma(\boldsymbol{u})$ we have the freedom to parameterize that curve differently in such a way that $\lambda(\boldsymbol{u})$ is constant over that curve, i.e. we are traveling at constant speed. The inequality in (18) can then be changed into an equality and we have

$$\min_{\gamma} \max_{\boldsymbol{u}} T(\boldsymbol{u}) = \frac{4}{\pi^2}(\min_{\gamma} L_{\boldsymbol{\gamma}})^2. \tag{19}$$

Hence we have reduced the problem to finding a curve $\boldsymbol{\gamma}(\boldsymbol{u})$ of minimal length with the constraint $\gamma_w(\boldsymbol{u}) = \sqrt{(M\boldsymbol{h})_w(\boldsymbol{u})}$ where $M(t_1, t_2, r)$ is given by

$$M = \begin{array}{c||c|c|c|} w\backslash\psi & \{0\} & \{1\} & \{0,1\} \\ \hline \emptyset & (1-t_1)(1-r) & (1-t_1)(1-r) & (1-t_2)^2 \\ \{0\} & t_1(1-r) & (1-t_1)r & t_2(1-t_2) \\ \{1\} & (1-t_1)r & t_1(1-r) & t_2(1-t_2) \\ \{0,1\} & t_1 r & t_1 r & t_2^2 \end{array}. \tag{20}$$

5.2 Geodesics

In general, the method to find length-minimizing curves is to solve the Euler-Lagrange differential equations for the geodesics of the appropriate metric. In our case the additional constraint $\gamma_w(\boldsymbol{u}) = \sqrt{(M\boldsymbol{h})_w(\boldsymbol{u})}$ makes things more difficult. The constraint can be interpreted in the following way. If we write $M = [m_1, m_2, m_3]$ then because of constraint (15) then we have that $\boldsymbol{g} = M\boldsymbol{h}$ is a convex combination of the three column vectors m_1, m_2, m_3. Hence the allowed space of $\boldsymbol{g}$ is anywhere inside the triangle shown in Fig. 3.

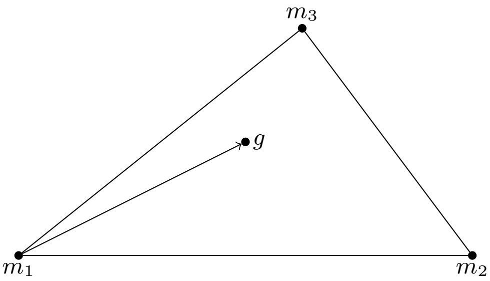


Fig. 3. The vector $\boldsymbol{g}$ is not allowed to lie outside the triangle

We switch from variables (u_1, u_2) to s_1, s_2 with $0 \leq s_1 \leq 1$ and $0 \leq s_2 \leq 1-s_1$.

$$\boldsymbol{g}(s_1, s_2) \triangleq m_1 + s_1(m_2 - m_1) + s_2(m_3 - m_1). \tag{21}$$

The marking assumption gives us that $\boldsymbol{u} = (1,0) \Rightarrow \boldsymbol{h} = (1,0,0)$ and $\boldsymbol{u} = (0,1) \Rightarrow \boldsymbol{h} = (0,1,0)$. In terms of $\boldsymbol{g}(s_1, s_2)$ this means $\boldsymbol{g}(0,0) = m_1$ and $\boldsymbol{g}(1,0) = m_2$. We are looking for the shortest path from the lower left corner (m_1) of the triangle to the lower right corner (m_2).

The infinitesimal change in $\mathrm{d}\gamma_w$ in terms of $(\mathrm{d}s_1, \mathrm{d}s_2)$ is given by

$$\mathrm{d}\gamma_w = \frac{\mathrm{d}g_w}{2\sqrt{g_w}} = \frac{(m_{2,w} - m_{1,w})\mathrm{d}s_1 + (m_{3,w} - m_{1,w})\mathrm{d}s_2}{2\sqrt{g_w}}. \tag{22}$$

This allows us to define the appropriate metric $G(s_1, s_2)$,

$$\|\mathrm{d}\boldsymbol{\gamma}\|^2 = G_{11}(\mathrm{d}s_1)^2 + G_{22}(\mathrm{d}s_2)^2 + 2G_{12}\mathrm{d}s_1\mathrm{d}s_2. \tag{23}$$

We use this metric to compute the geodesics (locally distance minimizing curves). See Appendix A for the details.

5.3 Finding the Shortest Path

We want to find the shortest path between m_1 and m_2 that is fully inside the triangle. If a direct geodesic between these two points exists we know that it is the optimal path; but this does not always happen. We encounter three possible cases, given in Fig. 4. In case A the direct geodesic is the shortest possible path. For cases B and C the optimal paths are shown in Fig. 5.

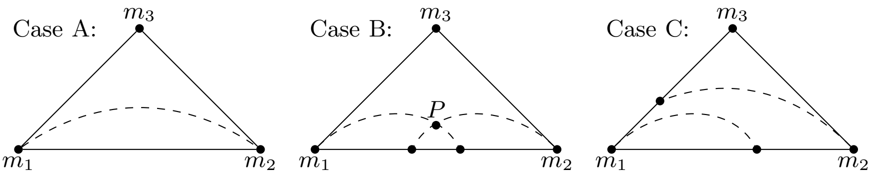


Fig. 4. In case A there exists a direct geodesic from m_1 to m_2. In case B the maximum-slope geodesics starting from m_1 and m_2 intersect in P. In case C they do not intersect.

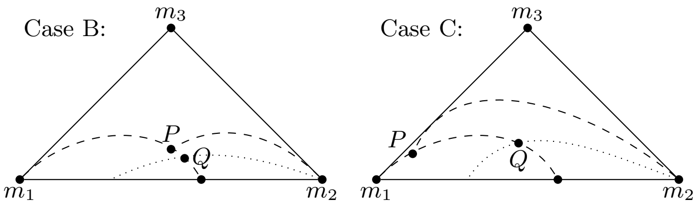


Fig. 5. The optimal path in both cases is $m_1 - P - m_2$ over the dashed lines (geodesics). In case C the geodesic from m_2 is the one which is tangent to the left side of the triangle.

Any geodesic starting from m_2 with a smaller initial slope eventually has to cross the maximum-slope geodesic from m_1 in a point Q. From Q the optimal path to m_1 is to follow the geodesic; but when you pass P you could have done better by simply going directly from m_2 to P on the geodesic.

Once we have the optimal path we can determine its length L_{opt} (see Appendix) and use it to compute the capacity,

$$C_{2,\infty}^{\text{CDM}} = \frac{1}{2c^2 \ln 2} \cdot \frac{4}{\pi^2} L_{\text{opt}}^2. \tag{24}$$

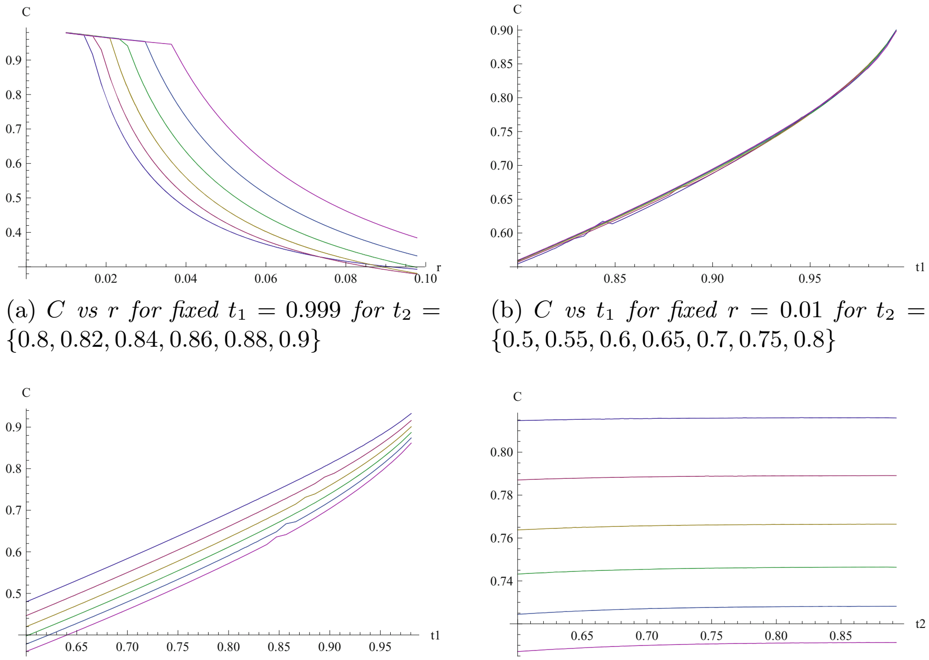
5.4 Results

In Fig. 6 we show plots of the ratio $C = C_{2,\infty}^{\text{CDM}} / C_{2,\infty}^{\text{RDM}}$ between the asymptotic capacities for the CDM and the RDM as a function of the parameters t_1, t_2, r. (For the binary alphabet $\mathcal{Q} = \{0, 1\}$, we have that r is the noise strength, t_1 is the probability of detecting a symbol α if the coalition used $\Psi = \{\alpha\}$, and t_2 is the probability of detecting α if the coalition used $\Psi = \{0, 1\}$). Several aspects of the graphs are easily understood and yield no surprises:

- Obviously, the capacity is an increasing function of t_1 and t_2, and a decreasing function of r. When the attack options become more powerful, the capacity goes down.
- For r close to zero and t_1 close to 1, the capacity has very weak dependence on t_2. This can be understood as follows. A small value of t_2 effectively means that the attackers create an erasure, which brings us to the Unreadable Digit Model. For large t_2 is it not advantageous for them to take $\Psi = \{0, 1\}$, since

> the detector will find both symbols, giving the tracer more information than taking $|\Psi| = 1$. The attackers will output a single symbol, which brings us back to the RDM. For $(r, t_1) \approx (0, 1)$ we are close to the Marking Assumption. When the MA holds, all the attack models for $q = 2$ are equivalent.

Other behaviour is more surprising. In Fig. 6a we see a transition from linear behavior as a function of r (with almost total insensitivity to t_2) to nonlinear behavior (with dependence on t_2). The transition point depends on t_2.



(a) *C vs r for fixed $t_1 = 0.999$ for $t_2 = \{0.8, 0.82, 0.84, 0.86, 0.88, 0.9\}$*

(b) *C vs t_1 for fixed $r = 0.01$ for $t_2 = \{0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8\}$*

(c) *C vs t_1 for fixed $t_2 = 0.5$ for $r = \{0.01, 0.02, 0.03, 0.04, 0.05\}$*

(d) *C vs t_2 for fixed $t_1 = 0.9$ for $r = \{0.01, 0.02, 0.03, 0.04, 0.05\}$*

Fig. 6. Numerics for $q = 2$. The ratio $C = C^{\text{CDM}}_{2,\infty} / C^{\text{RDM}}_{2,\infty}$ is plotted on the vertical axis.

6 Discussion

We have investigated the asymptotic channel capacity $C^{\text{CDM}}_{q,\infty}$ in the Combined Digit Model. For general alphabet size q it turns out to be very difficult to compute this quantity. We have shown how the asymptotic capacity for the RDM [4] follows as a limiting case of the CDM. In the general case, $C^{\text{CDM}}_{q,\infty}$ can be expressed as the solution of a min-max game (13) where the payoff function is the trace of the metric induced by the mapping γ from $\sqrt{p_\alpha}$-space to $\sqrt{g_w}$-space. The CDM parameters r and $t_1, \cdots, t_q$ give rise to a constraint $\boldsymbol{g} = M\boldsymbol{h}$ on $\boldsymbol{g}$ which prevents the application of the solution method of [4]. For the binary alphabet we have shown that the problem reduces to finding a constrained geodesic between two points. Our numerical results do not contain significant surprises. They confirm the intuition in the vicinity of the Marking Assumption, $(r, t_1) \approx (0, 1)$. In this regime $C^{\mathrm{CDM}}_{2,\infty}$ is practically independent of t_2. The transitions in Fig. 6a are not intuitively clear. The study of these details and of larger alphabets $q \geq 3$ is left for future work.

Acknowledgements. We kindly thank Antonino Simone, Jan-Jaap Oosterwijk, Benne de Weger, Thijs Laarhoven and Jeroen Doumen for useful discussions. This research was done as part of the CREST project, funded by STW.

References

1. Amiri, E., Tardos, G.: High rate fingerprinting codes and the fingerprinting capacity. In: SODA 2009, pp. 336–345 (2009)
2. Anthapadmanabhan, N.P., Barg, A., Dumer, I.: Fingerprinting capacity under the marking assumption. IEEE Transaction on Information Theory – Special Issue on Information-theoretic Security 54(6), 2678–2689
3. Blayer, O., Tassa, T.: Improved versions of Tardos' fingerprinting scheme. Designs, Codes and Cryptography 48(1), 79–103 (2008)
4. Boesten, D., Škorić, B.: Asymptotic Fingerprinting Capacity for Non-binary Alphabets. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 1–13. Springer, Heidelberg (2011)
5. Charpentier, A., Xie, F., Fontaine, C., Furon, T.: Expectation maximization decoding of Tardos probabilistic fingerprinting code. In: Media Forensics and Security. SPIE Proceedings, vol. 7254, p. 72540 (2009)
6. Huang, Y.W., Moulin, P.: Saddle-point solution of the fingerprinting capacity game under the marking assumption. In: Proc. IEEE International Symposium on Information Theory, ISIT (2009)
7. Kuribayashi, M.: Tardos"s Fingerprinting Code over AWGN Channel. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 103–117. Springer, Heidelberg (2010)
8. Kuribayashi, M.: A New Soft Decision Tracing Algorithm for Binary Fingerprinting Codes. In: Iwata, T., Nishigaki, M. (eds.) IWSEC 2011. LNCS, vol. 7038, pp. 1–15. Springer, Heidelberg (2011)
9. Kuribayashi, M., Akashi, N., Morii, M.: On the systematic generation of Tardos's fingerprinting codes. In: MMSP 2008, pp. 748–753 (2008)
10. Laarhoven, T., de Weger, B.M.M.: Optimal symmetric Tardos traitor tracing schemes (2011), http://arxiv.org/abs/1107.3441
11. Meerwald, P., Furon, T.: Towards Joint Tardos Decoding: The 'Don Quixote' Algorithm. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) IH 2011. LNCS, vol. 6958, pp. 28–42. Springer, Heidelberg (2011)
12. Moulin, P.: Universal fingerprinting: Capacity and random-coding exponents. In Preprint arXiv:0801.3837v2 (2008), http://arxiv.org/abs/0801.3837
13. Nuida, K., Fujitsu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., Imai, H.: An improvement of discrete Tardos fingerprinting codes. Des. Codes Cryptography 52(3), 339–362 (2009)
14. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimal probabilistic fingerprinting codes using optimal finite random variables related to numerical quadrature. CoRR, abs/cs/0610036 (2006)

15. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC 2003, pp. 116–125 (2003)
16. Škorić, B., Katzenbeisser, S., Celik, M.U.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
17. Škorić, B., Katzenbeisser, S., Schaathun, H.G., Celik, M.U.: Tardos Fingerprinting Codes in the Combined Digit Model. IEEE Transactions on Information Forensics and Security 6(3), 906–919 (2011)
18. Škorić, B., Vladimirova, T.U., Celik, M.U., Talstra, J.C.: Tardos fingerprinting is better than we thought. IEEE Trans. on Inf. Theory 54(8), 3663–3676 (2008)
19. Xie, F., Furon, T., Fontaine, C.: On-off keying modulation and Tardos fingerprinting. In: MM&Sec 2008, pp. 101–106 (2008)

A Solving the Geodesic Equations

The metric $G(s_1, s_2)$ is a 2×2 symmetric matrix whose components can be derived from equations (22) and (23):

$$G_{ij}(s_1, s_2) = \frac{1}{4} \sum_w \frac{(m_{i+1,w} - m_{1,w})\,(m_{j+1,w} - m_{1,w})}{m_{1,w} + s_1(m_{2,w} - m_{1,w}) + s_2(m_{3,w} - m_{1,w})}, \tag{25}$$

with $i, j \in \{1, 2\}$. The Christoffel symbols Γ^i_{jk} for this metric are defined as

$$\Gamma^i_{jk} \triangleq \frac{1}{2} \sum_{m=1}^{2} G^{-1}_{im} \left(\frac{\partial G_{jm}}{\partial s_k} + \frac{\partial G_{km}}{\partial s_j} - \frac{\partial G_{jk}}{\partial s_m} \right) \tag{26}$$

where G^{-1} is the matrix inverse. We are looking for a shortest curve $(s_1(x), s_2(x))$ with $x \in \mathbb{R}$ from the point $(s_1, s_2) = (0, 0)$ to $(1, 0)$. The geodesic equations read

$$\begin{aligned}
s_1'(x) &= k_1(x) \\
s_2'(x) &= k_2(x) \\
k_1'(x) &= -\Gamma^1_{11} k_1^2(x) - 2\Gamma^1_{12} k_1(x) k_2(x) - \Gamma^1_{22} k_2^2(x) \\
k_2'(x) &= -\Gamma^2_{11} k_1^2(x) - 2\Gamma^2_{12} k_1(x) k_2(x) - \Gamma^2_{22} k_2^2(x).
\end{aligned} \tag{27}$$

Once we specify the initial conditions for $s_1(0), s_2(0), k_1(0), k_2(0)$ we can solve (27) numerically to obtain the geodesic curves starting at $(s_1(0), s_2(0))$ with initial 'velocity' vector $(k_1(0), k_2(0))$.

Bias Equalizer for Binary Probabilistic Fingerprinting Codes

Minoru Kuribayashi

Graduate School of Engineering, Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo, 657-8501 Japan
kminoru@kobe-u.ac.jp

Abstract. In binary probabilistic fingerprinting codes, the number of symbols "0" and "1" is generally balanced because of the design of the codeword. After a collusion attack, the balance of symbols is not always assured in a pirated codeword. It is reported in [8] that if the number of symbols in a pirated codeword is unbalanced, then a tracing algorithm can be improved by equalizing the unbalanced symbols. In this study, such a bias equalizer is customized for probabilistic fingerprinting codes utilizing the encoding parameters. Although the proposed bias equalizer is highly dependent on collusion strategies, it can improve the performance of traceability for some typical strategies to which the conventional bias equalizer can not be applied.

1 Introduction

Digital fingerprinting [17] is used to trace illegal users, where a unique ID known as a digital fingerprint is embedded into a content before distribution. When a suspicious copy is found, the owner can identify illegal users by extracting the fingerprint. Since each user purchases a content involving his own fingerprint, the fingerprinted copy slightly differs with each other. Therefore, a coalition of users will combine their differently marked copies of the same content for the purpose of removing or changing the original fingerprint. To counter this threat, coding theory has produced a number of collusion resistant codes under the well-known principle referred to as the marking assumption.

Tardos [13] has proposed a probabilistic fingerprinting code which has a length of theoretically minimal order with respect to the number of colluders. Theoretical analyses about the Tardos code yield more efficient probabilistic fingerprinting codes improving the traceability, code length, and so on. Among the variants of the Tardos code, Nuida et al. [11] studied the parameters to generate the codewords of the Tardos code which follow a continuous distribution, and presented a discrete version in attempts to reduce the code length and the required memory amount without degrading the traceability.

Under the marking assumption, an optimal detector is presented in [9] from the information theoretic analysis of collusion strategy. If we can identify the collusion strategy from a pirated codeword, the optimal detector is the best

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 269–283, 2013.

one for binary fingerprinting codes. However, in a generic situation, the marking assumption is not always valid, and the assumption should be modified to a relaxed version. Recently, the relaxation of the marking assumption has been employed in the analysis of the Tardos code and its variants [5],[6],[7],[10],[18],[15]. In [7], a pirated copy is produced by collusion attack and it is further distorted by additive white Gaussian noise (AWGN), which is called "relaxed marking assumption" in this paper. Because no robust watermarking scheme avoids an injection of noise into a pirated copy, it is reasonable to assume that the noisy channel is modeled by AWGN.

An important factor in the generation of binary fingerprinting codewords is the symmetry of the underlying bias distribution. It means that the number of symbols "1" in a codeword is expected to be equal to that of symbols "0", though the probability that a symbol at a certain element is biased in a codeword. Without the knowledge of the symbol values embedded into a digital content, the number of symbols in a pirated codeword also follows this rule. However, if colluders happen to get each symbol value of their codewords, they can employ aggressive collusion strategies to break down the rule in a pirated codeword. On the other hand, the bias of symbols in a pirated codeword is exploited to calculate weights for correlation scores in a tracing algorithm in [8]. It is noticed that the number of symbols "1" and "0" is easily derived by observing the symbols in a pirated codeword. Its experimental results revealed that the performance was improved only when the number of symbols "1" and "0" became imbalanced.

In this paper, we study the bias equalizer proposed in [8], and simplify the method under a noiseless case. Based on the simplified method, we investigate the mechanism of the equalization steps, and we extend our study to the noisy case. Then, we propose the bias equalizer based on the encoding parameters of Nuida code to improve the traceability under a constant false-positive probability. Since the bias distribution of the Nuida code is discrete, the encoding parameters related to the bias distribution are finite. Considering the finite candidates for the discrete version of the bias distribution, symbols in a pirated codeword is classified into groups, and their corresponding weighting parameters are calculated from the imbalance of the symbols "1" and "0" in their groups. The imbalance in each group must be occurred not only for specific collusion strategies, but also for a generic case. It comes from the biased probability assigned for the group. Hence, the proposed bias equalizer eliminates the limitation of the applicability in [8]. The effect of the proposed detector is evaluated by experiments under some typical collusion strategies. The experimental results reveal that the proposed equalizer effectively enhances the performance of tracing algorithm and outperforms the conventional bias equalizer. Under the relaxed marking assumption, we evaluate the performance of the proposed bias equalizer and the optimal detector. From our experiments, it is revealed that the proposed detector outperforms the above optimal detector for some collusion strategies.

2 Fingerprinting Code

In this study, we focus on binary fingerprinting codes, especially for the Tardos code [13] and the Nuida code [12]. In this section, we review these codes and related works.

2.1 Probabilistic Code

Let L be a length of codeword, and N be the number of users in a fingerprinting system. If at most c users collude to produce a pirated copy, at least one of them should be identified with a negligibly small false-positive probability, which is a requirement for a fingerprinitng code.

The Tardos code has a length of theoretically minimal order with respect to the number of colluders. The binary codeword of j-th user is denoted by $X_{j,i} \in \{0, 1\}$, $(1 \le i \le L)$, where $X_{j,i}$ is generated from an independently and identically distributed random number with a probability p_i such that $\Pr[X_{j,i} = 1] = p_i$ and $\Pr[X_{j,i} = 0] = 1 - p_i$. This probability p_i follows a continuous distribution $\mathcal{P}$ over an open unit interval $(0, 1)$, which is called *bias distribution.*

In order to improve the performance of the Tardos code, Nuida et al. [11] presented a discrete version of the bias distribution, which is customized for a given number c of colluders. Let $L_k(t) = \left(\frac{d}{dt}\right)^k (t^2 - 1)^k/(k!2^k)$ be the k-th Legendre polynomial, and put $\tilde{L}_k(t) = L_k(2t - 1)$. Then we define $\mathcal{P}_{2k-1}^{GL} = \mathcal{P}_{2k}^{GL}$ to be the finite probability distribution on the set of the k zeroes of $\tilde{L}_k$ such that each value p is taken with probability $\gamma\big(p(1-p)\big)^{-3/2}\tilde{L}_{k'}(p)^{-2}$, where γ is the normalization constant making the sum of the probability equal to 1. Since the output values of $\mathcal{P}_c^{GL}$ and the corresponding output probabilities are not necessarily rational, they are represented by approximated numbers. The numerical examples are shown in Table 1, where p and q denote the output values and their emerging probabilities, respectively. Based on those parameters for a given c, the actual probabilities $p_i, (1 \le i \le L)$ are fixed in a fingerprinting system. Except for the bias distribution, the Nuida code employs the same encoding mechanism as the Tardos code.

Table 1. Examples of the discrete version of Nuida code's bias distribution

c	p	q	c	p	q
1,2	0.50000	1.00000	7,8	0.06943	0.24833
3,4	0.21132	0.50000		0.33001	0.25167
	0.78868	0.50000		0.66999	0.25167
5,6	0.11270	0.33201		0.93057	0.24833
	0.50000	0.33598			
	0.88730	0.33201			

2.2 Tracing Algorithm

Suppose that $\tilde{c}(\leq c)$ malicious users out of N users collude to produce a pirated codeword $\boldsymbol{y} = (y_1, \ldots, y_L)$, $y_i \in \{0,1\}$. A tracing algorithm calculates a score for i-th bit of j-th user, and then sums them up as the total score $S^{(j)}$ of j-th user. Because the original correlation score adds each score only when $y_i = 1$, half of the elements in a pirated codeword is discarded. Considering the symmetry of $\mathcal{P}$, Škorić et al. [14] proposed a symmetric version of the correlation score:

$$S^{(j)} = \sum_{i=1}^{L} (2y_i - 1)U_{j,i}, \tag{1}$$

where

$$U_{j,i} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}} & (X_{j,i} = 1) \\ -\sqrt{\frac{p_i}{1-p_i}} & (X_{j,i} = 0). \end{cases} \tag{2}$$

If the score $S^{(j)}$ exceeds a threshold Z, the user is determined as guilty. Such a tracing algorithm is called "catch-many" type explained in [17]. The maximum allowed probability of accusing a fixed innocent user is denoted by ϵ_1, and the total false-positive probability is by $\eta = 1 - (1 - \epsilon_1)^{N-c} \approx N\epsilon_1$. The false-negative probability denoted by ϵ_2 is coupled to ϵ_1 according to $\epsilon_2 = \epsilon_1^{c/4}$ in [13]. By decoupling ϵ_1 from ϵ_2, the tracing algorithm can detect more colluders under a constant ϵ_1 and L [16][2].

A simple approach to estimate the false-positive probability ϵ_1 is to perform the Monte Carlo simulation. Indeed, it is not easy in general because of the heavy computational costs for estimating a tiny probability. Furon et al. [5] proposed an efficient method for estimating the probability of rare events. The method can estimate the false-positive probability ϵ_1 for a given threshold Z with a reasonable comutational cost, which means that the method calculates the map $\epsilon_1 = F(Z)$. By iteratively performing the estimating method, an objective threshold for a given ϵ_1 can be obtained.

For the Nuida code [11], its original tracing algorithm outputs only one guilty user whose score becomes maximum, which type is called "catch-one". Due to the similarity with the Tardos code, the catch-many tracing algorithm of the Tardos code can be applied to the Nuida code. The report in [6] stated that the performance of the Nuida code is better than that of the Tardos code when the catch-many tracing algorithm is used. Under a same code length and a same number of colluders, it is experimentally measured that the correlation score of the Nuida code is higher than that of the Tardos code.

2.3 Collusion Attack

A group of colluders is denoted by $\mathcal{C} = \{j_1, j_2, \ldots, j_{\tilde{c}}\}$. The collusion attack is the process of taking sequences $X_{j_t,i}, (1 \leq t \leq \tilde{c})$ as inputs and yielding the pirated sequence $\boldsymbol{y}$ as an output. The marking assumption states that the colluders have

$y_i \in \{X_{j_t,i}\}$. It implies that they cannot change the bit in the position where all of $X_{j_t,i}$ is identical.

In [5], the collusion attack is described by the parameter vector: $\boldsymbol{\theta}_{\tilde{\boldsymbol{c}}} = (\theta_0, \cdots, \theta_{\tilde{c}})$ with $\theta_\rho = \Pr[y_i = 1 | \varPhi = \rho], (0 \leq \rho \leq \tilde{c})$, where the random variable $\varPhi \in \{0, \cdots, \tilde{c}\}$ denotes the number of symbol "1" in the colluders' copies at a given index. The marking assumption enforces that $\theta_0 = 0$ and $\theta_{\tilde{c}} = 1$. The positions where symbols belong to $\theta_0 = 0$ and $\theta_{\tilde{c}} = 1$ are called "undetectable position", and the others are "detectable position".

In a watermarking research community, it is assumed that each symbol of a codeword is embedded into a small segment of a content, and colluders can compare their segments. In this attack model, the colluders can notice differences between segments, but they cannot know which segment contains symbol "1". Because of the symmetry of $\mathcal{P}$, symbols "1" and "0" play a symmetric role, and satisfy the following conditional probabilities:

$$\begin{aligned} \Pr[y_i = 1 | \varPhi = \rho] &= \Pr[y_i = 0 | \varPhi = \tilde{c} - \rho] \\ &= 1 - \Pr[y_i = 1 | \varPhi = \tilde{c} - \rho]. \end{aligned} \tag{3}$$

Hence,

$$\theta_\rho = 1 - \theta_{\tilde{c} - \rho}. \tag{4}$$

Notice that $\theta_{c/2} = 0.5$ for even $\tilde{c}$. Some typical examples are "majority", "minority", "random" attacks. It is reported in [5] that some collusion strategies have a deeper impact on the traceability than others and the Worst Case Attack(WCA) is defined as the collusion attack minimizing the rate of the code.

In a cryptography's community, however, it is assumed that the colluders' symbols are identical [3]. Under this scenario, colluders can employ more active collusion strategies by releasing the constraints given by Eq.(4). "all-0" and "all-1" strategies are the typical ones.

On the other hand, the attack strategies are not limited to the above types in a realistic situation such that a codeword is binary and each bit is embedded into one of segments of a digital content without overlapping using a robust watermarking scheme. It is reasonable to assume that each bit is embedded into a segment using an antipodal signal: $\hat{X}_{j,i} = 2X_{j,i} - 1$, namely it is binary phase shift keying(BPSK) modulation. In this case, colluders can apply the other attack strategy at the detectable positions. Since each bit of codeword of $\hat{\boldsymbol{y}}$ is one of $\{-1, 1\}$ after the BPSK modulation, it is possible for colluders to alter the signal amplitude of each element from the signal processing point of view. One simple example is averaging attack that $\hat{y}_i = \sum \hat{X}_{j,i} / \tilde{c}$, we call this attack by "average". Considering the removal of fingerprint signal, a worst case may be $\hat{y}_i = 0$. At the detectable position, it is sufficient to average only two segments whose $\hat{X}_{j,i}$ are different with each other, which attack is denoted by "average2".

Even if a robust watermarking method is used to embed the binary fingerprint code into digital contents, it must be degraded by attacks. In [7], it is assumed that a pirated copy is produced by a collusion attack and is further distorted by the Gaussian noise. Hence, a pirated codeword is represented by

$$\boldsymbol{y'} = \hat{\boldsymbol{y}} + \boldsymbol{e} \tag{5}$$

where $\boldsymbol{e}$ is the additive white Gaussian noise. In this case, symbols at the undetectable position may be changed by the additive noise, hence, the marking assumption is not valid under the noisy model. The above attack assumption is called “relaxed marking assumption” in this paper.

2.4 Related Works

It is reported in [5] that the WCA satisfies the constraint given by Eq.(4). It means that colluders do not need to know which symbol is indeed in the i-th segment. On the contrary, if a collusion strategy ignores the constraint, there may exist more effective detection strategies than the original one.

The Tardos' accusation process is independent with respect to the collusion strategies for codes of infinite length. The symmetric version proposed by Škorić et al. [14] also follows the independency. In [4] and [9], a Maximum A Posterior (MAP) decoder have been proposed by calculating a correlation score considering the collusion strategy:

$$S^{(j)} = \sum_{i=1}^{L} \log \frac{\Pr[y_i | X_{j,i}, \boldsymbol{\theta}_{\tilde{c}}]}{\Pr[y_i | \boldsymbol{\theta}_{\tilde{c}}]}. \tag{6}$$

This MAP detector is optimally discriminative in the Neyman-Pearson theorem point of view. However, there is a difficulty to realize such a detector because it requires the knowledge of $\tilde{c}$ and $\boldsymbol{\theta}_{\tilde{c}}$ in advance.

In [4], the collusion strategy $\boldsymbol{\theta}_{\tilde{c}}$ is estimated by applying the Expectation-Maximization (EM) algorithm [1]. However, the experimental results reveal that the accuracy of the estimation is not sufficiently high. In [9], the detector is generalized by finding the maximum of the MAP score for finite possible collusion strategies. The detector seems the best tracing algorithm to the best of our knowledge. Nevertheless, the estimation of collusion strategy is under investigation and the impact of mismatch has to be studied. Under the relaxed marking assumption, the white Gaussian noise may restrict the availability of the MAP detector because each symbol y'_i of a distorted codeword is rounded into “1” or “0” symbol before performing the above MAP detector.

3 Bias Equalizer

First, we suppose a noiseless case for the convenience of discussion. Let $\mathcal{Y}_1$ and $\mathcal{Y}_0$ be the set of indices i satisfying $y_i = 1$ and $y_i = 0$, respectively. Then, the numbers of elements in $\mathcal{Y}_1$ and $\mathcal{Y}_0$ are denoted by L_1 and L_0, respectively, where $L_1 + L_0 = L$. Because of the symmetry of a bias distribution, it is expected to be $L_1 = L_0$ unless the colluders do not know the actual values $X_{j,i}$ of their codewords. However, when they happen to get the values contained in segments, they can perform more active collusion strategies such as “all-0” and “all-1”. It is not surprising that L_1 is not always equal to L_0 in a real situation. Because of the probabilistic generation of codewords, there must be fluctuations between the number of symbols “1” and “0”. The imbalance is exploited to improve the traceability by introducing a weighting parameter on the calculation of correlation score.

For convenience, $S^{(j)}$ given by Eq.(1) is rewritten by

$$S^{(j)} = \sum_{i \in \mathcal{Y}_1} U_{j,i} - \sum_{i \in \mathcal{Y}_0} U_{j,i}, \tag{7}$$

The first term is the score related to $y_i = 1$ and the second one is to $y_i = 0$. When $L_1 = L_0$, these two terms are balanced, and hence, their summand are equal. Once the balance breaks down, the influence from one term dominates the other one. In order to equalize the bias of these numbers, $S^{(j)}$ is modified in [8]. It is noted that L_1 and L_0 are easily derived from the observation of $\boldsymbol{y}$ by counting the symbols “1” and “0”. Then, $S^{(j)}$ is modified by these parameters as follows:

$$S^{(j)} = \frac{1}{L_1} \sum_{i \in \mathcal{Y}_1} U_{j,i} - \frac{1}{L_0} \sum_{i \in \mathcal{Y}_0} U_{j,i}. \tag{8}$$

At a judgment, a threshold Z corresponding for the above correlation score $S^{(j)}$ is calculated for a given false-positive probability ϵ_1 using the rare event simulator $F(Z)$ explained in Sect.2.2.

Under the relaxed marking assumption, the above correlation score $S^{(j)}$ must be accommodated for considering the effect of additive white Gaussian noise. In [8], the detector first estimates the collusion channel regarded as a Gaussian Mixture Model (GMM) by performing the EM algorithm. The probability density function $pdf(y'_i)$ of distorted symbol y'_i is denoted by

$$pdf(y'_i) = \sum_{k=0}^{m-1} a_k \mathcal{N}(y'_i; \mu_k, \sigma_k^2), \tag{9}$$

where

$$\mathcal{N}(y'_i; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(y'_i - \mu)^2}{2\sigma^2} \right), \tag{10}$$

m is the number of Gaussian components, and $\sum_{k=0}^{m-1} a_k = 1$. If the “average” or “average2” attack is performed, the number of Gaussian components is at most $m = 3$; otherwise, $m = 2$ for collusion strategies under the marking assumption. When $m = 3$, the EM algorithm must estimate the following five parameters: a_0, a_1, a_2, μ_2 and $\sigma_e^2 (= \sigma_0^2 = \sigma_1^2 = \sigma_2^2)$. On the other hand, among these five parameters, a_2 and μ_2 are omitted when $m = 2$. Because $\hat{y}_i = \pm 1$ is distorted by a white Gaussian noise, there are at least two mean values $\mu_0 = -1$ and $\mu_1 = 1$. Then, the correlation score $S_i^{(j)}$ before equalization is derived from the following equation:

$$S^{(j)} = \sum_{i=1}^{L} \frac{a_1 \mathcal{N}(y'_i; \mu_1, \sigma_e^2) - a_0 \mathcal{N}(y'_i; \mu_0, \sigma_e^2)}{\displaystyle\sum_{k=0}^{m-1} a_k \mathcal{N}(y'_i; \mu_k, \sigma_e^2)} U_{j,i}. \tag{11}$$

After equalization, the above equation is modified as follows [8]:

$$S^{(j)} = \sum_{i=1}^{L} \frac{\mathcal{N}(y_i'; \mu_1, \sigma_e^2) - \mathcal{N}(y_i'; \mu_0, \sigma_e^2)}{\displaystyle\sum_{k=0}^{m-1} a_k \mathcal{N}(y_i'; \mu_k, \sigma_e^2)} U_{j,i}. \tag{12}$$

Note that $a_0 = L_0/L$ and $a_1 = L_1/L$ when $m = 2$. In this case, the above equation can be rewritten by

$$S^{(j)} = L\left(\frac{1}{L_1}\sum_{i=1}^{L} y_{i,1}^{ML} U_{j,i} - \frac{1}{L_0}\sum_{i=1}^{L} y_{i,0}^{ML} U_{j,i}\right). \tag{13}$$

where

$$y_{i,\nu}^{ML} = \frac{a_\nu \mathcal{N}(y_i'; \mu_\nu, \sigma_e^2)}{\displaystyle\sum_{k=0}^{m-1} a_k \mathcal{N}(y_i'; \mu_k, \sigma_e^2)} \tag{14}$$

for $\nu = 0$ or 1. By scaling down the above score by a factor of L, we obtain a simplified formula:

$$S^{(j)} = \frac{1}{L_1} \sum_{i \in \{\mathcal{Y}_0 \cup \mathcal{Y}_1\}} y_{i,1}^{ML} U_{j,i} - \frac{1}{L_0} \sum_{i \in \{\mathcal{Y}_0 \cup \mathcal{Y}_1\}} y_{i,0}^{ML} U_{j,i}. \tag{15}$$

4 Proposed Tracing Algorithm

The bias equalizer explained in the previous section can improve the traceability effectively only when the imbalance of symbols "1" and "0" is observed in a whole codeword. The idea of bias equalizer is also applicable for the Nuida code. In this section, we further study biases in its codeword and propose a new bias equalizer customized for the Nuida code. Because of its discrete bias distribution, it is possible to classify each symbol of a codeword into some groups corresponding to the probabilities p in Table 1. The proposed method calculates weighting parameters for those groups by observing the number of symbols within those groups.

Let n_c be the number of candidates of p_i for the Nuida code (e.g. $n_c = 4$ when $c = 7, 8$ as shown in Table 1). Hence, p_i can be classified into n_c groups. The number of elements in each group is expected to be $q_\xi L, (1 \leq \xi \leq n_c)$, where q_ξ is the emerging probability of p_i in the group(See Table 1). For convenience, the number of elements is denoted by ℓ_ξ where $\ell_\xi \geq 0$ and $\sum_{\xi=1}^{n_c} \ell_\xi = L$. Since p_i is not always 1/2 for $c > 2$, the number of symbols "1" and "0" in each group is imbalanced even for innocent users' codewords as well as a pirated codeword. Such a bias is exploited to the tracing algorithm in our method.

Let $\mathcal{Y}_{\xi,1}$ and $\mathcal{Y}_{\xi,0}$ be the set of indices i satisfying $y_i = 1$ and $y_i = 0$ in the ξ-th group, respectively. Similar to the representation in Eq.(8), the numbers of symbols "1" and "0" are denoted by $\ell_{\xi,1}$ and $\ell_{\xi,0}$, respectively. Notice that $\ell_{\xi,1}+\ell_{\xi,0}=\ell_\xi$. Using those parameters, the correlation score $S_\xi^{(j)}$ for ξ-th group is calculated as follows:

$$S_\xi^{(j)}=\frac{1}{\ell_{\xi,1}}\sum_{i\in\mathcal{Y}_{\xi,1}}U_{j,i}-\frac{1}{\ell_{\xi,0}}\sum_{i\in\mathcal{Y}_{\xi,0}}U_{j,i}. \tag{16}$$

The influence from the first term in the above equation is equivalent to that from the second term because of the above weighting parameters. However, these weighting parameters are not always valid since the number $\ell_{\xi,0}$ or $\ell_{\xi,1}$ happens to be zero with a non-negligible probability. Considering the weighting ratio between the first and second terms in the above equation, the above weighting parameters must be changed. Based on this consideration, $S_\xi^{(j)}$ is modified by

$$S_\xi^{(j)}=\ell_{\xi,0}\sum_{i\in\mathcal{Y}_{\xi,1}}U_{j,i}-\ell_{\xi,1}\sum_{i\in\mathcal{Y}_{\xi,0}}U_{j,i}. \tag{17}$$

After the above conversion, although the score $S_\xi^{(j)}$ in the above equation is changed from the value derived from Eq.(16), the weighting ratio between the first and second terms is unchanged. The above modification only considers the bias between the symbols "1" and "0" in a group. Here, it is noticed that the number ℓ_ξ of symbols in ξ-th group is different, namely, there is a bias among groups to be equalized. Therefore, the total score is calculated by

$$S^{(j)}=\sum_{\xi=1}^{n_c}\frac{S_\xi^{(j)}}{\ell_\xi} \tag{18}$$

It is worth-mentioning that the proposed bias equalizer only observes the symbols in a pirated codeword and cancels the bias using the encoding parameters p_i.

Now, let us consider the relaxed marking assumption for the above correlation score. Similar to the discussion in the previous section, the correlation score $S_\xi^{(j)}$ can be calculated using Eqs.(14), (15), and (17).

$$S_\xi^{(j)}=\ell_{\xi,0}\sum_{i\in\{\mathcal{Y}_{\xi,0}\cup\mathcal{Y}_{\xi,1}\}}y_{i,1}^{ML}U_{j,i}-\ell_{\xi,1}\sum_{i\in\{\mathcal{Y}_{\xi,0}\cup\mathcal{Y}_{\xi,1}\}}y_{i,0}^{ML}U_{j,i}. \tag{19}$$

In a noisy case, the number of symbols is not derived directly from the observation of y'_i, $(1\le i\le L)$. After quantizing y'_i to

$$\tilde{y}'_i=\begin{cases}1 & \text{if } y'_i\ge 0\\ 0 & \text{otherwise}\end{cases}, \tag{20}$$

we count the numbers $\ell_{\xi,1}$ and $\ell_{\xi,0}$ from $\tilde{y}'_i$, $(i\in\{\mathcal{Y}_{\xi,0}\cup\mathcal{Y}_{\xi,1}\})$. With the increase of noise energy, however, the above classification is not always valid. Because of the simplicity, $\ell_{\xi,1}$ and $\ell_{\xi,0}$ are derived using the above classification in this paper. Strictly speaking, the probability density function of y'_i in ξ-th group is represented by

$$pdf(y'_i, \xi) = \frac{\ell_{\xi,1}}{\ell_\xi}\mathcal{N}(y'_i; 1, \sigma_e^2) + \frac{\ell_{\xi,0}}{\ell_\xi}\mathcal{N}(y'_i; -1, \sigma_e^2). \tag{21}$$

Hence, the exploitation of the EM algorithm will be a better choice for estimating $\ell_{\xi,1}$ and $\ell_{\xi,0}$, which is left for our future work.

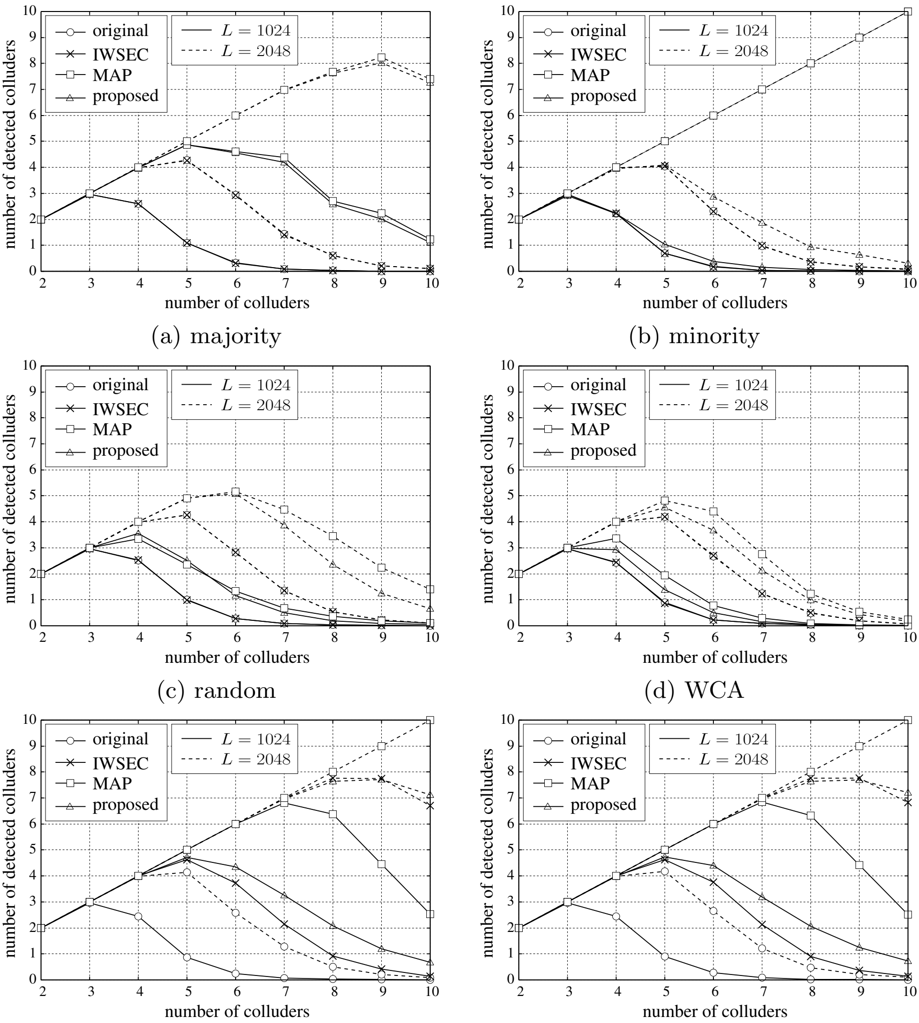
5 Experimental Results

For the comparison of the performance of proposed method, we perform the Monte Carlo simulation such that pirated codewords are produced by collusion attack on randomly selected 10^3 combinations of $\tilde{c}$ colluders. The number of users is $N = 10^4$ in this experiment. By giving a false-positive probability $\epsilon_1 = 10^{-8}$, the corresponding threshold Z is calculated by the rare event simulator. Under this condition, the total false-positive probability is approximated to be $\eta \approx N\epsilon_1 = 10^{-4}$. The maximum number of colluders is set to be $c = 8$ for Nuida code in this experiment.

First, the traceability is evaluated for 6 collusion strategies under a marking assumption. We compare the traceability of the proposed detector with those of the original symmetric detector [14], the detector with the bias equalizer explained in Sect. 3, and the MAP detector[1] [9], which are denoted by "original", "IWSEC", and "MAP", respectively. In this experiment, we assume that the MAP detector knows the number $\tilde{c}$ of colluders and the vector $\boldsymbol{\theta}_{\tilde{c}}$, namely, it is regarded as the optimal detector under the noiseless case. Figure 1 shows the number of detected colluders using the length $L = 1024, 2048$. It is observed that the performance of the conventional detector (IWSEC) is very close to that of original detector under four strategies: majority, minority, random, and WCA. It is because the number of symbols "1" and "0" in a whole pirated codeword is balanced after those attacks. On the other hand, the proposed detector improves the traceability for those attacks. It means that the proposed detector effectively enhances the traceability. Especially for the majority strategy, the traceability of the proposed detector is very close to that of the MAP detector. The proposed detector also improves the traceability against the collusion strategies all-0 and all-1 compared with the conventional detector. Similar results are derived for longer L, so they are omitted to show.

It is noted that the total false-positive probabilities of those detectors are constant and equal in this experiment. For the confirmation, the number of false-positive detection is measured for 10^5 trials of Monte Carlo simulation. Because of the limitation of space, the results of the MAP detector and the proposed detector only for the majority strategy are plotted in Fig. 2. It is observed that the false-positive probability of the proposed detector is slightly less than $\eta \approx 10^{-4}$

[1] We use the following source code for the evaluation of the MAP detector `http://www.irisa.fr/texmex/people/furon/fp.zip`



(a) majority

(b) minority

(c) random

(d) WCA

(e) all-0

(f) all-1

Fig. 1. Comparison of the number of detected colluders in a noiseless case

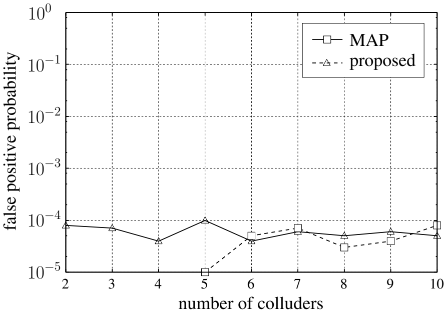


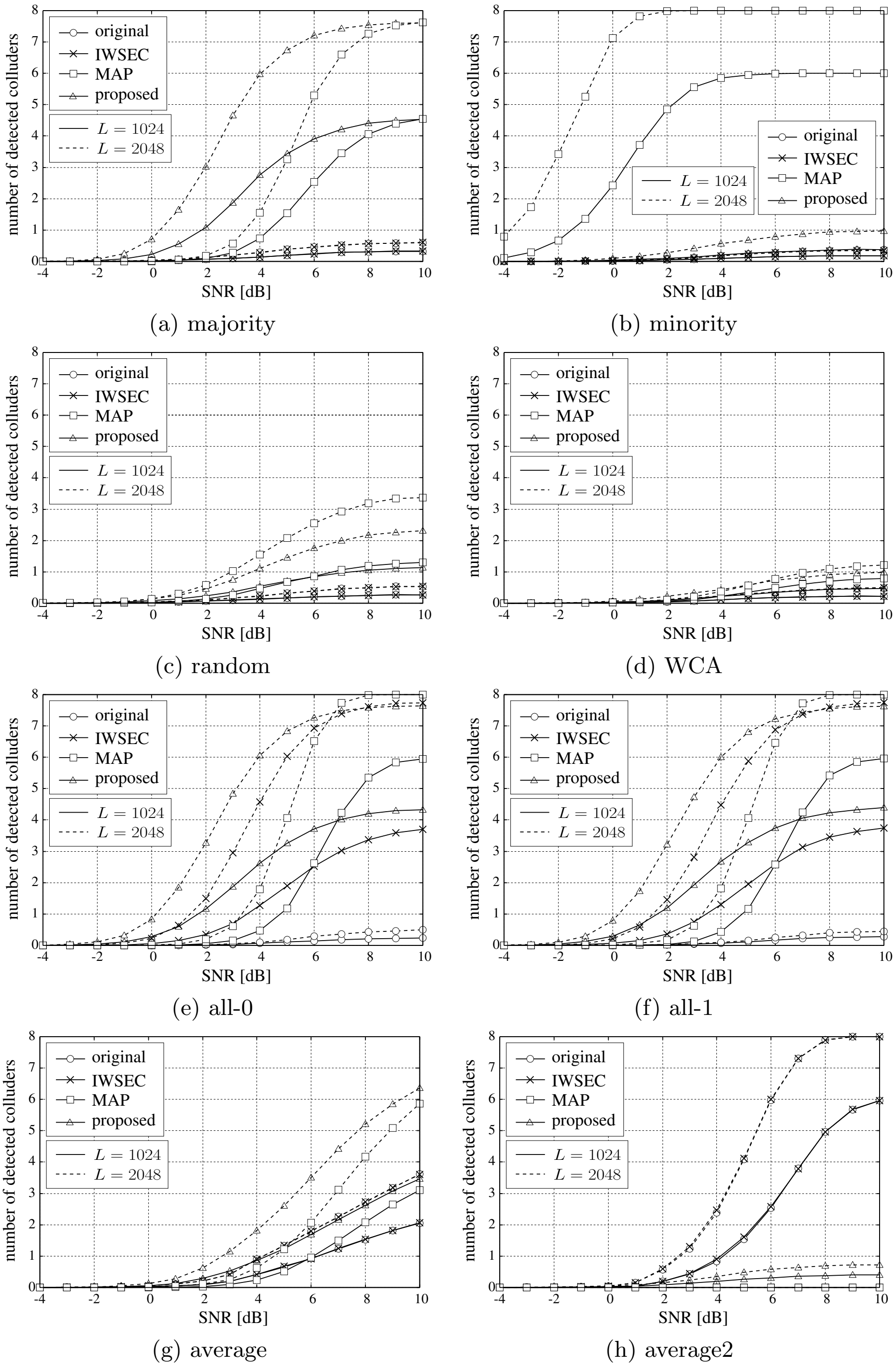
Fig. 2. Comparison of false-positive probability when $L = 1024$

regardless of the number of colluders, while the probability of the MAP detector becomes much lower with the decrease of the number of colluders. In any case, we confirmed that the false-positive probability of the proposed detector was independent on the collusion strategies and the number of colluders.

Next, we evaluate the performance under the relaxed marking assumption. In this experiment, a pirated codeword is produced by $\tilde{c} = 6$ colluders' codewords when $L = 1024$, and by $\tilde{c} = 8$ when $L = 2048$, and it is distorted by a white Gaussian noise. The number of detected colluders versus SNR [dB] is plotted in Fig.3. Because of the low traceability under "WCA" and "random" strategies, their false-negative probability is plotted in Fig.4 for the comparison of the performance when $L = 1024$. It is observed that the performance of the MAP detector is rapidly dropped as the decrease of SNR for the collusion strategies under the marking assumption. It means that the proposed detector should be selected rather than the MAP detector when the amount of noise added to a pirated codeword increases. As shown in Fig.5, the false-positive probability is still constant against the majority strategy, and the similar results are derived for other strategies.

On the other hand, interesting results are derived for averaging strategies. The proposed detector is better than the others for the "average" strategy while the conventional "original" and "IWSEC" detectors are better for the "average2" strategy when the SNR is within the range between -4 and 10 [dB]. With the decrease of the amount of noise, it is revealed that the conventional ones can catch all colluders involved in a pirated codeword. By changing the weighting parameters in Eq.(19), we check the traceability against these attacks. Because of the lack of space, we omit to show the numerical result, but we can say that it is required to design a detector considering the effects on a codeword caused by these averaging strategies.

Since we assume that the Gaussian noise is added to a pirated codeword after the collusion attack, the elements $y'_i = \sum \hat{X}_{j,i}/\tilde{c} + e_i$, $1 \leq i \leq L$ are input to a detector, where e_i is a noise. If $e_i = 0$, the "average" strategy is coincident with the "majority" strategy at the MAP detector because each y'_i is classified into one of ± 1 symbols. When $\sum \hat{X}_{j,i}/\tilde{c} = 0$, the noise e_i significantly affects



(a) majority

(b) minority

(c) random

(d) WCA

(e) all-0

(f) all-1

(g) average

(h) average2

Fig. 3. Comparison of the number of detected colluders under the relaxed marking assumption, where $\tilde{c} = 6, 8$ for $L = 1024, 2048$, respectively

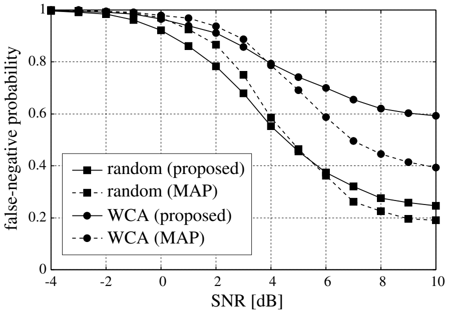


Fig. 4. Comparison of false-negative probability where $\tilde{c} = 6$ for $L = 1024$

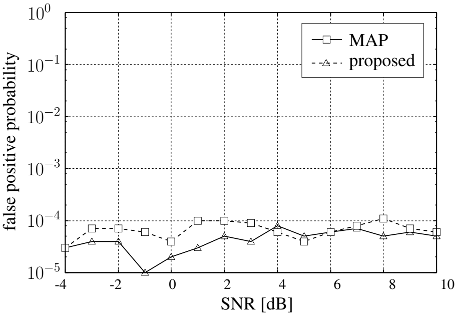


Fig. 5. Comparison of false-positive probability where $\tilde{c} = 6$ for $L = 1024$

the classification even if it is very small. Hence, the performance of MAP detector becomes lower. For the "average2" strategy, half of symbols at detectable positions are wrongly classified at the MAP detector, but $\boldsymbol{\theta}_{\tilde{c}}$ is designed as the "majority" strategy in this experiment. Hence, it is required to estimate $\boldsymbol{\theta}_{\tilde{c}}$ correctly to obtain the maximum performance of the MAP detector. Under the relaxed marking assumption, it is found that colluders can modify a symbol $\hat{y}_i$ in a pirated codeword to the range $[-1, 1]$ without an addition of noise, and hence, we come up against a difficulty in the modeling of distorted codeword, especially for the noisy case.

6 Concluding Remarks

In this paper, we proposed a bias equalizer for Nuida's fingerprinting code using the discrete bias distribution. When the number of symbols "1" and "0" in a pirated codeword is imbalanced, the proposed detector can catch more colluders than the conventional method. Furthermore, even if the number of symbols are balanced in a whole codeword, it is imbalanced for each set of symbols related to the corresponding probability p_i. Such a bias is equalized in the proposed detector to improve the performance. Although the performance of the proposed detector is lower than that of the MAP detector under a marking assumption, it is not always true under the relaxed marking assumption. The MAP detector should consider the analogue value of a distorted codeword as well as the type of collusion strategy.

The effect of bias equalizer is under investigation in a current version. The detailed and theoretical analysis is left for our future work. In addition, the design of an optimal detector under the relaxed marking assumption is still an open problem. The proposed bias equalizer is highly customized for the Nuida code because its discrete bias distribution is exploited. However, if the continuous bias distribution of the Tardos code is divided into small ones, it is possible to apply the proposed method to the Tardos code.

Acknowledgment. This research was partially supported by the Ministry of Education, Culture, Sports Science and Technology, Grant-in-Aid for Young Scientists (B) (24760299).

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
2. Blayer, O., Tassa, T.: Improved versions of Tardos' fingerprinting scheme. Designs, Codes and Cryptography 48(1), 79–103 (2008)
3. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Trating traitors. IEEE Trans. Inform. Theory 46(3), 893–910 (2000)
4. Furon, T., Preire, L.P.: EM decoding of Tardos traitor tracing codes. In: ACM Multimedia and Security, pp. 99–106 (2009)
5. Furon, T., Pérez-Freire, L., Guyader, A., Cérou, F.: Estimating the Minimal Length of Tardos Code. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 176–190. Springer, Heidelberg (2009)
6. Kuribayashi, M.: Experimental Assessment of Probabilistic Fingerprinting Codes over AWGN Channel. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) IWSEC 2010. LNCS, vol. 6434, pp. 117–132. Springer, Heidelberg (2010)
7. Kuribayashi, M.: Tardos"s Fingerprinting Code over AWGN Channel. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 103–117. Springer, Heidelberg (2010)
8. Kuribayashi, M.: A New Soft Decision Tracing Algorithm for Binary Fingerprinting Codes. In: Iwata, T., Nishigaki, M. (eds.) IWSEC 2011. LNCS, vol. 7038, pp. 1–15. Springer, Heidelberg (2011)
9. Meerwald, P., Furon, T.: Towards joint decoding of Tardos fingerprinting codes. CoRR abs/1104.5616 (2011), http://arxiv.org/abs/1104.5616
10. Nuida, K.: A General Conversion Method of Fingerprint Codes to (More) Robust Fingerprint Codes against Bit Erasure. In: Kurosawa, K. (ed.) ICITS 2009. LNCS, vol. 5973, pp. 194–212. Springer, Heidelberg (2010)
11. Nuida, K., Fujitu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., Imai, H.: An improvement of discrete Tardos fingerprinting codes. Designs, Codes and Cryptography 52(3), 339–362 (2009)
12. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimization of Tardos's Fingerprinting Codes in a Viewpoint of Memory Amount. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 279–293. Springer, Heidelberg (2008)
13. Tardos, G.: Optimal probabilistic fingerprint codes. J. ACM 55(2), 1–24 (2008)
14. Škorić, B., Katzenbeisser, S., Celik, M.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
15. Škorić, B., Katzenbeisser, S., Schaathun, H., Celik, M.: Tardos fingerprinting codes in the combined digit model. IEEE Trans. Inform. Forensics and Security 6(3), 906–919 (2011)
16. Škorić, B., Vladimirova, T.U., Celik, M., Talstra, J.C.: Tardos fingerprinting is better than we thought. IEEE Trans. Inform. Theory 54(8), 3663–3676 (2008)
17. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion resistant fingerprinting for multimedia. IEEE Signal Processing Mag., 15–27 (2004)
18. Xie, F., Furon, T., Fontaine, C.: On-off keying modulation and Tardos fingerprinting. In: MMSec 2008: Proc. ACM Multimedia and Security, pp. 101–106 (2008)

Author Index

Aliaga, Daniel G. 94

Bas, Patrick 207
Boesten, Dion 255
Böhme, Rainer 125

Cao, Xiaochun 110
Chao, Yi-Liu 94
Chen, Chenglong 1
Chen, Licong 63
Chuang, Wei-Hong 16
Cogranne, Rémi 46
Cornu, Philippe 46

Danezis, George 191
Du, Ling 110
Dutta, Aveek 160

Fillatre, Lionel 46
Fridrich, Jessica 31, 78
Fu, Huazhu 110
Furon, Teddy 207

Grunwald, Dirk 160

Huang, Jiwu 1, 240
Huang, Rongbin 1

Käsper, Emilia 191
Kiayias, Aggelos 142, 176
Kodovský, Jan 31, 78
Kuribayashi, Minoru 269

Ni, Jiangqun 1
Nikiforov, Igor 46

Retraint, Florent 46
Russell, Alexander 142

Saha, Dola 160
Samari, Katerina 176
Schöttle, Pascal 125
Shashidhar, Narasimha 142
Shi, Yun Q. 63
Sicker, Douglas 160
Škorić, Boris 255
Sutthiwan, Patchara 63

Wu, Min 16

Xiang, Shijun 224

Zhang, Muhua 110
Zheng, Peijia 240
Zitzmann, Cathel 46