Tomáš Filler
Tomáš Pevný
Scott Craver
Andrew Ker (Eds.)

LNCS 6958

Information Hiding

13th International Conference, IH 2011
Prague, Czech Republic, May 2011
Revised Selected Papers

Lecture Notes in Computer Science 6958

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Tomáš Filler Tomáš Pevný
Scott Craver Andrew Ker (Eds.)

Information Hiding

13th International Conference, IH 2011
Prague, Czech Republic, May 18-20, 2011
Revised Selected Papers

Springer

Volume Editors

Tomáš Filler
Digimarc Corporation
9405 Gemini Drive
Beaverton, OR, 97008, USA
E-mail: tomas.filler@digimarc.com

Tomáš Pevný
Czech Technical University
Faculty of Electrical Engineering, Department of Cybernetics
Karlovo namesti 13
121 35 Prague 2, Czech Republic
E-mail: pevnak@gmail.com

Scott Craver
SUNY Binghamton
T. J. Watson School, Department of Electrical and Computer Engineering
Binghamton, NY 13902, USA
E-mail: scraver@binghamton.edu

Andrew Ker
University of Oxford, Department of Computer Science
Wolfson Building, Parks Road
Oxford OX1 3QD, UK
E-mail: Andrew.Ker@comlab.ox.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-24177-2 e-ISBN 978-3-642-24178-9
DOI 10.1007/978-3-642-24178-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011936237

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 International Hiding Conference was founded 15 years ago, with the first conference held in Cambridge, UK, in 1996. Since then, the conference locations have alternated between Europe and North America. In 2011, during May 18–20, we had the pleasure of hosting the 13th Information Hiding Conference in Prague, Czech Republic. The 60 attendees had the opportunity to enjoy Prague in springtime as well as inspiring presentations and fruitfull discussions with colleagues.

The International Hiding Conference has a tradition in attracting researchers from many closely related fields including digital watermarking, steganography and steganalysis, anonymity and privacy, covert and subliminal channels, fingerprinting and embedding codes, multimedia forensics and counter-forensics, as well as theoretical aspects of information hiding and detection. In 2011, the Program Committee reviewed 69 papers, using a double-blind system with at least 3 reviewers per paper. Then, each paper was carefully discussed until consensus was reached, leading to 23 accepted papers (33% acceptance rate), all published in these proceedings.

The invited speaker was Bernhard Schölkopf, who presented his thoughts on why kernel methods (and support vector machines in particular) are so popular and where they are heading. He also discussed some recent developments in two-sample and independence testing as well as applications in different domains.

At this point, we would like to thank everyone, who helped to organize the conference, namely, Jakub Havránek from the Mediaform agency and Bára Jeníková from CVUT in Prague. We also wish to thank the following companies and agencies for their contribution to the success of this conference: European Office of Aerospace Research and Development, Air Force Office of Scientific Research, United States Air Force Research Laboratory (www.london.af.mil), the Office of Naval Research Global (www.onr.navy.mil), Digimarc Corporation (www.digimarc.com), Technicolor (www.technicolor.com), and organizers of IH 2008 in santa Barbara, CA, USA. Without their generous financial support, the organization would have been very difficult.

July 2011

Tomáš Filler
Tomáš Pevný
Scott Craver
Andrew Ker

Organization

13th Information Hiding Conference
May 18–20, 2011, Prague (Czech Republic)

General Chair

Tomáš Pevný	Czech Technical University, Czech Republic

Program Chairs

Tomáš Filler	SUNY Binghamton / Digimarc Corp., USA
Scott Craver	SUNY Binghamton, USA
Andrew Ker	University of Oxford, UK

Program Committee

Ross Anderson	University of Cambridge, UK
Mauro Barni	Università di Siena, Italy
Patrick Bas	CNRS, France
Rainer Böhme	University of Münster, Germany
Franǫis Cayre	GIPSA-lab/Grenoble INP, France
Ee-Chien Chang	National University of Singapore, Singapore
Christian Collberg	University of Arizona, USA
Ingemar J. Cox	University College London, UK
George Danezis	Microsoft Research Cambridge, UK
Gwenaël Doërr	Technicolor, France
Jessica Fridrich	SUNY Binghamton, USA
Teddy Furon	INRIA, France
Neil F. Johnson	Booz Allen Hamilton and JJTC, USA
Stefan Katzenbeisser	TU Darmstadt, Germany
Darko Kirovski	Microsoft Research, USA
John McHugh	University of North Carolina, USA and RedJack, LLC.
Ira S. Moskowitz	Naval Research Laboratory, USA
Ahmad-Reza Sadeghi	Ruhr-Universität Bochum, Germany
Rei Safavi-Naini	University of Calgary, Canada
Phil Sallee	Booz Allen Hamilton, USA
Berry Schoenmakers	TU Eindhoven, The Netherlands
Kaushal Solanki	Mayachitra Inc., USA
Kenneth Sullivan	Mayachitra Inc., USA
Paul Syverson	Naval Research Laboratory, USA

Local Organization

Jakub Havránek	Mediaform, Czech Republic
Barbora Jeníková	Czech Technical University, Czech Republic

External Reviewer

Boris Škorić	Eindhoven University of Technology, The Netherlands

Sponsoring Institutions

European Office of Aerospace Research and Development
Office of Naval Research
Digimarc Corporation, USA
Technicolor, France

Table of Contents

Fingerprinting

Special Session on BOSS Contest

Anonymity and Privacy

Steganography and Steganalysis

Watermarking

Digital Rights Management and Digital Forensics

Data Hiding in Unusual Content

Asymptotic Fingerprinting Capacity for Non-binary Alphabets

Dion Boesten and Boris Škorić

Eindhoven University of Technology

Abstract. We compute the channel capacity of non-binary fingerprinting under the Marking Assumption, in the limit of large coalition size c. The solution for the binary case was found by Huang and Moulin. They showed that asymptotically, the capacity is $1/(c^2 2 \ln 2)$, the interleaving attack is optimal and the arcsine distribution is the optimal bias distribution.

In this paper we prove that the asymptotic capacity for general alphabet size q is $(q-1)/(c^2 2 \ln q)$. Our proof technique does not reveal the optimal attack or bias distribution. The fact that the capacity is an increasing function of q shows that there is a real gain in going to non-binary alphabets.

1 Introduction

1.1 Collusion Resistant Watermarking

Watermarking provides a means for tracing the origin and distribution of digital data. Before distribution of digital content, the content is modified by applying an imperceptible watermark (WM), embedded using a watermarking algorithm. Once an unauthorized copy of the content is found, it is possible to trace those users who participated in its creation. This process is known as 'forensic watermarking'. Reliable tracing requires resilience against attacks that aim to remove the WM. Collusion attacks, where several users cooperate, are a particular threat: differences between their versions of the content tell them where the WM is located. Coding theory has produced 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. The interface between the layers is usually specified in terms of the *Marking Assumption*, which states that the colluders are able to perform modifications only in those segments where they received different WMs. These segments are called detectable positions.

Many collusion resistant codes have been proposed in the literature. Most notable is the Tardos code [13], which achieves the asymptotically optimal proportionality $m \propto c^2$, with m the code length. Tardos introduced a two-step

[1] The 'segments' are defined in a very broad sense. They may be coefficients in any representation of the content (codec).

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 1–13, 2011.

stochastic procedure for generating binary codewords: (i) For each segment a bias is randomly drawn from some distribution F. (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 alphabets in [14].

1.2 Related Work: Channel Capacity

In the original Tardos scheme [13] and many later improvements and generalisations (e.g. [16,14,3,10,9,4,15,17]), 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 discussion 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 [2,8,7], 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 F.

It was conjectured [7] that the binary capacity is asymptotically given by $1/(c^2 2 \ln 2)$. The conjecture was proved in [1,6]. Amiri and Tardos [1] developed an 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.

1.3 Contributions and Outline

We prove for alphabet size q that the asymptotic fingerprinting capacity is $\frac{q-1}{c^2 2 \ln q}$. Our proof makes use of the fact that the value of the maximin game can be found by considering the minimax game instead (i.e. in the reverse order). This proof does not reveal the asymptotically optimal collusion strategy and bias distribution of the maximin game.

In Section 2 we introduce notation, discuss the information-theoretic payoff game and present lemmas that will be used later. In Section 3 we analyze the properties of the payoff function in the large-c limit. We solve the minimax game in Section 4. In Section 5 we discuss the benefits of larger alphabets.

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. 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. If $\boldsymbol{p}$ and $\boldsymbol{\sigma}$ are two vectors of length n then by $\boldsymbol{p}^{\boldsymbol{\sigma}}$ we denote $\prod_{i=1}^{n} p_i^{\sigma_i}$. If c is a positive integer and $\boldsymbol{\sigma}$ is a vector of length n of nonnegative integers with sum equal to c then $\binom{c}{\boldsymbol{\sigma}}$ denotes the multinomial coefficient $\frac{c!}{\sigma_1!\sigma_2!\ldots\sigma_n!}$. 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 denoted by $\sum_x$. 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.

2.2 Fingerprinting with Per-Segment Symbol Biases

Tardos [13] 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 [14]. 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. (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: $\mathbb{P}[X_{ij} = \alpha] = p_\alpha^{(j)}$. The code generation has independence of all columns and rows.

2.3 The Collusion Attack

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 $\sum_\alpha \sigma_\alpha^{(j)} = c$ for all j. From now on we will drop the segment index j, since all segments are independent. For given $\boldsymbol{p}$, the vector $\boldsymbol{\Sigma}$ is multinomial-distributed,

$$\Lambda_{\boldsymbol{\sigma}|\boldsymbol{p}} \triangleq \text{Prob}[\boldsymbol{\Sigma} = \boldsymbol{\sigma}|\boldsymbol{P} = \boldsymbol{p}] = \binom{c}{\boldsymbol{\sigma}} \boldsymbol{p}^{\boldsymbol{\sigma}}. \tag{1}$$

The colluders' goal is to produce a symbol Y that does not incriminate them. It has been shown 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 $\theta_{y|\boldsymbol{\sigma}} \in [0,1]$, which are defined as:

$$\theta_{y|\boldsymbol{\sigma}} \triangleq \text{Prob}[Y = y \mid \boldsymbol{\Sigma} = \boldsymbol{\sigma}]. \tag{2}$$

For all $\boldsymbol{\sigma}$, conservation of probability gives $\sum_y \theta_{y|\boldsymbol{\sigma}} = 1$. Due to the Marking Assumption, $\sigma_\alpha = 0$ implies $\theta_{\alpha|\boldsymbol{\sigma}} = 0$ and $\sigma_\alpha = c$ implies $\theta_{\alpha|\boldsymbol{\sigma}} = 1$. The so called *interleaving attack* is defined as $\theta_{\alpha|\boldsymbol{\sigma}} = \sigma_\alpha/c$.

2.4 Collusion Channel and Fingerprinting Capacity

The attack can be interpreted as a noisy channel with input $\boldsymbol{\Sigma}$ and output Y. 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 $C_c(q)$ for a coalition of size c and alphabet size q is equal to the optimal value of the following two-player game:

$$C_c(q) = \max_F \min_{\boldsymbol{\theta}} \frac{1}{c} I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P}) = \max_F \min_{\boldsymbol{\theta}} \frac{1}{c} \int F(\boldsymbol{p}) I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) d^q\boldsymbol{p}. \quad (3)$$

Here the information is measured in q-ary symbols. Our aim is to compute the fingerprinting capacity $C_c(q)$ in the limit ($n \to \infty$, $c \to \infty$).

2.5 Alternative Mutual Information Game

The payoff function of the game (3) is the mutual information $I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P})$. It is convex in $\boldsymbol{\theta}$ (see e.g. [5]) and linear in F. This allows us to apply Sion's minimax theorem (Lemma 1), yielding

$$\max_F \min_{\boldsymbol{\theta}} I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P}) = \min_{\boldsymbol{\theta}} \max_F I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P}) \quad (4)$$

$$= \min_{\boldsymbol{\theta}} \max_{\boldsymbol{p}} I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) \quad (5)$$

where the last equality follows from the fact that the maximization over F in (4) results in a delta distribution located at the maximum of the payoff function. The game (3) is what happens in reality, but by solving the alternative game (5) we will obtain the asymptotic fingerprinting capacity.

2.6 Useful Lemmas

The following lemmas will prove useful for our analysis of the asymptotic fingerprinting game.

Lemma 1 (Sion's minimax theorem [12]). *Let $\mathcal{X}$ be a compact convex subset of a linear topological space and $\mathcal{Y}$ a convex subset of a linear topological space. Let $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a function with*

- *$f(x, \cdot)$ upper semicontinuous and quasiconcave on $\mathcal{Y}$, $\forall x \in \mathcal{X}$*
- *$f(\cdot, y)$ lower semicontinuous and quasi-convex on $\mathcal{X}$, $\forall y \in \mathcal{Y}$*

then $\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} f(x, y) = \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} f(x, y)$.

Lemma 2. *Let M be a real $n \times n$ matrix. Then $M^T M$ is a symmetric matrix with nonnegative eigenvalues. Being symmetric, $M^T M$ has mutually orthogonal eigenvectors. Furthermore, for any two eigenvectors $\boldsymbol{v}_1 \perp \boldsymbol{v}_2$ of $M^T M$ we have $M\boldsymbol{v}_1 \perp M\boldsymbol{v}_2$.*

Proof: M^TM is symmetric because we have $(M^TM)^T = M^T(M^T)^T = M^TM$. For an eigenvector $\boldsymbol{v}$ of M^TM, corresponding to eigenvalue λ, the expression $\boldsymbol{v}^TM^TM\boldsymbol{v}$ can on the one hand be evaluated to $\boldsymbol{v}^T\lambda\boldsymbol{v} = \lambda\|\boldsymbol{v}\|^2$, and on the other hand to $\|M\boldsymbol{v}\|^2 \geq 0$. This proves that $\lambda \geq 0$. Finally, any symmetric matrix has an orthogonal eigensystem. For two different eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ of M^TM, with $\boldsymbol{v}_1 \perp \boldsymbol{v}_2$, the expression $\boldsymbol{v}_1^TM^TM\boldsymbol{v}_2$ can on the one hand be evaluated to $\boldsymbol{v}_1^T\lambda_2\boldsymbol{v}_2 = 0$, and on the other hand to $(M\boldsymbol{v}_1)^T(M\boldsymbol{v}_2)$. This proves $M\boldsymbol{v}_1 \perp M\boldsymbol{v}_2$. □

Lemma 3. *Let $\mathcal{V}$ be a set that is homeomorphic to a (higher-dimenional) ball. Let $\partial\mathcal{V}$ be the boundary of $\mathcal{V}$. Let $f : \mathcal{V} \to \mathcal{V}$ be a differentiable function such that $\partial\mathcal{V}$ is surjectively mapped to $\partial\mathcal{V}$. Then f is surjective.*

Proof sketch: A differentiable function that surjectively maps the edge $\partial\mathcal{V}$ to itself can deform existing holes in $\mathcal{V}$ but cannot create new holes. Since $\mathcal{V}$ does not contain any holes, neither does $f(\mathcal{V})$. □

Lemma 4 (Arithmetic Mean - Geometric Mean (AM-GM) inequality). *For any $n \in \mathbb{N}$ and any list $x_1, x_2, \ldots, x_n$ of nonnegative real numbers it holds that $\frac{1}{n}\sum_{i=1}^{n} x_i \geq \sqrt[n]{x_1x_2\ldots x_n}$.*

3 Analysis of the Asymptotic Fingerprinting Game

3.1 Continuum Limit of the Attack Strategy

As in [6] we assume that the attack strategy satisfies the following condition in the limit $c \to \infty$. There exists a set of bounded and twice differentiable functions $g_y : [0,1]^q \to [0,1]$, with $y \in \mathcal{Q}$, such that

1. $g_\alpha(\boldsymbol{\sigma}/c) = \theta_{\alpha|\boldsymbol{\sigma}}$ for all α, $\boldsymbol{\sigma}$
2. $x_\alpha = 0$ implies $g_\alpha(\boldsymbol{x}) = 0$
3. $\sum_\alpha x_\alpha = 1$ implies $\sum_\alpha g_\alpha(\boldsymbol{x}) = 1$.

3.2 Mutual Information

We introduce the notation $\tau_{y|\boldsymbol{p}} \triangleq \text{Prob}[Y = y|\boldsymbol{P} = \boldsymbol{p}] = \sum_{\boldsymbol{\sigma}} \theta_{y|\boldsymbol{\sigma}}\Lambda_{\boldsymbol{\sigma}|\boldsymbol{p}} = \mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}\left[\theta_{y|\boldsymbol{\Sigma}}\right]$. The mutual information can then be expressed as:

$$I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P}) = \sum_y \sum_{\boldsymbol{\sigma}} \theta_{y|\boldsymbol{\sigma}}\Lambda_{\boldsymbol{\sigma}|\boldsymbol{p}} \log_q\left(\frac{\theta_{y|\boldsymbol{\sigma}}}{\tau_{y|\boldsymbol{p}}}\right) \tag{6}$$

where we take the base-q logarithm because we measure information in q-ary symbols. Using the continuum assumption on the strategy we can write

$$I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) = \sum_y \sum_{\boldsymbol{\sigma}} \Lambda_{\boldsymbol{\sigma}|\boldsymbol{p}} g_y(\frac{\boldsymbol{\sigma}}{c}) \log_q\left(\frac{g_y(\boldsymbol{\sigma}/c)}{\mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}\left[g_y(\boldsymbol{\Sigma}/c)\right]}\right). \tag{7}$$

3.3 Taylor Approximation and the Asymptotic Fingerprinting Game

For large c, the multinomial-distributed variable $\boldsymbol{\Sigma}$ tends towards its mean $c\boldsymbol{p}$ with shrinking relative variance. Therefore we do a Taylor expansion[2] of $\boldsymbol{g}$ around the point $\frac{\boldsymbol{\sigma}}{c} = \boldsymbol{p}$:

$$g_y\Big(\frac{\boldsymbol{\sigma}}{c}\Big) = g_y(\boldsymbol{p}) + \frac{1}{c}\sum_\alpha \frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha}(\sigma_\alpha - cp_\alpha) + \frac{1}{2c^2}\sum_{\alpha\beta}(\sigma_\alpha - cp_\alpha)(\sigma_\beta - cp_\beta)\frac{\partial^2 g_y(\boldsymbol{p})}{\partial p_\alpha \partial p_\beta} + \dots \tag{8}$$

We introduce the notation K for the (scaled) covariance matrix of the multinomial-distributed $\boldsymbol{\Sigma}$,

$$K_{\alpha\beta} \triangleq \frac{1}{c}\mathrm{Cov}\left(\Sigma_\alpha, \Sigma_\beta\right) = \delta_{\alpha\beta}p_\alpha - p_\alpha p_\beta. \tag{9}$$

For $\tau_{y|\boldsymbol{p}}$ we then get

$$\tau_{y|\boldsymbol{p}} = \mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{p}}\left[g_y\left(\frac{\boldsymbol{\Sigma}}{c}\right)\right] = g_y(\boldsymbol{p}) + \frac{1}{2c}\sum_{\alpha\beta} K_{\alpha\beta}\frac{\partial^2 g_y(\boldsymbol{p})}{\partial p_\alpha \partial p_\beta} + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right). \tag{10}$$

The term containing the 1st derivative disappears because $\mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{p}}\left[\boldsymbol{\Sigma} - c\boldsymbol{p}\right] = 0$. The $\mathcal{O}(1/c\sqrt{c})$ comes from the fact that $\left(\boldsymbol{\Sigma} - c\boldsymbol{p}\right)^n$ with $n \geq 2$ yields a result of order $c^{n/2}$ when the expectation over $\boldsymbol{\Sigma}$ is taken. Now we have all the ingredients to do an expansion of $I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$ in terms of powers of $\frac{1}{c}$. The details are given in Appendix 5.

$$I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p}) = \frac{T(\boldsymbol{p})}{2c \ln q} + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{11}$$

$$T(\boldsymbol{p}) \triangleq \sum_y \frac{1}{g_y(\boldsymbol{p})}\sum_{\alpha\beta} K_{\alpha\beta}\frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha}\frac{\partial g_y(\boldsymbol{p})}{\partial p_\beta}. \tag{12}$$

Note that $T(\boldsymbol{p})$ can be related to Fisher Information.[3] The asymptotic fingerprinting game for $c \to \infty$ can now be stated as

$$C_c(q) = \frac{1}{2c^2 \ln q}\max_F \min_{\boldsymbol{g}} \int F(\boldsymbol{p})T(\boldsymbol{p})d^q\boldsymbol{p}. \tag{13}$$

[2] Some care must be taken in using partial derivatives $\partial/\partial p_\beta$ of $\boldsymbol{g}$. The use of $\boldsymbol{g}$ as a continuum limit of $\boldsymbol{\theta}$ is introduced on the hyperplane $\sum_\alpha p_\alpha = 1$, but writing down a derivative forces us to define $\boldsymbol{g}(\boldsymbol{p})$ outside the hyperplane as well. We have a lot of freedom to do so, which we will exploit in Section 3.5.

[3] We can write $T(\boldsymbol{p}) = \mathrm{Tr}[K(\boldsymbol{p})\mathcal{I}(\boldsymbol{p})]$, with $\mathcal{I}$ the Fisher information of Y conditioned on the $\boldsymbol{p}$ vector, $\mathcal{I}_{\alpha\beta}(\boldsymbol{p}) \triangleq \sum_y g_y(\boldsymbol{p})\left(\frac{\partial \ln g_y(\boldsymbol{p})}{\partial p_\alpha}\right)\left(\frac{\partial \ln g_y(\boldsymbol{p})}{\partial p_\beta}\right)$.

3.4 Change of Variables

Substitution of K (9) into (12) gives

$$T(\boldsymbol{p}) = \sum_y \frac{1}{g_y(\boldsymbol{p})} \left\{ \sum_\alpha p_\alpha \left(\frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha} \right)^2 - \left(\sum_\alpha p_\alpha \frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha} \right)^2 \right\}. \tag{14}$$

Now we make a change of variables $p_\alpha = u_\alpha^2$ and $g_\alpha(\boldsymbol{p}) = \gamma_\alpha^2(\boldsymbol{u})$, with $u_\alpha \in [0,1]$, $\gamma_\alpha(\boldsymbol{u}) \in [0,1]$. The hyperplane $\sum_\alpha p_\alpha = 1$ becomes the hypersphere $\sum_\alpha u_\alpha^2 = 1$. For $\boldsymbol{u}$ on the hypersphere we must have $\sum_\alpha \gamma_\alpha^2(\boldsymbol{u}) = 1$. Due to the Marking Assumption, $u_\alpha = 0$ implies $\gamma_\alpha(\boldsymbol{u}) = 0$. The change of variables induces the probability distribution $\Phi(\boldsymbol{u})$ on the variable $\boldsymbol{u}$,

$$\Phi(\boldsymbol{u}) \triangleq F(\boldsymbol{p}(\boldsymbol{u})) \prod_\alpha (2u_\alpha). \tag{15}$$

In terms of the new variables we have a much simplified expression,

$$T(\boldsymbol{u}) = \sum_y \left\{ \|\nabla \gamma_y\|^2 - (\boldsymbol{u} \cdot \nabla \gamma_y)^2 \right\}. \tag{16}$$

where ∇ stands for the gradient $\partial / \partial \boldsymbol{u}$.

3.5 Choosing $\boldsymbol{\gamma}$ Outside the Hypersphere

The function $\boldsymbol{g}(\boldsymbol{p})$ was introduced on the hyperplane $\sum_\alpha p_\alpha = 1$, but taking derivatives $\partial / \partial p_\alpha$ forces us to define $\boldsymbol{g}$ elsewhere too. In the new variables this means we have to define $\boldsymbol{\gamma}(\boldsymbol{u})$ not only on the hypersphere 'surface' $\|\boldsymbol{u}\| = 1$ but also just outside of this surface. Any choice will do, as long as it is sufficiently smooth. A very useful choice is to make $\boldsymbol{\gamma}$ independent of $\|\boldsymbol{u}\|$, i.e. dependent only on the 'angular' coordinates in the surface. Then we have the nice property $\boldsymbol{u} \cdot \nabla \gamma_y = 0$ for all $y \in \mathcal{Q}$, so that (16) simplifies to

$$T(\boldsymbol{u}) = \sum_\alpha \|\nabla \gamma_\alpha\|^2 \tag{17}$$

and the asymptotic fingerprinting game to

$$C_c(q) = \frac{1}{2c^2 \ln q} \max_\Phi \min_{\boldsymbol{\gamma}} \int \Phi(\boldsymbol{u}) T(\boldsymbol{u}) d^q \boldsymbol{u}. \tag{18}$$

3.6 Huang and Moulin's Next Step

At this point [6] proceeds by applying the Cauchy-Schwartz inequality in a very clever way. In our notation this gives

$$\max_\Phi \min_{\boldsymbol{\gamma}} \int \Phi(\boldsymbol{u}) T(\boldsymbol{u}) d^q \boldsymbol{u} \geq \max_\Phi \frac{1}{\int \frac{1}{\Phi(\boldsymbol{u})} d^q \boldsymbol{u}} \min_{\boldsymbol{\gamma}} [\int \sqrt{T(\boldsymbol{u})} d^q \boldsymbol{u}]^2, \tag{19}$$

with equality when T is proportional to $1/\Phi^2$. For the binary alphabet ($q = 2$), the integral $\int \sqrt{T(\boldsymbol{u})} d^q\boldsymbol{u}$ becomes a known constant independent of the strategy $\boldsymbol{\gamma}$. That causes the minimization over $\boldsymbol{\gamma}$ to disappear: The equality in (19) can then be achieved and the entire game can be solved, yielding the arcsine bias distribution and interleaving attack as the optimum. For $q \geq 3$, however, the integral becomes dependent on the strategy $\boldsymbol{\gamma}$, and the steps of [6] cannot be applied.

4 Asymptotic Solution of the Alternative Game

Our aim is to solve the alternative game to (18), see Section 2.5.

$$C_c(q) = \frac{1}{2c^2 \ln q} \min_{\boldsymbol{\gamma}} \max_{\boldsymbol{u}} T(\boldsymbol{u}). \tag{20}$$

First we prove a lower bound on $\max_{\boldsymbol{u}} T(\boldsymbol{u})$ for any strategy $\boldsymbol{\gamma}$. Then we show the existence of a strategy which attains this lower bound. The first part of the proof is stated in the following theorem.

Theorem 1. *For any strategy $\boldsymbol{\gamma}$ satisfying the Marking Assumption ($u_\alpha = 0 \implies \gamma_\alpha(\boldsymbol{u}) = 0$) and conservation of probability ($\|\boldsymbol{u}\| = 1 \implies \|\boldsymbol{\gamma}(\boldsymbol{u})\| = 1$) the following inequality holds:*

$$\max_{\boldsymbol{u}:\, \boldsymbol{u}\geq 0, \|\boldsymbol{u}\|=1} T(\boldsymbol{u}) \geq q - 1. \tag{21}$$

Proof: We start with the definition of the Jacobian matrix $J(\boldsymbol{u})$:

$$J_{\alpha\beta}(\boldsymbol{u}) \triangleq \frac{\partial \gamma_\alpha(\boldsymbol{u})}{\partial u_\beta}. \tag{22}$$

In this way we can write:

$$T(\boldsymbol{u}) = \mathrm{Tr}(J^T J). \tag{23}$$

The matrix J has rank at most $q-1$, because of our choice $\boldsymbol{u} \cdot \nabla \gamma_y = 0$ which can be rewritten as $J\boldsymbol{u} = 0$. That implies that the rank of $J^T J$ is also at most $q-1$. Let $\lambda_1(\boldsymbol{u}), \lambda_2(\boldsymbol{u}), \ldots, \lambda_{q-1}(\boldsymbol{u})$ be the nonzero eigenvalues of $J^T J$. Then

$$T(\boldsymbol{u}) = \sum_{i=1}^{q-1} \lambda_i(\boldsymbol{u}). \tag{24}$$

Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{q-1}$ be the unit-length eigenvectors of $J^T J$ and let $\mathrm{d}\boldsymbol{u}_{(1)}$, $\mathrm{d}\boldsymbol{u}_{(2)}$, …, $\mathrm{d}\boldsymbol{u}_{(q-1)}$ be infinitesimal displacements in the directions of these eigenvectors, i.e. $\mathrm{d}\boldsymbol{u}_{(i)} \propto \boldsymbol{v}_i$. According to Lemma 2 the eigenvectors are mutually orthogonal. Thus we can write the $(q-1)$-dimensional 'surface' element $\mathrm{d}S_{\boldsymbol{u}}$ of the hypersphere in terms of these displacements:

$$\mathrm{d}S_{\boldsymbol{u}} = \prod_{i=1}^{q-1} \|\mathrm{d}\boldsymbol{u}_{(i)}\|. \tag{25}$$

Any change $\mathrm{d}\boldsymbol{u}$ results in a change $\mathrm{d}\boldsymbol{\gamma} = J\mathrm{d}\boldsymbol{u}$. Hence we have $\mathrm{d}\boldsymbol{\gamma}_{(i)} = J\mathrm{d}\boldsymbol{u}_{(i)}$. By Lemma 2, the displacements $\mathrm{d}\boldsymbol{\gamma}_{(1)}, \mathrm{d}\boldsymbol{\gamma}_{(2)}, \ldots, \mathrm{d}\boldsymbol{\gamma}_{(q-1)}$ are mutually orthogonal and we can express the $(q-1)$-dimensional 'surface' element $\mathrm{d}S_{\boldsymbol{\gamma}}$ as

$$\mathrm{d}S_{\boldsymbol{\gamma}} = \prod_{i=1}^{q-1} \|\mathrm{d}\boldsymbol{\gamma}_{(i)}\| = \prod_{i=1}^{q-1} \sqrt{\|J\mathrm{d}\boldsymbol{u}_{(i)}\|^2} \tag{26}$$

$$= \prod_{i=1}^{q-1} \sqrt{\mathrm{d}\boldsymbol{u}_{(i)}^T J^T J \mathrm{d}\boldsymbol{u}_{(i)}} = \prod_{i=1}^{q-1} \|\mathrm{d}\boldsymbol{u}_{(i)}\| \sqrt{\lambda_i} \tag{27}$$

$$= \mathrm{d}S_{\boldsymbol{u}} \prod_{i=1}^{q-1} \sqrt{\lambda_i}. \tag{28}$$

We define the spatial average over $\boldsymbol{u}$ as $\mathrm{Av}_u[f(\boldsymbol{u})] \triangleq \int f(\boldsymbol{u})\ \mathrm{d}S_u / \int \mathrm{d}S_u$. We then have

$$\mathrm{Av}_u[\sqrt{\lambda_1\lambda_2\ldots\lambda_{q-1}}] = \frac{\int dS_{\boldsymbol{u}} \sqrt{\lambda_1\lambda_2\ldots\lambda_{q-1}}}{\int dS_{\boldsymbol{u}}} = \frac{\int dS_{\boldsymbol{\gamma}}}{\int dS_{\boldsymbol{u}}} \geq 1 \tag{29}$$

where the inequality follows from Lemma 3 applied to the mapping $\boldsymbol{\gamma}(\boldsymbol{u})$. (The hypersphere orthant $\|\boldsymbol{u}\| = 1$, $\boldsymbol{u} \geq 0$ is closed and contains no holes; the $\boldsymbol{\gamma}$ was defined as being twice differentiable; the edge of the hypersphere orthant is given by the pieces where $u_i = 0$ for some i; these pieces are mapped to themselves due to the Marking Assumption. The edges of the edges are obtained by setting further components of $\boldsymbol{u}$ to zero, etc. Each of these sub-edges is also mapped to itself due to the Marking Assumption. In the one-dimensional sub-sub-edge we apply the intermediate value theorem, which proves surjectivity. From there we recursively apply Lemma 3 to increasing dimensions, finally reaching dimension $q-1$).

Since the spatial average is greater than or equal to 1 there must exist a point $\boldsymbol{u}_*$ where $\sqrt{\lambda_1(\boldsymbol{u}_*)\lambda_2(\boldsymbol{u}_*)\ldots\lambda_{q-1}(\boldsymbol{u}_*)} \geq 1$. Now we apply Lemma 4,

$$T(\boldsymbol{u}_*) = \sum_{i=1}^{q-1} \lambda_i(\boldsymbol{u}_*) \geq (q-1)\ \sqrt[q-1]{\lambda_1(\boldsymbol{u}_*)\lambda_2(\boldsymbol{u}_*)\ldots\lambda_{q-1}(\boldsymbol{u}_*)} \geq q-1. \tag{30}$$

The last inequality holds since $\sqrt{x} \geq 1$ implies $\sqrt[q-1]{x} \geq 1$. Finally $\max_{\boldsymbol{u}} T(\boldsymbol{u}) \geq T(\boldsymbol{u}_*) \geq q-1$. □

Next we show the existence of a strategy which attains this lower bound.

Theorem 2. *Let the interleaving attack $\boldsymbol{\gamma}$ be extended beyond the hypersphere $\|\boldsymbol{u}\| = 1$ as $\gamma_y(\boldsymbol{u}) = \frac{u_y}{\|u\|}$, satisfying $\boldsymbol{u} \cdot \nabla\gamma_y = 0$ for all y. For the interleaving attack we then have $T(\boldsymbol{u}) = q-1$ for all $\boldsymbol{u} \geq 0, \|\boldsymbol{u}\| = 1$.*

Proof:

$$\frac{\partial \gamma_y(\boldsymbol{u})}{\partial u_\alpha} = \frac{\delta_{y\alpha}}{\|u\|} - \frac{u_y u_\alpha}{\|u\|^3}. \tag{31}$$

$$\begin{aligned} T(\boldsymbol{u}) &= \sum_y \|\nabla \gamma_y(\boldsymbol{u})\|^2 = \sum_y \sum_\alpha \left(\frac{\delta_{y\alpha}}{\|u\|} - \frac{u_y u_\alpha}{\|u\|^3} \right)^2 \\ &= \sum_y \left\{ \frac{1}{\|u\|^2} - \frac{u_y^2}{\|u^4\|} \right\} = \frac{q-1}{\|u\|^2} \end{aligned} \tag{32}$$

where we used the property $\delta_{y\alpha}^2 = \delta_{y\alpha}$. For $\|u\| = 1$ it follows that $T(\boldsymbol{u}) = q - 1$. □

These two theorems together give the solution of the min-max game (20). The main result of this paper is stated in the following theorem:

Theorem 3. *The asymptotic fingerprinting capacity $C_c^\infty(q)$ in the limit $c \to \infty$ for an alphabet of size q is given by*

$$C_c^\infty(q) = \frac{q-1}{2c^2 \ln q}. \tag{33}$$

Proof: For any strategy $\boldsymbol{\gamma}$, Theorem 1 shows that $\max_{\boldsymbol{u}} T(\boldsymbol{u}) \geq q - 1$. As shown in Theorem 2, the interleaving attack has $T(\boldsymbol{u}) = q - 1$ independent of $\boldsymbol{u}$, demonstrating that the equality in Theorem 1 can be satisfied. Hence

$$\min_{\boldsymbol{\gamma}} \max_{\boldsymbol{u}} T(\boldsymbol{u}) = q - 1 \tag{34}$$

is the solution of the min-max game. By Sion's theorem this is also the pay-off solution to the max-min game, as shown in Section 2.5. Substitution into (20) yields the final result. □

Remark: When the attack strategy is interleaving, all distribution functions $\Phi(\boldsymbol{u})$ are equivalent in the expression $\int \Phi(\boldsymbol{u}) T(\boldsymbol{u}) \mathrm{d}^q \boldsymbol{u}$, since $T(\boldsymbol{u})$ then is constant, yielding $\int \Phi(\boldsymbol{u})(q-1)\mathrm{d}^q \boldsymbol{u} = q-1$. We emphasize again that the *min-max* solution gives no information about the optimal $\boldsymbol{\gamma}$ and Φ in the *max-min* game.

5 Discussion

We have proven that the asymptotic channel capacity is $C_c^\infty(q) = \frac{q-1}{c^2 2 \ln q}$. This is an increasing function of q; hence there is an advantage in choosing a large alphabet whenever the details of the watermarking system allow it.

The capacity is an upper bound on the achievable rate of (reliable) codes, where the rate measures which fraction of the occupied 'space' confers actual information. The higher the fraction, the better, independent of the nature of the symbols. Thus the rate (and channel capacity) provides a fair comparison between codes that have different q.

The obvious next question is how to construct a q-ary scheme that achieves capacity. We expect that a straightforward generalization of the Amiri-Tardos scheme [1] will do it. Constructions with more practical accusation algorithms, like [14], do not achieve capacity but have already shown that non-binary codes achieve higher rates than their binary counterparts.

When it comes to increasing q, one has to be cautious for various reasons.

- The actually achievable value of q is determined by the watermark embedding technique and the attack mechanism at the signal processing level. Consider for instance a $q = 8$ code implemented in such a way that a q-ary symbol is embedded in the form of three parts (bits) that can be attacked independently. Then the Marking Assumption will no longer hold in the $q = 8$ context, and the 'real' alphabet size is in fact 2.
- A large q can cause problems for accusation schemes that use an accusation sum as defined in [14] or similar. As long as the probability distributions of the accusation sums are approximately Gaussian, the accusation works well. It was shown in [11] that increasing q causes the tails of the probability distribution to slowly become less Gaussian, which is bad for the code rate. On the other hand, the tails become more Gaussian with increasing c. This leads us to believe that for this type of accusation there is an optimal q as a function of c.

The proof technique used in this paper does not reveal the asymptotically optimal bias distribution and attack strategy. This is left as a subject for future work. We expect that the interleaving attack is optimal in the max-min game as well.

Acknowledgements. Discussions with Jan de Graaf, Antonino Simone, Jan-Jaap Oosterwijk, Benne de Weger and Jeroen Doumen are gratefully acknowledged. We thank Teddy Furon for calling our attention to the Fisher Information. This work was done as part of the STW CREST project.

References

1. Amiri, E., Tardos, G.: High rate fingerprinting codes and the fingerprinting capacity. In: ACM-SIAM Symposium on Discrete Algorithms (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. Charpentier, A., Xie, F., Fontaine, C., Furon, T.: Expectation maximization decoding of Tardos probabilistic fingerprinting code. In: Media Forensics and Security 2009, p. 72540 (2009)
5. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley Series in Telecommunications. Wiley & Sons, Chichester (1991)

6. Huang, Y.-W., Moulin, P.: Maximin optimality of the arcsine fingerprinting distribution and the interleaving attack for large coalitions. In: IEEE Workshop on Information Forensics and Security, WIFS (2010)
7. Huang, Y.-W., Moulin, P.: Saddle-point solution of the fingerprinting capacity game under the marking assumption. In: IEEE International Symposium on Information Theory (ISIT) 2009, pp. 2256–2260 (2009)
8. Moulin, P.: Universal fingerprinting: Capacity and random-coding exponents. In: IEEE International Symposium on Information Theory (ISIT) 2008, pp. 220–224 (2008), http://arxiv.org/abs/0801.3837v2
9. Nuida, K., Fujitsu, 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)
10. 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)
11. Simone, A., Škorić, B.: Accusation probabilities in Tardos codes. In: Benelux Workshop on Information and System Security, WISSEC (2010), http://eprint.iacr.org/2010/472
12. Sion, M.: On general minimax theorems. Pacific Journal of Mathematics 8(1), 171–176 (1958)
13. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC 2003, pp. 116–125 (2003)
14. Škorić, B., Katzenbeisser, S., Celik, M.U.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
15. Škorić, B., Katzenbeisser, S., Schaathun, H.G., Celik, M.U.: Tardos fingerprinting codes in the Combined Digit Model. In: IEEE Workshop on Information Forensics and Security (WIFS) 2009, pp. 41–45 (2009)
16. 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)
17. Xie, F., Furon, T., Fontaine, C.: On-off keying modulation and Tardos fingerprinting. In: MM&Sec 2008, pp. 101–106 (2008)

Appendix: Taylor Expansion of $I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$

We compute the leading order term of $I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$ from (7) with respect to powers of $\frac{1}{c}$. We write $\log_q g_y = \ln g_y / \ln q$ and, using (8), $\ln g_y(\boldsymbol{\sigma}/c) = \ln[g_y(\boldsymbol{p}) + \epsilon_y] = \ln g_y(\boldsymbol{p}) + \ln(1 + \epsilon_y / g_y(\boldsymbol{p}))$, where we have introduced the shorthand notation

$$\epsilon_y \triangleq \frac{1}{c} \sum_{\alpha} \frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha} (\Sigma_\alpha - c p_\alpha) + \frac{1}{2c^2} \sum_{\alpha\beta} (\Sigma_\alpha - c p_\alpha)(\Sigma_\beta - c p_\beta) \frac{\partial^2 g_y(\boldsymbol{p})}{\partial p_\alpha \partial p_\beta} + \dots \quad (35)$$

Higher derivative terms are omitted since they contain higher powers of $1/c$ (even after the expectation over $\boldsymbol{\Sigma}$ is taken). Next we apply the Taylor expansion $\ln(1+x) = x - \frac{x^2}{2} + \cdots$, resulting in

$$\ln g_y(\frac{\boldsymbol{\Sigma}}{c}) = \ln g_y(\boldsymbol{p}) + \frac{\epsilon_y}{g_y(\boldsymbol{p})} - \frac{\epsilon_y^2}{2 g_y^2(\boldsymbol{p})} + \dots \quad (36)$$

where we stop after the second order term since that is already of order $\frac{1}{c}$ when we take the expectation over $\boldsymbol{\Sigma}$. Using (10) we get

$$\ln \tau_{y|\boldsymbol{p}} = \ln g_y(\boldsymbol{p}) + \frac{\zeta_y}{g_y(\boldsymbol{p})} + \ldots, \tag{37}$$

$$\zeta_y \triangleq \frac{1}{2c} \sum_{\alpha\beta} K_{\alpha\beta} \frac{\partial^2 g_y(\boldsymbol{p})}{\partial p_\alpha \partial p_\beta} + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{38}$$

Now we combine all the ingredients,

$$g_y\left(\frac{\boldsymbol{\Sigma}}{c}\right) \ln\left(\frac{g_y\left(\frac{\boldsymbol{\Sigma}}{c}\right)}{\tau_{y|\boldsymbol{p}}}\right) = (g_y(\boldsymbol{p}) + \epsilon_y + \ldots)\left(\frac{\epsilon_y - \zeta_y}{g_y(\boldsymbol{p})} - \frac{\epsilon_y^2}{2g_y^2(\boldsymbol{p})} + \ldots\right) \tag{39}$$

where in the first factor we stop at ϵ_y because when the expectation over $\boldsymbol{\Sigma}$ is applied, ϵ_y^2 gives at least a factor of $\frac{1}{c}$ and the terms in the second factor give at least a factor of $\frac{1}{\sqrt{c}}$.

Now $\mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}[\epsilon_y - \zeta_y] = 0$ because $\mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}[\boldsymbol{\Sigma} - c\boldsymbol{p}] = 0$ and ζ_y was defined as the expectation over $\boldsymbol{\Sigma}$ of the second term in (35). The expectation of the product $\mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}[\epsilon_y \zeta_y]$ is of order $\frac{1}{c^2}$ and so we drop it as well. The only remaining part of order $\frac{1}{c}$ in (39) is $\frac{\epsilon_y^2}{2g_y(\boldsymbol{p})}$ and hence we end up with:

$$I(Y; \boldsymbol{\Sigma} \mid \boldsymbol{P} = \boldsymbol{p})$$

$$= \frac{1}{2\ln q} \sum_y \frac{1}{g_y(\boldsymbol{p})} \mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}\left[\epsilon_y^2\right] + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{40}$$

$$= \frac{1}{2c^2 \ln q} \sum_y \frac{1}{g_y(\boldsymbol{p})} \mathbb{E}_{\boldsymbol{\Sigma}|\boldsymbol{P}=\boldsymbol{p}}\left[\left(\sum_\alpha \frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha}(\Sigma_\alpha - cp_\alpha)\right)^2\right] + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{41}$$

$$= \frac{1}{2c \ln q} \sum_y \frac{1}{g_y(\boldsymbol{p})} \sum_{\alpha\beta} K_{\alpha\beta} \frac{\partial g_y(\boldsymbol{p})}{\partial p_\alpha} \frac{\partial g_y(\boldsymbol{p})}{\partial p_\beta} + \mathcal{O}\left(\frac{1}{c\sqrt{c}}\right) \tag{42}$$

where in the second step we expanded ϵ_y^2 and took the square of only the first term in (35) because the other combination of terms give rise to higher powers of $\frac{1}{c}$.

Asymptotically False-Positive-Maximizing Attack on Non-binary Tardos Codes

Antonino Simone and Boris Škorić

Eindhoven University of Technology

Abstract. We use a method recently introduced by us to study accusation probabilities for non-binary Tardos fingerprinting codes. We generalize the pre-computation steps in this approach to include a broad class of collusion attack strategies. We analytically derive properties of a special attack that asymptotically maximizes false accusation probabilities. We present numerical results on sufficient code lengths for this attack, and explain the abrupt transitions that occur in these results.

1 Introduction

1.1 Collusion Attacks against Forensic Watermarking

Watermarking provides a means for tracing the origin and distribution of digital data. Before distribution of digital content, the content is modified by applying an imperceptible watermark (WM), embedded using a watermarking algorithm. Once an unauthorized copy of the content is found, it is possible to trace those users who participated in its creation. This process is known as 'forensic watermarking'. Reliable tracing requires resilience against attacks that aim to remove the WM. Collusion attacks, where a group of pirates cooperate, are a particular threat: differences between their versions of the content tell them where the WM is located. Coding theory has produced a number of collusion-resistant codes. The resulting system has two layers [5,9]: The coding layer determines which message to embed and protects against collusion attacks. The underlying watermarking layer hides symbols of the code in segments of the content. The interface between the layers is usually specified in terms of the *Marking Assumption* plus additional assumptions that are referred to as a 'model'. The Marking Assumption states that the colluders are able to perform modifications only in those segments where they received different WMs. These segments are called detectable positions. The 'model' specifies the kind of symbol manipulations that the attackers are able to perform *in detectable positions*. In the *Restricted Digit Model* (RDM) the attackers must choose one of the symbols that they have received. The *unreadable digit model* also allows for erasures. In the *arbitrary digit model* the attackers can choose arbitrary symbols, while the *general digit model* additionally allows erasures.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 14–27, 2011.

1.2 Tardos Codes

Many collusion resistant codes have been proposed in the literature. Most notable are the Boneh-Shaw construction [3] and the by now famous Tardos code [12]. The former uses a concatenation of an inner code with a random outer code, while the latter one is a fully randomized binary code. In Tardos' original paper [12] a binary code was given achieving length $m = 100c_0^2\lceil \ln \frac{1}{\varepsilon_1} \rceil$, along with a proof that $m \propto c_0^2$ is asympotically optimal for large coalitions, for all alphabet sizes. Here c_0 denotes the number of colluders to be resisted, and ε_1 is the maximum allowed probability of accusing a fixed innocent user. Tardos' original construction had two unfortunate design choices which caused the high proportionality constant 100. (i) The false negative probability ε_2 (not accusing any attacker) was set as $\varepsilon_2 = \varepsilon_1^{c_0/4}$, even though $\varepsilon_2 \ll \varepsilon_1$ is highly unusual in the context of content distribution; a deterring effect is achieved already at $\varepsilon_2 \approx \frac{1}{2}$, while ε_1 needs to be very small. In the subsequent literature (e.g. [15,2]) the ε_2 was decoupled from ε_1, substantially reducing m. (ii) The symbols 0 and 1 were not treated equally. Only segments where the attackers produce a 1 were taken into account. This ignores 50% of all information. A fully symbol-symmetric version of the scheme was given in [13], leading to a further improvement of m by a factor 4. A further improvement was achieved in [8]. The code construction contains a step where a bias parameter is randomly set for each segment. In Tardos' original construction the probability density function (pdf) for the bias is a continuous function. In [8] a class of discrete distributions was given that performs better than the original pdf against finite coalition sizes. In [16,14] the Marking Assumption was relaxed, and the accusation algorithm of the nonbinary Tardos code was modified to effectively cope with signal processing attacks such as averaging and addition of noise.

All the above mentioned work followed the so-called 'simple decoder' approach, i.e. an accusation score is computed for each user, and if it exceeds a certain threshold, he is considered suspicious. One can also use a 'joint decoder' which computes scores for sets of users. Amiri and Tardos [1] have given a capacity-achieving joint decoder construction for the binary code. (Capacity refers to the information-theoretic treatment [11,7,6] of the attack as a channel.) However, the construction is rather impractical, requiring computations for many candidate coalitions. In [13] the binary construction was generalized to q-ary alphabets, in the simple decoder approach. In the RDM, the transition to a larger alphabet size has benefits beyond the mere fact that a q-ary symbol carries $\log_2 q$ bits of information.

1.3 The Gaussian Approximation

The Gaussian approximation, introduced in [15], is a useful tool in the analysis of Tardos codes. The assumption is that the accusations are normal-distributed. The analysis is then drastically simplified; in the RDM the scheme's performance is almost completely determined by a single parameter, the average accusation $\tilde{\mu}$ of the coalition (per segment). The sufficient code length against a coalition

of size c is $m = (2/\tilde{\mu}^2)c^2 \ln(1/\varepsilon_1)$. The Gaussian assumption is motivated by the Central Limit Theorem (CLT): An accusation score consists of a sum of i.i.d. per-segment contributions. When many of these get added, the result is close to normal-distributed: the pdf is close to Gaussian in a region around the average, and deviates from Gaussian in the tails. The larger m is, the wider this central region. In [15,13] it was argued that in many practical cases the central region is sufficiently wide to allow for application of the Gaussian approximation. In [10] a semi-analytical method was developed for determining the exact shape of the pdf of innocent users' accusations, without simulations. This is especially useful in the case of very low accusation probabilities, where simulations would be very time-consuming. The false accusation probabilities were studied for two attacks: majority voting and interleaving.

1.4 Contributions

We discuss the simple decoder in the RDM, choosing $\varepsilon_2 \approx \frac{1}{2}$. We follow the approach of [10] for computing false accusation probabilities. Our contribution is threefold:

1. We prove a number of theorems (Theorems 1–3) that allow efficient computation of pdfs for more general attacks than the ones treated in [10].

2. We identify which attack minimizes the all-important[1] parameter $\tilde{\mu}$. It was shown in [10] that the majority voting attack achieves this for certain parameter settings, but we consider more general parameter values. We derive some basic properties of the attack.

3. We present numerical results for the $\tilde{\mu}$-minimizing attack. When the coalition is small the graphs contain sharp transitions; we explain these transitions as an effect of the abrupt changes in pdf shape when the attack turns from majority voting into minority voting.

2 Notation and Preliminaries

We briefly describe the q-ary version of the Tardos code as introduced in [13] and the method of [10] to compute innocent accusation probabilities.

2.1 The q-ary Tardos Code

The number of symbols in a codeword is m. The number of users is n. The alphabet is $\mathcal{Q}$, with size q. $X_{ji} \in \mathcal{Q}$ stands for the i'th symbol in the codeword of user j. The whole matrix of codewords is denoted as X.

Two-step code generation. m vectors $\boldsymbol{p}^{(i)} \in [0,1]^q$ are independently drawn according to a distribution F, with

$$F(\boldsymbol{p}) = \delta(1 - \sum_{\beta \in \mathcal{Q}} p_\beta) \cdot \frac{1}{B(\kappa \mathbf{1}_q)} \prod_{\alpha \in \mathcal{Q}} p_\alpha^{-1+\kappa}. \tag{1}$$

[1] Asymptotically for large m, the $\tilde{\mu}$-minimizing attack is the 'worst case' attack in the RDM in the sense that the false accusation probability is maximized.

Here $\mathbf{1}_q$ stands for the vector $(1, \cdots, 1)$ of length q, $\delta(\cdot)$ is the Dirac delta function, and B is the generalized Beta function. κ is a positive constant. For $v_1, \cdots, v_n > 0$ the Beta function is defined as[2]

$$B(\boldsymbol{v}) = \int_0^1 \mathrm{d}x^n \; \delta(1 - \sum_{a=1}^{n} x_a) \prod_{b=1}^{n} x_b^{-1+v_b} = \frac{\prod_{a=1}^{n} \Gamma(v_a)}{\Gamma(\sum_{b=1}^{n} v_b)}. \tag{2}$$

All elements X_{ji} are drawn independently according to $\Pr[X_{ji} = \alpha | \boldsymbol{p}^{(i)}] = p_\alpha^{(i)}$.

Attack. The coalition is $\mathcal{C}$, with size c. The i'th segment of the pirated content contains a symbol $y_i \in \mathcal{Q}$. We define vectors $\boldsymbol{\sigma}^{(i)} \in \mathbb{N}^q$ as

$$\sigma_\alpha^{(i)} \triangleq |\{j \in \mathcal{C} : X_{ji} = \alpha\}| \tag{3}$$

satisfying $\sum_{\alpha \in \mathcal{Q}} \sigma_\alpha^{(i)} = c$. In words: $\sigma_\alpha^{(i)}$ counts how many colluders have received symbol α in segment i. The attack strategy may be probabilistic. As usual, it is assumed that this strategy is column-symmetric, symbol-symmetric and attacker-symmetric. It is expressed as probabilities $\theta_{y|\boldsymbol{\sigma}}$ that apply independently for each segment. Omitting the column index,

$$\Pr[y|\boldsymbol{\sigma}] = \theta_{y|\boldsymbol{\sigma}}. \tag{4}$$

Accusation. The watermark detector sees the symbols y_i. For each user j, the *accusation sum* S_j is computed,

$$S_j = \sum_{i=1}^{m} S_j^{(i)} \quad \text{where} \quad S_j^{(i)} = g_{[X_{ji}==y_i]}(p_{y_i}^{(i)}), \tag{5}$$

where the expression $[X_{ji} == y_i]$ evaluates to 1 if $X_{ji} = y_i$ and to 0 otherwise, and the functions g_0 and g_1 are defined as

$$g_1(p) \triangleq \sqrt{\frac{1-p}{p}} \quad ; \quad g_0(p) \triangleq -\sqrt{\frac{p}{1-p}}. \tag{6}$$

The total accusation of the coalition is $S := \sum_{j \in \mathcal{C}} S_j$. The choice (6) is the unique choice that satisfies

$$p g_1(p) + (1-p) g_0(p) = 0 \quad ; \quad p[g_1(p)]^2 + (1-p)[g_0(p)]^2 = 1. \tag{7}$$

This has been shown to have optimal properties for $q = 2$ [4,15]. Its unique properties (7) also hold for $q \geq 3$; that is the main motivation for using (6). A user is 'accused' if his accusation sum exceeds a threshold Z, i.e. $S_j > Z$.

The parameter $\tilde{\mu}$ is defined as $\frac{1}{m}\mathbb{E}[S]$, where $\mathbb{E}$ stands for the expectation value over all random variables. The $\tilde{\mu}$ depends on q, κ, the collusion strategy, and weakly on c. In the limit of large c it converges to a finite value, and the code length scales as $c^2/\tilde{\mu}^2$.

[2] This is also known as a Dirichlet integral. The ordinary Beta function ($n = 2$) is $B(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x+y)$.

2.2 Marginal Distributions and Strategy Parametrization

Because of the independence between segments, the segment index will be dropped from this point onward. For given $\boldsymbol{p}$, the vector $\boldsymbol{\sigma}$ is multinomial-distributed, $\mathbb{P}(\boldsymbol{\sigma}|\boldsymbol{p}) = \binom{c}{\boldsymbol{\sigma}} \prod_\alpha p_\alpha^{\sigma_\alpha}$. Averaged over $\boldsymbol{p}$, the $\boldsymbol{\sigma}$ has distribution $\mathbb{P}(\boldsymbol{\sigma}) = \binom{c}{\boldsymbol{\sigma}} \frac{B(\kappa \mathbf{1}_q + \boldsymbol{\sigma})}{B(\kappa \mathbf{1}_q)}$. Two important marginals were given in [10]. First, the marginal probability $\mathbb{P}_1(b) \triangleq \Pr[\sigma_\alpha = b]$ for one arbitrary component α,

$$\mathbb{P}_1(b) = \binom{c}{b} \frac{B(\kappa + b, \kappa[q-1] + c - b)}{B(\kappa, \kappa[q-1])}. \tag{8}$$

Second, given that $\sigma_\alpha = b$, the probability that the remaining $q-1$ components of the vector $\boldsymbol{\sigma}$ are given by $\boldsymbol{x}$,

$$\mathbb{P}_{q-1}(\boldsymbol{x}|b) = \binom{c-b}{\boldsymbol{x}} \frac{B(\kappa \mathbf{1}_{q-1} + \boldsymbol{x})}{B(\kappa \mathbf{1}_{q-1})}. \tag{9}$$

It is always implicit that $\sum_{\beta \in \mathcal{Q} \setminus \{\alpha\}} x_\beta = c - b$.

An alternative parametrization was introduced for the collusion strategy, which exploits the fact that (i) $\theta_{\alpha|\boldsymbol{\sigma}}$ is invariant under permutation of the symbols $\neq \alpha$; (ii) $\theta_{\alpha|\boldsymbol{\sigma}}$ depends on α only through the value of σ_α.

$$\Psi_b(\boldsymbol{x}) \triangleq \theta_{\alpha|\boldsymbol{\sigma}} \text{ given that } \sigma_\alpha = b \text{ and } \boldsymbol{x} = \text{the other components of } \boldsymbol{\sigma}. \tag{10}$$

Thus, $\Psi_b(\boldsymbol{x})$ is the probability that the pirates choose a symbol that they have seen b times, given that the other symbols' occurences are $\boldsymbol{x}$. Strategy-dependent parameters K_b were introduced as follows,

$$K_b \triangleq \mathbb{E}_{\boldsymbol{x}|b} \Psi_b(\boldsymbol{x}) = \sum_{\boldsymbol{x}} \mathbb{P}_{q-1}(\boldsymbol{x}|b) \Psi_b(\boldsymbol{x}). \tag{11}$$

Due to the marking assumption $K_0 = 0$ and $K_c = 1$. The K_b obey the sum rule $q \sum_{b=0}^{c} K_b \mathbb{P}_1(b) = 1$. Efficient pre-computation of the K_b parameters can speed up the computation of a number of quantities of interest, among which the $\tilde{\mu}$ parameter. It was shown that $\tilde{\mu}$ can be expressed as

$$\tilde{\mu} = \sum_{\boldsymbol{\sigma}} \mathbb{P}(\boldsymbol{\sigma}) \sum_{\alpha \in \mathcal{Q}} \theta_{\alpha|\boldsymbol{\sigma}} T(\sigma_\alpha) = q \sum_{b=0}^{c} K_b \mathbb{P}_1(b) T(b), \tag{12}$$

where

$$T(b) \triangleq \left\{ \tfrac{1}{2} - \kappa + \frac{b}{c}(\kappa q - 1) \right\} c \frac{\Gamma(b + \kappa - \frac{1}{2})}{\Gamma(b + \kappa)} \frac{\Gamma(c - b + \kappa[q-1] - \frac{1}{2})}{\Gamma(c - b + \kappa[q-1])}. \tag{13}$$

2.3 Method for Computing False Accusation Probabilities

The method of [10] is based on the convolution rule for generating functions (Fourier transforms): Let $A_1 \sim f_1$ and $A_2 \sim f_2$ be continuous random variables,

and let $\tilde{f}_1$, $\tilde{f}_2$ be the Fourier transforms of the respective pdfs. Let $A = A_1 + A_2$. Then the easiest way to compute the pdf of A (say Φ) is to use the fact that $\tilde{\Phi}(k) = \tilde{f}_1(k)\tilde{f}_2(k)$. If m i.i.d. variables $A_i \sim \varphi$ are added, $A = \sum_i A_i$, then the pdf of A is found using $\tilde{\Phi}(k) = [\tilde{\varphi}(k)]^m$. In [10] the pdf φ was derived for an innocent user's one-segment accusation $S_j^{(i)}$. The Fourier transform was found to be

$$\tilde{\varphi}(k) = \frac{2q}{B(\kappa, \kappa[q-1])} \sum_{b=1}^{c} \binom{c}{b} K_b \cdot \left[\Lambda(d_b, v_b; k) + \Lambda(v_b - 1, d_b + 1; -k) \right], \quad (14)$$

with

$$d_b \triangleq b + \kappa \quad ; \quad v_b \triangleq c - b + \kappa[q-1] + 1$$

$$\Lambda(d, v; k) = (-ik)^{2v} \Gamma(-2v)\ {}_1F_2(v+d; v+\tfrac{1}{2}, v+1; \frac{k^2}{4}) + \tfrac{1}{2} \sum_{j=0}^{\infty} \frac{(ik)^j}{j!} B(d+\frac{j}{2}, v-\frac{j}{2}).$$

Using this result for $\tilde{\varphi}$ it is then possible to cast the expression $\tilde{\varphi}^m$ in the following special form,

$$\left[\tilde{\varphi}(\frac{k}{\sqrt{m}}) \right]^m = e^{-\frac{1}{2}k^2} \left[1 + \sum_{t=0}^{\infty} \omega_t(m) (i \operatorname{sgn} k)^{\alpha_t} |k|^{\nu_t} \right], \quad (15)$$

where α_t are real numbers; the coefficients $\omega_t(m)$ are real; the powers ν_t satisfy $\nu_0 > 2$, $\nu_{t+1} > \nu_t$. In general the ν_t are not all integer. The ω_t decrease with increasing m as $m^{-\nu_t/6}$ or faster. Computing all the α_t, ω_t, ν_t up to a certain cutoff $t = t_{\max}$ is straightforward but laborious, and leads to huge expressions if done analytically; it is best done numerically, e.g. using series operations in Mathematica. Once all these coefficients are known, the false accusation probability is computed as follows. Let R_m be a function defined as $R_m(\tilde{Z}) := \Pr[S_j > \tilde{Z}\sqrt{m}]$ (for innocent j). Let Ω be the corresponding function in case the pdf of S_j is Gaussian, $\Omega(\tilde{Z}) = \frac{1}{2}\mathrm{Erfc}(\tilde{Z}/\sqrt{2})$. The $R_m(\tilde{Z})$ is computed by first doing a reverse Fourier transform on $[\tilde{\varphi}(k/\sqrt{m})]^m$ expressed as (15) to find the pdf of the total accusation, and then integrating over all accusation values that exceed the threshold Z. After some algebra [10] the result is

$$R_m(\tilde{Z}) = \Omega(\tilde{Z}) + \frac{1}{\pi} \sum_{t=0}^{\infty} \omega_t(m) \Gamma(\nu_t) 2^{\nu_t/2} \mathrm{Im} \left[i^{-\alpha_t} H_{-\nu_t}(i\tilde{Z}/\sqrt{2}) \right]. \quad (16)$$

Here H is the Hermite function. It holds that $\lim_{m\to\infty} R_m(\tilde{Z}) = \Omega(\tilde{Z})$. For a good numerical approximation it suffices to take terms up to some cutoff $t_{\max}$. The required $t_{\max}$ is a decreasing function of m.

3 Our Results

3.1 Computing K_b for Several Classes of Colluder Strategy

Our first contribution is a prescription for efficiently computing the K_b parameters for more general colluder strategies than those studied in [10]. We consider

the strategy parametrization $\Psi_b(\boldsymbol{x})$ with $b \neq 0$. The vector $\boldsymbol{x} \in \mathbb{N}^{q-1}$ can contain several entries equal to b. The number of such entries will be denoted as ℓ. (The dependence of ℓ on b and $\boldsymbol{x}$ is suppressed in the notation for the sake of brevity.) The number of remaining entries is $r \triangleq q-1-\ell$. These entries will be denoted as $\boldsymbol{z} = (z_1, \cdots, z_r)$, with $z_j \neq b$ by definition. Any strategy possessing the symmetries mentioned in Section 2 can be parametrized as a function $\Psi_b(\boldsymbol{x})$ which in turn can be expressed as a function of b, ℓ and $\boldsymbol{z}$; it is invariant under permutation of the entries in $\boldsymbol{z}$. We will concentrate on the following 'factorizable' classes of attack, each one a sub-class of the previous one.

Class 1: $\Psi_b(\boldsymbol{x})$ is of the form $w(b,\ell)\prod_{k=1}^{r} W(b,\ell,z_k)$

Class 2: $\Psi_b(\boldsymbol{x})$ is of the form $\frac{w(b)}{\ell+1}\prod_{k=1}^{r} W(b,z_k)$

Class 3: $\Psi_b(\boldsymbol{x})$ is of the form $\frac{1}{\ell+1}\prod_{k=1}^{r} W(b,z_k)$, with $W(b,z_k) \in \{0,1\}$ and $W(b,z_k) + W(z_k,b) = 1$. By definition $W(b,0) = 1$.

Class 1 merely restricts the dependence on $\boldsymbol{z}$ to a form factorizable in the components z_k. This is a very broad class, and contains e.g. the interleaving attack ($\theta_{\alpha|\boldsymbol{\sigma}} = \frac{\sigma_\alpha}{c}$, $\Psi_b(\boldsymbol{x}) = \frac{b}{c}$) which has no dependence on $\boldsymbol{z}$.

Class 2 puts a further restriction on the ℓ-dependence. The factor $1/(\ell+1)$ implies that symbols with equal occurrence have equal probability of being selected by the colluders. (There are $\ell+1$ symbols that occur b times.)

Class 3 restricts the function W to a binary 'comparison' of its two arguments: $\Psi_b(\boldsymbol{x})$ is nonzero only if for the attackers b is 'better' than z_k for all k, i.e. $W(b,z_k) = 1$. An example of such a strategy is majority voting, where $\Psi_b(\boldsymbol{x}) = 0$ if there exists a k such that $z_k > b$, and $\Psi_b(\boldsymbol{x}) = \frac{1}{\ell+1}$ if $z_k < b$ for all k. Class 3 also contains minority voting, and in fact any strategy which uses a strict ordering or 'ranking' of the occurrence counters b, z_k. (Here a zero always counts as 'worse' than nonzero.)

Our motivation for introducing classes 1 and 2 is mainly technical, since they affect to which extent the K_b parameters can be computed analytically. In the next section we will see that class 3 captures not only majority/minority voting but also the $\tilde{\mu}$-reducing attack.

Theorem 1. *Let $N_b \in \mathbb{N}$ satisfy $N_b > \max\{c-b, |c-bq|, (c-b)(q-2)\}$. Let $\tau_b \triangleq e^{i2\pi/N_b}$, and let*

$$G_{ba\ell} \triangleq \sum_{z\in\{0,\ldots,c-b\}\setminus\{b\}} \frac{\Gamma(\kappa+z)W(b,\ell,z)}{\tau_b^{az} z!}, \qquad v_{ba} \triangleq \frac{\Gamma(\kappa+b)}{\tau_b^{ab} b!}. \tag{17}$$

Then for strategies in class 1 it holds that

$$K_b = \frac{(c-b)!}{N_b\Gamma(c-b+\kappa[q-1])B(\kappa\mathbf{1}_{q-1})} \sum_{a=0}^{N_b-1} \tau_b^{a(c-b)} \sum_{\ell=0}^{q-1} \binom{q-1}{\ell} G_{ba\ell}^{q-1-\ell} w(b,\ell) v_{ba}^{\ell}.$$

Theorem 2. *For strategies in class 2 the quantity $G_{ba\ell}$ as defined in (17) does not depend on ℓ and can be denoted as G_{ba} (with $W(b,\ell,z)$ replaced by $W(b,z)$). It then holds that*

$$K_b = \frac{b!(c-b)!\; w(b)}{qN_b\Gamma(\kappa+b)\Gamma(c-b+\kappa[q-1])B(\kappa\mathbf{1}_{q-1})} \sum_{a=0}^{N_b-1} \tau_b^{ac}\left[(G_{ba}+v_{ba})^q - G_{ba}^q\right].$$

Theorem 3. *For strategies in class 3, Theorem 2 holds, where* $w(b) = 1$ *and* G_{ba} *can be expressed as*

$$G_{ba} = \sum_{\substack{z\in\{0,\ldots,c-b\}\setminus\{b\} \\ W(b,z)=1}} \frac{\Gamma(\kappa+z)}{\tau_b^{az} z!}. \tag{18}$$

The proofs of Theorems 1–3 are given in the Appendix. Without these theorems, straightforward computation of K_b following (11) would require a full sum over $\boldsymbol{x}$, which for large c comprises $\mathcal{O}(c^{q-2}/(q-1)!)$ different terms. ($q-1$ variables $\leq c-b$, with one constraint, and with permutation symmetry. We neglect the dependence on b.) Theorem 1 reduces the number of terms to $\mathcal{O}(q^2c^2)$ at worst; a factor c from computing G_{ba}, a factor q from $\sum_\ell$ and a factor N_b from $\sum_a$, with $N_b < qc$. In Theorem 2 the ℓ-sum is eliminated, resulting in $\mathcal{O}(qc^2)$ terms.

We conclude that, for $q \geq 5$ and large c, Theorems 1 and 2 can significantly reduce the time required to compute the K_b parameters.[3] A further reduction occurs in Class 3 if the $W(b,z)$ function is zero for many z.

3.2 The $\tilde{\mu}$-Minimizing Attack

Asymptotically for large code lengths the colluder strategy has negligible impact on the Gaussian shape of the innocent (and guilty) accusation pdf. For $q \geq 3$ the main impact of their strategy is on the value of the statistical parameter $\tilde{\mu}$. (For the binary symmetric scheme with $\kappa = \frac{1}{2}$, the $\tilde{\mu}$ is fixed at $\frac{2}{\pi}$; the attackers cannot change it. Then the strategy's impact on the pdf shape is *not* negligible.)

Hence for $q \geq 3$ the strategy that minimizes $\tilde{\mu}$ is asymptotically a 'worst-case' attack in the sense of maximizing the false positive probability. This was already argued in [13], and it was shown how the attackers can minimize $\tilde{\mu}$. From the first expression in (12) it is evident that, for a given $\boldsymbol{\sigma}$, the attackers must choose the symbol y such that $T(\sigma_y)$ is minimal, i.e. $y = \arg\min_\alpha T(\sigma_\alpha)$. In case of a tie it does not matter which of the best symbols is chosen, and without loss of generality we impose symbol symmetry, i.e. if the minimum $T(\sigma_\alpha)$ is shared by N different symbols, then each of these symbols will have probability $1/N$ of being elected. Note that this strategy fits in class 3. The function $W(b,z_k)$ evaluates to 1 if $T(b) < T(z_k)$ and to 0 otherwise.[4]

Let us introduce the notation $x = b/c$, $x \in (0,1)$. Then for large c we have [10]

$$T(cx) \approx \frac{\frac{1}{2} - \kappa + x(\kappa q - 1)}{\sqrt{x(1-x)}}. \tag{19}$$

[3] To get some feeling for the orders of magnitude: The crossover point where $qc^2 = c^{q-2}/(q-1)!$ lies at $c = 120$, 27, 18, 15, 13, for $q =$5, 6, 7, 8, 9 respectively.

[4] For $x, y \in \mathbb{N}$, with $x \neq y$, it does not occur in general that $T(x) = T(y)$. The only way to make this happen is to choose κ in a very special way as a function of q and c. W.l.o.g. we assume that κ is not such a pathological case.

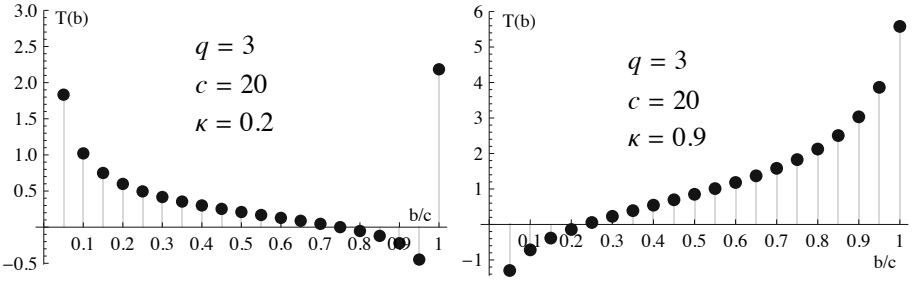


Fig. 1. The function $T(b)$ for $q = 3$, $c = 20$ and two values κ outside $(\frac{1}{2[q-1]}, \frac{1}{2})$

From (19) we deduce some elementary properties of the function T.

- If $\kappa < \frac{1}{2(q-1)}$ then T is monotonically decreasing, and $T(b)$ may become negative at large b.
- If $\kappa > \frac{1}{2}$, then T is monotonically increasing, and $T(b)$ may become negative at small b.
- For κ in between those values, $T(b)$ is nonnegative and has a minimum at $\frac{b}{c} \approx \frac{1}{q-2}(\frac{1}{2\kappa} - 1)$.

We expect that the existence of negative $T(b)$ values has a very bad impact on $\tilde{\mu}$ (from the accuser's point of view), and hence that κ is best chosen in the interval $(\frac{1}{2(q-1)}, \frac{1}{2})$.

Fig. 1 shows the function $T(b)$ for two values of κ outside this 'safe' interval. For $\kappa = 0.2$ it is indeed the case that $T(b) < 0$ at large b, and for $\kappa = 0.9$ at small b. Note that $T(c)$ is always positive due to the Marking Assumption. For small κ, the $T(b)$-ranking of the points is clearly such that majority voting is the best strategy; similarly, for large κ minority voting is best. For intermediate values of κ a more complicated ranking will occur.

3.3 Numerical Results for the $\tilde{\mu}$-Minimizing Attack

In [10] the $\tilde{\mu}$-minimizing attack was studied for a restricted parameter range, $\kappa \approx 1/q$. For such a choice of κ the strategy reduces to majority voting. We study a *broader range*, applying the full $\tilde{\mu}$-minimizing attack. We use Theorem 3 to precompute the K_b and then (14), (15) and (16) to compute the false accusation probability R_m as a function of the accusation threshold. We found that keeping terms in the expansion with $\nu_t \leq 37$ gave stable results.

For a comparison with [10], we set $\varepsilon_1 = 10^{-10}$, and search for the smallest codelength m_* for which it holds that $R_m(\tilde{\mu}\sqrt{m}/c) \leq \varepsilon_1$. The special choice $\tilde{Z} = \tilde{\mu}\sqrt{m}/c$ puts the threshold at the expectation value of a colluder's accusation. As a result the probability of a false negative error is $\approx \frac{1}{2}$. Our results for m_* are consistent with the numbers given in [10].

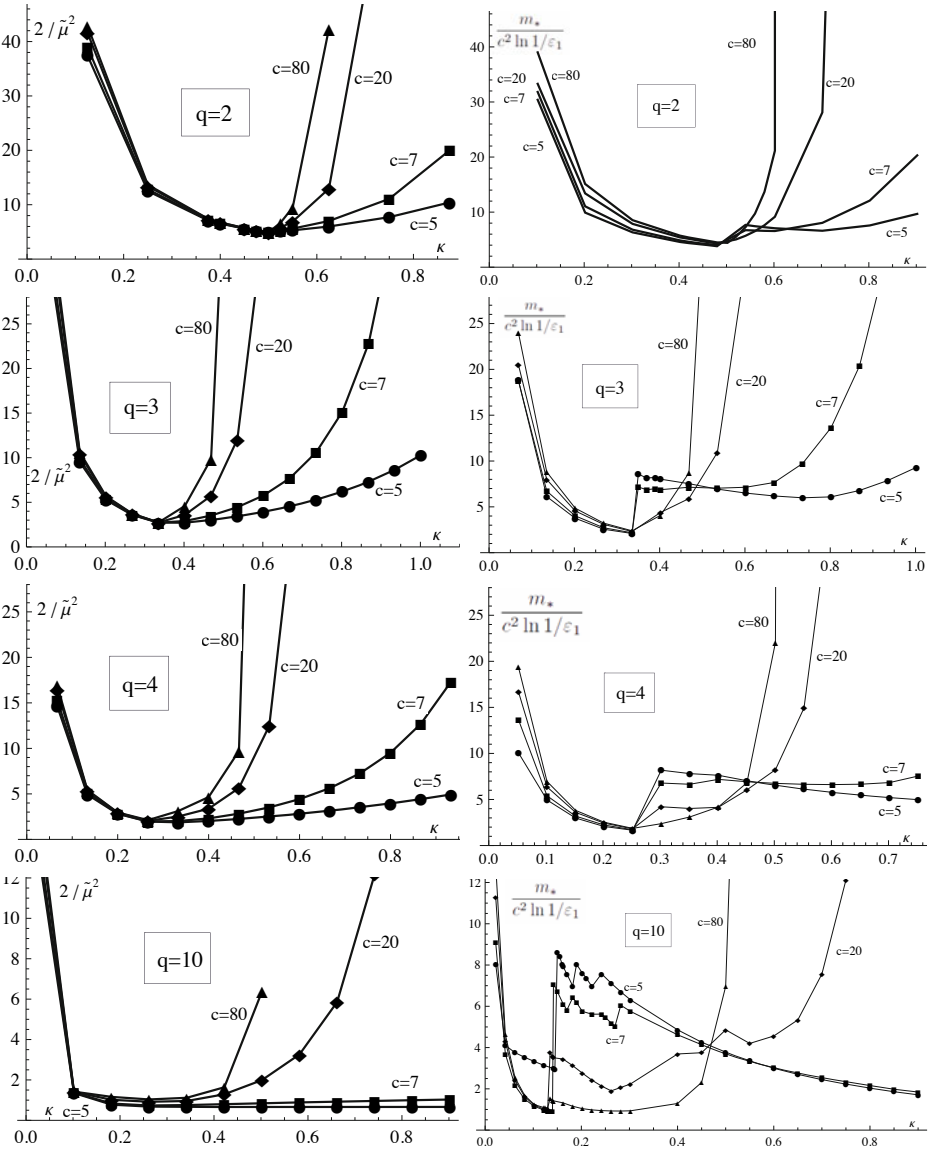


Fig. 2. Numerical results for the $\tilde{\mu}$-minimizing attack. $\varepsilon_1 = 10^{-10}$. **Left:** The Gaussian-limit code length constant $\frac{2}{\tilde{\mu}^2}$ as a function of κ, for various q and c. **Right:** The sufficient code length m_*, scaled by the factor $c^2 \ln(1/\varepsilon_1)$ for easy comparison to the Gaussian limit.

In Fig. 2 we present graphs of $2/\tilde{\mu}^2$ as a function of κ for various q, c.[5] If the accusation pdf is Gaussian, then the quantity $2/\tilde{\mu}^2$ is very close to the proportionality constant in the equation $m \propto c^2 \ln(1/\varepsilon_1)$. We also plot $\frac{m_*}{c^2 \ln(1/\varepsilon_1)}$ as a function of κ for various q, c. Any discrepancy between the $\tilde{\mu}$ and m_* plots is caused by non-Gaussian tail shapes.

In the plots on the left we see that the attack becomes very powerful (very large $2/\tilde{\mu}^2$) around $\kappa = \frac{1}{2}$, especially for large coalitions. This can be understood from the fact that the $T(b)$ values are decreasing, and some even becoming negative for $\kappa > \frac{1}{2}$, as discussed in Section 3.2. This effect becomes weaker when q increases. The plots also show a strong deterioration of the scheme's performance when κ approaches $\frac{1}{2(q-1)}$, as expected.

For small and large κ, the left and right graphs show roughly the same behaviour. In the middle of the κ-range, however, the m_* is very irregular. We think that this is caused by rapid changes in the 'ranking' of b values induced by the function $T(b)$; there is a transition from majority voting (at small κ) to minority voting (at large κ). It was shown in [10] that (i) majority voting causes a more Gaussian tail shape than minority voting; (ii) increasing κ makes the tail more Gaussian. These two effects together explain the m_* graphs in Fig. 2: first, the transition for majority voting to minority voting makes the tail less

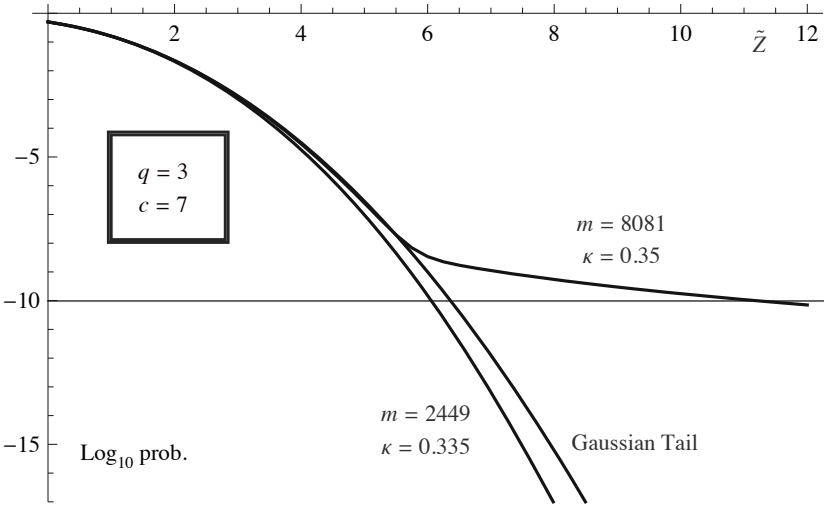


Fig. 3. Accusation probability for a fixed innocent user as a function of the (scaled) accusation threshold $\tilde{Z} = Z/\sqrt{m}$. The attack is the $\tilde{\mu}$-minimizing attack. The graph shows the Gaussian limit, and two parameter settings which correspond to 'before' and 'after' a sharp transition.

[5] The $\tilde{\mu}$ can become negative. These points are not plotted, as they represent a situation where the accusation scheme totally fails, and there exists no sufficient code length m_*.

Gaussian (hence increasing m_*), and then increasing κ gradually makes the tail more Gaussian again (reducing m_*).

In Fig. 3 we show the shape of the false accusation pdf of both sides of the transition in the $q = 3$, $c = 7$ plot. For the smaller κ the curve is better than Gaussian up to false accusation probabilities of better than 10^{-17}. For the larger κ the curve becomes worse than Gaussian around 10^{-8}, which lies significantly above the desired 10^{-10}. The transition from majority to minority voting is cleanest for $q = 2$, and was already shown in [13] to lie precisely at $\kappa = \frac{1}{2}$ for all c. For $q \geq 3$ it depends on c and is less easy to pinpoint.

4 Discussion

We have tested the pdf computation method of [10] for a large range of parameter values and for the various 'rankings' that are part of the $\tilde{\mu}$-minimizing attack. The method has performed well under all these conditions.

Our results reveal the subtle interplay between the average colluder accusation $\tilde{\mu}$ and the shape of the pdf of an innocent user's accusation sum. The sharp transitions that occur in Fig. 2 show that there is a κ-range (to the left of the transition) where the $\tilde{\mu}$-reducing attack is not optimal for small coalitions. It is not yet clear what the optimal attack would be there, but certainly it has to be an attack that concentrates more on the pdf shape than on $\tilde{\mu}$, e.g. the minority voting or the interleaving attack.

For large coalitions the pdfs are very close to Gaussian. From the optimum points m_* as a function of κ we see that it can be advantageous to use an alphabet size $q > 2$ (even if a non-binary symbol occupies $\log_2 q$ times more 'space' in the content than a binary symbol).

The results in this paper specifically pertain to the 'simple decoder' accusation algorithm introduced in [13]. We do not expect that the asymptotically optimal attack on $\tilde{\mu}$ is also optimal against information-theoretic accusations like [1]; there we expect the interleaving attack $\theta_{\alpha|\boldsymbol{\sigma}} = \sigma_\alpha/c$ to be optimal.

Acknowledgements. Discussions with Dion Boesten, Jan-Jaap Oosterwijk, Benne de Weger and Jeroen Doumen are gratefully acknowledged.

References

1. Amiri, E., Tardos, G.: High rate fingerprinting codes and the fingerprinting capacity. In: SODA 2009, pp. 336–345 (2009)
2. Blayer, O., Tassa, T.: Improved versions of Tardos' fingerprinting scheme. Designs, Codes and Cryptography 48(1), 79–103 (2008)
3. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory 44(5), 1897–1905 (1998)
4. Furon, T., Guyader, A., Cérou, F.: On the design and optimization of tardos probabilistic fingerprinting codes. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 341–356. Springer, Heidelberg (2008)

5. He, S., Wu, M.: Joint coding and embedding techniques for multimedia fingerprinting. TIFS 1, 231–248 (2006)
6. Huang, Y.W., Moulin, P.: Saddle-point solution of the fingerprinting capacity game under the marking assumption. In: ISIT 2009 (2009)
7. Moulin, P.: Universal fingerprinting: Capacity and random-coding exponents. Preprint arXiv:0801.3837v2 (2008)
8. 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)
9. Schaathun, H.G.: On error-correcting fingerprinting codes for use with watermarking. Multimedia Systems 13(5-6), 331–344 (2008)
10. Simone, A., Škorić, B.: Accusation probabilities in Tardos codes. In: Benelux Workshop on Information and System Security, WISSEC 2010 (2010), http://eprint.iacr.org/2010/472
11. Somekh-Baruch, A., Merhav, N.: On the capacity game of private fingerprinting systems under collusion attacks. IEEE Trans. Inform. Theory 51, 884–899 (2005)
12. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC 2003, pp. 116–125 (2003)
13. Škorić, B., Katzenbeisser, S., Celik, M.U.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
14. Škorić, B., Katzenbeisser, S., Schaathun, H.G., Celik, M.U.: Tardos fingerprinting codes in the combined digit model. In: IEEE Workshop on Information Forensics and Security (WIFS) 2009, pp. 41–45 (2009)
15. 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)
16. Xie, F., Furon, T., Fontaine, C.: On-off keying modulation and Tardos fingerprinting. In: MM&Sec 2008, pp. 101–106 (2008)

Appendix: Proofs

Proof of Theorem 1

We start from (11), with $\mathbb{P}_{q-1}$ defined in (9), and reorganize the $\boldsymbol{x}$-sum to take the multiplicity ℓ into account:

$$\sum_{\boldsymbol{x}}[\cdots] \to \sum_{\ell=0}^{\ell_{\max}} \binom{q-1}{\ell} \sum_{\boldsymbol{z}\in(\{0,\ldots,c-b\}\setminus\{b\})^r} \delta_{0,c-b(\ell+1)-\sum_{k=1}^r z_k}[\cdots]$$

$$= \sum_{\ell=0}^{\ell_{\max}} \binom{q-1}{\ell} \sum_{z_1\in\{0,\ldots,c-b\}\setminus\{b\}} \cdots \sum_{z_r\in\{0,\ldots,c-b\}\setminus\{b\}} \delta_{0,c-b(\ell+1)-\sum_{k=1}^r z_k}[\cdots]$$

where δ is the Kronecker delta, and $\ell_{\max} = \min\{q-1, \lfloor\frac{c-b}{b}\rfloor\}$. The factor $\binom{q-1}{\ell}$ pops up because the summand in (11) is fully symmetric under permutations of $\boldsymbol{x}$. The Kronecker delta takes care of the constraint that the components of $\boldsymbol{z}$ add up to $c-b-\ell b$.

If $\ell_{\max} = \lfloor\frac{c-b}{b}\rfloor$ and the sum over ℓ is extended beyond $\ell_{\max}$, then all the additional terms are zero, because the Kronecker delta condition cannot be satisfied. (The $\sum_k z_k$ would have to become negative.) Hence we are free to replace

the upper summation bound $\ell_{\max}$ by $q-1$ without changing the result of the sum.

Next we use a sum representation of the Kronecker δ as follows,

$$\delta_{0,s} = \frac{1}{N_b} \sum_{a=0}^{N_b-1} (e^{i2\pi/N_b})^{as}, \tag{20}$$

with $s = c - b(l+1) - \sum_k z_k$. This is a correct representation only if N_b is larger than the maximum $|s|$ that can occur. The most positive possible value of s is attained at $(\ell = 0, \boldsymbol{z} = 0)$, namely $s = c - b$. The most negative value (s_{neg}) is attained when $z_k = c - b$ for all k. Since there are $r = q - 1 - \ell$ components in $\boldsymbol{z}$, we have $s_{\text{neg}} = \min_\ell[c - b(\ell+1) - (q-1-\ell)(c-b)]$. The function is linear in ℓ, so there are only two candidates: the extreme values $\ell = 0$ and $\ell = q-1$, which yield $|s_{\text{neg}}| = (q-2)(c-b)$ and $|s_{\text{neg}}| = |c - bq|$ respectively. Hence N_b has to be larger than $\max\{c-b, (q-2)(c-b), |c-bq|\}$.

Our expression for K_b now contains sums over ℓ, z_k and a. We shift the a-sum completely to the left. Next we write

$$B(\kappa \mathbf{1}_{q-1} + \boldsymbol{x}) = \frac{[\Gamma(\kappa+b)]^\ell \prod_{k=1}^{q-1-\ell} \Gamma(\kappa + z_k)}{\Gamma(c - b + \kappa[q-1])}, \tag{21}$$

$$\binom{c-b}{\boldsymbol{x}} = \frac{(c-b)!}{[b!]^\ell \prod_{k=1}^{q-1-\ell} z_k!}. \tag{22}$$

All the expressions depending on the z_k variables are fully factorized; the part of the summand that contains the z_k is given by

$$\prod_{k=1}^{q-1-\ell} \left[\sum_{z_k \in \{0,\ldots,c-b\} \setminus \{b\}} \frac{W(b,\ell,z_k)\Gamma(\kappa+z_k)}{z_k!\, \tau_b^{a z_k}} \right] = (G_{ba\ell})^{q-1-\ell}. \tag{23}$$

Theorem 1 follows after some elementary rewriting. □

Proof of Theorem 2

We start from K_b as given by Theorem 1. The $G_{ba\ell}$ becomes G_{ba}, so the factor G_{ba}^{q-1} can be moved out of the ℓ-sum. The $w(b,\ell)$ becomes $w(b)/(\ell+1)$ and $w(b)$ can also be moved out of the ℓ-sum. The remaining sum is $\sum_{\ell=0}^{q-1} \binom{q-1}{\ell} \frac{1}{\ell+1} (v_{ba}/G_{ba})^\ell$ which evaluates to $[(G_{ba}+v_{ba})^q - G_{ba}^q] G_{ba}^{1-q}/(q v_{ba})$. Theorem 2 follows after substituting the definition of v_{ba} and some rewriting. □

Proof of Theorem 3

In (17) the $W(b,\ell,z)$ becomes $W(b,z)$. The definition of class 3 specifies that $W(b,z)$ is either 1 or 0. The result (18) trivially follows. □

Towards Joint Tardos Decoding: The 'Don Quixote' Algorithm

Peter Meerwald* and Teddy Furon

INRIA Rennes Bretagne Atlantique,
Campus de Beaulieu, Rennes, France
{peter.meerwald,teddy.furon}@inria.fr

Abstract. 'Don Quixote' is a new accusation process for Tardos traitor tracing codes which is, as far as we know, the first practical implementation of joint decoding. The first key idea is to iteratively prune the list of potential colluders to keep the computational effort tractable while going from single, to pair,... to t-subset joint decoding. At the same time, we include users accused in previous iterations as side-information to build a more discriminative test. The second idea, coming from the field of mismatched decoders and compound channels, is to use a linear decoder based on the worst case perceived collusion channel. The decoder is tested under two accusation policies: to catch one colluder, or to catch as many colluders as possible. The probability of false positive is controlled thanks to a rare event estimator. We describe a fast implementation supporting millions of users and compare our results with two recent fingerprinting codes.

Keywords: traitor tracing, fingerprinting, transactional watermarking, joint decoder.

1 Introduction

Traitor tracing or active fingerprinting has witnessed a flurry of research efforts since the invention of the now well-celebrated Tardos codes [13]. The codes of G. Tardos are optimal in the sense that the code length m necessary to fulfill the following requirements (n users, c colluders, probability of accusing an innocent below P_{fp}) has the minimum scaling in $O(c^2 \log n P_{\mathsf{fp}}^{-1})$. The accusation process (more precisely its symmetric version proposed by B. Skoric *et al.* [12]) is based on a scoring per user, so-called accusation sum, whose statistics only depend on the collusion size c, but not on the collusion attack (*e.g.* minority vote, majority vote, interleaving, etc). The alternative accusation strategy has also been tested: the accusation process estimates the collusion attack in order to resort to a matched scoring which is more discriminative [10].

However, these two previous strategies pertain to the same family: the single decoders, *i.e.* processes computing a score per user independently of the other

* Funded by national project ANR MEDIEVALS ANR-07-AM-005.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 28–42, 2011.

codewords and finally accusing users whose score is above a given threshold. Another family is that of the joint decoders, *i.e.* processes computing a score per subset of t users. As far as we know, K. Nuida was the first to propose *and* experiment some sort of a joint decoder [7]. The accusation algorithm only works for very limited collusion size and it doesn't scale well when the number of users n is more than some hundreds. Indeed, so far joint decoders are of particular interest only in theoretical analysis of fingerprinting. P. Moulin [6], and, independently, E. Amiri and G. Tardos [2] show that the capacity of fingerprinting is given by a *maxmin* game whose pay-off is the mutual information $I(Y; X^c|P) \cdot c^{-1}$ where Y is a r.v. representing the symbol decoded from the pirated copy, P is the r.v. denoting the secret of the code, and $X^c = \{X_{j_1}, \ldots, X_{j_c}\}$ is the set of the c symbols assigned to the colluders.

Both papers proposed a joint decoder based on the empirical mutual information computed on the joint type of the observations $(\mathbf{y}, \boldsymbol{\varphi}, \mathbf{p})$ where $\boldsymbol{\varphi} = \sum_{k=1}^{t} \mathbf{x}_{j_k}$ (termed accumulated codeword in the sequel) for the t-subset of users $\{j_1, \ldots, j_t\}$, $t \leq c$. Note that these papers are theoretical studies and that they do not contain any experiment. A practical implementation is hampered by two drawbacks: this is not a linear decoder [1, Def. 1] and the complexity is proportional to $\binom{n}{t}$, the number of t-subsets, *i.e.* in $O(n^t)$. In the quest of practical implementations of this theoretical decoder, 'Don Quixote' is a milestone based on two key ideas: (i) there is no need to compute a score for all t-subsets if we can invent a mechanism preselecting a small number of suspects who are the most likely guilty users; (ii) a linear decoder allowing a fast implementation of the scoring function.

These ideas are indeed not easily translated into practical algorithms. In real life scenarios such as Video-on-Demand portals, m bits are in copies of a movie, which are afterwards distributed to n clients (the parameters (m, n) vary from one Work to another). The collusion size c is neither known at the code construction nor at the accusation side. Therefore, one never knows how the rate $m^{-1} \log n$ compares to the theoretical capacity which depends on c. However, for (i), we need to identify suspects whenever it is possible (*i.e.* when the rate is below capacity) while guaranteeing a probability of false alarm P_{fp}. Efficient linear decoders pertaining to (ii) are based on likelihood ratio, which can't be computed in practice since we do not know the collusion strategy. Fortunately, information theorist have recently come up with a very elegant solution providing universal linear decoders performing well (*i.e. capacity achieving*) over a family of channels while ignoring the active channel [1]. This theory of compound channel fits very well with the traitor tracing framework.

2 Structure of the Don Quixote Algorithm

Before detailing the structure of the proposed decoder, let us briefly remind the construction of a Tardos code. Let (m, n) be the length and the size of the code. First, draw randomly m variables $\mathbf{p} = (p(1), \ldots, p(m))^T$ s.t. $p \overset{\text{i.i.d.}}{\sim} f(p)$. Then draw randomly and independently the elements of the j-th codeword

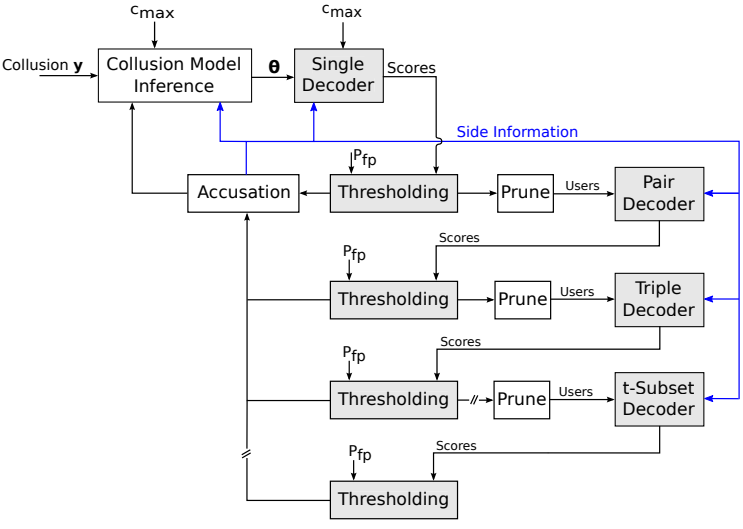


Fig. 1. Overview of the iterative, side-informed joint Tardos fingerprint decoder

$\mathbf{x}_j = (x_j(1), \ldots, x_j(m))$ s.t. $\mathbb{P}(x_j(i) = 1) = p(i)$. We chose the pdf $f(p)$ recommended by G. Tardos.

The iterative architecture of our joint decoder is sketched in Figure 1. For sake of clarity, we postpone details about the computation of scores and the thresholding (shaded blocks of Fig. 1) to the next sections. The decoder can be employed in *catch-one* and *catch-many* traitor tracing scenarios [14]. In the first case, our iterative decoder simply stops after the first accusation; in the later case, the decoder stops when c_{max} accusations have been made or when no further accusations can safely be made using the t_{max}-subset decoder ($2 \leq t_{\mathsf{max}} \leq c_{\mathsf{max}}$).

2.1 Iterative, Joint Decoding

The theoretical papers [2,6] tell us that scores computed from subsets of t users are more discriminative as t increases, provided that $t \leq c$, c being the real collusion size. Discriminative means that scores of subsets of innocent users are statistically more separated from scores for guilty users – the Kullback-Leibler distance between their pdf is significantly higher. Our point is that, indeed, hybrid subsets containing κ colluders and $t - \kappa$ innocents have also greater scores in expectation as κ increases. Therefore, by pruning out users involved in subsets of small score, we are likely maintaining a list of suspects with a good number of colluders.

At the beginning, $\mathcal{X}$ is the set of n codewords or users. The t-th iteration of our algorithm takes a set of codewords $\mathcal{X}^{(t)} \subseteq \mathcal{X}$ and computes the scores for each subset of t codewords from $\mathcal{X}^{(t)}$. Denote $n^{(t)} = |\mathcal{X}^{(t)}|$, there are then $\binom{n^{(t)}}{t} = O((n^{(t)})^t)$ scores to be computed. For instance, in the first iteration, $\mathcal{X}^{(1)} = \mathcal{X}$

Table 1. Maximal number $p^{(t)}$ of suspected users input to the joint t-subset decoder versus total number of subsets without pruning out users for $n = 10^6$

Subset size (t)	2	3	4	5	6	7	8
Users suspected ($p^{(t)}$)	3 000	300	103	58	41	33	29
Computed subsets $\binom{p^{(t)}}{t}$	4 498 500	4 455 100	4 421 275	4 582 116	4 496 388	4 272 048	4 292 145
Total subsets $\binom{n}{t}$	$\sim 10^{11}$	$\sim 10^{17}$	$\sim 10^{22}$	$\sim 10^{27}$	$\sim 10^{33}$	$\sim 10^{38}$	$\sim 10^{43}$

and the scores are just the $n^{(1)} = n$ outputs of a single decoder. We assume to have the computation power scaling as $O(n)$ such that this first iteration is feasible. The key idea is to gradually reduce $n^{(t)}$ such that the computation of scores remains tractable. For instance, if $n^{(t)} = O(n^{1/t})$ then the t-th iteration relies on a $O(n)$ scores computation just like the first iteration.

During each iteration, some users might be deemed guilty (cf. Section 2.4) and added to the side information (cf. Section 3.1).

The main operation is to construct the subset $\mathcal{X}^{(t+1)}$ of suspects to be passed to the following iteration. Suspects are users so far neither accused nor declared as innocent. The users get ranked (with guilty users most likely placed in top positions) and the first $n^{(t+1)}$ users compose the set $\mathcal{X}^{(t+1)}$ while the others are discarded (*i.e.* considered as innocents). The $(t+1)$-th iteration starts by computing scores for subset of size $t+1$ from $\mathcal{X}^{(t+1)}$, see Section 3.3 for details. For $t > 1$, the size $n^{(t+1)} \leq p^{(t+1)}$ where $p^{(t+1)}$ is the upper size and runtime limit that our computer can handle. Table 1 gives values s.t. the number of subsets is kept approximately constant at about 4 500 000. The choices for $p^{(t)}$ are presumably not optimal – other values may allow a better distribution of resources – but a necessaary trade-off between the computational effort and the decoding performance.

2.2 Pruning Out

First Iteration. The first iteration computes a score per user: the bigger the score, the more likely the user is a colluder. If some conditions are met, users with the highest scores might be accused (cf. Section 2.4). Users are ranked according to their score in decreasing order. The first $n^{(2)} \leq p^{(2)}$ users are included in the set $\mathcal{X}^{(2)}$.

***t*-th Iteration, $t > 1$.** Once the scores for all t-subsets are computed, they are ranked in decreasing order. Again, if accusations can be safely made, some users from the first-ranked subset are deemed guilty. The others are included in set $\mathcal{X}^{(t+1)}$. The algorithm browses down the sorted list of subsets and includes their users in $\mathcal{X}^{(t+1)}$ if they have not been already included and if they have not been accused. This stops when $n^{(t+1)} = p^{(t+1)}$ (the users are listed in arbitrary order in a t-subset, therefore for the last subset under suspicion, the last users might be relaxed while the first are suspected) or when the last subset of the sorted list has been analyzed.

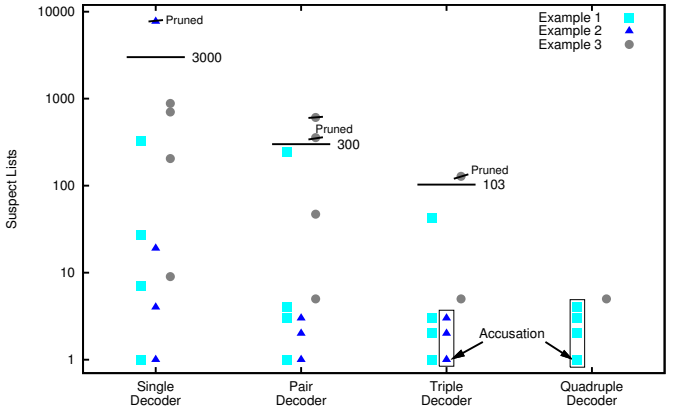


Fig. 2. Examples of the iterative decoding and pruning process for $c = 4$

Figure 2 illustrates the iterative decoding and pruning process where the positions of the colluders are marked with different symbols in three examples. The first stage denoted *single decoder* represents the list of suspects/users ranked according to their accusation scores after single decoding. Since no user's score is above the accusation threshold, the process continues in the following iteration. Only suspects ranked within the first 3 000 positions are passed to the *pair decoder* and the illustration shows the suspect list ranked after computing the scores of user pairs. Users within a pair are ordered according to the criterion defined in Section 2.4. After pruning – now the list is limited to 300 positions – the remaining suspects are fed to the *triple decoder*. Note that the positions of the colluders generally move towards the top of the list (the bottom of the illustration) with each iteration as shown in the examples. This observation allows us to reduce the suspect list at each iteration while likely retaining the colluders. On the other hand, colluders may be discarded, as visualized in *Examples 2 & 3*. Pruning is the necessary trade-off to reduce the computational burden of the decoder. The users of the subset whose score is the highest and above the threshold are framed in the illustration. Their top-ranked user is accused and added to the side information.

2.3 Enumerating all t-Subsets

The joint t-subset decoder has to enumerate all $\binom{n^{(t)}}{t}$ subsets and compute the corresponding scores. One way to implement the generation of all t-subsets of $\mathcal{X}^{(t)}$ is the *revolving door* algorithm[1] [9] which changes exactly one element of the subset at each enumeration step.

In particular, the score of the k-th t-subset $\mathbf{t}_k$ only depends on the accumulated codewords $\boldsymbol{\varphi}_k = \sum_{j_\ell \in \mathbf{t}_k} \mathbf{x}_{j_\ell}$. The revolving door is initialized with the first

[1] Termed algorithm **R** by Knuth [5, Chap. 7.2.1.3].

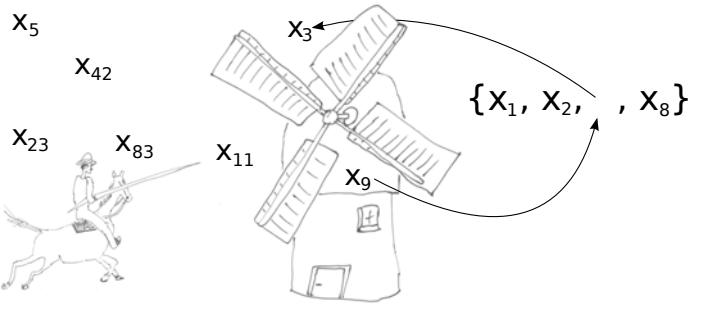


Fig. 3. Illustration of the revolving door algorithm for $t = 4$

subset $\mathbf{t}_1 = \{j_1, \ldots, j_t\}$ whose accumulated codeword is $\boldsymbol{\varphi}_1$. At each step, the algorithm replaces one user $j_\dagger \in \mathbf{t}_k$ with a new user $j_\star$ and computes the updated code sequence relating to the combination $\mathbf{t}_{k+1}$ as $\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \mathbf{x}_{j_\dagger} + \mathbf{x}_{j_\star}$. Fig. 3 provides an illustration. The benefit of the *resolving door* is that the computational effort to generate a t-subset and its associated accumulated codeword is independent of size t.

2.4 Accusation

Let $\mathbf{t}^\diamond$ denote the subset with the highest score. We accuse one user of the t-subset $\mathbf{t}^\diamond$, only if its score is greater than a threshold: $s_{\mathbf{t}^\diamond} > \tau$. The computation of threshold τ is explained hereafter in Section 3.4. The thresholding operation ensures that subsets with score above τ contain at least one colluder with a very high probability. Assume now that this condition is met. Obviously, for the first iteration, $t = 1$ and the single user in subset $\mathbf{t}^\diamond$ is accused. For $t > 1$, we propose the following method. In order to identify and accuse the most probable traitor in $\mathbf{t}^\diamond$, we record for each user $j \in \mathcal{X}^{(t)}$ the subset leading to that user's highest score:

$$\mathbf{t}_j^\diamond = \arg\max_{\mathbf{t}} \{s_{\mathbf{t}} \mid j \in \mathbf{t}\}. \tag{1}$$

Next, we count how often each user $j_k \in \mathbf{t}^\diamond$ appears in the recorded subsets $\{\mathbf{t}_j^\diamond\}_{j \in \mathcal{X}^{(t)}}$ and denote this value a_{j_k}. Finally, we accuse the user $j^\diamond$ appearing most often:

$$j^\diamond = \arg\max_{j \in \mathbf{t}^\diamond} a_j. \tag{2}$$

3 Technical Details

This section describes four remaining operations: side-information, inferences about the collusion model, scoring, and the thresholding.

3.1 Side Information

The knowledge of the identity of some colluders is beneficial in two operations: derivation of better inferences about the collusion channel, and derivation of more discriminative scores. It is well known in estimation and detection theory that conditioning (*i.e.* to side-inform with prior knowledge) is always helpful on average. Denote by $\mathcal{X}_{\mathsf{SI}}$ the set of accused users (subscript SI denotes Side Information). At the beginning, $\mathcal{X}_{\mathsf{SI}} = \emptyset$. If a user is accused as described in Section 2.4, then he is removed from $\mathcal{X}^{(t)}$ and included into $\mathcal{X}_{\mathsf{SI}}$. If nobody is accused, then iteration $(t+1)$ starts with $\mathcal{X}^{(t+1)}$. If someone is accused, then iteration t is not over. Since new side information is available, we can benefit from it right away. A new inference process is run with the new $\mathcal{X}_{\mathsf{SI}}$, and the scores for t-subsets are computed again with the new inference, the new conditioning $\mathcal{X}_{\mathsf{SI}}$ and over the new set $\mathcal{X}^{(t)}$. The t-th iteration breaks this loop whenever no additional colluder is identified.

3.2 Inferences about the Collusion Model

A long tradition in Tardos traitor tracing codes is to model the attack led by c colluders by a vector $\boldsymbol{\theta}^{(c)} = (\theta_0^{(c)}, \theta_1^{(c)}, \ldots, \theta_c^{(c)})$ where $\theta_\sigma^{(c)} = \mathbb{P}(y_i = 1 | \sum_{k=1}^{c} x_{j_k}(i) = \sigma)$ [10]. In words, when the colluders have σ symbols '1' over c, they flip a coin of bias $\theta_\sigma^{(c)}$ to decide whether they put symbol '1' or a '0' in the pirated sequence. The marking assumption holds if $\theta_0^{(c)} = 1 - \theta_c^{(c)} = 0$. The main difficulty is that $\boldsymbol{\theta}^{(c)}$ cannot be estimated from the observations $(\mathbf{y}, \mathbf{p})$ since, for any integer $c' > c$ there exists $\boldsymbol{\theta}^{(c')}$ s.t. $\mathbb{P}(y = 1|p, \boldsymbol{\theta}^{(c')}) = \mathbb{P}(y = 1|p, \boldsymbol{\theta}^{(c)}), \forall p \in (0, 1)$. We call $\boldsymbol{\theta}^{(c')}$ the equivalent attack of $\boldsymbol{\theta}^{(c)}$ of size c'. The parameter of the model cannot be identified except if we were knowing the collusion size c. We chose the maximum log-likelihood estimator (MLE) for a given $\hat{c}$:

$$\hat{\boldsymbol{\theta}}^{(\hat{c})} = \underset{\boldsymbol{\theta} \in [0,1]^{\hat{c}+1} \;\; s.t. \;\; \theta(0)^{(\hat{c})}=0, \theta(\hat{c})^{(\hat{c})}=1}{\arg\max} \log \mathbb{P}(\mathbf{y}|\mathbf{p}, \mathcal{X}_{\mathsf{SI}}, \boldsymbol{\theta}), \tag{3}$$

with $\mathbb{P}(\mathbf{y}|\mathbf{p}, \boldsymbol{\theta}) = \prod_{i=1}^{m} \mathbb{P}(y(i)|p(i), \mathcal{X}_{\mathsf{SI}}, \boldsymbol{\theta})$. Section 3.3 details the computation of this likelihood. However, due to the lack of identifiability, this approach cannot estimate c, but only $\hat{\boldsymbol{\theta}}^{(\hat{c})}$ for a given $\hat{c}$. For a long enough code, the MLE accurately finds $\boldsymbol{\theta}^{(c)}$ if $\hat{c} = c$, or its equivalent attack of size $\hat{c}$ if $\hat{c} > c$; yet there is no way to distinguish the two cases.

We have experimentally noticed that scores based on $\hat{\boldsymbol{\theta}}^{(\hat{c})}$ are only slightly less powerful than the optimal ones (based on the real $\boldsymbol{\theta}^{(c)}$) provided that $\hat{c}$ is bigger than the real c. Therefore, we assume that $c < c_{\mathsf{max}}$ and set $\hat{c} = c_{\mathsf{max}}$. We estimate not the real collusion parameter but its equivalent attack of size c_{max} (this is why we rather speak of collusion inference than collusion estimation).

The theory of compound channel justifies this approach. Suppose $c < c_{\mathsf{max}}$ and consider the family of collusions $\{\boldsymbol{\theta}^{(c')}\}_{c'=c}^{c_{\mathsf{max}}}$ gathering the real collusion $\boldsymbol{\theta}^{(c)}$ and

its equivalent attacks of size from $c+1$ to c_{max}. It can be shown[2] that this family is a one-sided compound channel [1, Def. 3, Eq.(8)]. Therefore by [1, Lemma 5], we know that a good (*i.e.* information theorists say *capacity achieving*) linear decoder is the maximum likelihood decoder tuned on the worst element of the family, which is in our case the collusion $\boldsymbol{\theta}^{(c_{\mathsf{max}})}$.

3.3 Score Computation

The score is just the log-likelihood ratio tuned on the inference $\hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}$. We give its most generic expression for a subset $\mathbf{t}$ of t users and side information $\mathcal{X}_{\mathsf{SI}}$ containing n_{SI} codewords of already accused users. Denote by $\boldsymbol{\rho}$ and $\boldsymbol{\varphi}$ the accumulated codewords of $\mathcal{X}_{\mathsf{SI}}$ and $\mathbf{t}$: $\boldsymbol{\rho} = \sum_{j\in\mathcal{X}_{\mathsf{SI}}} \mathbf{x}_j$ and $\boldsymbol{\varphi} = \sum_{j\in\mathbf{t}} \mathbf{x}_j$. We have $\forall i \in [m]$, $0 \le \rho(i) \le n_{\mathsf{SI}}$ and $0 \le \varphi(i) \le t$.

Denote $\mathcal{H}_0$ the hypothesis where subset $\mathbf{t}$ is composed of innocent users. Then $\mathbf{y}$ is statistically independent from its codewords which in turn only depend on $\mathbf{p}$:

$$\mathcal{H}_0 : \quad \mathbb{P}(\mathbf{y}, \{\mathbf{x}_j\}_{j\in\mathbf{t}} | \mathbf{p}, \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}, \mathcal{X}_{\mathsf{SI}}) = \mathbb{P}(\mathbf{y}|\mathbf{p}, \boldsymbol{\theta}, \mathcal{X}_{\mathsf{SI}}) \mathbb{P}(\{\mathbf{x}_j\}_{j\in\mathbf{t}}|\mathbf{p}) \tag{4}$$

Denote $\mathcal{H}_1$ the alternative where subset $\mathbf{t}$ is composed of colluders. Then $\mathbf{y}$ is statistically dependent of its codewords:

$$\mathcal{H}_1 : \quad \mathbb{P}(\mathbf{y}, \{\mathbf{x}_j\}_{j\in\mathbf{t}} | \mathbf{p}, \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}, \mathcal{X}_{\mathsf{SI}}) = \mathbb{P}(\mathbf{y}|\{\mathbf{x}_j\}_{j\in\mathbf{t}}, \mathbf{p}, \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}, \mathcal{X}_{\mathsf{SI}}) \mathbb{P}(\{\mathbf{x}_j\}_{j\in\mathbf{t}}|\mathbf{p}) \tag{5}$$

All these sequences are composed of independent r.v. thanks to the code construction and the memoryless nature of the collusion. Moreover, the collusion only depends on the number of symbol '1' present in the codewords of a subset, *i.e.* the accumulated codeword. Therefore, the score of subset $\mathbf{t}$ is just the log-ratio of the two previous probability expressions which simplifies to:

$$s = \sum_{y(i)=1} \log \frac{\alpha(i)}{\beta(i)} + \sum_{y(i)=0} \log \frac{1-\alpha(i)}{1-\beta(i)}, \tag{6}$$

with the following expressions:

$$\alpha(i) = \mathbb{P}(y=1|(\varphi(i), t), (\rho(i), n_{\mathsf{SI}}), p(i), \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}) = P(\varphi(i)+\rho(i), t+n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})})$$
$$\beta(i) = \mathbb{P}(y=1|(\rho(i), n_{\mathsf{SI}}), p(i), \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}) = P(\rho(i), n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})})$$

and function $P(\cdot)$ is defined by:

$$P(u, v, p, \hat{\boldsymbol{\theta}}^{(c_{\mathsf{max}})}) = \sum_{\sigma=u}^{c_{\mathsf{max}}-v+u} \hat{\theta}(\sigma)^{(c_{\mathsf{max}})} \binom{c_{\mathsf{max}}-v}{\sigma-u} p^{\sigma-u} (1-p)^{c_{\mathsf{max}}-v-\sigma+u} \tag{7}$$

This expression is compact, involved, but very generic. In words, it gives the probability that $y=1$ knowing that the symbol '1' has been distributed to users

[2] The journal version of this paper contains the proof.

with probability p, the collusion model $\hat{\boldsymbol{\theta}}^{(c_{\max})}$, and the identity of v colluders who have u symbol '1' and $v-u$ symbol '0' at this index.

The inference on the collusion model searches for a $\boldsymbol{\theta}^{(c_{\max})}$ maximizing the following likelihood:

$$\log \mathbb{P}(\mathbf{y}|\mathbf{p},\boldsymbol{\theta},\mathcal{X}_{\mathsf{SI}}) = \sum_{y(i)=1} \log \beta_i + \sum_{y(i)=0} \log 1-\beta_i. \tag{8}$$

In the first iteration, a single decoder is used: $t=1$ and $\boldsymbol{\varphi} = \mathbf{x}_j$ for user j. If nobody has been deemed guilty so far, then $\rho(i) = n_{\mathsf{SI}} = 0$, $\forall i \in [m]$. The t-th iteration works on subsets of size t. However, our scoring is only defined if $t + n_{\mathsf{SI}} \leq c_{\max}$. Therefore, for a given size of side-information, we cannot conceive score for subset of size bigger than $t_{\max} = c_{\max} - n_{\mathsf{SI}}$. This implies that in the catch-all scenario, the maximal number of iterations depends on how fast $\mathcal{X}_{\mathsf{SI}}$ grows.

3.4 Thresholding

The issue here is the translation of the scores into probabilities. At a given iteration and a given state of the side information, all the subset scores are computed in the same deterministic way. The idea is to generate subsets composed of new codewords and to compute their scores. We are then sure to observe scores of subset of innocents since these codewords have not been used to forge $\mathbf{y}$. With a Monte Carlo simulation, we can estimate the probability that the score of an innocent subset is bigger than threshold τ, or the other way around, the threshold τ such that this probability is below ϵ. This approach works whatever the way scores are computed.

In the first iteration, the subset is just a singleton, the codeword of one user, and that user is either innocent either guilty. Therefore, users whose scores are above the threshold are accused and included in $\mathcal{X}_{\mathsf{SI}}$. Denote P_{fp} the total probability of false positive, *i.e.* accusing at least one innocent, and ϵ the probability of wrongly accusing a given innocent user. Since the codewords are i.i.d. and $c \ll n$, we have $P_{\mathsf{fp}} = 1-(1-\epsilon)^{n-c} \approx n\epsilon$. P_{fp} is stipulated in the requirements and we fix $\epsilon = {}^{P_{\mathsf{fp}}}/_{n}$. In the t-th iteration ($t>1$), the same Monte Carlo simulation over subsets of size t is run. It estimates the threshold s.t. the score of a subset of innocents is greater than τ only with a probability ϵ. In other words, scores above τ indicate subset with at least one colluder. A further analysis identifies and accuses the most likely one among the t users (cf. Section 2.4). Again, ϵ should be set as low as ${}^{P_{\mathsf{fp}}}/_{\binom{n}{t}}$ to control the total probability of false alarm.

The only problem is that a large n implies a very low probability ϵ for both cases ($t=1$ and $t>1$), and a Monte Carlo simulation is then bad at estimating accurately threshold τ. This is the reason why we implemented an numerical estimator based on rare event analysis [3].

4 Experimental Results

The Tardos decoder is implemented in C++ and compiled using GNU `g++` version 4.4.5 on a x86 Ubuntu/Linux 10.10 system with `-O3 -march=native`

`-fomit-frame-pointer -mfpmath=sse`. The estimation of $\boldsymbol{\theta}$ uses approximate vectorized single-precision floating point arithmetic and Shigeo Mitsunari's fast approximative log() function[3]; the remaining components are implemented with double-precision. Pseudo-random numbers are generated with the SIMD-oriented Fast Mersenne Twister (dSFMT)[4] [11].

All runtime results are reported for a single core of a x86 Intel Core2 CPU (E6700) clocked at 2.6 GHz with 2 GB of memory running Ubuntu/Linux 10.10.

The joint decoder receives lists of suspects whose length are upper bounded by values of Table 1.

4.1 Catch-One Scenario

Here the aim is to catch the most like colluder – this is the tracing scenario most commonly considered in the literature. We compare our single and joint decoder performance against the results provided by Nuida *et al.* [8]. These authors assumed that c is known for the code construction and the decoding. For a fair comparison, our decoder uses this assumption: $c_{\mathsf{max}} = c$.

The experimental setup considers $n = 1\,000\,000$ users and $c \in \{2, 3, 4, 6, 8\}$ colluders performing *worst-case* attack [4]. In Fig. 4, we plot the empirical probability of error $P_{\mathsf{e}} = P_{\mathsf{fp}} + P_{\mathsf{fn}}$ obtained by running at least 10 000 experiments for each setting versus the code length m. The false-positive error is controlled by thresholding based on rare-event simulation, $P_{\mathsf{fp}} = 10^{-3}$. For shorter code length, almost exclusively false-negative errors occur. As expected, we observe a huge decoding performance improvement for the joint decoder over the single decoder. The advantage is much more pronounced when a larger number of colluders collaborates.

Table 2 compares the code length to obtain an error rate of $P_{\mathsf{e}} = P_{\mathsf{fp}} + P_{\mathsf{fn}} = 10^{-3}$ for our proposed Tardos decoders with the results reported by Nuida *et al.* [8] under marking assumption. While the joint decoder only marginally improves the decoding performance for two colluders, it almost halves the code length for four colluders.

The column *hypothetical* of Table 2 reports simulation results of a joint decoder that knows the identity of the colluders and just computes scoring and thresholding for the colluders. The simulation allows to judge the performance gap between the proposed joint decoder operating on a *pruned* list of suspects (potentially discarding colluders) and the unconstrained joint decoder.

Figure 5 (a) shows in which iteration the first out of $c = 4$ colluders is successfully identified for varying code length, $P_{\mathsf{fp}} = 10^{-3}$. The benefit of joint decoding is best seen for intermediate code lengths between $m = 352$ and $m = 864$. For longer codes the single decoder is sufficient to make the first accusation. Figure 5 (b) illustrates the average runtime in seconds for score computation and

[3] Available from `http://homepage1.nifty.com/herumi/soft/fmath.html`, version of February 16, 2010.

[4] Available from `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/SFMT/`, version 2.1

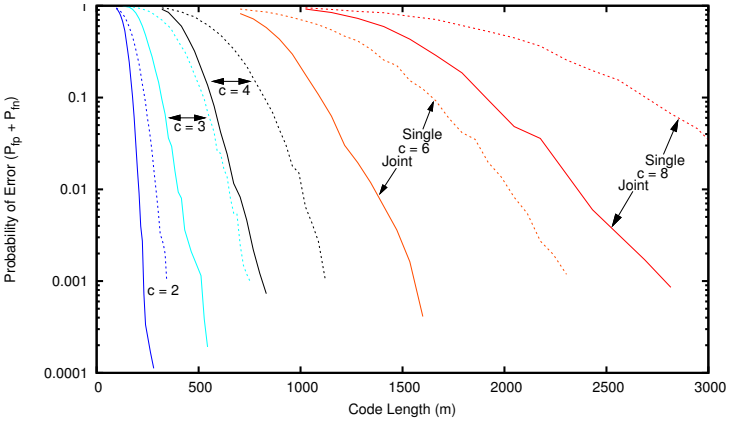


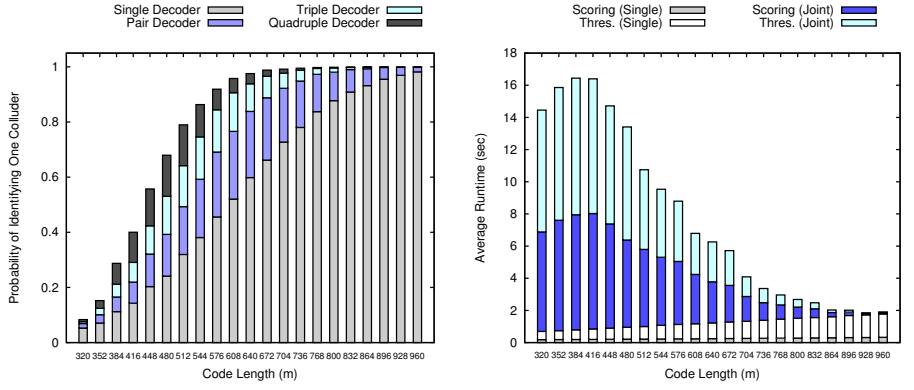
Fig. 4. Probability of error versus code lengths for the single and joint decoder for different collusion sizes; $n = 10^6$, *worst-case* attack

Table 2. Code length comparison for $n = 10^6$, *worst-case* attack, $P_e = 10^{-3}$

Colluders (c)	Nuida *et al.* [8]	Proposed Decoder		Hypothetical
		Single	Joint	
2	253	~ 344	~ 232	~ 232
3	877	~ 752	~ 512	~ 400
4	1454	~ 1120	~ 784	~ 720
6	3640	~ 2304	~ 1568	~ 1440
8	6815	~ 3712	~ 2688	~ 2432

thresholding for the single and joint decoders in that scenario. For short code length all decoding stages (up to $t = 4$) have to be run – often unsuccessfully. A significant amount of the execution time is spent in thresholding relative to scoring for the number of computed subsets, $\binom{p^{(t)}}{t} \sim 4\,500\,000$.

In Fig. 6 we plot the probability of correctly identifying one colluder and the iteration number leading to this accusation. This time, we vary the number of score computations performed in each iteration from 10^5 to 10^9 by controlling the suspect list sizes $\{p^{(t)}\}$. The rightmost results relate to the hypothetical joint decoder which does not have to enumerate all combinations but just computes the accusation scores for the colluders. Surprisingly, a significant difference in accusation performance can only be observed at the last iteration (*i.e.* the quadruple decoder). An equal weighting of the computation resources over the iterations is certainly not optimal. This experiment seems to conclude that the emphasis should be put on the last iterations. Yet, it is not clear what the optimal resources distribution is.



(a) Accusations (b) Runtime

Fig. 5. Iteration making the first accusation (a), and average runtime in seconds for score computation and thresholding for the single and joint decoders (b); $n = 10^6$, $c = 4$, *worst-case* attack

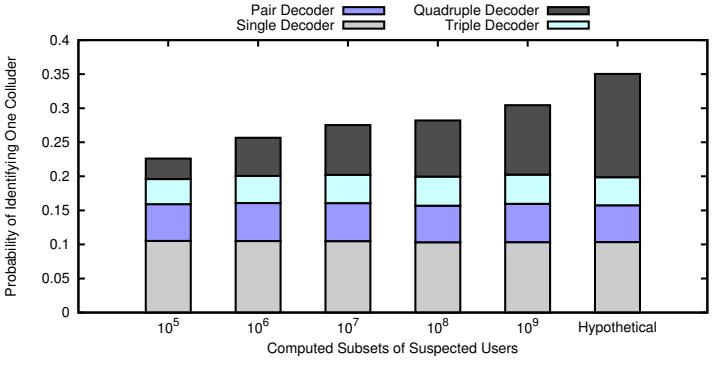


Fig. 6. Iteration making the first accusation with different number of subsets; $n = 10^6$, $m = 384$, $c = 4$, *worst-case* attack

4.2 Catch-Many Scenario

We now consider the more realistic case where the code length m is fixed. The only assumption at the decoder side is that $c \leq c_{\mathsf{max}}$. The aim is to identify as many colluders as possible. Figure 7 shows the average number of identified colluders by the symmetric Tardos single decoder, our non-iterated single, the iterative side-informed single and our iterative side-informed joint decoders. The experimental setup considers $n = 1\,000\,000$ users, code length $m = 2048$, and *worst-case* collusion attack carried out by between two and eight colluders. The global probability of false positive is fixed to $P_{\mathsf{fp}} = 10^{-3}$. The performance advantage of the more sophisticated decoders is evident. The joint decoder has a good chance to catch most of the colluders even when $c = 8$.

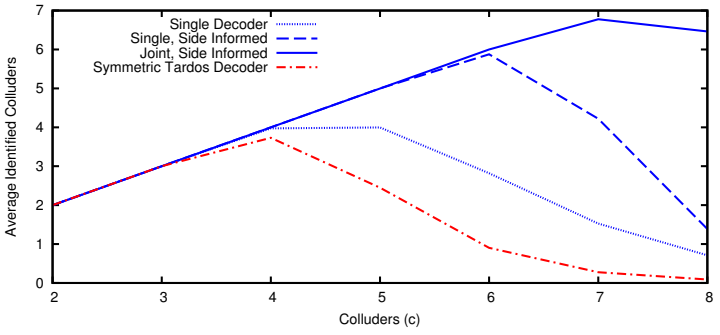
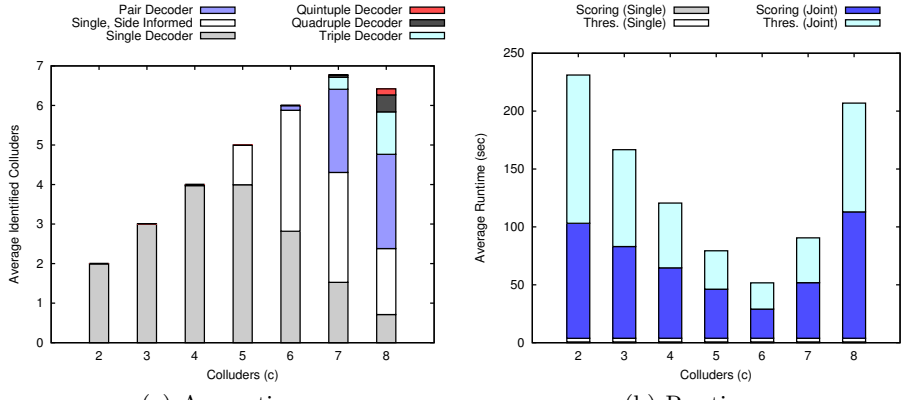


Fig. 7. Average identified traitors for different number of colluders performing *worst-case* attack; $n = 10^6$, $m = 2048$, $P_{\mathsf{fp}} = 10^{-3}$, $c_{\mathsf{max}} = 8$



(a) Accusations (b) Runtime

Fig. 8. Average number of accusations per iteration (a); average runtime in seconds for score computation and thresholding for the single and joint decoders (b); $n = 10^6$, $m = 2048$, *worst-case* attack, $P_{\mathsf{fp}} = 10^{-3}$

In Fig. 8 (a) we analyse the average number of accusations made per iteration (same setup as above). Figure 8 (b) shows the average runtime in seconds accounted for score computation and thresholding of the iterative single and joint decoders. The longest runtimes are observed for $c = 2$ and $c = 8$. In the first case, both colluders are caught by the single decoder, yet the remaining iterations up to the 6-subset decoder have to be run since $c_{\mathsf{max}} = 8$.

4.3 Runtime Performance Analysis

Table 3 provides average runtime results in seconds split up per decoder component for two traitor tracing scenarios with $n = 10\,000$ and $n = 1\,000\,000$ users. The runtime for the collusion model estimation and refinement is negligible and independent of the number of users, $O(c \cdot m)$.

Table 3. Average runtime in seconds per decoder component; the number of joint accusation score computations is fixed to approximately 4.5 million

Avg. Runtime (sec) / Decoder Component	$n = 10\,000$		$n = 1\,000\,000$	
	$m = 320$	$m = 640$	$m = 320$	$m = 640$
Collusion Model ($\hat{\boldsymbol{\theta}}^{(4)}$)	0.00	0.01	0.00	0.01
Single Decoder (s_j)	0.01	0.01	0.17	0.23
Thresholding	0.48	0.90	0.51	0.98
Pair Decoder	2.34	4.46	2.38	4.53
Thresholding	1.48	2.69	1.68	3.07
Triple Decoder	2.26	4.41	2.29	4.45
Thresholding	2.32	4.18	2.79	5.00
Quadruple Decoder	2.19	4.43	2.20	4.42
Thresholding	3.17	5.76	4.02	6.96
Total	14.34	26.85	16.04	29.65

Single decoding can be efficiently implemented to compute more than ten million scores for a code of length $m = 320$ per second. The complexity is $O(n \cdot m + n \cdot \log n)$. The second term relates to sorting the results which consumes a substantial parts of the runtime for small m. The runtime contribution of the joint decoding stage clearly depends on the size of pruned list of suspects, $O(m \cdot p)$ and is independent of the subset size t thanks to the *revolving door* enumeration method. Our implementation performs almost two million joint score computations per second.

Thresholding accounts for more than half of the runtime in the experimental setups investigated in this work. However, this is not a serious issue for applications with a large user base or when p becomes large. Thresholding depends on the subset size t because a large number of random codeword combinations must be generated and because we seek lower probability level in $O(P_{\mathsf{fp}}/n^t)$. Therefore, the complexity is in $O(m \cdot t^2 \cdot \log(n/P_{\mathsf{fp}}))$.

Note that all runtime results have been obtained with single CPU core although a parallel implementation can be easily achieved. The score computation (Eq. 6) has been implemented using pre-computed weights which reduce the computation effort to a single table lookup for each codeword symbol and the accumulation of the values.

5 Conclusion

'Don Quixote' is built on three main pillars. Joint decoding is made affordable by an iterative algorithm pruning out users that are likely not guilty. The theory of compound channel gives fast linear and discriminative scores. The rare event simulation guarantees the reliability of the accusation by controlling the probability of false positive. The collusion size and process are nuisance parameters that are neither needed for the construction of the code, nor at the accusation side. The decoding performance is at the forefront of the state of the art.

References

1. Abbe, E., Zheng, L.: Linear universal decoding for compound channels. IEEE Transactions on Information Theory 56(12), 5999–6013 (2010)
2. Amiri, E., Tardos, G.: High rate fingerprinting codes and the fingerprinting capacity. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, pp. 336–345. SIAM, New York (2009)
3. Cérou, F., Furon, T., Guyader, A.: Experimental assessment of the reliability for watermarking and fingerprinting schemes. EURASIP Jounal on Information Security (2008), iD 414962, 12 pages
4. Furon, T., Pérez-Freire, L.: Worst case attacks against binary probabilistic traitor tracing codes. In: Proc. First IEEE Int. Workshop on Information Forensics and Security, London, UK, pp. 46–50 (December 2009)
5. Knuth, D.E.: The Art of Computer Programming, Generating All Combinations and Partitions, vol. 4. Addison-Wesley, Reading (2005); Fascicle 3
6. Moulin, P.: Universal fingerprinting: Capacity and random-coding exponents. In: Proc. IEEE International Symposium on Information Theory, ISIT 2008, Toronto, ON, Canada, pp. 220–224 (July 2008)
7. Nuida, K.: Short collusion-secure fingerprint codes against three pirates. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 86–102. Springer, Heidelberg (2010)
8. Nuida, K., Fujitsu, 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), http://eprint.iacr.org/2008/338
9. Payne, W.H., Ives, F.M.: Combination generators. ACM Transactions on Mathematical Software 5(2), 163–172 (1979)
10. Pérez-Freire, L., Furon, T.: Blind decoder for binary probabilistic traitor tracing codes. In: Proc. First IEEE Int. Workshop on Information Forensics and Security, London, UK, pp. 56–60 (December 2009)
11. Saito, M., Matsumoto, M.: A PRNG specialized in double precision floating point numbers using an affine transition. In: Proc. Eighth Int. Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, MCQMC 2008, pp. 589–602. Springer, Montréal (2008)
12. Skoric, B., Katzenbeisser, S., Celik, M.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
13. Tardos, G.: Optimal probabilistic fingerprint codes. In: Proc. 35th ACM Symposium on Theory of Computing, San Diego, CA, USA, pp. 116–125 (2003), http://www.renyi.hu/~tardos/publications.html
14. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion-resistant fingerprinting for Multimedia. IEEE Signal Processing Magazine 21(2), 15–27 (2004)

An Asymmetric Fingerprinting Scheme Based on Tardos Codes

Ana Charpentier[1,⋆], Caroline Fontaine[2], Teddy Furon[1], and Ingemar Cox[3]

[1] INRIA-Rennes research center, Campus de Beaulieu, Rennes, France
[2] CNRS/Lab-STICC/CID, Télécom Bretagne/ITI, Brest, France
[3] University College London, Dpt. of Computer Science, London, United Kingdom

Abstract. Asymmetric fingerprinting protocols are designed to prevent an untrustworthy Provider incriminating an innocent Buyer. These protocols enable the Buyer to generate their own fingerprint by themself, and ensure that the Provider never has access to the Buyer's copy of the Work. Until recently, such protocols were not practical because the collusion-resistant codes they rely on were too long. However, the advent of Tardos codes means that the probabilistic collusion-resistant codes are now sufficiently short that asymmetric fingerprint codes should, in theory, be practical.

Unfortunately, previous asymmetric fingerprinting protocols cannot be directly applied to Tardos codes, because generation of the Tardos codes depends on a secret vector that is only known to the Provider. This knowledge allows an untrustworthy Provider to attack traditional asymmetric fingerprinting protocols. We describe this attack, and then propose a new asymmetric fingerprinting protocol, specifically designed for Tardos codes.

1 Introduction

This paper considers a problem arising in the fingerprinting of digital content. In this context, a fingerprint is a binary code that is inserted into a Work for the purpose of protecting it from unauthorized use, or, more precisely, for the purpose of identifying individuals responsible for its unauthorized use. In such a scenario, it is assumed that two or more users may collude in order to try to hide their identities. Under the *marking assumption* [2], colluders cannot alter those bits of the code that are identical for all colluders. However, where bits differ across colluders, these bits may be assigned arbitrary values. A key problem is resistance to collusion, i.e. if c users create a pirated copy of the Work, its tampered fingerprint (i) should not implicate innocent users, and (ii) should identify at least one of the colluders.

This problem has received considerable attention since Boneh and Shaw [2] discussed it. They introduced the concept of a c-secure code such that the probability of framing an innocent user is lower than ϵ. Unfortunately, the length of

⋆ Supported by National Project MEDIEVALS ANR-07-AM-005.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 43–58, 2011.

their codes, $O(c^4 \log(\frac{n}{\epsilon}) \log(\frac{1}{\epsilon}))$ where n is the number of users, was too long to be practical. Following Boneh and Shaw's paper, there has been considerable effort to design shorter codes. In 2003, Tardos [19] proposed an efficient code construction that, for the first time, reduced the code length to the theoretical lower bound, $O(c^2 \log(\frac{n}{\epsilon}))$, thereby making such codes practical. Tardos codes are currently the state-of-the-art for collusion-resistant fingerprinting.

Contemporaneously, some papers considered the scenario where the Provider is untrustworthy. Given knowledge of a Buyer's fingerprint, the Provider creates a pirated copy of a Work, implicating the innocent Buyer. To prevent this, Pfitzman and Schunter [16] first introduced the concept of asymmetric fingerprinting in which the Provider does not need to know the Buyer's fingerprint. The Buyer first commits to a secret (the fingerprint) that only he/she knows. The Buyer and Provider then follow a protocol which results in the Buyer receiving a copy of the Work with his/her secret fingerprint (and some additional information coming from the Provider) embedded within it. The Provider does not learn the Buyer's secret, and cannot therefore create a forgery. Unfortunately, the early implementations of this concept were not practical due to the very long length of the collusion resistant codes. The advent of Tardos codes has reduced the length of the collusion resistant codes to a practical size. However, generation of these codes depends on a probability distribution based on a secret vector that is only known to the Provider. This knowledge is sufficient for the Provider to circumvent traditional asymmetric fingerprinting protocols.

In the next Section, we briefly summarize the design of Tardos codes. We then describe how an untrustworthy Provider, with knowledge of the secret vector needed to generate the Tardos codes, can false accuse an innocent Buyer. Section 3 then describes a new asymmetric fingerprinting protocol specific to the use of Tardos codes, that prevents both the Buyer and the Provider from cheating. Practical aspects of the fingerprints embedding and accusation are discussed in Section 4, while security and efficiency of the whole scheme are discussed in Section 6.

2 Untrustworthy Provider with the Tardos Code

For readers unfamiliar with Tardos codes, we now provide a brief introduction. Further details can be found in [18].

2.1 Introduction to Tardos Codes

Let n denote the number of buyers, and m the length of the collusion-resistant codes. The fingerprints can then be arranged as a binary $n \times m$ matrix $\mathbf{X}$, where Buyer j's binary fingerprint is the jth row of the matrix, i.e. $\mathbf{X}_j = (X_{j1}, X_{j2}, \ldots, X_{jm})$.

To generate this matrix, m real numbers $p_i \in [t, 1-t]$ are generated, each of them being randomly and independently drawn according to the probability density function $f : [t, 1-t] \to \mathbb{R}^+$ with $f(z) = \kappa(t)(z(1-z))^{-1/2}$ and

$\kappa(t)^{-1} = \int_t^{1-t}(z(1-z))^{-1/2}dz$. The parameter $t \ll 1$ is referred to as the cutoff whose value is around $1/300c$. The resulting vector, $\mathbf{p} = (p_1, \ldots, p_m)$ is a secret only known by the Provider. Each element of the matrix $\mathbf{X}$ is then independently randomly drawn, such that the probability that the element X_{ji} is set to symbol '1' is $\mathbb{P}(X_{ji} = 1) = p_i$. The collusion-resistant fingerprint, $\mathbf{X}_j$, is then embedded into Buyer j's copy of the Work. This embedding can be accomplished by a variety of watermarking techniques.

When an unauthorized copy is found, a binary sequence, $\mathbf{Y}$, is extracted from the copy thanks to the watermark decoder. Due to collusion and possible distortions such as transcoding, this binary sequence is unlikely to exactly match one of the fingerprints in the matrix $\mathbf{X}$. To determine if Buyer j is involved in the creation of the unauthorized copy, a score, referred to as an accusation score, S_j is computed. If this score is greater than a given threshold Z, then Buyer j is considered to have colluded. The value of the threshold Z theoretically guarantees that the probability of accusing an innocent person is below a significance level, ϵ.

The scores are computed according to an accusation function g, reflecting the impact of the correlation between the fingerprint $\mathbf{X}_j$, associated with Buyer j, and the decoded sequence $\mathbf{Y}$:

$$S_j = G(\mathbf{Y}, \mathbf{X}_j, \mathbf{p}) = \sum_{i=1}^{m} g(Y_i, X_{ji}, p_i). \tag{1}$$

In the usual symmetric codes [18], the function g is constrained (for example, for an innocent person, the expectation of the score is zero and its variance is m), giving $g(1,1,p) = g(0,0,1-p) = -g(0,1,p) = -g(1,0,1-p) = \sqrt{\frac{1-p}{p}}$.

2.2 Untrustworthy Content Provider

We now consider the case where the Provider is no longer trusted, and wishes to frame Buyer j. There are a number of scenarios, depending on the knowledge available to the Provider. We briefly outline these and discuss our specific scenario in detail.

The Provider Knows the Buyer's Fingerprint and How to Embed the Corresponding Watermark. This scenario provides no protection to the Buyer. The Provider can simply watermark a Work with the fingerprint of Buyer j, place the Work in an incriminating location and then accuse Buyer j.

The Provider Knows the Buyer's Fingerprint. In this scenario the Provider does not have the ability to watermark a Work. Instead, upon a Provider's request, a trusted Technology Provider embeds the fingerprint into a Work and sends the fingerprinted Work to the Buyer. We emphasize that the Technology Provider is trusted, and as such, the Provider cannot embed the same fingerprint into a Work and have it delivered to two different users, one of which is colluding with the Provider to frame the other user. If the Technology Provider were not trusted, we would be back to the previous scenario.

All the Provider needs is fingerprinted copies from $c \geq 3$ fake users or colluders. There is nothing special about the particular fingerprints. For a given Buyer j, whom the Provider wishes to frame, the Provider knows where the elements of the Buyer's fingerprint $X_{ji} = 1$. This happens with probability p_i. At least one of the accomplices has the same symbol as the Buyer with a probability of $1 - (1 - p_i)^c$. Therefore, given that the Provider knows the Buyer's fingerprint, $\mathbf{X}_j$, the accomplices can forge a sequence very similar to the fingerprint of Buyer j. More specifically, if $Y_i = X_{ji}$ whenever the marking assumption allows it, then the forgery is such that, in expectation, the score of Buyer j becomes:

$$\begin{aligned} S_j &= m \int_t^{1-t} f(p) \left[p(1-(1-p)^c) g(1,1,p) + (1-p)(1-p^c) g(0,0,p) \right. \\ &\quad \left. + \; p(1-p)^c g(0,1,p) + (1-p)p^c g(1,0,p) \right] dp \\ &= 2m\kappa(t) \left((1-2t) - 2\frac{(1-t)^{c+1} - t^{c+1}}{c+1} \right) \approx 2m\kappa(t)(1 - \frac{2}{c+1}) \end{aligned} \tag{2}$$

In comparison, the colluders have scores equalling $2m\kappa(t)c^{-1}$ in expectation. This means that with only $c = 3$ accomplices, the score of Buyer j is bigger than the ones of the colluders, which are bigger than Z if the code is long enough to face a collusion of size 3 (depending on the parameters (n, ϵ)). The Provider sends $(\mathbf{X}_j, \mathbf{Y}, \mathbf{p}, Z)$ to the Judge as an evidence to accuse Buyer j. This attack is just an example, there certainly exists a better way to frame an innocent.

The Provider Knows the Bias Vector p. The previous two scenarios demonstrate that the Provider must not know the fingerprints of the Buyers, if the Buyers are to be protected. This is well known in the literature of asymmetric fingerprinting. However, another threat occurs when dealing with Tardos codes. In this scenario, the Provider has no knowledge of the Buyer's fingerprint, nor the underlying watermark method. We therefore assume that the Provider cannot forge an unauthorized copy, either on his/her own or with accomplices. On receipt of a pirated copy, the sequence is extracted by the trusted Technology Provider. Given the extracted sequence $\mathbf{Y}$, the scores of all Buyers are computed using Equation (1). It is here that the Provider can lie, since the probabilities in $\mathbf{p}$ are only known by the Provider.

Specifically, an untrustworthy Provider can create a fake vector of probabilities $\hat{\mathbf{p}}$ that implicates Buyer j. However, the distribution $f(p)$ is publicly known, so the question becomes how to generate a $\hat{\mathbf{p}}$ that (i) implicates Buyer j, and (ii) has an arbitrarily high probability of been drawn from the distribution $f(p)$?

The following method shows that it is simple to do so. However, we do not claim that this attack is unique or optimal. Let us focus on a column where $p_i = p$ and $Y_i = X_{j,i}$. The true summand in Equation (1) is $g(1,1,p)$ or $g(0,0,p)$ (with equal probability). Suppose that the content provider replaces the secret value p by a fake secret $\hat{p}$ which is drawn independently according to f. On average, this summand takes the new value:

$$\Delta(t) = \int_t^{1-t} f(\hat{p}) \frac{g(1,1,\hat{p}) + g(0,0,\hat{p})}{2} d\hat{p} = \kappa(t) \ln \frac{1-t}{t}.$$

For a cutoff $t = 1/900$ (recommended by G. Tardos to fight against 3 colluders), $\kappa(t) \approx \pi^{-1}$ and the numerical value is surprisingly high: $\Delta(1/900) \approx 2.16$. Suppose now that the content provider applies the same strategy on an index i where $Y_i \neq X_{j,i}$. Then the expectation is the opposite. However, in a Tardos code, even for an innocent Buyer j, the proportion α of indices where symbols Y_i and $X_{j,i}$ agree is above $1/2$ for common collusion strategies. For instance, with an interleaving collusion attack [18], $\alpha = 3/4$ whatever the collusion size c.

Based on this fact, we propose the following attack. The Provider computes the score for all Buyers, which on average equals 0 for innocent Buyers and $2m\kappa(t)c^{-1}$ for the colluders [18]. The Provider initializes $\hat{\mathbf{p}} = \mathbf{p}$. Then, he/she randomly selects a column i and randomly draws a fake secret $\hat{p_i} \sim f$. He/She re-computes the score of Buyer j with this fake secret and iterates selecting a different column until S_j is above the threshold Z. On average, $m(c\kappa(t)^{-1}\Delta(t)(\alpha - 1/2))^{-1}$ secret values p_i need to be changed in this way, e.g. only 20% of the code length if the copy has been made using an interleaving attack.

Figure 1 illustrates this attack for the case where the code length is $m = 1000$ and the number of colluders is $c = 3$. The solid coloured lines depict the accusation scores of 10 randomly selected innocent buyers. We observe that after 20 to 30% of the elements of $\mathbf{p}$ have been altered, the accusation scores of the innocent Buyers exceed the *original* scores of the colluders. In fact, the colluders' accusation scores also increase. However, we are not concerned by the highest score, but rather by the fact that the Provider is able to exhibit a couple $(\hat{\mathbf{p}}, \mathbf{X}_j)$ such that $S_j > Z$. Thus, it is sufficient to raise the score of the innocent Buyer, even if this raises all other Buyers' scores as well.

Randomly selecting some p_i's (independently from $\mathbf{X}_j$ and $\mathbf{Y}$) and re-drawing them according to the same law ensures that $\hat{p_i} \sim f$, $\forall i$. Therefore, the Judge observing $\hat{\mathbf{p}}$ cannot distinguish the forgery. For this reason, the Judge might request to see the matrix $\mathbf{X}$ to statistically test whether the elements of $\mathbf{X}$ are drawn from the distribution $\hat{\mathbf{p}}$. In this case, the Provider can give a fake matrix $\hat{\mathbf{X}}$ where the columns whose p_i have been modified are re-drawn such that $\mathbb{P}(X_{ki} = 1) = \hat{p_i}$, $\forall k \neq j$. The only way to prevent this deception would be for the Judge to randomly asked an innocent Buyer $k \neq j$ for his copy in order to verify the authenticity of $\hat{\mathbf{X}}$. This latter step seems somewhat odd. We arrive at the strange situation where the Judge has to contact innocent buyers when Buyer j is accused.

3 An Asymmetric Tardos Code Construction

The previous section underlines the difficulty of constructing an asymmetric fingerprinting protocol using Tardos codes. The constraints are:

- The Provider should not know the fingerprints.
- The Provider should not change the secret $\mathbf{p}$ used for the code construction during the accusation score computation.
- The Buyer should know neither the secret $\mathbf{p}$ nor the fingerprint of any other user.

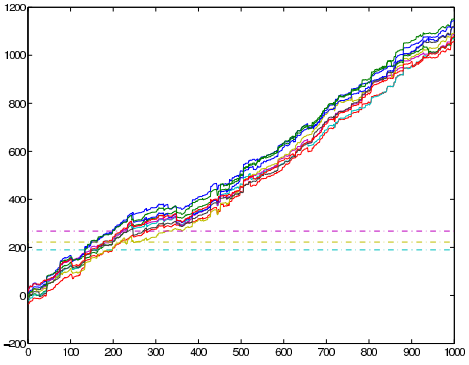


Fig. 1. Accusation score as a function of the number of changed elements of the vector $\mathbf{p}$ for the case where $m = 1000$ and $c = 3$. The solid coloured lines show how the accusation scores of 10 randomly selected innocent buyers increases. The dotted horizontal lines show the original scores for the colluders before the modification.

- His fingerprint must be drawn according to the statistical distribution induced by $\mathbf{p}$.
- The Buyer should not be able to modify his fingerprint.

These constraints prevent the application of previous asymmetric fingerprinting schemes to a Tardos code. This section proposes a solution to this problem, which consists of two phases: the generation of the fingerprint and the disclosure of a halfword. Both phases rely on a primitive which we present first.

3.1 Pick a Card, Any Card!

What we need is a scheme that enables a receiver $\mathbf{R}$ to pick k elements at random in a list of N elements provided by a sender $\mathbf{S}$, in such a way that:

1. $\mathbf{R}$ gets elements that belong to the list;
2. $\mathbf{R}$ does not get any information on the elements he did not pick;
3. $\mathbf{S}$ does not know which elements have been picked.

Functionally speaking, this is precisely what is called *Oblivious Transfer* by cryptographers. A k-out-of-N Oblivious Transfer protocol is denoted by OT_k^N. In the literature we can find OT_1^2, OT_1^N and OT_k^N protocols. When $k \geq 1$, if the k elements are picked one-by-one adaptively, we speak of *adaptive OT protocols*, denoted by $OT_{k\times 1}^N$; if they are picked simultaneously, we speak of *non-adaptive OT protocols*, simply denoted OT_k^N .

Technically speaking, the oblivious transfer problem has been independently tackled by two communities. First, Cryptographers have been working on it since 1981. We will refer to this quite long and mature framework as "traditional" OT. Second, in 2001 other researchers proposed a different approach based on *Commutative Encryption* and *Two-lock Cryptosystems*. Both are considered and

discussed in Sec. 4, according to their respective advantages. We provide more details on the use of OT protocols based on Commutative Encryption or Two-lock crypto-systems, as they are less known but particularly interesting in our case.

3.2 Phase 1: Generation of the Fingerprint

Fingerprint generation consists of two steps. During Step 1, the Provider generates lists from the secret $\mathbf{p}$, and commits them in order to avoid any *a posteriori* cheating. During Step 2, the Buyer picks elements in the lists to generate his own fingerprint. This step is addressed by oblivious transfer protocols.

Step 1. We use the commutative encryption protocol m times to generate the fingerprint of the j-th Buyer $\mathbf{X}_j = (X_{j,1}, \ldots, X_{j,m})$. $\mathbf{S}$ is the Provider, and $\mathbf{R}$ is Buyer j. The Provider generates a secret vector $\mathbf{p}$ for a Tardos code. Each p_i is quantized such that $p_i = L_i/N$ with $L_i \in [N-1]$.

For a given index i, the objects are the concatenation of a binary symbol and a text string. There are only two versions of an object in list $\mathcal{C}_i$. For L_i objects, $O_{k,i} = (1 \| \mathtt{ref}_{1,i})$, and $O_{k,i} = (0 \| \mathtt{ref}_{0,i})$ for the $N - L_i$ remaining ones. The use of the text strings $\{\mathtt{ref}_{X,i}\}$ depends on the content distribution mode as detailed in Sec. 4.3. The object $O_{k,i}$ is committed with key $K_{k,i}$ and stored in the list $\mathcal{C}_i = \{C_{k,i}\}_{k=1}^{N}$. There are thus as many different lists $\mathcal{C}_i$ as the length m of the fingerprint. These lists are the same for all buyers, and are published in a public Write Once Read Many (WORM) directory [**?**] whose access is granted to all users. As the name, nobody can modify or erase what is initially written in a WORM directory, but anyone can read from it.

$$
\begin{array}{lcccl}
p_1 & \overset{Quantize}{\longrightarrow} & (0\|\mathtt{ref}_{0,1}, 1\|\mathtt{ref}_{1,1}, \ldots, 1\|\mathtt{ref}_{1,1}) & \overset{Commit}{\longrightarrow} & \mathcal{C}_1 = (C_{1,1}, \ldots, C_{N,1}) \\
p_2 & \longrightarrow & (0\|\mathtt{ref}_{0,2}, 0\|\mathtt{ref}_{0,2}, \ldots, 0\|\mathtt{ref}_{0,2}) & \longrightarrow & \mathcal{C}_2 = (C_{1,2}, \ldots, C_{N,2}) \\
 & & \vdots & & \\
p_m & \longrightarrow & (1\|\mathtt{ref}_{1,m}, 0\|\mathtt{ref}_{0,m}, \ldots, 1\|\mathtt{ref}_{1,m}) & \longrightarrow & \mathcal{C}_m = (C_{1,m}, \ldots, C_{N,m})
\end{array}
$$

Fig. 2. The lists $\mathcal{C}_i = \{C_{k,i}\}_{k=1}^{N}$ are stored in a WORM

Step 2. If we use a traditional Oblivious Transfer protocol, the Buyer and Provider run it to get the corresponding key $K_{\mathsf{ind}(j,i),i}$: the Provider proposes the list of the keys $\{\pi_j(k) \| K_{\pi_j(k),i}\}$ and the Buyer picks one with an OT_1^N. This key allows him to open one of the commitments $C_{\pi_j(k),i}$. Provider and Buyer will have to keep in a log file some elements of the exchange in order to run the Phase 2. It is specific to the *OT* protocol and we have not studied this problem in detail.

Let us now describe how to solve the problem with a Commutative Encryption scheme. Contrary to the $\mathcal{C}$-lists, the $\mathcal{D}$-lists are made specific to a given Buyer

j. The Provider picks a secret key S_j and a permutation $\pi_j(.)$ over $[N]$. The Buyer is given a list $\mathcal{D}_{j,i} = \{D_{j,i,k} = \mathbf{CE}(S_j, (\pi_j(k)\|K_{\pi_j(k),i}))\}_{k=1}^{N}$. Therefore, the lists $\{\mathcal{C}_i\}_{i=1}^{m}$ are common for all users, whereas the lists $\{\mathcal{D}_{j,i}\}_{i=1}^{m}$ are specific to Buyer j. We have introduced here a slight change with respect to protocol 4.1, i.e. the permutation π_j whose role is explained below. Buyer j chooses one object in the list, say the $k(j,i)$-th object. He/she sends the corresponding ciphertext $U_{k(j,i),i} = \mathbf{CE}(R_{j,i}, D_{j,i,k(j,i)})$ decrypted by the provider with S_j and sent back to the Buyer who, at the end, gets the index $\mathsf{ind}(j,i) = \pi_j(k(j,i))$ and the key $K_{\mathsf{ind}(j,i),i}$, which grants him/her the access to the object $O_{\mathsf{ind}(j,i),i}$, stored in encrypted form in the WORM. It contains the symbol $b_{\mathsf{ind}(j,i),i}$. This becomes the value of the i-th bit of his/her fingerprint, $X_{j,i} = b_{\mathsf{ind}(i,j),i}$, which equals '1' with probability p_i. The provider keeps in a log file the values of S_j and $U_{k(j,i),i}$, and the user keeps $R_{j,i}$ in his/her records.

	P		B
1.	**Commit** $C_{\pi_j(k),i} = \mathbf{Com}(K_{\pi_j(k),i}, O_{\pi_j(k),i})$		
	Only one time for all users and all bits		
2.	$D_{j,i,k} = \mathbf{CE}(S_j, (\pi_j(k)\|K_{\pi_j(k),i}))$		
3.		$\longleftarrow$	$U_{k(j,i),i} = \mathbf{CE}(R_{j,i}, D_{j,i,k(j,i)})$
4.	$V = \mathbf{CE}^{-1}(S_j, U_{k(j,i),i})$	$\longrightarrow$	
5.			$\pi_j(k)\|K_{\pi_j(k),i} = \mathbf{CE}^{-1}(R_{j,i}, V)$
6.			$O_{\pi_j(k),i} = \mathbf{E}^{-1}(K_{\pi_j(k),i}, C_{\pi_j(k),i})$

Fig. 3. Generation of a fingerprint bit using the Commutative Encryption Scheme

3.3 Phase 2: Disclosure of the Halfword

The accusation process detailed in Sec. 4.4 allows the Provider to list a set of suspected users to be forwarded to the judge for verification. After phase 1 is completed, the Provider orders Buyer j to reveal $m_h < m$ bits of his fingerprint. These disclosed symbols compose the so-called halfword [16]. The following facts must be enforced: Buyer j does not know which bits of his/her fingerprint are disclosed even if the Provider asks for the same bit indices to all the users. The Provider discloses m_h bits of the fingerprints without revealing any knowledge about the others. Of course, Buyer j refuses to follow the protocol for more than m_h objects.

Commutative Encryption. Again, we propose to use the double-blind random selection protocol of Sec. 3.1. Now, Buyer j plays the role of **S**, and the Provider the role of **R**, $N = m$, and object $O_i = (R_{i,j}\|\mathtt{alea}_{i,j})$. These items are the m secret keys selected by Buyer j during phase 1 (Sec. 3.2) concatenated with random strings $\mathtt{alea}_{i,j}$ to be created by Buyer j. This alea finds its use during the personalization of the content (see Sec. 4.3). Following the protocol, the Provider selects m_h such object. The decryption of message $U_{k(i,j),j}$ received during phase 1 thanks to the disclosure of the key $R_{i,j}$ yields $D_{i,j,k(i,j)}$ which

in turn is decrypted with key S_j, provides the index of the selected object, otherwise the protocol stops. This prevents a colluder from denying the symbol of his fingerprint and from copying the symbol of an accomplice. At the end, the Provider learns which item was picked by Buyer j at index i. Therefore, he/she ends up with m_h couples $(X_{j,i}, \mathtt{alea}_{k(i,j),i})$ associated to a given Buyer j.

Generic Oblivious Transfer protocols. At phase 2, any $OT^N_{k\times 1}$ can be used to allow the Provider to get m_h objects from the list of the $O_i = (R_{i,j} \| \mathtt{alea}_{i,j})$ owned by the Buyer. The problem is if another OT scheme was used at the precedent step, there is no such things as the $R_{i,j}$ values. In order to prevent the Buyer from denying the symbol of his fingerprint, the $R_{i,j}$ values have to be replaced by a number which was part of the exchange during the generation of the fingerprint. This element is specific to the OT protocol.

4 Implementation Details

The previous section has detailed the core of our scheme which is the construction of the codewords based on oblivious transfer. This section deals with the details of this primitive and the remaining elements, namely the watermarking of video content, the distribution and the accusation process.

4.1 Details of the Oblivious Transfer Protocol

This protocol can be implemented by two approaches, 'classical' Oblivious transfer and Commutative encryption, which have been studied with different security models. Both are interesting for us, and we will now summarize them and discuss their usefulness

Traditional Oblivious Transfer protocols. Oblivious Transfer Protocols have been introduced by cryptographers in [17] and led to a huge number of papers in the cryptographic community, *e.g.* [13,5,10]. These protocols are studied in the same framework as multi-party computation. Their security is studied under different models below, listed from the weakest to the strongest: honest-but-curious model (where no one cheats during the protocol execution), half simulation (introduced by [14], cheating sender or cheating receiver studied separately; local security study), full simulation (introduced in [3], studying cheating sender and receiver globally; global security study). In addition, the UC (Universally Composable) model has been introduced in [4] to study the behavior and security of protocols that are based on concurrent and composable cryptographic primitives.

Oblivious Transfer based on Commutative Encryption. An encryption primitive **CE** is said to be a *Commutative Encryption* if for any two keys k_R and k_S and any plaintext m, we have (usual definition in the literature)

$$\mathbf{CE}(k_R, \mathbf{CE}(k_S, m)) = \mathbf{CE}(k_S, \mathbf{CE}(k_R, m)). \tag{3}$$

Based on such a primitive, a *Commutative Encryption Scheme (CES)* can be defined as follows [1].

1. Let m_1, m_2, ...,m_N be the N inputs of the Sender **S**. **S** chooses N secret keys K_1, K_2, ..., K_N for a symmetric cryptosystem **E** (*e.g.* AES, DES) and a key k_S for the commutative encryption primitive **CE**. **S** provides

$$\begin{aligned} C_1 &= \mathbf{E}(K_1, m_1)\ , \quad & D_1 &= \mathbf{CE}(k_S, K_1) \\ C_2 &= \mathbf{E}(K_2, m_2)\ , \quad & D_2 &= \mathbf{CE}(k_S, K_2) \\ &\dots \quad & &\dots \\ C_N &= \mathbf{E}(K_N, m_N)\ , \quad & D_N &= \mathbf{CE}(k_S, K_N) \end{aligned}$$

   Note that the couples $\langle C_j,\ D_j \rangle$ can be publicly accessed.

2. Now, let us assume that the receiver **R** wants to pick the i-th element of the list. **R** loads $\langle C_i,\ D_i \rangle$ and chooses a secret key k_R for **CE**. He encrypts D_i with it and sends the result $U = \mathbf{CE}(k_R, D_i)$ to **S**.
3. **S** decrypts U with S and sends $W = \mathbf{CE}^{-1}(k_S, U)$ to **R**. **R** computes K_i, and can get to $m_i = \mathbf{E}^{-1}(K_i, C_i)$.

A *Two-lock Cryptosystem* is a variant that uses two different primitives **CE1** and **CE2** instead of **CE**:

$$\mathbf{CE1}(k_R, \mathbf{CE2}(k_S, m)) = \mathbf{CE2}(k_S, \mathbf{CE1}(k_R, m)). \tag{4}$$

Both approaches are interesting for us, as we will discuss now. First of all, the security of Oblivious Transfer Protocols has been much stronger studied than the one of the Commutative Encryption Schemes. Hence, we will use them each time it is possible, leaning on well known protocols.

But, at some steps of the protocol we prefer to use Commutative Encryption Schemes, as its structure fits really well to our purpose. It is for example the case during fingerprint generation, as we also want the Provider to commit on the lists elements, which correspond to the secret vector Tardos accusation will rely on. This ensures that the same secret vector will be used during the accusation process. Such commitments are easily included in a Commutative Encryption Scheme, it is more difficult in a traditional Oblivious Transfer protocol. In addition, we use some elements exchanged during the course of the protocol in phase 1 (Sec. 3.2) to ensure the correct conduct of the Phase 2 (Sec. 3.3).

Designing the right Commutative Encryption Scheme is not so easy, as the literature does not provide us a scheme that fulfill our requirements. First of all, notice that using a symmetric or asymmetric encryption primitive as **CE**, or in the variant scheme **CE1** and **CE2**, does not matter here, functionally speaking, as encryption and decryption will be performed by the same person. Hence, only security and eventually efficiency may guide our choice. Of course, we would like to use the most secure encryption primitives. The highest security level, *unconditional security* is only reached by the One-Time Pad, and cannot be achieved here because it would require to use a different key for each encryption whereas here the same key k_S is used to encrypt all the keys K_i. Hence, *semantic security* is the best security class we might achieve [9,20,7]. Moreover, semantic security

is necessary in our case, because we have to encrypt binary symbols and do not want the Receiver to be able to distinguish encrypted 0's from encrypted 1's during both the fingerprint generation or the halfword disclosure steps. This implies the use of a probabilistic encryption scheme. Unfortunately, semantic security has not yet been tackled in the Commutative Encryption literature [1,11,21]. Nevertheless, semantic security should be achieved in a near future, making this kind of OT particularly interesting for us.

Concerning the variant called Two-lock Cryptosystem, a few implementations have been proposed: a first one based on the Knapsack problem [21], which has been broken [22], a second one based on the discrete logarithm problem [21], and a third one based on RSA [11]. None of them achieve semantic security at the moment.

4.2 Watermarking

A nowadays trend is the application of fingerprinting to premium video contents. Premium means movies in very high quality available for home cinema shortly after their release in theaters. Personalization of the copies are usually done as follows: Before distribution, the content is divided into sequential blocks (e.g. Group of Pictures of few seconds of a video). Offline, a robust watermarking technique creates two versions of some blocks embedding the symbol '0' and respectively '1'. This is done by the Technology Provider. Quality is very important for premium movies and watermarking under that constraint involves a lot of processing. This motivates this offline preprocessing.

In some scenarios (screeners for jurys, marketing, blu-ray discs, premium downloads), the physical medium storage or bandwitth is so large that both versions of the blocks are encrypted and transmitted to the software client or the device of the Buyers. This latter is trusted and the strings $\{\texttt{ref}_{X,i}\}$ it got from phase 1 are parameters needed to get access to the i-th block watermarked with symbol X.

4.3 Content Personalization at the Server Side

As for Video On Demand where the client is not trusted, personalization of the content is usually made at the server side, which raises an issue since the Provider doesn't know user fingerprints. There exist Buyer-Seller protocols for embedding a sequence $\mathbf{X}_j$ into a content c_o without disclosing $\mathbf{X}_j$ to the Seller and c_o to the Buyer. They are based on homomorphic encryption scheme and work with some specific implementations of spread spectrum [12] or Quantization Index Modulation watermarking [6]. In other words, not any watermarking technique can be used, and this is not the route we have chosen so far. Due to space limitations, a brief sketch of the adaptation of [6] is presented hereafter.

Let $\mathbf{c}_i^{(0)} = (c_{i,1}^{(0)}, \ldots, c_{i,Q}^{(0)})$ be the Q quantized components (like pixels, DCT coefficients, portion of streams etc) of the i-th content block watermarked with symbol '0' (resp. $\mathbf{c}_i^{(1)}$ with symbol '1'). Denote $\mathbf{d}_i = \mathbf{c}_i^{(1)} - \mathbf{c}_i^{(0)}$. Assume as in

[6, Sect. 5], an additive homomorphic and probabilistic encryption $E[.]$ such as the Pallier cryptosystem. Buyer j has a pair of public/private keys (pk_j, sk_j) and sends $(E_{pk_j}[X_{j,1}], \ldots, E_{pk_j}[X_{j,m}])$. The provider sends him/her the ciphers

$$E_{pk_j}[c_{i,\ell}^{(0)}].E_{pk_j}[X_{j,i}]^{d_{i,\ell}}, \forall (i,\ell) \in [m] \times [Q]. \tag{5}$$

Thanks to the homomorphism, Buyer j decrypts this with sk_j into $c_{i,\ell}^{(0)}$ if $X_{j,i} = 0$, $c_{i,\ell}^{(1)}$ if $X_{j,i} = 1$. Since $X_{j,i}$ is constant for the Q components of the i-th block, a lot of bandwidth and computer power will be saved with a composite signal representation as detailed in [6, Sect. 3.2.2].

A crucial step in this kind of Buyer-Seller protocols is to prove to the Provider that what is sent by the Buyer is indeed the encryption of bits, and moreover bits of the Buyer's fingerprint. This usually involves complex zero-knowledge subprotocols [12,6]. Here, we avoid this complexity by taking advantage of the fact that the Provider already knows some bits of the fingerprint $\mathbf{X}_j$, i.e. those belonging to the halfword (see Sec. 3.3), and the Buyers do not know the indices of these bits. Therefore, in $m_v < m_h$ random indices of the halfword, the Provider asks Buyer j to open his/her commitment. For one such index i_v, Buyer j reveals the random value r_{i_v} of the probabilistic Pallier encryption (with the notation of [6]). The Provider computes $g^{X_{j,i_v}} h^{r_{i_v}} \mod N$ and verifies it equals the i_v-th cipher, which Buyer j pretended to be $E_{pk_j}[X_{j,i}]$.

One drawback of this simple verification scheme is that the Buyer discovers m_v indices of the halfword. This may give rise to more elaborated collusion attacks. For example, Buyer j, as a colluder, could try to enforce $Y_{i_v} \neq X_{j,i_v}$ when attempting to forge a pirated copy. Further discussion of this is beyond the scope of this paper.

This approach may also introduce a threat to the Buyer. An untrustworthy Provider can ask to open the commitments of non-halfword bits in order to disclose bits he/she is not supposed to know. For this reason, the Provider needs to send $\texttt{alea}_{k(i_v,j),i_v}$ as defined in Sec. 3.3 to show Buyer j that his/her verification duly occurs on a halfword bit.

4.4 The Accusation Procedure

The accusation is straightforward and similar to other fingerprinting protocols. A Scouting Agency is in charge of catching a forgery. The Technology Provider decodes the watermark and extracts sequence $\mathbf{Y}$ from the pirated content. The Provider computes the halfscores by applying Eq. (1) only on the halfwords. This produces a list of suspects, e.g. those users whose score is above a threshold, or those users with the highest scores.

Of course, this list cannot be trusted, since the Provider may be untrustworthy. The list is therefore sent to a third party, referred to as the Judge, who first verifies the computation of the halfscores. If different values are found, the Provider is black-listed. Otherwise, the Judge computes the scores of the full fingerprint.

To do so, the Judge needs the secret $\mathbf{p}$: he/she asks the Provider for the keys $\{K_{k,i}\}$, $\forall(k,i) \in [N] \times [m]$ and thereby obtains from the WORM all the objects $\{O_{k,i}\}$, and the true values of $(p_1, \ldots, p_m)$. The Judge must also request suspected Buyer j for the keys $R_{j,i}$ in order to decrypt the messages $U_{k(j,i),i}$ in $D_{i,j,k(i,j)}$ which reveal which object Buyer j picked during the i-th round of Sec. 3.2 and whence $X_{j,i}$. Finally, the Judge accuses the user whose score over the full length fingerprint is above a given threshold (related to a probability of false alarm).

5 Discussion

5.1 Security

Suppose first that the Provider is honest and denote by c the collusion size. A reliable tracing capability on the halfwords is needed to avoid false alarms. Therefore, as proven by G. Tardos, $m_h = O(c^2 \log n\epsilon^{-1})$, where ϵ is the probability of suspecting some innocent Buyers. Moreover, successful collusions are avoided if there are secret values such that $p_i < c^{-1}$ or $p_i > 1 - c^{-1}$(see [8]). Therefore, N should be sufficiently big, around a hundred, to resist against collusion of size of some tens. During the generation of the fingerprint in Sec. 3.2, permutation $\pi_j(.)$ makes sure that Buyer j randomly picks up a bit '1' with probability $p_i = L_i/N$ as needed in the Tardos code. In particular, a colluder cannot benefit from the discoveries made by his accomplices.

We now analyze why colluders would cheat during the watermarking of their version of the Work described in Sec. 4.3. By comparing their fingerprints, they see indices where they all have the same symbols, be it '0' or '1'. As explained in the introduction, they won't be able to alter those bits in the tampered fingerprint except if they cheat during the watermarking: If their fingerprint bits at index i all equal '1', one of them must pretend he/she has a '0' in this position. If they succeed to do so for all these positions, they will able to forge a pirated copy with a null fingerprint for instance.

How many times do the colluders need to cheat? With probability p_i^c (resp. $(1-p_i)^c$), they all have bit '1' (resp. '0') at index i. Thus, there are on average $m_c(c) = m\int_t^{1-t}(p^c + (1-p)^c)f(p)dp$ such indices. The Provider asks for a bit verification with probability m_v/m_h. The probability of a successful attack for a collusion of size c is therefore $(1 - m_v/m_h)^{m_c(c)}$. Our numerical simulations (see figure 4 (a)) show that m_v shouldn't be more than 50 bits for typical code length and collusion size below a hundred. Thus, m_v is well below m_h.

Suppose now that the Provider is dishonest. The fact that the m lists $\mathcal{C}_i$, $\forall i \in [m]$ are public and not modifiable prevents the Provider from altering them for a specific Buyer in order to frame him/her afterwards. Moreover, it will raise the Judge's suspicion if the empirical distribution of the p_i is not close to the pdf f. Yet, biases can be introduced on the probabilities for the symbols of the colluders' fingerprint only if there is a coalition between them and the untrustworthy Provider. For instance, the Provider can choose a permutation such that by selecting the first item (resp. the last one) in the list $\mathcal{D}_{j,i}$ an accomplice colluder

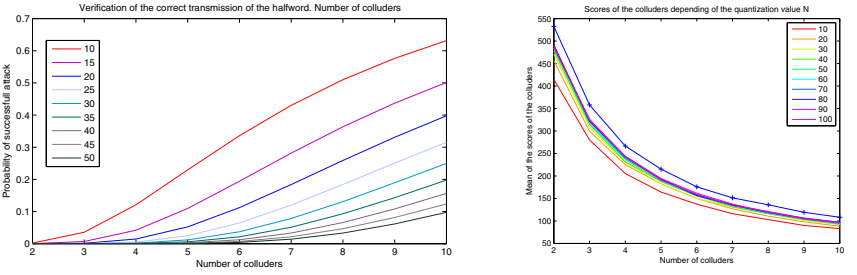


Fig. 4. (a) m_v goes from 10 to 50 by 5, $m = 3000$ and $m_h = 1500$. (b) N goes from 10 to 100 and $m = 1500$. The gray curve with crosses is for the unquantified Tardos code.

is sure to pick up a symbol '1' (resp. '0'). This ruins the tracing property of the code, but this does not allow the Provider to frame an innocent. First, it is guaranteed that **p** used in Eq. (1) is the one which generated the code. Second, the Provider and his accomplices colluders must ignore a significant part of the fingerprints of innocent Buyers. To this end, $m - m_h$ must also be in order of $O(c^2 \log n\epsilon^{-1})$. If this holds, the Judge is able to take a reliable decision while discarding the halfword part of the fingerprint. Consequently, $m \approx 2m_h$, our protocol has doubled the typical code length, which is still in $O(c^2 \log n\epsilon^{-1})$.

5.2 Efficiency

Parameters. The parameters of the Tardos code are chosen according to the formulas linking length, number of colluders, and number of users. We have found out that the value m_v doesn't need to be more than 50, see Sec. 4. We consider the value N, the quantization parameter, with the interleaving collusion attack. In the figure 4 (b), we can see that up to a small value of N (around 20), there is no gain of efficiency. The red line shows that the results with the unquantized Tardos parameters remain better.

Complexity. The cost of phase 1 is $m \times N$ commitments for the lists that will be stored in the Worm file, and $mn \times (N+4)$ exponentiations for the OT phase. Regarding the use of a non specific OT, still $m \times N$ commitments, plus the cost of mn 1-out-of-N Oblivious Transfers. This cost depends of course of the chosen protocol, it is in $O(N)$ for a lot of protocols. For Phase 2, the cost is that of an m_h-out-of-m Oblivious transfer. If this OT is performed with the use of a *Commutative Encryption*, the cost is $2m+4m_h$ for the communication, and $4m_h$ rounds, for another OT scheme, the communication is in $O(m)$ and the number of rounds depends of the protocol, it is usually in $O(m_h)$.

6 Conclusion

Tardos codes are currently the state-of-the-art in collusion-resistant fingerprinting. However, the previous asymmetric fingerprint protocols cannot be applied to this particular construction. There are mainly two difficulties. First, the Buyer has to generate his/her secret fingerprint but according to vector $\mathbf{p}$, which is kept secret by the Provider. Second, the secret $\mathbf{p}$ used in the accusation process must be the same as the one which generated the fingerprints.

We have proposed the first asymmetric fingerprinting protocol dedicated to Tardos codes. The construction of the fingerprints and their embedding within pieces of Work do not need a trusted third party. Note, however, that during the accusation stage, a trusted third party is necessary like in any asymmetric fingerprinting scheme we are aware of. Further work is needed to determine if such a third party can be eliminated. In particular, we anticipate that some form of secure multi-party computation can be applied.

We considered two forms of oblivious transfer protocols, the first based on traditional cryptographic techniques and the second based on less well known Commutative Encryption or Two-Lock crypto-systems. These latter techniques are less mature than traditional Oblivious Transfer protocols in terms of security, but offers interesting properties that are convenient to our application. Further work is needed to improve their semantic security, so that their advantages do not come at the cost of decreased security.

Acknowledgement. We would like to thank Boris Škorić, and the three anonymous reviewers for their useful comments, which helped to improve the presentation of our results.

References

1. Bao, F., Deng, R.H., Feng, P.: An efficient and practical scheme for privacy protection in the E-commerce of digital goods. In: Won, D. (ed.) ICISC 2000. LNCS, vol. 2015, pp. 162–170. Springer, Heidelberg (2001)
2. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory (1998)
3. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007)
4. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd IEEE Symposium on Foundations of Computer Science, pp. 136–145. IEEE, Los Alamitos (2002)
5. Chu, C., Tzeng, W.: Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 172–183. Springer, Heidelberg (2005)
6. Deng, M., Bianchi, T., Piva, A., Preneel, B.: An efficient Buyer-Seller watermarking protocol based on composite signal representation. In: ACM MM&Sec 2009, pp. 9–18 (2009)

7. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security 15 (2007)
8. Furon, T., Pérez-Freire, L.: Worst case attack against binary probabilistic traitor tracing codes. In: IEEE WIFS 2009, pp. 46–50 (2009)
9. Goldreich, O.: Foundations of cryptography: Basic applications. Cambridge Univ. Pr., Cambridge (2004)
10. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Heidelberg (2007)
11. Huang, H., Chang, C.: A new design for efficient t-out-n oblivious transfer scheme (2005)
12. Kuribayashi, M.: On the Implementation of Spread Spectrum Fingerprinting in Asymmetric Cryptographic Protocol. EURASIP Journal on Inf. Security (2010)
13. Naor, M., Pinkas, B.: Oblivious transfer with adaptive queries. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, p. 791. Springer, Heidelberg (1999)
14. Naor, M., Pinkas, B.: Computationally secure oblivious transfer. Journal of Cryptology 18(1), 1–35 (2005)
15. Oprea, A., Bowers, K.D.: Authentic time-stamps for archival storage. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 136–151. Springer, Heidelberg (2009)
16. Pfitzmann, B., Schunter, M.: Asymmetric fingerprinting. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 84–95. Springer, Heidelberg (1996)
17. Rabin, M.: How to exchange secrets by oblivious transfer. Tech. rep., Technical Report TR-81, Harvard Aiken Computation Laboratory (1981)
18. Skoric, B., Katzenbeisser, S., Celik, M.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography 46(2), 137–166 (2008)
19. Tardos, G.: Optimal probabilistic fingerprint codes. In: STOC 2003, pp. 116–125. ACM, New York (2003), http://www.renyi.hu/~tardos/publications.html
20. van Tilborg, H.: Encyclopedia of cryptography and security. Springer, Heidelberg (2005)
21. Wu, Q., Zhang, J., Wang, Y.: Practical t-out-n oblivious transfer and its applications. In: Qing, S., Gollmann, D., Zhou, J. (eds.) ICICS 2003. LNCS, vol. 2836, pp. 226–237. Springer, Heidelberg (2003)
22. Zhang, B., Wu, H., Feng, D., Bao, F.: Cryptanalysis of a knapsack based two-lock cryptosystem. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 303–309. Springer, Heidelberg (2004)

"Break Our Steganographic System": The Ins and Outs of Organizing BOSS

Patrick Bas[1], Tomáš Filler[2], and Tomáš Pevný[3]

[1] CNRS - LAGIS, Lille, France
patrick.bas@ec-lille.fr
[2] State University of New York at Binghamton, NY, USA
tomas.filler@gmail.com
[3] Czech Technical University in Prague, Czech Republic
pevnak@gmail.com

Abstract. This paper summarizes the first international challenge on steganalysis called BOSS (an acronym for *Break Our Steganographic System*). We explain the motivations behind the organization of the contest, its rules together with reasons for them, and the steganographic algorithm developed for the contest. Since the image databases created for the contest significantly influenced the development of the contest, they are described in a great detail. Paper also presents detailed analysis of results submitted to the challenge. One of the main difficulty the participants had to deal with was the discrepancy between training and testing source of images – the so-called cover-source mismatch, which forced the participants to design steganalyzers robust w.r.t. a specific source of images. We also point to other practical issues related to designing steganographic systems and give several suggestions for future contests in steganalysis.

1 BOSS: Break Our Steganographic System

During the years 2005 and 2007, the data-hiding community supported by the European Network of Excellence in Cryptology (ECRYPT) launched two watermarking challenges, BOWS [13] and BOWS-2 [1] (abbreviations of *Break Our Watermarking System*). The purpose of participants of both challenges was to break watermarking systems under different scenarios. The purpose of organizers was not only to assess the robustness and the security of different watermarking schemes in the environment similar to real application, but to increase the interest in watermarking and to boost the research progress within the field. Both watermarking contests showed to be popular (BOWS/BOWS2 played more than 300/150 domains and 10/15 participants respectively were ranked), and novel approaches towards breaking watermarking systems were derived during them. This, combined with a thrill associated with organization and participation, inspired us to organize the BOSS (Break Our Steganographic System) challenge.

The most important motivation for the contest was to investigate whether content-adaptive steganography improves steganographic security for empirical

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 59–70, 2011.

covers. For the purpose of this contest, a new spatial-domain content-adaptive algorithm called HUGO (Highly Undetectable steGO) was invented [12]. The fact that in adaptive steganography the selection channel (placement of embedding changes) is publicly known, albeit in a probabilistic form, could in theory be exploited by an attacker. Adaptive schemes introduce more embedding changes than non-adaptive schemes because some pixels are almost forbidden from being modified, which causes an adaptive scheme to embed with a larger change rate than a non-adaptive one. On the other hand, the changes are driven to hard-to-model regions, because the change rate is not an appropriate measure of statistical detectability as it puts the same weight to all pixels. As compared by the state-of-the-art available in mid 2010, HUGO was largely resistant to steganalysis up to 0.4 bits per pixel in 512×512 grayscale images.

The other incentive for organizing the challenge was a hope to encourage the development of new approaches toward steganalysis, pointing to important deadlocks in steganalysis and hopefully finding solutions to them, finding weaknesses of the proposed steganographic system, and finally raising interest in steganalysis and steganography. While running the contest, we became aware of a similar contest organized within the computer vision community [7].

This paper serves as an introduction to a series of papers [4,5,6] describing the attacks on HUGO. Here, we describe the contest, image databases, and the HUGO algorithm to give the papers uniform notation and background.

1.1 Requirements and Rules

In order for BOSS challenge to be attractive and fruitful for the community, we have obeyed the following conditions and limitations.

- All participants were ranked by a scalar criterion, the accuracy of detection on a database of $1,000$ 512×512 grayscale images called BOSSRank. Each image in the BOSSRank database was chosen to contain secret message of size $104,857$ bits (0.4 bits per pixel) with probability 50% (naturally the list of stego and cover images was kept secret).
- In order to ensure that all participants start with the same degree of knowledge about the steganographic system used in the contest, we started the contest with a warm-up phase on June 28, 2010. The very same day the steganographic algorithm HUGO was presented at the International Hiding Conference 2010. For the warm-up phase, we also released the source code of the embedding algorithm. To simplify the steganalysis, a training database of $7,518$ 512×512 grayscale images (the BOSSBase*)* was released along with an implementation of the state-of-the-art feature set (the Cross Domain Features (CDF) [10]) for blind steganalysis. The motivation leading to supply this material, especially the description and implementation of the embedding algorithm, came from the Kerckhoffs' principle.
- We wanted all participants to have an easy access to the score of their predictions, yet prevent them to perform an oracle attack[1] on the evaluation

[1] A method to reach 100% accuracy by learning the true classification of BOSSRank from a very large number of carefully constructed predictions.

Table 1. Camera models and number of images in BOSSBase v0.92 and BOSSRank

Camera model	# of images in *BOSSBase*	# of images in BOSSRank
Leica M9	2267	847
Canon EOS DIGITAL REBEL XSi	1607	0
PENTAX K20D	1398	0
Canon EOS 400D DIGITAL	1354	0
Canon EOS 7D	1354	0
NIKON D70	1033	0
Canon EOS 40D	61	0
Panasonic Lumix DMC-FZ50	0	153

system. To achieve both requirements, the hosting server `http://www.agents.cz/boss` allowed to upload a prediction on BOSSRank once every three days for every IP address. Moreover, the provided score was computed from a subset of 900 randomly selected images. If the detection accuracy was above 65%, the participants could enter the the Hall of Fame.

– To impose a deadline for the participants, the challenge was divided into two phases. The warm-up phase started on June 28, 2010 and ended on September 9, 2010 by publishing the BOSSRank image database used to evaluate the participants. This was immediately followed by a four-month-long period, during which the challenge took its place. The challenge was originally scheduled to end on December 15, 2010, but it was later extended to January 10, 2011.

1.2 Source of Cover Images for BOSS

The BOSS webpage offered two databases of images, the BOSSBase and the BOSSRank.

BOSSBase was composed of $9,074$ never-compressed cover images coming from 7 different cameras.[2] This database was provided as the source of cover images used for the development of steganalyzers. All images were created from full-resolution color images in RAW format (CR2 or DNG). The images were first resized so that the smaller side was 512 pixels long, then they were cropped to 512×512 pixels, and finally converted to grayscale. The whole process was published in a script along with the original images in RAW format and their EXIF headers. Table 1 shows the actual number of images for each camera.

The BOSSRank database was composed of $1,000$ 512×512 grayscale images obtained by the same processing script. 482 of them were randomly chosen to carry the secret payload of approximately 0.4 bpp while keeping the rest without

[2] The *BOSSBase* was released in three phases. On June 28, 2010, the version 0.90 containing 7518 images was released. When the challenged moved to its second phase, the version 0.92 was released with 9074 images. Finally, the version 1.0 containing 10000 images was released in May 2011.

any payload. Participants did not know that 847 images were obtained by Leica M9 in RAW format and 153 images came from Panasonic Lumix DMC-FZ50 captured directly in JPEG[3] format.

The fact that images in both databases came from slightly different sources lead to interesting consequences on steganalyzers trained purely on the BOSS-Base. Although created unintentionally, this cover source mismatch forced the participants to deal with the situation, where the exact source of cover images is not fully known, a problem which surely happens in practice when detecting steganographic communication. Designing steganalyzers which are robust to the cover-source mismatch was one of the main challenges which the participants very quickly realized.

2 HUGO, The Embedding Algorithm for BOSS

The HUGO (Highly Undetectable steGO) algorithm used in the contest hides messages into least significant bits of grayscale images represented in the spatial domain. It was designed to follow the minimum-embedding-impact principle, where we embed a given message while minimizing a distortion calculated between cover and stego images. This strategy allows to decompose its design into two parts: the design of *image model* and the *coder*. The role of the image model is to generate a space in which the distance between points leads to a good distortion function. This function is subsequently used by the coder to determine the exact cover elements that need to be changed in order to communicate the message. In addition, the *optimal coder* minimizes the average distortion calculated over different messages of the same length. The relationship between the size of the payload (embedding rate) and the average distortion is often called the rate–distortion bound. Due to recent development in coding techniques [2,3], we believe that larger gains (in secure payload for example) can be achieved by designing distortion functions more adaptively to the image content instead of by changing the coder. From this reason, when designing HUGO we have focused on the image model.

The image model was largely inspired by the Subtractive Pixel Adjacency Matrix (SPAM) steganalytic features [11], but steps have been taken to avoid over-fitting to a particular feature set [9]. The original publication [12] describes and analyzes several different versions of the algorithm. Here, the most powerful version used in the BOSS competition is described.

2.1 HUGO's Image Model

For the purpose of embedding, each image $\mathbf{X} = (x_{i,j}) \in \mathcal{X} \triangleq \{0, \dots, 255\}^{n_1 \times n_2}$ of size $n_1 \times n_2$ pixels is represented by a feature vector computed from eight three-dimensional co-occurrence matrices obtained from differences of horizontally,

[3] Initially we wanted to use images only from one of the camera in *BOSSBase*, but because of the lack of time we had to use another camera that was not in the training database.

vertically, and diagonally neighboring pairs of pixels. The (d_1, d_2, d_3)th entry of the empirical horizontal co-occurrence matrix calculated from $\mathbf{X}$ is defined as

$$C^{\mathbf{X},\rightarrow}_{d_1,d_2,d_3} = \frac{1}{n_1(n_2-2)} \left|\{(i,j)|D^{\rightarrow}_{i,j} = d_1 \wedge D^{\rightarrow}_{i,j+1} = d_2 \wedge D^{\rightarrow}_{i,j+2} = d_3\}\right|, \quad (1)$$

where $d_1, d_2, d_3 \in [-T, -T+1, \ldots, T]$, $D^{\rightarrow}_{i,j} = x_{i,j} - x_{i,j+1}$ when $|x_{i,j} - x_{i,j+1}| \leq T$. Differences greater that T, $|x_{i,j} - x_{i,j+1}| > T$, are not considered in the model. Co-occurrence matrices for other directions, $k \in \{\leftarrow, \uparrow, \downarrow, \searrow, \nwarrow, \swarrow, \nearrow\}$ are defined analogically. The feature vector defining the image model is $(\mathbf{F}^{\mathbf{X}}, \mathbf{G}^{\mathbf{X}}) \in \mathbb{R}^{2(2T+1)^3}$ with

$$F^{\mathbf{X}}_{d_1,d_2,d_3} = \sum_{k\in\{\rightarrow,\leftarrow,\uparrow,\downarrow\}} C^{\mathbf{X},k}_{d_1,d_2,d_3}, \qquad G^{\mathbf{X}}_{d_1,d_2,d_3} = \sum_{k\in\{\searrow,\nwarrow,\swarrow,\nearrow\}} C^{\mathbf{X},k}_{d_1,d_2,d_3}. \quad (2)$$

The embedding distortion between cover $\mathbf{X}$ and stego image $\mathbf{Y}$, $D(\mathbf{X}, \mathbf{Y})$, is a weighted L_1-norm between their feature vectors:

$$D(\mathbf{X},\mathbf{Y}) = \sum_{d_1,d_2,d_3=-T}^{T} \Big[w(d_1,d_2,d_3) \left|F^{\mathbf{X}}_{d_1,d_2,d_3} - F^{\mathbf{Y}}_{d_1,d_2,d_3}\right| + \\ + w(d_1,d_2,d_3) \left|G^{\mathbf{X}}_{d_1,d_2,d_3} - G^{\mathbf{Y}}_{d_1,d_2,d_3}\right| \Big], \quad (3)$$

where the weights $w(d_1, d_2, d_3)$ quantify the detectability of an embedding change in the (d_1, d_2, d_3)th element of $\mathbf{F}$ and $\mathbf{G}$. The weights were heuristically chosen as

$$w(d_1,d_2,d_3) = \left(\sqrt{d_1^2 + d_2^2 + d_3^2} + \sigma\right)^{-\gamma}, \quad (4)$$

where σ and γ are scalar parameters. For the BOSS challenge, the parameters were set to $\sigma = 1$, $\gamma = 1$, and $T = 90$.

2.2 Embedding

The practical implementation of HUGO embeds the message in pixel's LSBs by using Syndrome-Trellis Code (STC), which were shown [3] to achieve near optimal rate–distortion performance. For the purpose of the challenge, only a simulator of HUGO with the STC coder replaced by a simulated optimal coder operating at the rate–distortion bound was released. This coder modifies ith pixel x_i to $y_i = \arg\min_{z\in\{x_i-1,x_i+1\}} D(\mathbf{X}, z\mathbf{X}_{\sim i})$ with probability

$$p_i = \Pr(Y_i = y_i) = \frac{1}{Z} e^{-\lambda D(\mathbf{X}, y_i\mathbf{X}_{\sim i})}, \quad (5)$$

where Z is a normalization factor and $y_i\mathbf{X}_{\sim i}$ denotes the cover image whose ith pixel has been modified to $Y_i = y_i$ and all other pixels were kept unchanged. The constant $\lambda \geq 0$ is determined by the condition

$$m = -\sum_i p_i \log_2 p_i + (1-p_i)\log_2(1-p_i), \quad (6)$$

which quantifies the desire the communicate m bit long message.

During embedding, whenever a pixel's LSB needs to be changed, the sender has a freedom to choose between a change by +1 or −1 (with the exception of boundaries of the dynamic range). The sender first chooses the direction that leads to a smaller distortion ([3]), embeds the message and then perform the embedding changes. Moreover, in strategy S2 (the most secure version of the algorithm), the embedding changes are performed sequentially and the sender recomputes the distortion at each pixel that is to be modified because some of the neighboring pixels might have already been changed. This step does not change the communicated message and enables HUGO to consider mutual interaction of embedding changes and thus further minimize the statistical detectability.

To illustrate the adaptivity of the algorithm, Figure 1 shows the average probability of changing each pixel in the Lena image[4] estimated by embedding 500 different messages of the same length using the simulated coding algorithm.

(a) 0.25 bpp (b) 0.5 bpp

Fig. 1. Probabilities of pixel being changed during embedding in the Lena image. Probabilities were estimated by embedding 500 different pseudo-random messages with sizes 0.25/0.5 bits per pixel (bpp).

3 Final Results and Analysis of the Submissions

From a large number of received submissions, only 3 participant teams have entered the Hall of Fame, namely A. Westfeld, the team of J. Fridrich called Hugobreakers and finaly the team of G. Gül & F. Kurugollu. Final competition results and scores: (1) Hugobreakers 80.3%, (2) Gül & Kurugollu 76.8%, and (3) A. Westfeld 67%. As can be seen from the number of unique IP addresses from which the BOSSRank image database was downloaded, many other researchers tried to play BOSS. Figure 2 shows the distribution of 142 unique IP addresses among different countries.

[4] Obtained from `http://en.wikipedia.org/wiki/File:Lenna.png`

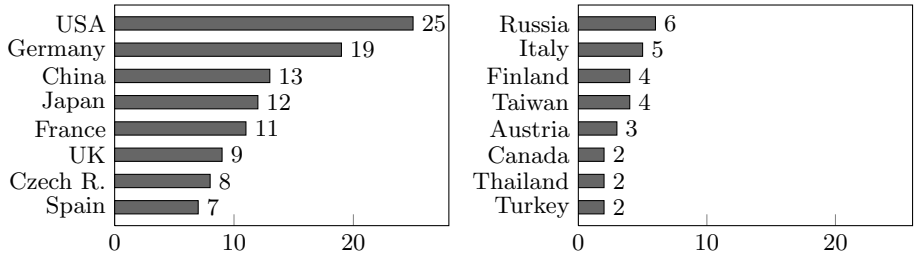


Fig. 2. Number of unique IP addresses from which the BOSSRank image database was downloaded during the contest. Total 142 IP addresses were recorded.

3.1 Cover-Source Mismatch

The cover-source-mismatch problem refers to a scenario, where images used for training the steganalyzer do not come from the same source as images w.r.t. which the steganalyzer is tested. If the source of images is very different and the steganalyzer is not robust with respect to this discrepancy, this can lead to decrease of the detection accuracy. By accident, the addition of pictures coming from a camera which was not used in BOSSBase has caused the cover-source mismatch problem.. Figure 3 shows the accuracy of submissions entered to the hall of fame according to the camera model. It can clearly be seen that all submissions are more accurate on images coming from the Leica M9 than on images captured by the Panasonic DMC-FZ50. The cover-source mismatch can be used to partly explain this phenomenon, the other reason might be that images coming from the DMC-FZ50 are more difficult the classify because of their contents.

The loss of accuracy is higher for steganalyzers developed by Hugobreakers than by other groups. It is also interesting to observe that on the beginning of

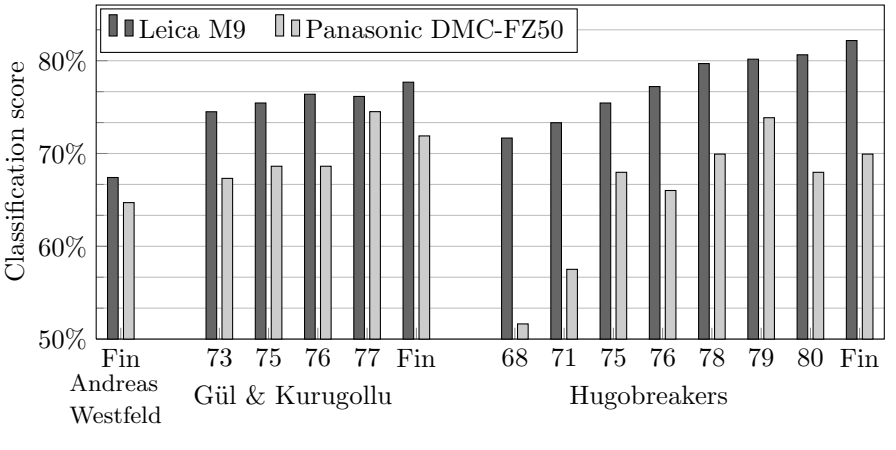


Fig. 3. Scores for each cameras for the different submissions in the Hall of Fame

the challenge, the accuracy of the first submission of Hugobreakers was nearly random on images coming from the Panasonic camera. From this analysis, it also appears that Gül & Kurugollu's steganalyzers were more immune to the problem of model mismatch than the classifier proposed by Hugobreakers.

To learn from this analysis more, it would be interesting to know the design of Hugobreakers' steganalyzers which scored at 71% and 75%, because between these two submissions, the cover-source mismatch was significantly reduced. Did this improvement come from training on a more diverse set of images, or it is due to different features or machine learning algorithm? Moreover, it should be also investigated, why steganalyzers of A. Westfeld and Gül & Kurugollu were more robust. Answers to these questions are important for building more robust and thus practically usable steganalyzers.

3.2 False Positives, False Negatives

We now extend the analysis from the previous subsection to false positive and false negative rates defined here as probability of cover image classified as stego and stego image classified as cover, respectively. Figure 4 shows these rates on BOSSRank together with rates on each camera separately for two best submissions of Hugobreakers and Gül & Kurugollu. We have noticed that Hugobreakers' steganalyzer suffered from very high false positive rate on images captured by the Panasonic camera. Their best submission had an almost 47% false positive rate, but only 8% false negative rate. Surprisingly, the final steganalyzer of Gül & Kurugollu did not exhibit such an imbalance between the false positive and false negative rates. Although the score used during the challenge evaluated overall accuracy of the steganalyzers, for the practical application, it is very

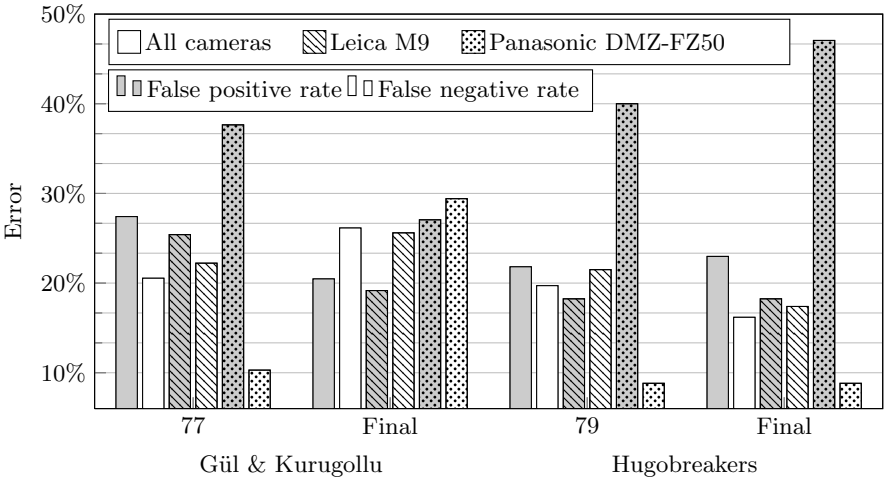


Fig. 4. False positive and false negative rates according to the camera for the four best submissions

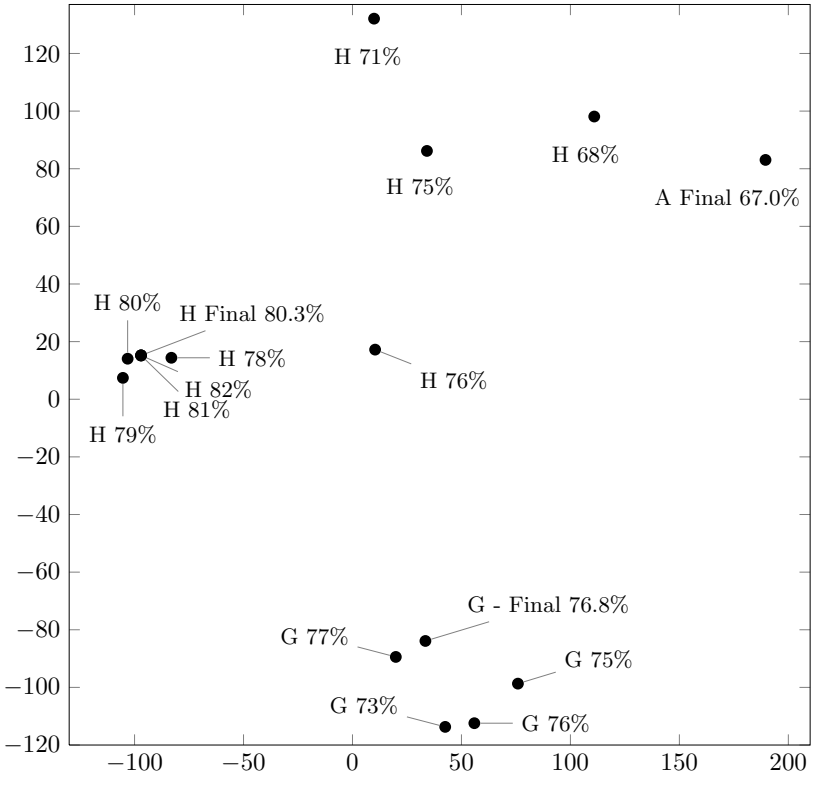


Fig. 5. MDS plot of submissions entered to Hall of fame. Legend: A — Andreas Westfeld, G — Gül & Kurugollu, and H — Hugobreakers. Each submission is labeled by the score as calculated on 900 random images measured at the time of submission. Final solutions are labeled by the score calculated w.r.t. the whole BOSSRank database.

important to limit the false positive rate. According to the results, the cover-source mismatch can make these errors even worse.

3.3 Clustering Analysis

Clustering analysis provides an interesting insight, how diverse were participants' submissions and how they evolved in time. Figure 5 shows an MDS plot of Hamming distances between submission vectors from the Hall of fame [8].[5] The MDS plot reveals that the initial detector of Hugobreakers (H 68%) was similar to the detector of A. Westfeld. Later, as the challenge progressed, Hugobreakers improved their detector and departed from the initial solution. Towards the end of the contest, Hugobreakers were merely tuning their detector, but no

[5] Multi-Dimensional Scaling (MDS) plot tries to map points from high-dimensional space to low-dimensional space such that distances between individual points are preserved.

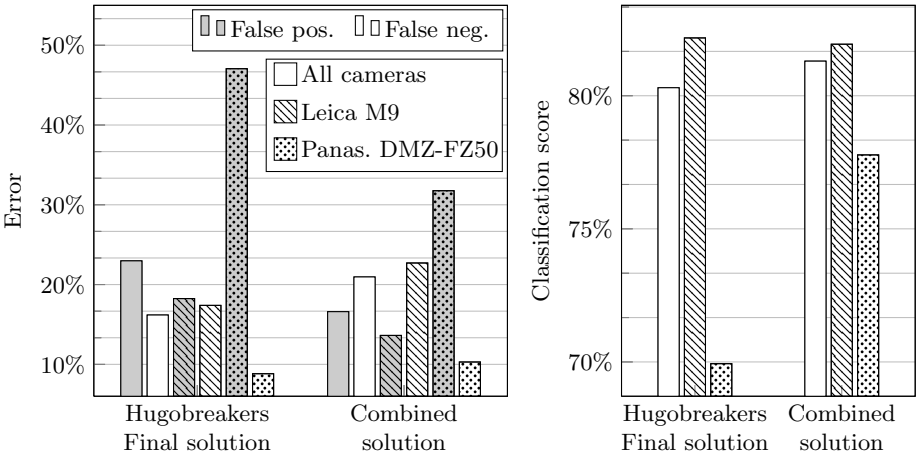


Fig. 6. Comparisons between the results of the collusion and the winner of the challenge

significant change has been introduced. This can be recognized by many submissions forming a tiny cluster. On the other hand, the detector developed by Gül & Kurugollu was from the very beginning different from detectors of other participants, as their submissions form a small compact cluster within the space.

It is interesting to see that Hugobreakers and Gül & Kurugollu have developed detectors with similar accuracy but independent errors. This is supported by the fact that only two images out of 1000 were always incorrectly classified (both images, image no. 174 and image no. 353, were false positives). In other words for 99.8% of the images there has been at least one submission in which the image was classified correctly. These suggest that the accuracy can be improved by fusing the classifiers developed in the contest as is shown in the next section.

4 Mixing Strategies

From the informed analysis done in the previous section, we noticed that the submission $\mathbf{h} = (h_1, \ldots, h_{1000}) \in \{0,1\}^{1000}$[6] of Hugobreakers scoring 79% is more immune to cover-source mismatch and false positive errors than their final submission $\mathbf{h}' = (h'_1, \ldots, h'_{1000}) \in \{0,1\}^{1000}$ scoring 80.3%. In order to decrease the false positive errors of the final solution we fuse the two submissions and define new vector $\mathbf{c} = (c_1, \ldots, c_{1000}) \in \{0,1\}^{1000}$ as:

$$c_i = \begin{cases} 1 & \text{if } h_i = 1 \text{ and } h'_i = 1 \text{ (both submissions call } i\text{th image stego)} \\ 0 & \text{otherwise.} \end{cases}$$

Figure 6 compares the performances of the collusion vector c with the best submissions of BOSS. This vector c achieves 81.3%, which is 1% more than the

[6] Element 0 (1) in the of the submission vector corresponds to cover (stego) prediction.

final score of Hugobreakers. The fused vector is also less sensitive to false positive errors and cover-source mismatch. Note however that this is an a posteriori submission using results from the test set and consequently it should be evaluated on other test sets in order to consider the comparison fair.

5 Conclusion and Perspectives

As can be seen from [4,5,6], BOSS challenge has stimulated research and forced the participants to deal with many challenging problems in steganalysis. The accuracy of detection of the HUGO algorithm, developed for the challenge, has increased from 65% to 81% for an embedding capacity of 0.4bpp and further improvement is to be expected. Moreover, according to the clustering analysis presented in this report, at least two different steganalyzers with similar performance have been developed which can lead to better results after the players exchange their ideas.

In possible extensions of HUGO, authors should consider avoiding the payload-limited sender regime, where the same amount of payload is embedded in every image. Instead, the stegosystem should try to embed different amount of payload based on the image content and possibly spread the payload among multiple cover objects, i.e., use batch steganography.

Besides that, BOSS challenge pointed out that cover-source mismatch is a significant problem for practical applications of steganalyzers based on a combination of steganalytic features and machine learning algorithms. We believe that the future research should focus to mitigate the cover source mismatch together with a problem of excessively high false positive rates. These findings also underline the need to develop a methodology to compare steganalyzers in a fair manner.

One of the incentives to organize BOSS was to investigate if steganalysis can exploit the knowledge of probability of pixel changes. For adaptive schemes, which represent current state-of-the-art in steganography, this probability is not uniform and can be well estimated from the stego image. Whether this fact presents any weakness has not been proved yet, but according to our knowledge, none of the successful participants of BOSS contest was able to utilize such information.

References

1. Bas, P., Furon, T.: BOWS-2 (July 2007), `http://bows2.gipsa-lab.inpg.fr`
2. Filler, T., Fridrich, J.: Gibbs construction in steganography. IEEE Transactions on Information Forensics and Security 5(4), 705–720 (2010)
3. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security (2010); under review
4. Fridrich, J., Goljan, M., Kodovský, J., Holub, V.: Steganalysis of spatially-adaptive steganography. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 102–117. Springer, Heidelberg (2011)

5. Fridrich, J., Kodovský, J., Goljan, M., Holub, V.: Breaking hugo - the process discovery. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 85–101. Springer, Heidelberg (2011)
6. Gul, G., Kurugoiiu, F.: A new methodology in steganalysis: Breaking highly undetactable steganograpy (hugo). In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 71–84. Springer, Heidelberg (2011)
7. Goldenstein, S., Boult, T.: The first IEEE workitorial on vision of the unseen (2008), http://www.liv.ic.unicamp.br/wvu/
8. Gower, J.: Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika 53(3-4), 325 (1966)
9. Kodovský, J., Fridrich, J.: On completeness of feature spaces in blind steganalysis. In: Ker, A.D., Dittmann, J., Fridrich, J. (eds.) Proceedings of the 10th ACM Multimedia & Security Workshop, Oxford, UK, September 22-23, pp. 123–132 (2008)
10. Kodovský, J., Pevný, T., Fridrich, J.: Modern steganalysis can detect YASS. 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, vol. 7541, pp. 02-01–02-11 (2010)
11. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. In: Dittmann, J., Craver, S., Fridrich, J. (eds.) Proceedings of the 11th ACM Multimedia & Security Workshop, Princeton, NJ, September 7-8, pp. 75–84 (2009)
12. 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)
13. Piva, A., Barni, M.: The first BOWS contest: Break our watermarking system. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, January 29-February 1, vol. 6505 (2007)

A New Methodology in Steganalysis: Breaking Highly Undetectable Steganograpy (HUGO)

Gokhan Gul[1] and Fatih Kurugollu[2]

[1] Signal Processing Group, Institute of Telecommunications,
Technische Universität Darmstadt, 64283 Darmstadt
ggul@spg.tu-darmstadt.de
[2] School of Electronics, Electrical Engineering and Computer Science,
Queen's University, Belfast, UK
f.kurugollu@qub.ac.uk

Abstract. This paper presents a new methodology for the steganalysis of digital images. In principle, the proposed method is applicable to any kind of steganography at any domain. Special interest is put on the steganalysis of Highly Undetectable Steganography (HUGO). The proposed method first extracts features via applying a function to the image, constructing the k variate probability density function (PDF) estimates, and downsampling it by a suitable downsampling algorithm. The extracted feature vectors are then further optimized in order to increase the detection performance and reduce the computational time. Finally using a supervised classification algorithm such as SVM, steganalysis is performed. The proposed method is capable of detecting BOSSRank image set with an accuracy of 85%.

1 Introduction

A perfectly secure steganography is a long quest for information hiding. The aim of steganography can be described using the prisoners' problem in which two prisoners try to devise an escape plan by means of an open communication channel monitored by a warden. In this sense this secret communication which uses innocuous digital media should be undetectable by the warden [1]. The earliest attempt on steganographic data hiding is based on LSB substitution which assumes the LSBs of the cover image correspond to the natural noise in the imaging process and hide the embedding from the warden. However this scheme is easily detected by histogram based targeted steganalysis methods because of the unbalanced embedding leaving footprints in the image histogram. One way to alleviate this problem is to increment or to decrement the LSBs according to the message rather than the simple substitution scheme. Although this LSB± method provides a balanced embedding which is difficult to detect by using only the image histogram it disturbs the underlying statistical information resulting from the strong correlation between the image pixels [1]. Therefore using this

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 71–84, 2011.

fact LSB± is also not a secure steganographic method and can be detected[1] [2], [3], [4], [5], [6].

The information theoretic bound for secure steganography was established by Cachin [7]. The main idea is that a perfect steganography should preserve the distribution of the cover images. In this context, a distance based on Kullback-Leibler divergence between the stego and the cover distributions was proposed to determine the security of the steganography method. If this distance is zero this means that the warden cannot differentiate the stego images from the cover onces since both distributions for stego and cover objects are identical. However it is not practical to design such stegonagraphic system under this strict bound because it is not a trivial task to determine the distribution of the cover images. Even the image set used in the system is confined to a known source, such as a single camera, it is nearly impossible to determine the cover distribution [1].

Because of this difficulty the steganography methods concentrate on some marginal models rather than exact ones used to model the whole cover set and they try to provide a secure steganographic system under this simplified model. In this context, secure steganographic data hiding is a game between the prisoner and the warden. While the prisoner tries to hide data by preserving some statistics the warden strives to detect the hidden message by deploying better models which are not preserved by the embedding. Based on this fact, one practical way to design a secure steganography method for the prisoner is to take into account the most successful steganalysis method and try to overcome it by means of the model which is not encompassed by this steganalyser.

One of the recent attempts, which uses this paradigm to design a very secure steganography system, is Highly Undetectable steGO (HUGO) method [8]. Its impact relies on using high dimensional image models which is not employed in steganography yet. For this purpose, HUGO takes into account SPAM features which have been recently proposed and are very successful on spatial domain LSB± embedding [6]. SPAM uses Markov transition probabilities calculated over difference images obtained by using neighboring pixels in 8 directions. First and second order transitions are taken into consideration by averaging horizontal and vertical directions yielding to one feature set and diagonal directions resulting to another one. Without any restriction on the dynamic range of the considered feature domain, this approach results in a very high dimensional embedding model. Using such high dimensional data in classification based steganalysis is problematic because of curse of dimensionality and related overfitting problems in the context of pattern recognition. Moreover a practical implementation is not a trivial task as well with this high dimensionality. To alleviate these problems SPAM calculates the probabilities in a restricted range [-T T]. By choosing T as 4 and 3 for $1st$ and $2nd$ order transitions respectively, SPAM has totally 162 and 686 features in both orders. HUGO employs this drawback to deploy embedding based on high dimensional models which is not a problem for steganography. For this aim, HUGO determines a distortion measure for each pixel individually.

[1] This work has been done while the first author was a visiting researcher at Queens University, Belfast.

This measure is actually a weighted sum of differences between the features used in SPAM derived from the cover and stego images. The significant point in the calculation is that the range for feature values are stretched to [-90 90]. The rationale behind this selection is that steganalysis should operate on a very large range such as [-90 90] to encompass the changes effectively. Then the model has more than 10^7 features which cannot be handled practically. The detectability of HUGO was tested against $1st$ and $2nd$ order SPAM, Wavelet Absolute Moments (WAM) and Cross Domain (CDF) based steganalytic features. The tests also showed that HUGO can embed 7 times longer messages than LSB± method in the same security level. The details of the method can be found in [8].

Generally, embedding in any steganography method is carried out in a single domain by preserving some statistics. However, preserving these statistics in one domain does not mean preserving other statistics in another domain as long as some model correction is not carried out in both domains. For example, HUGO preserves the model only derived from the domain in which SPAM features are extracted successfully. It is vulnerable against steganalysis using other models elicited from different domains.

In this paper, we propose a general approach for image steganalysis which uses different domains with different modalities to combat against steganalysis aware steganography like HUGO. Therefore steganography will find it difficult to stretch all domains. This work was carried out under Break Our Steganographic System (BOSS) contest which aims to evaluate the security of HUGO. The contest provided all information about the HUGO including the embedding algorithm and two image databases corresponding to training (BOSSBase) and testing (BOSSRank). The BOSSBase contains 9074 cover as well as 0.4 rate embedded stego images captured by using seven different digital cameras. BOSSRank image set, on the other hand, provides 1000 images (518 cover and 482 stego) for the testing which were taken by (an)unknown camera(s). The best detection rate obtained with CDF, 65%, was the minimum accuracy required to enter to the hall of fame.

The rest of the paper is organized as follows. In the next section the proposed method is presented and it is described how it can be used to break HUGO. Section 3 elaborates the training and feature set optimization by taking into account the cover-source mismatch problem as well as the computational feasibility of feature selection. The results are presented in section 4 while section 5 concludes the paper.

2 Steganalytic System Description

2.1 Proposed Methodology

In this section, we define a steganalytic system which can be applied to any kind of steganography although our particular interest was on the HUGO algorithm. The main idea is as follows: "*If HUGO is capable of preserving the higher order statistics on the difference image domain, can we find some other domains as*

well as some functions which are applied to these domains so that after an optimization process, high performance detection is possible". We tested the whole methodology on the HUGO algorithm as explained in the following sections.

2.2 Steganalysis against HUGO

For the proposed methodology, one has to find only the free parameters of the system. Let's assume that a real function $f\colon \mathbb{R}^m \to \mathbb{R}^n$ is applied to the image I. Then, there can be three different conditions namely, $m > n$, $m < n$ and $m = n$. The function f on the other hand can be linear or non-linear. In this work, we addressed all conditions considering the proposed methodology. Denoting $ds(\cdot)$ as a downsampling function and pdf_k as the k-variate PDF estimate, then basically we are trying to find

$$V_{k,i} = ds\left(pdf_k\left(f_i\left(I\right)\right)\right), \tag{1}$$

for i different functions where $V_{k,i}$ is the $(k,i)th$ feature vector. A downsampling function $ds(\cdot)$ is especially necessary in order to get rid of the dimensionality problem as well as to prevent from the sparseness problem of high variate PDF estimates when we combine the features with a classifier. As a result, we can reduce the steganalysis problem to find a suitable downsampling function $ds(\cdot)$ (as well as its parameters), and an f function which is applied to the image I and the k, defining the dimensionality of the PDF function. For a given training set of a steganography algorithm, and for a chosen f there is always a k and some parameters of a downsampling function which optimizes the detection performance. As default, we consider two types of downsampling functions. The first downsampling function sums the two neighboring PDF values and thus reduces the dimensionality by two at each iteration. The second downsampling function takes the average of the pixel values which are *symmetrically* situated in a PDF[2]. For larger PDF estimates, the symmetry-downsampling function gets complicated. However we were able to use it for k=1, 2, ...5.

Once we define a suitable f function, then it is becoming easy to optimize the $ds(\cdot)$ and the k. In our first tests we explored that

$$f = sort\left(M * I\right), \tag{2}$$

is a good way to start with a matrix M where $*$ corresponds to the convolution. The sort operation sorts the resulting data such that after the pdf_k and $ds(\cdot)$ operations, the detection is maximized over the training set. Well-known sorting functions considered in this work are horizonal, vertical, diagonal and minor diagonal scannings. However a sort function is not restricted to those four as it covers any permutations in general. It is important that a sorting function sorts the data in a form that it is as much correlated as possible for the neighboring pixels. Because the embedding noise destroys the correlations in an image and

[2] The symmetry mentioned in the text is the central symmetry, e.g., {0 0 1}, {0 1 0} and {1 0 0} indexes are summed up and divided by 3.

Table 1. Non-linear Filtering Matrices

M_{nl_1}	$\alpha_2+\alpha_7-2\alpha_0,\ \alpha_4+\alpha_5-2\alpha_0,\ \min\{\alpha_0-\alpha_1,\alpha_0-\alpha_3,\alpha_0-\alpha_6,\alpha_0-\alpha_8\}$ $\min\{\alpha_0-\alpha_2,\alpha_0-\alpha_4,\alpha_0-\alpha_5,\alpha_0-\alpha_7\}$
M_{nl_2}	$\min\{\alpha_1+\alpha_2+\alpha_3+\alpha_5+\alpha_8-5\alpha_0,\alpha_1+\alpha_4+\alpha_6+\alpha_7+\alpha_8-5\alpha_0,$ $\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_6-5\alpha_0,\alpha_3+\alpha_5+\alpha_6+\alpha_7+\alpha_8-5\alpha_0\},$ $\min\{\alpha_2+\alpha_4-2\alpha_0,\alpha_2+\alpha_5-2\alpha_0,\alpha_5+\alpha_7-2\alpha_0,\alpha_4+\alpha_7-2\alpha_0\}$ $\min\{\alpha_2+\alpha_4+\alpha_7-3\alpha_0,\alpha_2+\alpha_5+\alpha_7-3\alpha_0,$ $\alpha_2+\alpha_4+\alpha_5-3\alpha_0,\alpha_4+\alpha_5+\alpha_7-3\alpha_0\}$ $\min\{\alpha_0-\alpha_1,\alpha_0-\alpha_3,\alpha_0-\alpha_8,\alpha_0-\alpha_6,\alpha_0-\alpha_2,\alpha_0-\alpha_4,\alpha_0-\alpha_5,\alpha_0-\alpha_7\}$

such a modeling helps us to detect the existence of the embedding noise. For the steganalysis of HUGO, we observed that the following M matrices are able to provide some detection,

$$M_1=[1\ \ -2\ 1]\ \ ,M_2=M_0*M_1=[-1\ 3\ \ -3\ 1]\,, \tag{3}$$

which are actually the convolutions of $M_0=[1\ \ -1]$. Due to the range problem M_0 is inevitably used for steganalysis, please cf. [6]. The dynamic range of the filtered data is cut to [-T T] to reduce the dimensionality of the PDF estimates as well as to prevent from the poor statistics coming from the complex image regions. Keeping in mind that dimensionalities above 1000 are difficult to classify in a considerable time slot with some sophisticated classifiers, we considered T = 3, 4, 5 and 6, the PDF dimensions $k=1,2,...6$, and the downsampling methods $ds(\cdot)$ (as well as their parameters) and the f functions defined above (along with the transposes of M_i, for $i=1,2$). We observed that the highest detection was possible for $k=4$, $T=5$ and for a regular downsampling which is iterated for 4 times. The dimensionality of features for any T,k,r thus becomes,

$$D=\left\lfloor\frac{(2T+1)^k}{2^r}\right\rfloor, \tag{4}$$

which is 915 for $k=4$. In order to take into account the correlations from one column to the next one at each matrix of a 4-variate PDF estimate, we scan the odd indexed columns in the reverse order to get a one-row matrix for the downsampling.

In addition to the linear functions, we also explored the potential of non-linear ones. For the sake of clarity, we only give the main idea and the considered feature vectors. The non-linear functions in general provide diversity and increase the performance of the steganalyser. Figure 1 shows a 3×3 image block which we used in the optimization, however the block size might also be chosen differently. Per block, we need to derive 4 different values which are used to construct the 4-variate pdf estimates. Table 1 gives the corresponding M functions. Note that for linear f functions, we obtain only one value per convolution instead of four for non-linear Ms. The PDF estimates are also calculated in a similar fashion.

α_1	α_2	α_3
α_4	α_0	α_5
α_6	α_7	α_8

Fig. 1. A 3×3 block of an image I

So far, we obtained 8 feature vectors from the linear and 2 from the non-linear f functions. These feature vectors are further optimized due to the dimensionality problem. In order to obtain a feature vector which provides a detection performance which is better then each single feature vector, we have an optimization problem,

$$\begin{aligned} V_4 =& \gamma_1 V^{ver}_{4,M_1} + \gamma_2 V^{hor}_{4,M_1} + \gamma_3 V^{ver}_{4,M_2} + \gamma_4 V^{hor}_{4,M_2} + \gamma_5 V^{ver}_{4,M_1^T} \\ &+ \gamma_6 V^{hor}_{4,M_1^T} + \gamma_7 V^{ver}_{4,M_2^T} + \gamma_8 V^{hor}_{4,M_2^T} + \gamma_9 V_{4,M_{nl_1}} + \gamma_{10} V_{4,M_{nl_2}} \end{aligned} \tag{5}$$

with the free parameters γ_i for $i = 1, 2..., 10$ so that the detection performance is maximized over a training set. There are several alternatives for the solution of this problem. It is important to know which classifier we use to optimize the parameters and how many iterations for the training and the test we need to obtain reliable detection results. For an optimum solution, we can launch an exhaustive search with a predetermined step size. The lowest resolution is for a step size equals to 1 ($\gamma_i = 0$ or $\gamma_i = 1$) and we already know that each feature vector in V_4 might be able to increase the detection performance. For a smaller step size and a higher precision search, the search space is very huge especially when we consider that we need at least an average of a couple of random tests with a simple classifier and we have only a limited computational power. Therefore, we developed an iterative solution for the above mentioned problem. We call it as extended Fisher Linear Discriminator (FLD) algorithm since its detection performance is lower bounded by FLD and its complexity is still of order $\mathcal{O}(n^2)$.

For a linear classifier, we have a set of linear equations as many as the number of images in the training set. By convention, half of the images are chosen as cover and the other half as stego. For the decision labels $l_i \in [-1\ 1]$, $i = 1, 2, ..., N$ where N is the total number of images in the training set, we have the equations $\alpha_1 f_{1,i} + \alpha_2 f_{2,i} + ... + \alpha_\rho f_{\rho,i} \in [-1\ 1]$, for the jth feature from the ith image $f_{j,i}$, where $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_\rho\}$ are the regression coefficients. For the problem in (5) we have, however, two set of variables, namely γ and α. This means, the overall linear equations have the following form:

$$\gamma_1 \sum_{j=1}^{\rho} \alpha_j f_{j,i} + \gamma_2 \sum_{j=\rho+1}^{2\rho} \alpha_j f_{j,i} + ... + \gamma_\kappa \sum_{j=\frac{\kappa\rho}{2}+1}^{\kappa\rho} \alpha_j f_{j,i} \in [-1\ 1] \tag{6}$$

which is $\kappa\rho$ regression coefficients for the linear system of equations. The detection performance is reduced considerably because N/ρ, the ratio of the number of images in the training set to the total number of features, is divided by a factor of κ. Our approach was to use the same α coefficients for each feature vector (i.e., for each $V_{4,(\cdot)}^{(\cdot)}$ in (5)). We have then,

$$\sum_{k=1}^{\rho} \alpha_k \sum_{j=1}^{\kappa} \gamma_j f_{(j-1)\rho+k,i} \in [-1\ 1], \tag{7}$$

which might be very fast solved iteratively. The main idea is that, in the first step we keep γ constant and solve the equations for α. In the next step, we keep α constant and determine γ according to linear least squares rule ($N > \rho$). For the initialization, $\gamma_j = 1 \forall j$ might be considered. This algorithm converges fast and is able to increase the detection performance, especially in case some feature vectors decrease the performance. We believe that two main factors restrict the performance of this algorithm: first, its high dependence on the initial conditions and second $\kappa << \rho$ in (7). Assume that our problem is a simple two class classification problem with two feature vectors, then the proposed algorithm can be used with a source separation algorithm. Each single feature vector is separated into several/many feature vectors with a dimensionality at least as much as that of the original feature vector. Then these feature vectors are weighted with γ and accordingly (7) is solved using the proposed iterative approach.

FLD classifier is a rather poor classifier in terms of detection performance when compared with more state of the art classifiers such as SVM. We use FLD especially for searching purposes due to its low computational complexity permitting a comprehensive search in a considerable time. The problem is that the chosen feature vectors with FLD are not always the best ones when we use them with an SVM classifier. We justified this with experiments and restricted ourselves to choose the best result among lower precision search with SVM and higher precision search with FLD. Due to limitations in the computational complexity, we used a very simple search algorithm. This algorithm first selects one feature vector, e.g., V_{4,M_1}^{ver}, for $\gamma_1 = 1$. Then, the increase in the detection performance is evaluated for all the remaining feature vectors when γ_i is varied in the interval [0 1] with a certain precision. The feature vector which increases the detection performance most along with its multiplicative term γ_i is selected and removed from the set of candidate feature vectors. This process is repeated as long as the detection performance no more increases. We were able to choose the step size as 0.1 for a linear classifier and 0.2 for an SVM classifier with an RBF kernel. The training and testing was carried out in a random fashion for 5 times when SVM is used whereas it was 20 when linear classifier was

Table 2. Selected set of parameters for (5)

γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8	γ_9	γ_{10}
1	0	0.8	0.8	0	0.8	0.6	0.4	1	0.8

considered[3]. The final detection performance was better for a low resolution search with SVM classifier. This is perhaps because we both search and evaluate the feature vectors using an SVM classifier. The corresponding γ is given in Table 2.

We performed the similar designs for k=3, 5 and 6 (both linear and non-linear functions are considered). However when they were used together with 4D features we have not seen any performance improvement. Then we used only non-linear functions when $k = 3$. There was a slight performance improvement (around 1-2%) with a linear classifier but when SVM is used the performance was degraded.

In the next step we considered $k = 2$. Again for $T = 5$ and for the function defined in (2),

$$M_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

is the filtering function. The optimized sorting functions were the diagonal and the minor diagonal scanning. After the scanning we used symmetry-downsampling for both PDF estimates. The final feature vector for $k = 2$ was the summation of both single feature vectors,

$$V_2 = V_{2,M_3}^{diag} + V_{2,M_3}^{mdiag} \tag{8}$$

which has a dimensionality of 66.

In addition to the features from $k = 2$ and $k = 4$, we also extracted features for $k = 1$. Especially, we tried to cover a wider range, [-128 127], which is not symmetric as for $k = 2$ and $k = 4$. The PDFs are extracted after the filtering with the following 4 linear filters M_0, M_0^T,

$$M_4 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{and} \quad \mathrm{M}_5 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Note that for $k = 1$ there is no need for a sorting function. Consequently, we obtained altogether 256 features for $k = 1$ by,

$$V_1 = V_{1,M_0} + V_{1,M_0^T} + V_{1,M_4} + V_{1,M_5}. \tag{9}$$

3 Training and Feature Set Optimization

In this section it is explained how the training set as well as the features can be optimized to increase the detection rate of the steganalyser.

[3] From the BOSSBase set V.92, randomly chosen 4500 images are used for training and 1000 for testing.

3.1 Construction of the Training Set and Cover-Source Mismatch Problem

In order to achieve higher detection performances in steganalysis, one of the challenging problems is to select a suitable training set. Especially, training set construction is not testing set independent. It has to be carefully collected keeping in mind some special characteristics of the images in the testing set. For example one can group the images according to image content. This will definitely be helpful for the steganalysis. One other idea might be dividing images according to their complexity by a metric such as entropy. Then instead of building one classifier, K classifiers can be built and used for the K groups in the testing set. Furthermore, one can build one classifier per image under test. However this needs a lot of effort. It is also clear that the best way to build the training set is to use the same camera which is used in capturing images in the testing set. If it is unavailable, one might try to determine the source of the camera from the EXIF files extracted from the image. If EXIF file is not available or maliciously deleted, we can still try to determine the source using some image forensic tools [9]. In case we have neither the original camera nor a camera from the same brand, then we might expect severe performance degradation when we run our steganalyser on the testing set. This is called *cover-source mismatch problem* and we need to indicate that it is really a challenging one.

The easiest solution for the cover-source mismatch problem comes inherently with our design, namely the quantity of the features put a default gap to the detection performance. The higher the variety of features, the higher the detection performance, and the lower the effect of the cover-source mismatch problem. In this way the features also gain a universal meaning. Apart from the features themselves, we propose to use the testing set images in the training set. In the BOSS contest we were given 1000 test images, however, of course it is questionable what had happened if we would have had only a few images in the testing set. Probably, there would not be enough information to use as a feedback to the steganalyser. The increase of the number of images in the testing set allows us to deduce more and more information about the camera characteristics and to use it more efficiently in the steganalysis.

We have two basic designs here: one is based on the singular value decomposition (SVD) which is called as *controlled* denoising and the other is based on the discrete cosine transform (DCT) which is named as *uncontrolled* denoising. The main idea behind the proposed approach is to inform the steganalyser with the type of the testing images. Since the testing set is composed both of the cover and stego images, the best way to alleviate this problem is to apply some denoising algorithms to estimate the cover image. The estimated image is then considered as a cover image and embedded with the same embedding rate used to create the test images (which is known in advance and equals to 0.4).

For the controlled denoising, we determine the SVD of the image and set the singular values to zero starting from the lowest one to the largest one as soon as we reach to a certain mean squared error (MSE) between the decomposed image and the original image. We selected this value as 0.1 which roughly equals to the

MSE of the embedding noise. Then the denoised image is obtained by an inverse SVD transform and rounding-clipping operation. For the uncontrolled denoising, DCT transform is applied to the raw image and the resulting coefficients are rounded. Then, the inverse transform with rounding-clipping in the spatial domain is utilized to obtain the denoised image.

A more difficult yet powerful design for the cover-source mismatch problem exists in case we are able to determine the source of the camera. In this case, the deviations from the camera which we will use to build the training set and the camera which is used to obtain the test images are due to different sensor noise, lens types, etc. Given the test images such as BOSSRank, this deviation can be modeled using camera fingerprints and this might lead to obtain training images much closer to the ones obtained by the original camera.

3.2 Feature Selection

After having obtained the raw features, further optimization is possible with a suitable feature selection algorithm. In this process, we aim to reduce the dimensionality of the features as well as to increase the overall detection performance. If a feature selection algorithm does not increase the performance, then either we might use another feature selection algorithm or we stick to the raw features. The exact solution of this problem is an exhaustive search and often impractical in a reasonable time span. Another alternatives, more pronounced in the literature, are Sequential Floating Forward Search (SFFS) and Sequential Floating Backward Search (SFBS) algorithms [10]. For each selected feature, excluding further additions and subtractions from the set, it is necessary to make at least ρ searches. Repeating this for s selected features gives at least $\mathcal{O}(\rho s)$ complexity. Along with this complexity, the structure of one by one feature selection is not suitable for the features around 1000. The grouping of the features might be desirable, which can reduce the complexity of SFFS or SFBS to a reasonable time. Although we expect that correlated features should be grouped together the problems such as the number of features in a group and/or the algorithm to use for grouping are still open questions. We do still not know if such a design would finally improve the performance. Therefore, we restricted ourselves to the simplest feature selection algorithms. An attractive and simple feature selection algorithm sorts the features according to their reliability by the use of a metric.

We consider two metrics here, which are p values of the ANOVA statistical tests [11] and the co-variance between a feature and the embedding rate, respectively. Similar to our previous experiences, when we determine the p values and take the first K best features accordingly, we have not seen any performance improvement. It might be because of the correlation between the selected features, i.e., a powerful classifier requires diverse information which is usually difficult to be provided by correlated features. On the other hand, we were able to increase the detection performance using co-variance as a metric. To calculate the co-variance between a feature and the embedding rate informs us if the feature is useful for the classification. We expect that good features should be correlated to the embedding changes. To find out which features were useful, we selected

Table 3. Detection Performances of the Single Feature Vectors With a Linear Classifier on the BOSSBase Image set

	# of features[1]	# of selected features[2]	Detection[1]	Detection[2]
k=1	256	121	70.0%	70.3%
k=2	66	×	61.5%	×
k=4	915	×	76.4%	×
$k = 2\&4$	981	957	77.2%	77.3%
All	1237	1078	84.1%	84.4%

an image containing both complex and smooth regions. This image is embedded with a random message by varying the embedding rates. Then, from each embeeded image, we extracted the combined features (V_2+V_4, altogether 981 features). Next, the co-variance between the features and the embedding rate is calculated and the features are sorted corresponding to their co-variances (in the decreasing order). In the final step we determined the detection performances with the best K features by adding them to the classification process one by one. Using an SVM classifier the best performance is attained by 957 features.

4 Experimental Results

4.1 Image Set and Parameter Selection

We consider the BOSSBase training image set (V.92) with 9074 images and the denoised image sets which are derived from the BOSSRank testing set [12] as explained in the previous section. We choose the first 5500 images from BOSSBase and use this subset to find the optimum parameters of the SVM classifier [13]. We consider a large space for the grid search algorithm, $C\&\gamma \in \{2^i | i \in \{-20, ..., 20\}\}$.[4] The training and testing sets never overlap for any of the experiments and iterated randomly for 5 times to get reliable parameters. Determined parameters were $C = 2^{20}$ and $\gamma = 2^{-1}$.

4.2 Simulation Results

In the first simulation, we obtained the detection performances of the feature vectors per k both with and without feature selection using a linear classifier. Randomly chosen 8074 images are used for training and the non-overlapping 1000 for the testing. Table 3 shows the corresponding average detection performances of single classifiers over 100 random training and testing. In Table 3, we can see that the feature selection improves the performance poorly. Actually the performance improvement is more significant with an SVM classifier. However we give the results for a linear classifier because of the high computational complexity of SVM (one single training and testing with an SVM takes more

[4] The notations C and γ in this sub-section are taken from [13].

Table 4. Detection Performances of the Designed Steganalyser on the BOSSRank Testing Set When Trained With Various Training Sets

BOSSBase	BOSSBase+BOSSRank$_{svd}$	BOSSBase+BOSSRank$_{dct}$	All
84.0%	82.3%	85.0%	84.9%

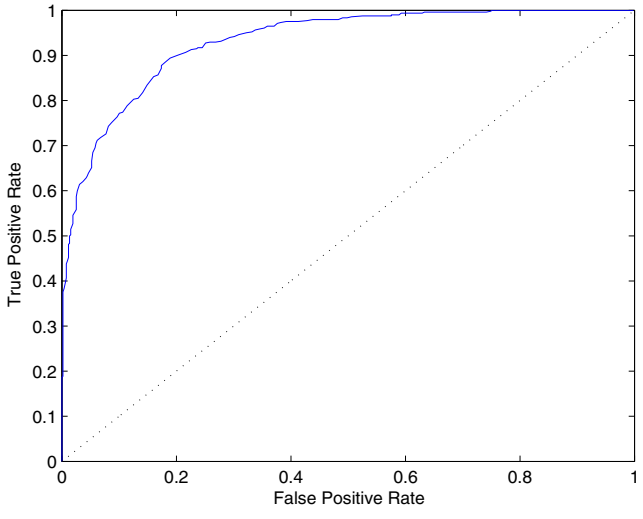


Fig. 2. ROC of the designed steganalyser for BOSSRank testing set

than half a day). It can be seen that when we consider feature selection and/or all the features for the classification performance is improved significantly. In the next experiment, we used SVM with the selected parameters ($C = 2^{20}$ and $\gamma = 2^{-1}$) and with the selected features (1078 features) for 20 random training and testing. The average detection performance that we obtained was 85,8%.

In our last experiment, we considered both the BOSSBase and the denoised 2000 images obtained from the BOSSRank testing set. Accordingly we wanted to see the effect of cover-source mismatch problem for the designed steganalyser especially when some information is transferred from the testing set to the training set via denoising. Table 4 shows the detection performance of the designed steganalyser for the BOSSRank image set. We need to note that there was never a feedback from the testing set to the designed classifier after the ground truth was revealed in the BOSS website. In Table 4, we can see that DCT based denoised 1000 images increased the detection performance of the classifier about 1% whereas the SVD based denoised images, however, decreased the performance. The reason is that SVD introduces always controlled same power of distortion in the denoising process (with a MSE equals to 0.1) and DCT introduces uncontrolled power of distortion which is on average lower than that of SVD (on average MSE about 0.08). We find the final results quite impressive because we

loose no more than 1% due to the cover-source mismatch problem which was around 3% when we considered the features only for $k = 2\&4$[5]. In Figure 2, the ROC curve for the BOSSRank set is given. The null hypothesis H_0 assumes that a given image contains no data embedding. The Area Under the Curve (AUC) is calculated as 93% whereas the false positive and false negative rates were 19,8% and 10.2% respectively.

5 Conclusions and Outlook

In this paper a novel steganalysis methodology is presented. It applies a function to the image under test, obtains the k variate PDF estimates and finally uses a suitable downsampling function. Having obtained the feature vectors, our methodology serves an extensive optimization process. We optimize the proposed model especially for the HUGO algorithm during the BOSS contest. Our observation is that HUGO leaves telltale effects on the filtered domain when filtering is especially highpass. This also removes most of the image content therefore it will be interesting to evaluate why HUGO is detectable for the second order derivatives in a pure theoretical work. In this work we considered both linear and non-linear filtering as a function applied to the image. Our main constraint was the dimensionality problem for a powerful classification therefore much effort is served in this area. Another problem which we dealt with was the cover-source mismatch problem. For the solution of this problem we proposed to use the denoised test images in the training process. This solution was able to increase the detection performance by 1%. Further performance improvement was obtained by a suitable, yet fast feature selection algortihm. After all efforts we were able to determine the BOSSRank with 85% accuracy which is only 0.8% less than the overall score which we obtained for the BOSSBase training image set. Finally we would like to mention that there are some following future works having very high potential to further increase the detection performance.

- ([5]) is not optimized with a powerful search algorithm such as an exhaustive search
- For $k = 2$, we have not used non-linear filtering and we have not considered a wider or a narrower range (only T=5 is considered)
- for $k = 1$, we have not used non-linear filtering, no optimization for linear combination parameters such as the one done in ([5]), and no range optimization
- No detailed denoising algorithms have been performed for the cover-source mismatch problem.
- No steganalytic attacks have been performed only on the saturated image regions

[5] There is no formal way of calculating the cover-source mismatch gap. In this work we consider the deviation of the BOSSRank score from the average score that we obtained from the BOSSBase image set.

References

1. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications, 1st edn. Cambridge University Press, Cambridge (2009)
2. Fridrich, J., Holotyak, T., Voloshynovskiy, S.: Blind Statistical Steganalysis of Additive Steganography Using Wavelet Higher Order Statistics. In: Proc. of the 9th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Salzburg, Austria, September 19-21 (2005)
3. Fridrich, J., Goljan, M.: Practical Steganalysis-State of the Art. In: Proc. SPIE Photonics West Electronic Imaging Security and Watermarking of Multimedia Contents, San Jose, California, vol. 4675, pp. 1–13 (January 2002)
4. Fridrich, J., Goljan, M., Du, R.: Detecting LSB Steganography in Color and Gray-Scale Images. IEEE Multimedia 8(4), 22–28 (2001)
5. Gul, G., Kurugollu, F.: SVD-Based Universal Spatial Domain Image Steganalysis. IEEE Transactions on Information Forensics and Security 5(2), 349–353 (2010)
6. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by Subtractive Pixel Adjacency Matrix. IEEE Transaction on Information Forensics and Security 5(2), 215–224 (2010)
7. Cachin, C.: An Information-Theoretic Model for Steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)
8. 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)
9. Gul, G., Avcibas, I.: Source Cell Phone Camera Identification Based on Singular Value Decomposition. In: First IEEE International Workshop on Information Forensics and Security, pp. 171–175 (December 2009)
10. Pudil, P., Novovicova, J., Kittler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters 15(2), 1119–1125 (1994)
11. Gul, G., Dirik, A.E., Avcibas, I.: Steganalytic Features for JPEG Compression-Based Perturbed Quantization. IEEE Signal Processing Letters 14(3), 204–208 (2007)
12. http://boss.gipsa-lab.grenoble-inp.fr/ (accessed Januray 31, 2011)
13. http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (accessed Januray 31, 2011)

Breaking HUGO – The Process Discovery

Jessica Fridrich, Jan Kodovský, Vojtěch Holub, and Miroslav Goljan

Department of ECE, SUNY Binghamton, NY, USA
{fridrich,jan.kodovsky,vholub1,mgoljan}@binghamton.edu

Abstract. This paper describes our experience with the BOSS competition in chronological order. The intention is to reveal all details of our effort focused on breaking HUGO – one of the most advanced steganographic systems ever published. We believe that researchers working in steganalysis of digital media and related fields will find it interesting, inspiring, and perhaps even entertaining to read about the details of our journey, including the dead ends, false hopes, surprises, obstacles, and lessons learned. This information is usually not found in technical papers that only show the final polished approach. This work accompanies our other paper in this volume [9].

1 Introduction

Competitions, such as BOSS (Break Our Steganographic System) [5] or BOWS (Break Our Watermarking System) [2] help focus the attention of the research community to a specific problem and thus advance the field by a large margin within a rather short time span. This is because challenges and competitive environment have always appealed to humans and also due to the fact that the participants do not need to formulate the problem (a task that is sometimes more important than the solution). Moreover, the competition guarantees that the results of different teams are comparable. For BOSS, the performance is evaluated using a single scalar value – the BOSSrank score.

According to our understanding, the original intention behind BOSS was to investigate whether content-adaptive steganography improves steganographic security for empirical covers in the form of raster, never-compressed images. The fact that in adaptive steganography the selection channel (placement of embedding changes) is publicly known, albeit in a probabilistic form, could in theory be exploited by an attacker. Adaptive schemes also introduce more embedding changes than non-adaptive schemes because some pixels are almost forbidden from being modified. Thus, an adaptive scheme will embed with a larger change rate than a non-adaptive one. On the other hand, the changes are constrained to those regions of images that are hard to model and thus the change rate is not an appropriate measure of statistical detectability as it puts the same weight to all pixels. The organizers of BOSS proposed a different distortion measure and argued that it better corresponds to detectability of embedding. To further substantiate their claim, the measure was incorporated in the steganographic algorithm HUGO (Highly Undetectable SteGO) [16] and the stego community was

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 85–101, 2011.

challenged to attack it. Preliminary tests with existing steganalyzers indeed indicated that HUGO is significantly more resistant to steganalysis than previous algorithms.

The BOSS competition, including the rules and the materials made available to the competitors, is described in a different paper in this volume [1]. Our team entered the competition at the end of August. This paper reveals the details of our investigation in chronological order. This technical narrative will hopefully be inspiring and maybe even amusing to those who tried to break HUGO and, in general, to all interested in steganalysis of digital media. Portraying our effort including the final results as well as our false beliefs and dead ends will convey those aspects of research work that is typically not found in technical papers. Our understanding of the field has evolved much over the last few months. We were forced to abandon established paradigms and reevaluate existing empirical truths. As a result, we learned quite a bit and we certainly hope that the reader of this paper will as well. This paper accompanies another paper [9] in this volume, which contains additional technical details of our final approach together with an extensive experimental section.

Everywhere in this article, lower-case boldface symbols are used for vectors and capital-case boldface symbols for matrices or higher-dimensional arrays. The symbols $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \{0, \dots, 255\}^{n_1 \times n_2}$ and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$ will always represent pixel values of grayscale cover and stego images with $n = n_1 n_2$ pixels. When the two-dimensional character of the pixel arrays is not important, for convenience, and hopefully without introducing any confusion, we index pixels with a single symbol instead of a pair. We will use $E[X]$ and $Var[X]$ for the expected value and variance of random variable X. For any $x \in \mathbb{R}$, the largest integer smaller than or equal to x is $\text{floor}(x)$. The detection accuracy of steganalyzers will always be evaluated on a test set never seen by the steganalyzer using a scalar score defined as

$$\rho \triangleq 1 - \min_{P_{\text{FA}}} \frac{1}{2}(P_{\text{FA}} + P_{\text{MD}}(P_{\text{FA}})), \tag{1}$$

where P_{FA} and P_{MD} are the probabilities of false alarm and missed detection. When the score is computed from BOSSrank images, it will always be referred to as the "BOSSrank score."

2 Early Ideas – Is the Public Selection Channel a Problem?

The very first idea that naturally lends itself is whether it is possible to somehow utilize the fact that the attacker can approximately determine the probabilities with which each pixel was changed during embedding. According to the folklore, revealing where embedding changes are made and where they are not may be a weakness of adaptive embedding that may be exploited.

HUGO modifies pixel x_i by ± 1 with probability $p_i^{\mathbf{X}}$ that can be determined from the cover image $\mathbf{X}$ and the payload.[1] Since the source code of HUGO

[1] For details, see [1] in this volume.

is public, one can easily extract the algorithm that computes the probabilities. However, when the image inspected by the attacker is a stego image, the probabilities computed from the stego image will in general be slightly different $p_i^{\mathbf{Y}} \neq p_i^{\mathbf{X}}$. Fig. 1 shows $p_i^{\mathbf{Y}}$ versus $p_i^{\mathbf{X}}$, $i = 1, \ldots, n$, for BOSSbase image no. 50. Overall, $p_i^{\mathbf{X}} \approx p_i^{\mathbf{Y}}$ with the largest relative errors for small $p_i^{\mathbf{X}}$. In particular, $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.05$ for 99.4% of pixels, $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.01$ for 85.2% pixels, and $|p_i^{\mathbf{Y}} - p_i^{\mathbf{X}}| \leq 0.001$ for 40.4% of pixels.

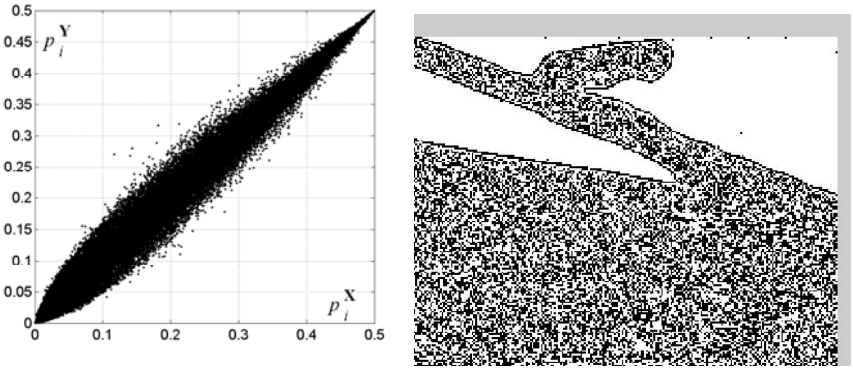


Fig. 1. Left: Probability of embedding change computed from the stego image, $p_i^{\mathbf{Y}}$, vs. $p_i^{\mathbf{X}}$ (for image no. 50 from BOSSbase). Right: LSB plane of the upper-right corner of image no. 235 from BOSSrank. The embedding changes are visible as black dots around the image boundary.

Because the payload is known and because $p_i^{\mathbf{X}} \approx p_i^{\mathbf{Y}}$, one could in theory derive (at least in expectation) the values of cover-image statistics, such as histograms or co-occurrence matrices. However, even if we succeeded in accurately estimating the cover-image statistics, using these estimates for steganalysis may still be problematic because HUGO does not introduce any easily detectable changes and we may not have any way of telling whether we are inspecting a cover or a stego image.

Having abandoned this direction, it is rather amusing that HUGO's embedding changes can be detected *visually* in seven images from BOSSrank – images no. 62, 195, 235, 396, 438, 948, and 983.[2] All seven images contain a region of pixels saturated either at 255 or at 0 while the rest of the image lacks any complex texture. Since HUGO was forced to embed 0.4 bpp in every image and since the probability of embedding in saturated areas is not completely zero, the embedding leaves suspicious salt-and-pepper noise in the least significant bit plane. An example is shown in Fig. 1 right. Notice that most of the visible

[2] These images were all classified correctly using our feature-based approach described below in this paper, thus the visual attack did not help us increase our BOSSrank score.

embedding changes are concentrated around the image boundary – most likely a consequence of how the embedding probabilities are computed at boundary pixels.

2.1 Detection by Correlation?

If we were able to estimate from the stego image whether a given pixel was modified by 1 or -1 with probability better than random guessing, we could detect HUGO (and ± 1) embedding using a correlation just like a spread-spectrum watermark. This idea is essentially identical to the Weighted Stego steganalysis [8]. Using $y_i = x_i + s_i$, $s_i \in \{-1, 0, 1\}$, we have $1/n \sum_i s_i^2 = \beta$, the change rate. (For HUGO with payload 0.4 bpp, $\beta \approx 0.1$, depending on the content.) Furthermore, let $\hat{x}_i = x_i + \Xi_i$ be an estimate of x_i from $\mathbf{Y}$ (e.g., $\hat{x}_{ij} = (y_{i,j-1} + y_{i,j+1})/2$), with Ξ_i being the estimation error. Assuming that the embedding change $\hat{s}_i$ can be estimated from $\mathbf{Y}$ with $\Pr(\hat{s}_i = s_i) = b > 1/2$, we now analyze the correlation for a cover and a stego image:

$$c = \sum_i (y_i - x_i)\hat{s}_i = \sum_i (s_i - \Xi_i)\hat{s}_i = \sum_i s_i \hat{s}_i - \sum_i \Xi_i \hat{s}_i. \tag{2}$$

When $\mathbf{Y}$ is a stego image and if $\boldsymbol{\Xi}$ and $\hat{\mathbf{s}}$ are uncorrelated, $E[c] = \beta(2b-1)n$ while for a cover image $\mathbf{X}$, $E[c] \approx 0$. Also, $Var[c] \propto n$ in both cases. This opens the possibility to detect embedding by thresholding c. This idea, however, hinges upon two assumptions – that we can estimate the direction of an embedding change with probability better than random guessing and the assumption of $\boldsymbol{\Xi}$ and $\hat{\mathbf{s}}$ being uncorrelated. While it is, indeed, possible to estimate $\hat{s}_i$ with probability $b > 1/2$, for example by testing if a change of y_{ij} by 1 or -1 decreases the sum $\sum_{0<|a|+|b|\leq 2} |y_{ij} - y_{i+a,j+b}|$, $\boldsymbol{\Xi}$ and $\hat{\mathbf{s}}$ are unfortunately correlated. The reason is the content-adaptive character of embedding. As a result, even though $E[c(\mathbf{Y})] > E[c(\mathbf{X})]$, it is not possible to find a threshold for c as it varies greatly across images. For some images, we observed the increase in correlation up to 60% but the average increase (over BOSSbase) was only 1.74%, which is by several orders of magnitude smaller than the variations of c across images.

3 Pixel Domain Is Not Useful, Right?

HUGO preserves complex statistics in a 10^7-dimensional feature space built from joint statistics of pixel differences on 7×7 neighborhoods. Thus, it may seem that features computed from differences between neighboring pixels will lead to weak detection simply because the embedding algorithm was designed to preserve statistics in this domain. One argument supporting this point of view is the experimental result reported in the original publication [16]: While the performance of the second-order SPAM feature set [15] (dimensionality 686) on HUGO is quite weak ($\rho = 58\%$), after augmenting SPAM with the DCT-based Cartesian-calibrated Pevný set [13] (dimensionality 548), the score improved to

$\rho = 65\%$.[3] This line of reasoning initially motivated us to compute features in an alternative domain, such as the wavelet domain. To this end, we decided to modify the WAM feature vector originally introduced in [10].

The WAM features are computed by first transforming the image to the wavelet domain using the Daubechis D8 wavelet, $(\mathbf{H}, \mathbf{V}, \mathbf{D}, \mathbf{L}) = W(\mathbf{X})$. When an undecimated transform is used, the first-level wavelet transform produces four subbands, $\mathbf{H}, \mathbf{V}, \mathbf{D}, \mathbf{L}$, of the same size as the original image. The three high-frequency subbands, $\mathbf{H} = (h_{ij}), \mathbf{V}, \mathbf{D}$, are denoised using the Wiener filter with variance σ_{W}^2:

$$\hat{h}_i = \frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2 + \sigma_{\mathrm{W}}^2} h_i, \tag{3}$$

where $\hat{\sigma}_i^2$ is the local variance at wavelet coefficient i estimated from its neighborhood. Finally, the WAM features, $\mu_m^{\mathrm{h}}, \mu_m^{\mathrm{v}}, \mu_m^{\mathrm{m}} \in \mathbb{R}^9$, are formed as nine central moments of their corresponding high-frequency subband noise residuals:

$$\mu_m^{\mathrm{h}} = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{h}_i - h_i - (\overline{\hat{\mathbf{H}} - \mathbf{H}}) \right|^m, \quad m \in \{1, \ldots, 9\}, \tag{4}$$

which gives a feature vector of dimension 27.

Our initial tests were done on BOSSbase 0.9 containing $2 \times 7,518$ images. The database was randomly divided into two equal-size subsets, one used for training and the other for testing. A Gaussian Support Vector Machine (G-SVM) was trained using standard five-fold crossvalidation on a multiplicative grid. The original WAM classifier with default $\sigma_{\mathrm{w}}^2 = 1/2$ gave the score of $\rho = 55.85\%$. To improve this rather weak performance, we decided to extend WAM by adding 27 moments (4) computed directly from the subbands $\mathbf{H}, \mathbf{V}, \mathbf{D}$ to inform the steganalyzer about the image content. This, indeed, makes sense to do for spatially-adaptive steganography. This content-informed WAM feature (WAMC) of dimensionality 54 reached the score of $\rho = 57.40\%$.

Exploring a different extension of WAM features, we augmented them with the same feature computed from an image re-embedded with the same payload of 0.4 bpp. This 54-dimensional vector (WAMre) produced a respectable $\rho = 59\%$.

The final and most significant improvement of WAM involved replacing the Wiener filter (3) with its adaptive version in which the fixed noise variance σ_{w}^2 was replaced with the variance of the stego noise s_i at pixel i, $\sigma_{\mathrm{w},i}^2 = p_i^{\mathbf{Y}}$. The best performance was achieved by merging the original 27 WAM features, 27 content features, and 27 WAM features obtained using the adaptive filter and adding to it the same set of 81 features from a re-embedded image (total of 162 features WAMCPre). The final performance is summarized in Table 1.

As part of our investigation of alternative embedding domains, we also tested the Cross-Domain Features (CDF) [13], which is a merger of the second-order SPAM with Cartesian-calibrated Pevný set (total dimensionality 1234). A G-SVM produced a prediction file with BOSSrank score of 68%, which is higher than the score of 65% obtained using the same feature set reported by the Czech

[3] This result was reported on the BOWS2 database [2].

Table 1. Performance of the WAM steganalyzer and its various extensions

Feature	Dimension	Score ρ [%]
WAM	27	55.85
WAMC	54	57.40
WAMre	54	59.00
WAMCPre	162	62.97

University Team. This difference is most likely caused by a different training set. While we trained on all images from BOSSbase 0.91, the Czech University Team trained on one half of this database.

4 Going Back to Pixel Domain

Even though alternative domains may be useful in steganalysis, the best detection is usually achieved by forming features *directly in the embedding domain*. This is where the embedding changes are localized and thus most pronounced. This strategy, originally coined in 2004 [6], was later confirmed in [6,10,17,15,18,4]. Because HUGO's embedding domain is known, after the early failures described in the previous two sections, we revisited the pixel domain and achieved a major breakthrough on September 23, 2010.

HUGO approximately preserves the joint distribution of first-order differences $r_{ij}^{(1)} = x_{i,j+1} - x_{ij}$ between four neighboring pixels – the co-occurrence of triples $(r_{ij}^{(1)}, r_{i,j+1}^{(1)}, r_{i,j+2}^{(1)})$ truncated[4] to a finite dynamic range, $r_{ij} \leftarrow \mathrm{trunc}_T(r_{ij})$, where $\mathrm{trunc}_T(x) = x$ when $x \in [-T, T]$ and $\mathrm{trunc}_T(x) = T\mathrm{sign}(x)$ otherwise. Thus, to detect traces of embedding, a fourth-order co-occurrence $(r_{ij}^{(1)}, r_{i,j+1}^{(1)}, r_{i,j+2}^{(1)}, r_{i,j+3}^{(1)})$ is needed. However, with increasing order of the co-occurrence its elements will be rather sparse when computed from small images and thus too noisy for steganalysis. The *key* idea and a major breakthrough in our effort to break HUGO was the realization that another way to form a statistic that spans more than four pixels is to use *higher-order pixel differences (residuals)*.

Because the second-order residuals, $r_{ij}^{(2)} = x_{i,j-1} - 2x_{ij} + x_{i,j+1}$, involve three pixels, one needs to consider the joint statistic of only three adjacent differences $(r_{ij}^{(2)}, r_{i,j+1}^{(2)}, r_{i,j+2}^{(2)})$. This keeps the co-occurrence matrix well-populated and thus useful for detection. The second-order residuals better remove content that is locally linear – while $r_{ij}^{(1)}$ may get out of the dynamic range $[-T, T]$ in locally linear regions, $r_{ij}^{(2)}$ may be mapped back inside the interval $[-T, T]$. One can also interpret $r_{ij}^{(2)} = 2(\hat{x}_{ij} - x_{ij})$, where $\hat{x}_{ij} - x_{ij}$ is the noise residual at pixel ij obtained using a simple denoising filter that predicts the value of the central pixel as an arithmetic average of its two closest neighbors: $\hat{x}_{ij} = \frac{1}{2}(x_{i,j-1} + x_{i,j+1})$. It is very important that the denoised value does not depend on the central pixel

[4] The truncation is an established way to keep the dimensionality low prior to forming joint statistics.

in any way, otherwise $\hat{x}_{ij}$ would be affected by the stego signal s_{ij}, which would thus be undesirably suppressed in $r_{ij}^{(2)}$.

Before describing the first successful feature set that gave us BOSSrank over 70%, we introduce four types of operators that can be applied to any two-dimensional array $\mathbf{A} = (a_{ij})$. The horizontal co-occurrence is a matrix $C^{\mathrm{h}}(\mathbf{A})$ whose (d_1, d_2, d_3)th element, $d_1, d_2, d_3 \in [-T, T]$, is

$$C^{\mathrm{h}}_{d_1 d_2 d_3}(\mathbf{A}) = |\{(i,j)|(a_{i,j}, a_{i,j+1}, a_{i,j+2}) = (d_1, d_2, d_3)\}|. \tag{5}$$

The operators C^{v}, C^{d}, and C^{m} are defined analogically.

After many initial experiments, we arrived at the following two feature vectors that allowed us to improve our BOSSrank score by a rather large margin. First, compute four second-order residuals at each pixel along the horizontal, vertical, diagonal, and minor diagonal direction:

$$\begin{aligned} r^{\mathrm{h}}_{ij} &= x_{i,j-1} - 2x_{ij} + x_{i,j+1}, & r^{\mathrm{v}}_{ij} &= x_{i-1,j} - 2x_{ij} + x_{i+1,j}, \\ r^{\mathrm{d}}_{ij} &= x_{i-1,j-1} - 2x_{ij} + x_{i+1,j+1}, & r^{\mathrm{m}}_{ij} &= x_{i-1,j+1} - 2x_{ij} + x_{i+1,j-1}. \end{aligned} \tag{6}$$

and then form the MIN and MAX residuals:

$$r^{\mathrm{MIN}}_{ij} = \mathrm{trunc}_T(\min\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}) \qquad r^{\mathrm{MAX}}_{ij} = \mathrm{trunc}_T(\max\{r^{\mathrm{h}}_{ij}, r^{\mathrm{v}}_{ij}, r^{\mathrm{d}}_{ij}, r^{\mathrm{m}}_{ij}\}). \tag{7}$$

The MINMAX feature vector is defined as

$$\mathbf{F}^{\mathrm{MINMAX}} = (C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MIN}}), C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MAX}})). \tag{8}$$

Since each cooccurrence matrix has $(2T+1)^3$ elements, $\mathbf{F}^{\mathrm{MINMAX}}$ has dimensionality of $2(2T+1)^3$.

By training the MINMAX feature vector with $T = 4$ using Fisher Linear Discriminant (FLD) on 9,074 cover and stego images from BOSSbase 0.91, we achieved a BOSSrank score of 71% on October 3, 2010.

The next discovery we made can be interpreted as a clever marginalization of the MINMAX vector for $T = 8$. Before forming r^{MIN}_{ij} and r^{MAX}_{ij}, the differences are quantized using a scalar quantizer $Q_q(x) = \mathrm{floor}(x/q)$ with q a positive integer:

$$\begin{aligned} &\mathbf{F}^{\mathrm{QUANT},q} = \\ &\left(C^{\mathrm{h}}(Q_q(\mathbf{R}^{\mathrm{MIN}})) + C^{\mathrm{v}}(Q_q(\mathbf{R}^{\mathrm{MIN}})), C^{\mathrm{h}}(Q_q(\mathbf{R}^{\mathrm{MAX}})) + C^{\mathrm{v}}(Q_q(\mathbf{R}^{\mathrm{MAX}}))\right). \end{aligned} \tag{9}$$

For $q = 2$, this QUANT feature can "see" twice as far as MINMAX but in a quantized manner to keep the dimensionality of the feature unchanged. By training a G-SVM on BOSSbase 0.91 on the 2,916-dimensional feature vector $(\mathbf{F}^{\mathrm{MINMAX}}, \mathbf{F}^{\mathrm{QUANT},2})$, with $T = 4$, we obtained a BOSSrank score of 73% on October 4, 2010.

On October 11, the organizers announced that the first 7,518 stego images from BOSSbase 0.9 and 0.91 were created with a different set of parameters ($\sigma = 10$, $\gamma = 4$, see [16] or [1] for details of the embedding algorithm) than

all BOSSrank stego images and the rest of the stego images in BOSSbase 0.91 (which were created with $\sigma = 1$, $\gamma = 1$). This change in parameters caused a mismatch between the training and testing stego sources. After recomputing the MINMAX and QUANT features on the correct stego images, on October 12 we achieved the score of 75% by merging the MINMAX and QUANT into a 2,916-dimensional feature set. Thus, the drop of performance due to this stego-source mismatch was 2%. To us, it was a HUGE difference even though the BOSS Team claimed on their blog on October 11 that HUGO behaves "similarly" for both choices of the parameters.

At this point, our team became confident that the 80% milestone was within reach by the end of October. We could not have been more wrong! Not only have we become hopelessly stuck at 75% for more than a month, but it would take us two and half months of very hard work to reach 80%. And we did so on December 23 with a feature vector of dimensionality $22,307$ trained on $2 \times 24,184$ images. To be able to train a classifier at this scale, we had to abandon SVMs and reinvent the entire machine learning approach. But before we get to that, in the next section we describe the Warden's nightmare.

5 The Dreaded Cover-Source Mismatch

The next logical step in our attack was to fine-tune our feature set by finding the optimal value of the threshold T, adding other versions of the features, and perhaps by training on a larger number of images. We also moved to a four-dimensional co-occurrence operator for the QUANT feature set, obtaining thus a 4,802-dimensional feature vector ($2 \times (2 \times 3 + 1)^4 = 4802$). To our big surprise, while we steadily improved detection accuracy on BOSSbase by adding more features, the BOSSrank score was moving in the *opposite* direction. We began facing the dreaded cover-source mismatch issue[5] – our classifier was trained on a different source of cover images (BOSSbase) than the source of BOSSrank images. Thus, as we optimized our detector on the training set, the performance on the testing set was steadily worsening. Our detector lacked what is recognized in detection theory as robustness.

Google search on "robust machine learning" returned publications that concerned only the case of training on noisy data or on data containing outliers. Our problem seemed different – we trained on one distribution and tested on another.

Perhaps using classifiers with less complicated decision boundary than the one produced by a G-SVM might help. The performance of a linear SVM (L-SVM), however, was consistently subpar to G-SVM and disturbingly comparable to the much simpler FLD classifier (see Table 2).

Another way to increase robustness, or so we thought, was to train on a larger set of images. We added to BOSSbase 0.91 another set of 6,500 images taken in raw format by 22 different cameras converted using the same script that was

[5] Cover source mismatch differs from overtraining as the latter refers to the lack of ability of the detector to generalize to unseen examples from the same source.

Table 2. BOSSrank score of the first successful feature sets, MINMAX and QUANT, for three different machine learning approaches

Feature	Dimensionality	Training set	G-SVM	L-SVM	FLD
MINMAX	1458	BOSSbase 0.92	73	70	71
QUANT	1458	BOSSbase 0.92	73	72	71
MINMAX+QUANT	2916	BOSSbase 0.92	75	72	71
MINMAX+QUANT	2916	BOSSbase+CAMERAS	71	70	-

used for creating the BOSSbase. Training on more images, however, seemed to make the BOSSrank score only worse (see the last row in Table 2).

The cover-source mismatch has been recognized by the research community as a serious issue that may complicate deployment of steganalysis in real life. The authors of [10] reported that the performance of the WAM steganalyzer on images could be vastly improved if the steganalyzer was trained on images from the exact same camera or, to a slightly lesser degree, on images from a camera of the same model. However, training WAM on a mixture of images from CAMERAS, the performance was significantly worse. The cover-source mismatch problem was also mentioned in the more recent publication [3], where the authors tested various steganalyzers on multiple sources for the ± 1 embedding. Thus, as the next logical step in our quest we decided to find out as much as possible about the source of covers for BOSSrank. We saw this as the only way to further improve our BOSSrank score.

5.1 Forensic Analysis of BOSSrank

On October 14, we extracted the sensor fingerprint [7] for each camera from BOSSbase (we did so from the resized grayscale 512×512 images). Then, we tested all BOSSrank images for the presence of the fingerprints. Only one camera tested positive – the Leica M9. Its fingerprint was found in approximately 490 images. We knew the source of one half of the database.

Visual inspection of BOSSrank images revealed that at least some portion of images was taken in the Pacific Northwest because many pictures contained license plates from the State of Oregon and Washington. One image (see Fig. 2 upper left) contained an address, which, after plugging it in GoogleMaps, returned the exact location – Portland, Oregon. And after the photographer was identified in a window pane reflection in image no. 558 (see Fig. 2 right), we knew what the camera was – Panasonic Lumix DMC-FZ50 – and it belonged to Tomáš Filler, a BOSS Team member.[6] However, we could not use this finding in competition because we relied on information other competitors did not have access to. Therefore, we closed our forensic investigation knowing that roughly one half (and potentially more) BOSSrank images were from Leica M9. The source

[6] The camera was confirmed by identifying its fingerprint in about 90 BOSSrank images. Here, we extracted the fingerprint from other images taken by Tomáš Filler during our previous trips to the SPIE conference.

of the remaining images in BOSSrank was declared unknown. All we needed to do now was to obtain more images from Leica.

Since stealing the camera from Patrick Bas seemed too dangerous and buying it too expensive ($7,000), we rented it from http://www.lensrentals.com/ for a week (October 23–30). The camera was rented with the standard 50mm lens.[7] After a grueling work with a heavy and boxy camera with no auto focus, we managed to take a total of 7,301 images in their original resolution of 18 megapixels. All images were processed using the BOSS conversion script and subsequently embedded with payload 0.4 bpp. After the MINMAX+QUANT features were extracted from them, we built two detectors – one G-SVM trained on all BOSSbase images that would be used for detection of all non-Leica images from BOSSrank, and the second G-SVM specifically trained on the union of our 7,301 Leica images and the 2,267 Leica images from BOSSbase. The decisions would then be merged into one prediction file. The result was quite disheartening – a measly 74% (BOSSrank score). We ran a couple of more experiments, such as training a G-SVM on a union of BOSSbase and 7,301 Leica images and testing the entire BOSSrank with it, but none of these experiments would produce a BOSSrank score higher than 74%.

This rather time-consuming exercise was an important lesson for us because we realized what makes a cover source and how hard it is to duplicate it. First, we took images with a different lens (50mm) than the BOSSbase images (35mm). The lens may have a significant impact on steganalysis because a longer focal length means lower depth of field, which implies less content in focus and more content slightly out of focus. Of course, an out-of-focus content is easier for the steganalyst.

The content of images has obviously a major influence on content-adaptive steganalysis. The cover source is a very complex entity that is affected by the lens, the environment in which pictures are taken and even the photographer's habits – stopping the lens more leads to a higher depth of field but also darker images with potentially more motion blur, while opening the lens leads to shorter exposures and less dark current but lower depth of field. Our images were all taken in the Fall in a little town of Binghamton in upstate New York. On the contrary, a large number of the Leica images in BOSSrank showed scenes with an ocean, ships, beaches, etc. As one of us sighed: "Binghamton in the Fall is a poor replacement for French Riviera." Consequently, it was rather foolish to think that we could duplicate the cover source.

6 Diversity Is Important

One important lesson we learned by now is that one should not be afraid of high feature dimensionality. After all, we successfully trained a 2,916-dimensional feature vector on $2 \times 9,074$ images and obtained a high BOSSrank score. However, scaling up the dimensionality simply by increasing the threshold T or the order of

[7] Only later it was pointed out to us that the lens information is in the EXIF headers of BOSSbase images. And the lens used for BOSSbase had a focal length of 35mm.

Fig. 2. Identifying the source of BOSSrank

the co-occurrence matrix did not lead to better results because the added features were increasingly sparsely populated. Thus, we refocused our effort to creating a more *diverse* feature set while keeping the dimensionality around 3,000, what seemed as a sweet spot for the given training set (BOSSbase). To this end, we used a lower threshold $T = 3$ and incorporated higher-order differences among neighboring pixels. One can easily extend the MINMAX and QUANT feature vectors (8) and (9) to higher-order residuals:

$$r_{ij}^{(3)} = x_{i,j-1} - 3x_{ij} + 3x_{i,j+1} - x_{i,j+2} \tag{10}$$

$$r_{ij}^{(4)} = -x_{i,j-2} + 4x_{i,j-1} - 6x_{ij} + 4x_{i,j+1} - x_{i,j+2}. \tag{11}$$

We also built features using fourth-order co-occurrence operators. To limit the growth of feature dimensionality, we used $T = 2$ for all fourth-order co-occurrences. This reasoning gave birth to the following 3,872-dimensional feature set SUM3 consisting of four different subsets (see Table 3).

Table 3. A merger of four feature sets, SUM3, computed from second- and third-order differences among pixels forming co-occurrence matrices of order 3 and 4. The feature dimensionality is 3,872.

Difference order q	Cooc. order T			Dimensionality
2nd	2	3	3	686
3rd	2	3	3	686
2nd	2	4	2	1,250
3rd	2	4	2	1,250

The strategy of increasing feature diversity was successful. By training a G-SVM on images from BOSSbase and with the feature set shown in Table 3 we obtained a BOSSrank score of 76% on November 13. The direction that was opening for us was clear – instead of blindly increasing the threshold and co-occurrence order, increase the feature diversity! For example, one could form higher-order residuals (differences) using two-dimensional kernels instead of one-dimensional or extract the residuals along edges to improve detection for textured images. The complexity of training a G-SVM, however, was beginning to limit the speed of development, while the performance of the much faster L-SVMs was subpar compared to G-SVMs. We needed an alternative machine learning tool that would enable faster development and testing of many ideas and combinations of features. Fortunately, our other research direction that we were simultaneously pursuing independently of the BOSS competition gave us just what we needed – an inexpensive, fast, and scalable machine learning approach.

7 Ensemble Classifiers – A Great Alternative to SVMs

In this section, we only provide a rather brief description, referring to [14] and our other paper in this volume [9] for a more detailed exposition of this methodology, experimental evaluation and comparison to SVMs as well as a discussion on the relationship of our approach to prior art in machine learning.

Starting with a feature set of full dimensionality d, we build a simple classifier (base learner), such as an FLD, on a randomly selected subset of $d_{\text{red}} \ll d$ features while using all training images. The classifier is a mapping $F : \mathbb{R}^d \to \{0, 1\}$, where 0 and 1 stand for cover and stego classes. This is repeated L times with a different random subset of the features. Consequently, we obtain L classifiers (FLDs) $F_1, \ldots, F_L$. Given a feature vector $\mathbf{b} \in \mathbb{R}^d$ from the testing set, the ensemble classifier makes a decision by fusing the individual decisions of all L FLDs, $F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})$. Although many fusion rules can certainly be used, we used simple voting as it gave us the same performance as more complicated rules.

To give the reader an idea about the savings, the ensemble classifier can be trained on $2 \times 9,074$ images with a $10,000$-dimensional feature set with $L = 31$ and $d_{\text{red}} = 1600$ and at the same time make decisions about the entire BOSSrank in about 7 minutes. This was achieved on a DELL Precision T1500 machine with 8GB of RAM and 8 Intel Cores i7 running at 2.93GHz. The same task when approached using a G-SVM takes substantially longer. Just obtaining the performance for a *single* grid point in cross-validation took between 2–17 hours, depending on the SVM parameters. Most importantly, however, the speed and simplicity of ensemble classifiers does not seem to compromise their performance. When comparing our BOSSrank scores obtained using the ensemble classifier and G-SVMs, the values were comparable and often in favor of the ensemble classifier. We view this approach to steganalysis as a viable fully-functional and scalable alternative to SVMs.

8 The Behemoths and the Final Attack – When 1% Seems Like Infinity

The scalability and low-complexity of the ensemble classifier enabled us to improve our BOSSrank score simply by gradually scaling up our features and training sets. On November 15, we reached the milestone of 77% with a set consisting of 5,330 features trained with $L = 31$ and $d_{\text{red}} = 1600$ on the entire BOSSbase. The set was obtained by adding the $1,458$-dimensional MINMAX vector with $T = 4$ to SUM3 (see Table 3).

On November 29, we added more features to our 5,330-dimensional set to form a feature vector with 9,288 elements. The added features were: 1) the QUANT feature vector (9) with $q = 2$ constructed from fourth-order residuals and a 4D co-occurrence (dimensionality 2×625) formed from horizontal and vertical samples as in (9) and 2) an equivalent of the QUANT feature (9) with $q = 2$ constructed from second-order residuals and a 4D co-occurrence of residuals arranged into a 2×2 square (2×625), and 3) a vector constructed from residuals computed using a translationally-invariant Ker–Böhme kernel [12]

$$\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix}, \tag{12}$$

and a 3D co-occurrence $C^{\text{h}}(\mathbf{R})+C^{\text{v}}(\mathbf{R})$ (729 features) and the same co-occurrence after quantizing the residual with $q = 2$ (another 729 features). All together, the new set had $5330 + 2500 + 1458 = 9288$ features. When trained on BOSSbase, this set produced a score of 76%. However, after enlarging the training set by adding images from CAMERAS to $2 \times (9074 + 6500) = 2 \times 15,574$ images, we obtained another Hall-of-Fame entry of 78% (again with $L = 31$ and $d = 1600$ as these parameters were becoming our "sweet spot"). This submission was an eye-opener. We learned that to maximize the BOSSrank score, we had to keep a certain balance between the feature dimensionality, d, and the number of images in the training set. Given $2N$ images for training, the best results were obtained when N was by 20–50% larger than d. Training on too few images or too many would make the BOSSrank score worse. And we observed this peculiar behavior until the end of the competition. We do not have a good explanation for this oddity but hypothesize that it is one of the strange consequences of the cover-source mismatch. This rule of thumb does NOT hold when the cover-source mismatch is absent. Without the mismatch, the detection accuracy simply keeps on improving with increased feature dimensionality (see our other paper [9] in this volume).

The rest of our record submissions are displayed in Fig. 3. The last three were achieved with $L = 51, 51, 71$ and $d_{\text{red}} = 2400$. The winning 24,993-dimensional feature set $\mathcal{B}$ is described in the Appendix. Our strategy was simple – keep on adding various types of features computed from different types of residuals and their quantized versions and scale the training set accordingly. We observed that the detection performance on BOSSrank was rather flat w.r.t. the parameters

of the ensemble classifiers L and d_{red}. With increasing feature dimensionality, we had to increase d_{red} from 1600 to 2400 or 2800, while the number of base learners, L, did not affect the performance as much and we kept it in the range 31–81. The individual predictions converged rather fast with increased L – for the winning submission, the prediction files for BOSSrank differed in only 37 images (for $L = 31$ and 51) and in 18 images for $L = 51$ and 81.

We have also tried increasing the dimensionality up to 37,859 and the training set to $2 \times 44,138$ images but we started observing a drop in BOSSrank. This may mean that we saturated our approach but a more likely explanation is that our saturation in performance was another consequence of the cover-source mismatch.

The winning submission we selected for the final ranking reached the score of 80.3%. After the ground truth was revealed, we found out that our best prediction file had a score of 80.5%.

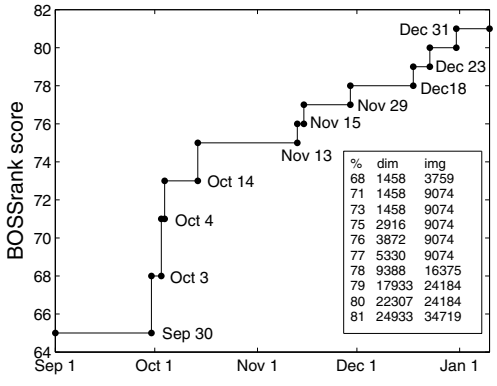


Fig. 3. Chronological development of our BOSSrank score. The table shows the feature dimensionality and the number of cover images on which the classifier was trained. Scores 77% and larger were obtained using ensemble classifiers.

9 What Have We Learned?

Quite a bit. First, there is no reason why steganalysts should frown at high-dimensional feature sets. To the contrary, we believe that high-dimensional features are a necessity to attack advanced steganography. The dimensionality could probably be reduced by clever marginalization, however, automatized design using ensemble classifiers is preferable to hand-crafting the features. The ensemble classifiers offer a scalable and quite simple classification with very similar performance to that of the much more complex SVMs.

The second important lesson is the existence of the Warden's nightmare – the cover-source mismatch that manifests when a detector optimized on one source

when applied on another experiences a loss of accuracy. Solving this problem appears to be extremely difficult because the mismatch can have too many forms. Just like robust statistics and robust versions of the likelihood-ratio test were developed to address the problems with robustness of optimal detectors and estimators, machine learning needs the same. Unfortunately, to the best knowledge of the authors very little appears to have been published on this important topic. If the BOSS oragizers had strictly adhered to the Kerckhoffs' principle, the cover source mismatch would never manifest and the competition would be more about breaking HUGO, which was perhaps the original motivation behind BOSS.

The steganalyst can improve the detection by training on a source with properties as close to the one from which the test images came. We tried to alleviate the negative impact of the cover-source mismatch by adding to BOSSbase all BOSSrank images after denoising (and pronouncing them as "covers") and all images after embedding in them payload of 0.4 bpp with HUGO (and pronouncing them as "stego"). The feature vectors of these 2×1000 images added to the training database should be rather close to the feature vectors of BOSSrank images, which might improve robustness to the cover source. We called this idea "training on a contaminated database" but were unable to improve our results with it. We plan to explore this rather interesting idea as part of our future effort.

Acknowledgements. The work on this paper was 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 to the Soukal's Family, Tice Lerner, Jim Pittaresi, Thomas Gloe, Peggy Goldsberry, and Jonathan Cohen for providing their images for research purposes and to the BOSS Team for setting up the BOSS competition. The authors would also like to thank their families for their great patience and understanding during those long months while the authors intensely and endlessly pondered about HUGO. The thanks go to Monika, Nicole, and Kathy Fridrich, and Fenix Garcia Tigreros.

Appendix – The Final 24,993-Dimensional Behemoth

For compactness, we use the following convention. Each feature set type is described using four parameters (s, q, m, T): s – the span of the difference used to compute the residual ($s = 3, 4, 5, \ldots$ for second-order residuals, third-order, etc.), q is the quantization step, m the order of the co-occurrence matrix, and T the truncation threshold. When a parameter is a set, the features are to be formed using all values from the set. The KB set was formed using (12) as described in Section 8. The SQUARE set is obtained from the MINMAX residual with

Table 4. The BOSS winner – the behemoth $\mathcal{B}$ of dimensionality 24,993

Feature type	Feature parameters (s, q, m, T)	Dimensionality
MINMAX	$(3, \{1, 2\}, 3, 3)$, $(4, \{2, 3\}, 3, 3)$, $(5, 6, 3, 3)$	5×686
	$(3, \{1, 2\}, \{5, 6\}, 1)$	$2 \times 486 + 2 \times 1458$
	$(3, 2, 4, 2)$, $(4, \{2, 3\}, 4, 2)$, $(5, \{1, 6\}, 4, 2)$	5×1250
	$(2, \{1, 2\}, 4, 2)$	2×1250
MARKOV	$(3, \{1, 2\}, 3, 3)$	2×686
KB	$(9, \{1, 2, 4\}, 3, 4)$	3×729
SQUARE	$(3, 2, 4, 2)$	1250
CALI	$(3, 2, 3, 3)$, $(4, \{2, 3\}, 3, 3)$	3×686
EDGE	$(6, \{1, 2, 4\}, 3, 3)$	3×686

co-occurrence elements formed by putting together residuals from 2×2 squares instead of straight lines. In the CALI set, prior to computing the features from the MINMAX residual, the image was convolved with an averaging 2×2 kernel to "erase" the embedding changes in a manner similar to calibration as proposed by Ker [11]. The residuals, $\mathbf{R}^{\text{EDGEMIN}}$ and $\mathbf{R}^{\text{EDGEMAX}}$, for EDGE were formed by taking MIN and MAX from residuals obtained using four directional kernels meant to follow edges in the image. An example of a kernel oriented along the minor-diagonal direction is:

$$\begin{pmatrix} -1 & 2 & -1 \\ 2 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}. \tag{13}$$

The final feature set for EDGE is formed as $C^{\text{h}}(\mathbf{R}^{\text{EDGEMIN}}) + C^{\text{v}}(\mathbf{R}^{\text{EDGEMIN}})$, $C^{\text{h}}(\mathbf{R}^{\text{EDGEMAX}}) + C^{\text{v}}(\mathbf{R}^{\text{EDGEMAX}})$.

References

1. Bas, P., Filler, T., Pevný, T.: Break our steganographic system – the ins and outs of organizing BOSS. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 59–70. Springer, Heidelberg (2011)
2. Bas, P., Furon, T.: BOWS-2 (July 2007), http://bows2.gipsa-lab.inpg.fr
3. Cancelli, G., Doërr, G., Cox, I.J., Barni, M.: A comparative study of ±1 steganalyzers. In: Proceedings IEEE International Workshop on Multimedia Signal Processing, Cairns, Australia, October 8-10, pp. 791–796 (2008)
4. Chen, C., Shi, Y.Q.: JPEG image steganalysis utilizing both intrablock and interblock correlations. In: IEEE International Symposium on Circuits and Systems, ISCAS 2008, pp. 3029–3032 (May 2008)
5. Filler, T., Pevný, T., Bas, P.: BOSS (July 2010), http://boss.gipsa-lab.grenoble-inp.fr/BOSSRank/
6. 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)
7. Fridrich, J.: Digital image forensic using sensor noise. IEEE Signal Processing Magazine 26(2), 26–37 (2009)

8. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI, San Jose, CA, January 19-22, vol. 5306, pp. 23–34 (2004)
9. Fridrich, J., Kodovský, J., Goljan, M., Holub, V.: Steganalysis of spatially-adaptive steganography. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 102–117. Springer, Heidelberg (2011)
10. Goljan, M., Fridrich, J., Holotyak, T.: New blind steganalysis and its implications. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, January 16-19, vol. 6072, pp. 1–13 (2006)
11. Ker, A.D.: Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters 12(6), 441–444 (2005)
12. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Delp, E.J., Wong, P.W. (eds.) 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)
13. Kodovský, J., Fridrich, J.: Calibration revisited. In: Dittmann, J., Craver, S., Fridrich, J. (eds.) Proceedings of the 11th ACM Multimedia & Security Workshop, Princeton, NJ, September 7-8, pp. 63–74 (2009)
14. Kodovský, J., Fridrich, J.: Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In: Memon, N.D., Delp, E.J., Wong, P.W., Dittmann, J. (eds.) Proceedings SPIE, Electronic Imaging, Watermarking, Security, and Forensics of Multimedia XIII, San Francisco, CA, January 23-26, vol. 7880 (2011)
15. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
16. 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)
17. Pevný, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, January 29-February 1, vol. 6505, pp. 3 1–3 14 (2007)
18. Shi, Y.Q., Chen, C., Chen, W.: A markov process based approach to effective attacking JPEG steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 249–264. Springer, Heidelberg (2007)

Steganalysis of Content-Adaptive Steganography in Spatial Domain

Jessica Fridrich, Jan Kodovský, Vojtěch Holub, and Miroslav Goljan

Department of ECE, SUNY Binghamton, NY, USA
{fridrich,jan.kodovsky,vholub1,mgoljan}@binghamton.edu

Abstract. Content-adaptive steganography constrains its embedding changes to those parts of covers that are difficult to model, such as textured or noisy regions. When combined with advanced coding techniques, adaptive steganographic methods can embed rather large payloads with low statistical detectability at least when measured using feature-based steganalyzers trained on a given cover source. The recently proposed steganographic algorithm HUGO is an example of this approach. The goal of this paper is to subject this newly proposed algorithm to analysis, identify features capable of detecting payload embedded using such schemes and obtain a better picture regarding the benefit of adaptive steganography with public selection channels. This work describes the technical details of our attack on HUGO as part of the BOSS challenge.

1 Introduction

Steganalysis is a signal detection problem – the task is to discover the presence of secretly embedded messages in objects, such as digital images or audio files. Since the dimensionality of digital media is typically very large, the detection is always preceded by dimensionality reduction – the objects are represented using a feature vector of a lower dimensionality. Steganalyzers are built in the feature space by training a classifier on a large database of cover and stego objects.

The main goal of this paper is to improve detection of adaptive steganography that makes embedding changes in hard-to-model regions of covers. A recent example of this type of steganography is HUGO [14]. Although this algorithm was designed for images in raster formats, the ideas can be applied to other domains and other media types. What distinguishes HUGO from other algorithms is that it approximately preserves a very high-dimensional feature vector and thus takes into consideration a large number of complex dependencies among neighboring pixels. With the help of advanced syndrome-coding techniques, HUGO embedding was reported undetectable using state-of-the-art steganalyzers even at rather large payloads [14].

It appears that as steganographers turn to feature spaces of very high dimension, steganalysts need to do the same to capture more subtle relationships among individual pixels. This brings about two major problems – how to form good high-dimensional feature sets and how to train classifiers in high dimensions

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 102–117, 2011.

with a limited number of training examples. To detect content-adaptive embedding, we need better models of local content, which could be achieved simply by adding more features. However, the dimensionality should be increased with care and one needs to make sure the features are *diverse* and *well populated* even in complex/textured regions. We propose to form the features as co-occurrences of image noise residuals obtained from higher-order local models of images.

The second problem presents a formidable challenge because training classifiers in high-dimensions requires a large number of examples to properly generalize to unknown images. However, it is not always easy or even possible for the Warden to obtain a sufficiently large number of examples from a given cover source. Additionally, training Support Vector Machines (SVMs) on a large number of examples in high-dimensional spaces can quickly become computationally prohibitive. To address these issues, we propose *ensemble classifiers* obtained by fusing decisions of base learners trained on random subspaces of the feature space. This machine learning approach is scalable and achieves accuracy comparable to SVMs. Its low complexity and scalability is especially convenient for rapid design and development – an attribute we view as vital for construction of practical steganalyzers as well as for winning steganography competitions.

The HUGO algorithm is described in [14] and a brief description also appears in [1] in this volume. In the next section, we introduce HOLMES – a strategy for constructing a large number of diverse features capable of detecting embedding changes in more complex parts of images. The ensemble classifier is detailed in Section 3, while all experiments are described in Section 4. We experimentally establish HUGO's detectability, compare its security with its non-adaptive ± 1 version, and contrast the performance of HOLMES to previous art. The paper is summarized in Section 5, where we also discuss the implications of our attack on design of future steganographic schemes.

Everywhere in this article, boldface symbols are used for vectors and capital-case boldface symbols for matrices or higher-dimensional arrays. The symbols $\mathbf{X} = (x_{ij}) \in \mathcal{X} = \{0, \ldots, 255\}^{n_1 \times n_2}$ and $\mathbf{Y} = (y_{ij}) \in \mathcal{X}$ will always represent pixel values of 8-bit grayscale cover and stego images with $n = n_1 n_2$ pixels.

2 The HOLMES Feature Set

Spatially-adaptive steganography makes embedding changes in those regions of the cover image that are hard to model, which makes the detection more difficult. On the other hand, the public selection channel could also be a weakness because the Warden can estimate the probability with which each pixel is modified. The authors of this paper were unable to utilize this information to improve their attack.

HUGO approximately preserves the joint statistic of differences between up to four neighboring pixels in four different directions. Thus, a better model is needed that can "see" farther than four pixels. We achieve this by working with higher-order noise residuals obtained by modeling the local content using polynomials.

2.1 Residuals

A popular way to design steganalysis methods is to extract the features not directly from the stego image $\mathbf{Y}$ but from a signal with a more favorable SNR – the image noise residual $\mathbf{R} = (r_{ij})$:

$$r_{ij} = y_{ij} - \text{Pred}(\mathcal{N}(\mathbf{Y}, i, j)), \tag{1}$$

where $\text{Pred}(\mathcal{N}(\mathbf{Y}, i, j))$ is an estimate of the cover image pixel x_{ij} from its neighborhood $\mathcal{N}(\mathbf{Y}, i, j)$.

A tempting option is to implement Pred(.) as a denoising filter. In fact, some previously proposed steganalysis features were designed exactly in this manner. In WAM [7], the predictor is the Wiener filter applied to wavelet coefficients. In [4], a shift-invariant linear predictor was used for an entire subband in a decomposition obtained using quadrature mirror filters. The problem with using denoising filters and linear filters, however, is that they place substantial weight on the central pixel being denoised / predicted. Consequently, the predicted value is generally a *biased* estimate of the cover pixel and the stego signal becomes suppressed in the residual (1). What is really needed for steganalysis is an unbiased estimate of the central pixel obtained from the neighboring pixels, *excluding* the pixel being estimated. The recently proposed SPAM feature set [13], as well as the earlier work [2,15], use the value of the neighboring pixel as the prediction:

$$\text{Pred}(\mathcal{N}(\mathbf{Y}, i, j)) = y_{i,j+1}. \tag{2}$$

While the noise residual $\mathbf{R}$ is confined to a narrower dynamic range when compared to $\mathbf{Y}$, it remains high-dimensional and cannot be used directly as a feature in machine learning. To reduce its dimensionality, features are usually constructed as some integral quantities. Considering the noise residual as a Markov chain, one can take its sample transition probability matrix [2,13,15] or the sample joint probability matrix (the co-occurrence matrix) as a feature. To capture higher-order dependencies among pixels, higher-order co-occurrence matrices are usually formed. However, the number of elements in 2D and 3D matrices rapidly increases and the bins become sparsely populated, making them less useful for steganalysis. This problem is usually resolved by marginalization before forming the co-occurrences – the residual is truncated, $r_{ij} \leftarrow \text{trunc}_T(r_{ij})$, where $\text{trunc}_T(x) = x$ when $x \in [-T, -T+1, \ldots, T]$, and $\text{trunc}_T(x) = T\text{sign}(x)$ otherwise. The truncation, however, introduces an undesirable information loss. Consider a locally linear part of an image, such as sky with a gradient of blue. The differences between neighboring pixels may be quite large due to the color gradient and thus end up being truncated despite the fact that this portion of an image is well modellable. Similar situation may occur around edges. Even though the content around the edge pixels may be quite complex, the values of pixels that follow the edge appear predictable using polynomial models (see Fig. 1).

These considerations motivated us to propose Higher-Order Local Model Estimators of Steganographic changes (HOLMES). Instead of the simplistic estimator (2), we compute the residuals using a family of local linear estimators.

Fig. 1. Close-up of a horizontal edge. Note that the grayscales in the horizontal direction are quite smooth and thus can be well approximated using polynomial models.

Table 1. Horizontal residuals from higher-order local models and their span s

Residual type	s	Horizontal residual $\mathbf{R}^{\mathrm{h}} = (r_{ij}^{\mathrm{h}})$
First order	2	$y_{i,j+1} - y_{ij}$
Second order	3	$y_{i,j-1} - 2y_{ij} + y_{i,j+1}$
Third order	4	$y_{i,j-1} - 3y_{ij} + 3y_{i,j+1} - y_{i,j+2}$
Fourth order	5	$-y_{i,j-2} + 4y_{i,j-1} - 6y_{ij} + 4y_{i,j+1} - y_{i,j+2}$
Fifth order	6	$-y_{i,j-2} + 5y_{i,j-1} - 10y_{ij} + 10y_{i,j+1} - 5y_{i,j+2} + y_{i,j+3}$
Sixth order	7	$y_{i,j-3} - 6y_{i,j-2} + 15y_{i,j-1} - 20y_{ij} + 15y_{i,j+1} - 6y_{i,j+2} + y_{i,j+3}$

The residuals in Table 1 are intentionally shown in their integer versions to avoid the need for rounding.

For example, the third and fourth order residuals can be derived from a locally quadratic model spanning three and four neighbors of the central pixel, respectively. They can also be interpreted as higher-order differences among neighboring pixels or discrete derivatives. The set of pixels involved in computing the residual is called a *clique* and its cardinality will be called *span* and always denoted s.

The residuals listed in Table 1 are all computed over horizontal cliques. The reader will readily supply the corresponding formulas for the vertical, diagonal, and minor-diagonal directions, $\mathbf{R}^{\mathrm{v}}$, $\mathbf{R}^{\mathrm{d}}$, $\mathbf{R}^{\mathrm{m}}$. There are numerous other possibilities how to define the residuals, each providing a different type of information. One particular case that turned out to be quite effective for attacking HUGO are the so-called MINMAX residuals:

$$r_{ij}^{\mathrm{MIN}} = \min\{r_{ij}^{\mathrm{h}}, r_{ij}^{\mathrm{v}}, r_{ij}^{\mathrm{d}}, r_{ij}^{\mathrm{m}}\}, \qquad r_{ij}^{\mathrm{MAX}} = \max\{r_{ij}^{\mathrm{h}}, r_{ij}^{\mathrm{v}}, r_{ij}^{\mathrm{d}}, r_{ij}^{\mathrm{m}}\}. \tag{3}$$

For pixel ij close to an edge, one of the MINMAX residuals will be large (in the direction perpendicular to the edge), while the other will likely be computed along the edge. Features built from these MINMAX residuals thus better adapt to textures and improve detection of adaptive embedding.

Of course, one can think of a myriad of other local predictors, such as the non-directional Ker–Böhme kernel [9] defined on 3×3 cliques:

$$\begin{aligned} r_{ij}^{\mathrm{KB}} = {} & 2y_{i-1,j} + 2y_{i+1,j} + 2y_{i,j-1} + 2y_{i,j+1} \\ & - y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j-1} - y_{i+1,j+1} - 4y_{ij} \end{aligned} \tag{4}$$

or directional kernels designed to model local image content around an edge (the model for a diagonal edge is shown in (5)) defined on cliques of span 6:

$$r_{ij}^{\mathrm{EDGE}} = 2y_{i-1,j} + 2y_{i,j+1} - y_{i-1,j-1} - y_{i-1,j+1} - y_{i+1,j+1} - y_{ij}. \tag{5}$$

Higher-order models better adjust to the local content and thus produce residuals with a more favorable SNR. Moreover, involving a clique of neighboring pixels in the linear combination "averages out" the embedding changes from the predicted value and thus further improves the prediction. According to our experience, even residuals of order as high as 5 or 6 provide useful information for steganalysis.

The reader will immediately notice that the higher-order predictors from Table 1 will have a larger dynamic range, which calls for a larger threshold T for their marginalization. To prevent rapid growth of feature dimensionality, the authors introduced quantized versions of the residuals:

$$Q_q(r_{ij}) = \mathrm{floor}\left(\frac{r_{ij}}{q}\right), \tag{6}$$

where q is a quantization step and $\mathrm{floor}(x)$ is the largest integer smaller than or equal to x. For small T, such as $T = 3$ or 4, the best detection is obtained by quantizing r_{ij} with the coefficient at the predicted pixel (see Section 4.1). In other words, for residuals of span 3–7, one should choose $q = 2, 3, 6, 10, 20$ (see Table 1).

The second-order quantized residual with $q = 2$ can be interpreted in another manner. Consider decreasing the dynamic range of the image by 50% by removing the LSB of each grayscale. The dynamic range of the resulting image is twice smaller and we also lost approximately 50% of all embedding changes – those that were LSB flips. However, the remaining changes are easier to detect due to the decreased dynamic range of the transformed image.

2.2 Features

Our features will be co-occurrence matrices formed from neighboring residual samples. To keep the notation compact, we introduce several different types of co-occurrence operators that can be applied to any two-dimensional array (residual) to produce a co-occurrence matrix or dimensionality $(2T+1)^m$, where m is the order of the co-occurrence. For example, the horizontal co-occurrence matrix of order m is

$$C^{\mathrm{h}}_{d_1\ldots d_m}(\mathbf{R}) = \Pr(r_{ij} = d_1 \wedge \ldots \wedge r_{i,j+m-1} = d_m), \quad d_1, \ldots, d_m \in [-T, \ldots, T]. \tag{7}$$

The operators $C^{\mathrm{v}}_{d_1\ldots d_m}$, $C^{\mathrm{d}}_{d_1\ldots d_m}$, and $C^{\mathrm{m}}_{d_1\ldots d_m}$ for the vertical (v), diagonal (d), and minor diagonal (m) directions are defined analogically. Note that forming the co-occurrence matrices makes sense even when r_{ij} is non-stationary. In fact, for natural images r_{ij} is a mixture – residuals in smooth regions fill out the neighborhood of $(d_1, \ldots, d_m) = (0, \ldots, 0)$, while residuals around vertical edges

will concentrate at the boundary of the matrix. Thus, different textures will likely occupy different parts of the co-occurrence matrix.

We will also make use of the fourth-order co-occurrence from residuals forming 2×2 squares:

$$C^{\mathrm{s}}_{d_1 \ldots d_4}(\mathbf{R}) = \Pr(r_{ij} = d_1 \wedge r_{i+1,j} = d_2 \wedge r_{i,j+1} = d_3 \wedge r_{i+1,j+1} = d_4). \quad (8)$$

There are many possibilities how to combine the residual and the co-occurrence operator to obtain features. And all combinations capture different relationships among pixels and are thus potentially useful for steganalysis. Certain combinations, however, provide little information. Since HUGO approximately preserves the joint probability distributions of differences between four neighboring pixels along all four directions, the matrices whose elements are computed from neighboring residuals whose union of cliques spans more than four pixels are more effective for steganalysis of HUGO. Thus, we require $s + m > 5$, where s is the span of the residual and m the co-occurrence order. For example, when working with first-order residuals ($s = 2$), we recommend to take co-occurrences of at least the fourth order, while for second-order residuals ($s = 3$) the third order may be sufficient.

Another pair of parameters that needs to be adjusted jointly is T and m. With larger m, one should correspondingly decrease T otherwise the co-occurrence matrix becomes too sparse and its elements become too noisy to provide useful detection statistic. It is worth mentioning that the marginals in the co-occurrence matrix may be as important (or even more important than) the inside of the matrix. According to our experience, even co-occurrences with $T = 1$ and $m \in \{5, 6\}$ still provide quite useful information for detection.

Based on a large number of experiments, we identified several combinations of residuals and co-occurrences that provided the best results. They are listed in Table 2. Each row corresponds to a feature type (a combination of a residual and a co-occurrence operator). All feature types between highlighted lines of the table are to be combined with all parameter sets in the second column. When a parameter is a set, e.g., $(3, \{1, 2\}, 3, 4))$, it means that the features are computed with both $(3, 1, 3, 4)$ and $(3, 2, 3, 4)$.

Table 2. Features formed by co-occurrence matrices and their parameters

Feature	Parameters (s, q, m, T)
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MIN}})$	$(3, \{1, 2\}, 3, 4)$,$(3, \{1, 2\}, 4, 2)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{MIN}})$	$(4, \{2, 3\}, 3, 4)$, $(4, \{2, 3\}, 4, 2)$
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MAX}})$	$(5, \{2, 3, 6\}, 3, 4)$,$(5, \{2, 3, 6\}, 4, 2)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{MAX}})$	$(6, \{5, 10\}, 3, 4)$,$(7, \{10, 20\}, 3, 4)$
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{h}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{v}})$	$(2, \{1, 2\}, 4, 2)$,$(3, 2, 5, 1)$,$(3, 2, 6, 1)$
$C^{\mathrm{d}}(\mathbf{R}^{\mathrm{d}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{m}})$	
$C^{\mathrm{h}}(\mathbf{R}^{\mathrm{KB}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{KB}})$	$(9, \{1, 2, 4\}, 3, 3)$
$C^{\mathrm{s}}(\mathbf{R}^{\mathrm{MIN}})$,$C^{\mathrm{s}}(\mathbf{R}^{\mathrm{MAX}})$	$(3, 2, 4, 2)$

The first four feature types in the table are computed from the MINMAX residuals. The matrices for the horizontal and vertical directions (and diagonal and minor diagonal directions) are added together to decrease dimensionality and provide a more stable statistic. The following two feature types can be thought of as sums of joint distributions of consecutive residuals modeled as Markov chains in each direction (they are similar in spirit to the SPAM feature set [13]), while the next one is computed from the Ker–Böhme residual (4).

This list should be taken as an example rather than a hard recommendation. The reader will easily come up with other forms of residuals and co-occurrence operators that may also lead to accurate detection of embedding. The steganalyst should select the individual sets so that they are diverse and complement each other as highly correlated features are undesirable. In practice, the size of the final feature set will be limited by the ability of the steganalyst to train a high-dimensional feature vector. If the dimensionality needs to be reduced, one can apply feature selection techniques or marginalize the set in some other way, for example by forming linear combinations of individual features.

The direction we adopted in this paper is to avoid hand design as much as possible and, instead, leave this job to the machine learning algorithm. We form a large feature set preferably consisting of a union of many *diverse* feature sets. Rather than mindlessly increasing the threshold T, we keep the threshold small and add more diverse feature sets by combining different types of residuals and co-occurrence operators. The emphasis here is on diversity and the ability of the features to "calibrate themselves" – to provide useful baseline information about each other [10]. For example, it makes sense to pair the parameter set $(s, q, m, T) = (3, 1, 3, 4)$ with $(3, 2, 3, 4)$ as the former provides more detailed information around the origin ($d_1 = d_2 = d_3 = 0$) while the same feature computed from the quantized residual "can see" twice as far before marginalizing the residuals.

Overall, our strategy for attacking HUGO is to assemble the feature set by merging multiple diverse subsets and let each subset contribute to the overall detection. In the next section, we supply the missing piece – a scalable machine-learning tool that can handle high-dimensional features and a large number of training examples with low complexity and good performance.

3 Ensemble Classifier

High feature dimensionality may negatively influence the complexity of training and classification as well as the ability of a classifier to generalize to previously unseen examples from the same source. Overcoming these problems becomes difficult especially when the class distinguishability is small and the number of examples from the cover source limited. Today, the machine learning tool of choice by steganalysts are kernelized SVMs, which are quite resistant to the curse of dimensionality. However, their complexity does not scale well and one can rather quickly run into memory and processing bottlenecks. The complexity is smaller for efficient implementations of linear SVMs but can become too large

as well if one desires to use linear SVMs as a development tool when many ideas need to be tested in a short period of time.

To lower the complexity, we decided to use ensemble classifiers based on fusing decisions of weak base learners trained on random subsets of the feature space. In order to make the supervised ensemble strategy work, the individual base learners have to be sufficiently diverse in the sense that they should make different errors on unseen data. The diversity is often more important than the accuracy of the individual classifiers, provided their performance is better than random guessing. From this point of view, overtrained base learners are not a big issue. In fact, ensemble classification is often applied to relatively weak and unstable classifiers since these yield higher diversity. It was shown that even fully overtrained base learners, when combined through a classification ensemble, may produce accuracy comparable to state-of-the-art techniques [3].

What makes ensemble classifiers especially attractive is that they scale well with dimensionality and the number of training examples and, according to our experience, their performance is comparable to that of Gaussian SVMs. Detailed description of ensemble classifiers, their analysis, and relationship to previous art appears in [11]. Here, we only provide a brief description. Starting with the full feature set of dimensionality d, the steganalyst first randomly selects $d_{\text{red}} \ll d$ features and trains a classifier (base learner) on them. The classifier is a mapping $F : \mathbb{R}^d \to \{0, 1\}$, where 0 stands for cover and 1 for stego.[1] This process is repeated L times, each time with a different random subset. As a result, L base learners, $F_1, \ldots, F_L$, are obtained. Given a feature $\mathbf{b} \in \mathbb{R}^d$ from the testing set, the final decision is obtained by fusing the decisions of all L individual base learners:

$$F_{\text{ens}}(\mathbf{b}) = \Im(F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})) \in \{0, 1\}, \tag{9}$$

where $\Im$ is some fusion rule.

Note that all classifiers in the algorithm are trained on feature spaces of a fixed dimension d_{red} that can be chosen to be significantly smaller than the full dimensionality d. Our base learners were the low-complexity Fisher Linear Discriminants (FLDs) and we used a simple voting for the fusion rule

$$\Im(F_1(\mathbf{b}), \ldots, F_L(\mathbf{b})) = \begin{cases} 1 & \text{when } \sum_{i=1}^{L} F_i(\mathbf{b}) > L/2 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The voting could be replaced by other aggregation rules. For example, when the decision boundary is a hyperplane, one can use the sum of projections on the normal vector of each classifier or the sum of likelihoods of each projection after fitting models to the projections of cover and stego images. Because in our experiments all three fusion strategies gave essentially identical results, we recommend using voting due to its simplicity. The individual classifiers should be adjusted to meet a desired performance criterion. In this paper, the decision threshold was always set to produce minimum overall average classification error

[1] F is really a map from $\mathbb{R}^{d_{\text{red}}} \to \{0, 1\}$ as each learner works with a subset of features.

$P_{\rm E} = \min_{P_{\rm FA}}(P_{\rm FA} + P_{\rm MD}(P_{\rm FA}))/2$ on the training data, which is the quantity that we also use to report the accuracy of detection in this paper.

4 Experiments

The main bulk of our experiments was carried out on BOSSbase 0.92 [5,1] containing 9,074 grayscale images originally 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. All tests were done by randomly dividing the BOSSbase into a training set of 8,074 images and a testing set of 1000 images. This split was repeated and the median value of $P_{\rm E}$ and its Mean Absolute Deviation (MAD) are what we report in graphs and tables. We remark that the selection of random feature subsets in our ensemble classifier was also different in each run.

4.1 Initial Tests

In our first set of experiments, we test the performance of selected individual feature sets listed in Table 3 to show the influence of the parameters (s, q, m, T) on the detection performance. The first set (MARKOV) is a direct equivalent of the second-order SPAM [13] with two differences – the first-order differences were replaced with second-order differences and the transitional probability matrix with the joint matrix (co-occurrence). It is rather interesting that by changing a *single line* of code SPAM turns into a significantly more powerful feature set – $P_{\rm E}$ has dropped from 42% [14] to 28.6%.[2] The second row of the table informs us that the detection is even better with the MINMAX residual, while the its quantized version shaves another 1% from $P_{\rm E}$. The next two rows are mergers of five sets of total dimensionality 7,290 and 6,250 for co-occurrence matrices of order $m = 3$ and 4 with $T = 4$ and $T = 2$, respectively. Adding features steadily leads to better performance.

The feature sets in the last two rows were quantized with q equal to the coefficient at x_{ij} in the higher-order residual (inspect Table 1) as this choice of q gave us the best performance. This is confirmed in Table 4 with the MINMAX residual with $s = 5$ (fourth-order residual) by showing $P_{\rm E}$ as a function of q while fixing all other parameters and variables ($L = 31$, $d_{\rm red} = 1000$).

According to our experiments on BOSSbase, adding more features generally leads to better detection. However, adding uninformative or dependent features will obviously decrease the detection accuracy. Clever marginalizations may also improve detection while keeping the dimensionality low. For example, we *added* all five co-occurrence matrices of third order listed in row 4 in Table 3 to form one 1458-dimensional vector. Then, we did the same with the features from row 5 to form a 1250-dimensional vector. Putting these two matrices together gave us

[2] This comparison is not really fair as the results were obtained on two different databases – BOWS2 vs. BOSSbase – while the latter appears somewhat easier to steganalyze.

Table 3. Performance of individual feature sets on BOSSbase 0.92. The acronyms MARKOV and MINMAX stand for co-occurrences $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{h}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{v}})$, $C^{\mathrm{d}}(\mathbf{R}^{\mathrm{d}}) + C^{\mathrm{m}}(\mathbf{R}^{\mathrm{m}})$, and $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MIN}})$, $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{MAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{MAX}})$, respectively. The quantization step in the last two sets was set to the coefficient at x_{ij} in the higher-order residual ($c = 2, 3, 6, 10, 20$ for $s = 3, 4, 5, 6, 7$).

Feature set	(s, q, m, T)	d	P_{E}	Best	Worst	L	d_{red}
MARKOV	$(3, 1, 3, 4)$	1458	28.6±0.9	25.5	31.0	31	1000
MINMAX	$(3, 1, 3, 4)$	1458	27.3±0.8	25.1	31.3	31	1000
MINMAX	$(3, 2, 3, 4)$	1458	26.2±1.2	23.2	28.4	31	1000
MINMAX	$(\{3, 4, 5, 6, 7\}, c, 3, 4)$	7290	20.0±0.8	17.8	22.6	81	1600
MINMAX	$(\{3, 4, 5, 6, 7\}, c, 4, 2)$	6250	20.9±0.4	19.0	23.5	81	1600

Table 4. Detection error P_{E} for the MINMAX feature set with parameters $(5, q, 3, 4)$ as a function of $q \in \{2, 4, 6, 8, 10, 12\}$. The best performance is achieved when q is equal to 6 – the coefficient at x_{ij} in the higher-order residual.

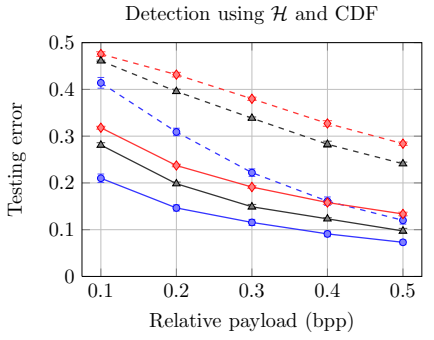
q	2	4	6	8	10	12
P_{E}	30.50	26.75	26.05	26.75	27.70	28.20

a $1458 + 1250 = 2708$-dimensional vector with $P_{\mathrm{E}} = 22\%$ under the same testing conditions (with $L = 81$ and $d_{\mathrm{red}} = 1600$). Obviously, adding feature sets is by no means the optimal operation and we prefer to leave the marginalization to an automated procedure instead of hand-tweaking. For experiments in the next section, we prepared a feature set by merging various combinations of residuals and co-occurrence matrices (the set is described in the Appendix).

4.2 Performance on BOSSbase

The purpose of experiments in this section is three-fold: to evaluate the detectability of HUGO, compare the HOLMES features and our ensemble classifier with the current state of the art – the CDF set [12], and to compare HUGO with non-adaptive ±1 embedding. Unless stated otherwise, all detectors were implemented using ensemble classifiers with FLDs as described in Section 3. We used a 33,963-dimensional feature set $\mathcal{H}$ implemented with $L = 81$ and $d_{\mathrm{red}} = 2800$ (see the Appendix). The CDF classifier used $L = 51$ and $d_{\mathrm{red}} = 500$. The values of d_{red} were determined by hand based on our experience.

All results are displayed in the self-explanatory Fig. 2. The CDF set has higher detection accuracy when implemented using a Gaussian SVM (G-SVM) instead of our ensemble classifier. However, unlike G-SVM, the ensemble classifier is capable of handling the high-dimensional HOLMES features which resulted in a consistently lower detection error P_{E} than the error for the CDF trained with a G-SVM. HUGO is confirmed to be more secure than non-adaptive ±1 embedding but the difference is less pronounced than what was reported in [14]. It is also interesting to compare the increase in detection accuracy for both algorithms and



α	HUGO			±1 embedding		
bpp	$\mathcal{H}$	CDF	G-SVM	$\mathcal{H}$	CDF	G-SVM
0.1	41.4	47.6	46.1	21.0	31.8	28.1
0.2	30.9	43.2	39.6	14.7	23.7	19.8
0.3	22.2	38.0	33.9	11.6	19.1	14.9
0.4	16.1	32.7	28.3	9.1	15.8	12.3
0.5	12.0	28.4	24.1	7.3	13.4	9.7

Fig. 2. Detection error P_{E} for HUGO and ±1 embedding for five relative payloads for the CDF and HOLMES classifiers. The error bars are MAD over 100 database splits 8074/1000. The CDF set was implemented with both our ensemble classifier and as a G-SVM (only 10 splits 8074/1000 were performed using G-SVM due to computational complexity).

feature sets. While the improvement for HUGO is about 5–12%, the detectability of ±1 embedding improved only by 2–7%.

Since BOSSbase images were resized to quite a small size, the correlations among neighboring pixels weaken significantly in textured regions, such as grass, sand, or foliage. Visual inspection confirmed that such textures start resembling random noise on the pixel level, which makes their steganalysis very difficult if possible at all since HUGO avoids regions where the content can be accurately modeled. To identify the type of images on which our classifier makes consistently correct and wrong decisions, we carried out the following experiment. Using the same setup with the HOLMES feature set $\mathcal{H}$, we repeated the random 8,074/1000 split of BOSSbase 1000 times (with $L = 81$ and $d_{\text{red}} = 2400$) and counted how many times a given cover image was classified as stego and vice versa. Each image $i \in \{1, \dots, 9074\}$ appeared in the testing set N_i times, where N_i is a binomial r.v. with mean 110 and standard deviation 9.9. Fig. 3 shows the probability $p_i = \delta_i / N_i$ of correctly detecting cover image i as cover (cover i was correctly classified δ_i times). In the figure, the BOSSbase is ordered by cameras. First, note that the detection heavily depends on the camera model. While cover images from Pentax can be classified with average accuracy of about 95%, images from Canon Rebel are significantly harder to classify (66%). This difference is most likely a combined effect of varying depth of field across both cameras (which is influenced by the lens), in-camera processing (some cameras denoise their images), the resizing script, and the environment in which the images were taken. All this forms the cover source and gives it unique properties that have a *major* effect on statistical detectability of embedding changes.

Second, notice that some cover images are persistently classified as stego (FAs) – the steganalyzer errs on them with probability 1. In fact, we identified 743 cover images that were *always* detected as stego and 674 stego images always detected as cover (MDs). Most of these images were highly textured and/or

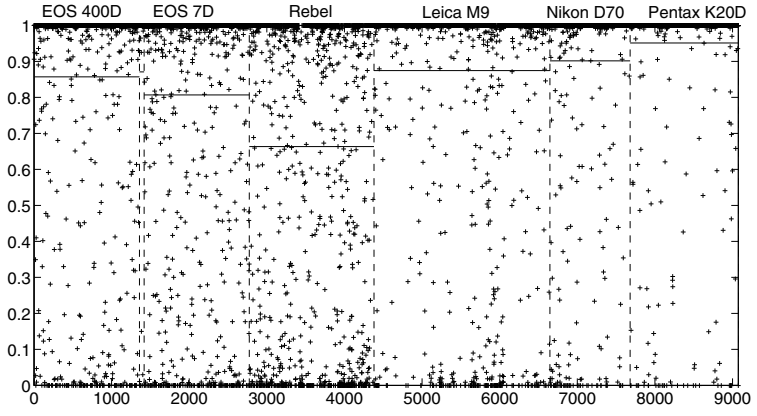


Fig. 3. Probability with which each cover image $i \in \{1, \ldots, 9074\}$ from BOSSbase was correctly classified as cover over 1000 random splits (8074/1000). The images are sorted by cameras. The average detection for each camera is displayed with a horizontal line.

with a large contrast, and many contained complex content, such as shots in a forest with many fine branches. The high dimensionality of the feature set and the relatively low number of training examples mean that some images will be located in sparsely populated regions of the feature space. The classifier generalizes to them but, due to lack of similar features in their neighborhood, the decision boundary is not likely to be well placed. As a result, some images are consistently misclassified.

Also, 6627 cover images were always correctly detected as cover and 6647 stego images were always detected as stego. The intersection of these two sets contains 4836 BEST images that were always detected correctly both in their cover and stego forms. These easiest-to-classify images did not contain many edges or textures, some were out-of-focus shots or shots with low depth of field and images with a small dynamic range of pixel values. Table 5 displays the average grayscale, average number of pixels saturated at 255, and average texture defined as $t = c \cdot \sum_{ij} |x_{ij} - x_{i,j-1}|$, with c being a scaling constant. Overall, images with a high number of saturated pixels and bright / textured images are harder to classify. Lower average grayscale is connected to a lower dynamic range, which indeed will make detection of embedding changes easier. The effect of saturated pixels, however, is more mysterious.

Table 5. Average grayscale, number of pixels saturated at 255, and texture for BEST, FAs, and MDs from BOSSbase

Images	Avg. gray	Avg. saturation	Texture
BEST	74.1	2046	1.73
FAs	101.3	4415	4.66
MDs	102.0	5952	3.95

4.3 Performance on BOSSrank

This section briefly discusses our performance on the BOSSrank set used for the BOSS competition [1]. It consists of 847 images taken by Leica M9 and 153 images from Panasonic Lumix DMC-FZ50. Total of 518 images were covers, while the remaining 482 were stego images embedded with relative payload 0.4 bpp.

The best score we achieved on BOSSrank was $1-P_{\rm E} = 80.3\%$ or $P_{\rm E} = 19.6\%$.[3] It was obtained for a submission generated from a 25,993-dimensional feature set trained on $34,719$ images[4] with $L = 31$ and $d_{\rm red} = 2400$. More details about this feature set and our experience with BOSS appear in our other paper in this volume [6]. The drop in performance w.r.t. our results on BOSSbase is caused by the cover-source mismatch and the lack of robustness of our ensemble classifier.[5] While our detector was trained on BOSSbase, BOSSrank images are coming from a different source. The Panasonic Lumix images are not in BOSSbase at all and they were taken in JPEG instead of the RAW format. While the Leica M9 is in BOSSbase, it forms only about 25% of the database (2267 images). The cover source mismatch is a serious issue for practical steganography as it lowers the detection accuracy and complicates controlling the error rates of practical detectors. The cover-source mismatch is also the reason why our detector that used the higher-dimensional set $\mathcal{H}$ performed worse on BOSSrank even though we observed the opposite for BOSSbase.

5 Conclusion

Modern steganographic algorithms, such as HUGO, hide messages by approximately preserving a high-dimensional representation of covers that captures many complex dependencies among individual cover elements. The embedding is thus naturally adaptive and confines the modifications to hard-to-model regions of covers. This is the reason why steganalyzers that work in feature spaces of low dimension do not detect this type of embedding well. A possible way to improve the detection is to work with high-dimensional features as well. The two key open problems are the formation of such feature spaces and machine learning whose complexity scales favorably with dimension.

In particular, it is not sufficient to blindly increase the feature dimensionality for example by increasing the order of co-occurrence matrices or their range (threshold). This way, we would be adding sparsely-populated (noisy) features with low detectability. In this paper, we propose a methodology called HOLMES for forming a diverse high-dimensional feature vector. It consists of two steps – computing several types of higher-order residuals and then forming co-occurrence matrices from their neighboring values in a standard fashion.

[3] Our error on Leica was 17.7% and 30.0% on Panasonic.
[4] All training images were obtained from RAW images using the same BOSS script.
[5] Other classifiers, including linear SVMs, Gaussian SVMs, and the FLD were equally susceptible to the cover-source mismatch.

The residuals should be computed in the embedding domain and using pixel predictors that only depend on the neighboring pixels but not the central pixel being predicted. We also discovered that good residuals for content-adaptive steganalysis may be obtained using non-linear processing as minimal and maximal values of residuals computed from several different directions – the MINMAX residual. The emphasis should be on high *diversity* of the features rather than dimensionality so that combining features improves detection.

Having formed a high-dimensional feature vector, we coin the use of ensemble classifiers obtained by fusing decisions of simple detectors implemented using the Fisher linear discriminant. They were a crucial element in our participation in BOSS as their low complexity, simplicity, and speed enabled rapid development and optimization of the feature set to maximize the performance.

To summarize our attack, we were unable to use the fact that for HUGO the probability of embedding changes at individual pixels can be approximately estimated. It does not appear that giving the Warden probabilistic information about the selection channel is a weakness. Another lesson learned is that, as the level of sophistication of steganographic schemes increases, steganalysis needs to use high-dimensional feature sets and scalable machine learning.

Our attack on HUGO also reveals quite useful information about steganography design. While the authors of HUGO did strive to preserve a high-dimensional feature vector, they scaled the dimensionality simply by increasing the threshold T. Most features in this high-dimensional feature vector are, however, quite uninformative and trying to preserve them eventually weakens the algorithm. Instead, the dimensionality needs to be increased by adding more diverse features. We expect the future versions of HUGO working with more diverse feature spaces, such as the set $\mathcal{H}$, to be significantly more secure to attacks.

Acknowledgements. The work on this paper was 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 to the Soukal's Family, Tice Lerner, Jim Pittaresi, Thomas Gloe, Peggy Goldsberry, and Jonathan Cohen for providing their images for research purposes and to the BOSS Team for setting up the BOSS competition.

Appendix – The Final Feature Set

Table 6. The final HOLMES feature set $\mathcal{H}$ of dimensionality 33,963

Feature type	(s, q, m, T)	Dimensionality
MINMAX	$(3,1,3,4)$, $(3,2,3,3)$, $(4,\{2,3\},3,3)$ $(5,\{2,6\},3,3)$, $(6,10,3,3)$, $(7,20,3,3)$	$1458 + 7 \times 686 + 10 \times 162$
MARKOV	$(3,\{1,2\},4,1)$, $(4,\{2,3\},4,1)$, $(5,\{2,6\},4,1)$ $(6,\{5,10\},4,1)$, $(7,\{10,20\},4,1)$	$1458 + 7 \times 686 + 10 \times 162$
MINMAX	$(3,2,5,1)$	2×243
MINMAX	$(2,\{1,2\},4,2)$	2×1250
KB	$(9,\{1,2,4\},3,4)$	3×729
SQUARE	$(3,2,4,1)$	2×162
CALI	$(3,2,3,4)$, $(4,2,3,4)$	2×1458
EDGE	$(6,\{1,2,4\},3,4)$	3×1458
MINMAX	$(\{3,4,5,6,7\},c,3,4)$ summed	$1458 + 1250$
MARKOV	$(\{3,4,5,6,7\},c,4,2)$ summed	$1458 + 1250$

All feature types in a block between two highlighted lines are to be combined with all parameter sets. The KB set was formed by $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{KB}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{KB}})$, where $\mathbf{R}^{\mathrm{KB}}$ is the residual (4). The SQUARE set is obtained from the MINMAX residual with co-occurrence operator (8). In the CALI set, prior to computing the features from the MINMAX residual, the image was convolved with an averaging 2×2 kernel $[1\ 1; 1\ 1]$ in an attempt to calibrate the features as in [8]. The residual for EDGE was formed using (5) as the minimum and maximum values along edges in four different directions (residual $\mathbf{R}^{\mathrm{EDGEMIN}}$ and $\mathbf{R}^{\mathrm{EDGEMAX}}$) and then applying $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{EDGEMIN}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{EDGEMIN}})$, $C^{\mathrm{h}}(\mathbf{R}^{\mathrm{EDGEMAX}}) + C^{\mathrm{v}}(\mathbf{R}^{\mathrm{EDGEMAX}})$. The last four sets were obtained as sums of all five sets whose parameters appear in the second column.

References

1. Bas, P., Filler, T., Pevný, T.: Break our steganographic system – the ins and outs of organizing BOSS. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 59–70. Springer, Heidelberg (2011)
2. Chen, C., Shi, Y.Q.: JPEG image steganalysis utilizing both intrablock and interblock correlations. In: IEEE International Symposium on Circuits and Systems, ISCAS 2008, pp. 3029–3032 (May 2008)
3. Cutler, A., Zhao, G.: PERT - perfect random tree ensembles. Computing Science and Statistics 33, 490–497 (2001)
4. 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)
5. Filler, T., Pevný, T., Bas, P.: BOSS (July 2010), http://boss.gipsa-lab.grenoble-inp.fr/BOSSRank/

6. Fridrich, J., Kodovský, J., Goljan, M., Holub, V.: Breaking HUGO – the process discovery. In: Filler, T., Pevný, T., Ker, A., Craver, S. (eds.) IH 2011. LNCS, vol. 6958, pp. 85–101. Springer, Heidelberg (2011)
7. Goljan, M., Fridrich, J., Holotyak, T.: New blind steganalysis and its implications. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII, San Jose, CA, January 16-19, vol. 6072, pp. 1–13 (2006)
8. Ker, A.D.: Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters 12(6), 441–444 (2005)
9. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Delp, E.J., Wong, P.W. (eds.) 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)
10. Kodovský, J., Fridrich, J.: Calibration revisited. In: Dittmann, J., Craver, S., Fridrich, J. (eds.) Proceedings of the 11th ACM Multimedia & Security Workshop, Princeton, NJ, September 7-8, pp. 63–74 (2009)
11. Kodovský, J., Fridrich, J.: Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In: Memon, N.D., Delp, E.J., Wong, P.W., Dittmann, J. (eds.) Proceedings SPIE, Electronic Imaging, Watermarking, Security, and Forensics of Multimedia XIII, San Francisco, CA, January 23-26, vol. 7880 (2011)
12. Kodovský, J., Pevný, T., Fridrich, J.: Modern steganalysis can detect YASS. 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, vol. 7541, pp. 02-01–02-11 (2010)
13. Pevný, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
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. Shi, Y.Q., Chen, C., Chen, W.: A markov process based approach to effective attacking JPEG steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 249–264. Springer, Heidelberg (2007)

I Have a DREAM! (DiffeRentially privatE smArt Metering)

Gergely Ács and Claude Castelluccia

INRIA Rhone Alpes, Montbonnot, France
{gergely.acs,claude.castelluccia}@inrialpes.fr

Abstract. This paper presents a new privacy-preserving smart metering system. Our scheme is private under the differential privacy model and therefore provides strong and provable guarantees. With our scheme, an (electricity) supplier can periodically collect data from smart meters and derive aggregated statistics without learning anything about the activities of individual households. For example, a supplier cannot tell from a user's trace whether or when he watched TV or turned on heating. Our scheme is simple, efficient and practical. Processing cost is very limited: smart meters only have to add noise to their data and encrypt the results with an efficient stream cipher.

1 Introduction

Several countries throughout the world are planning to deploy smart meters in households in the very near future. The main motivation, for governments and electricity suppliers, is to be able to match consumption with generation. Traditional electrical meters only measure total consumption on a given period of time (i.e., one month or one year). As such, they do not provide accurate information of when the energy was consumed. Smart meters, instead, monitor and report consumption in intervals of few minutes. They allow the utility provider to monitor, almost in real-time, consumption and possibly adjust generation and prices according to the demand. Billing customers by how much is consumed and at what time of day will probably change consumption habits to help matching consumption with generation. In the longer term, with the advent of smart appliances, it is expected that the smart grid will remotely control selected appliances to reduce demand.

Problem statement: Although smart metering might help improving energy management, it creates many new privacy problems [2]. Smart meters provide very accurate consumption data to electricity providers. As the interval of data collected by smart meters decreases, the ability to disaggregate low-resolution data increases. Analyzing high-resolution consumption data, Nonintrusive Appliance Load Monitoring (NALM) [11] can be used to identify a remarkable number of electric appliances (e.g., water heaters, well pumps, furnace blowers, refrigerators, and air conditioners) employing exhaustive appliance signature libraries. Researchers are now focusing on the myriad of small electric devices around the home such as personal computers, laser printers, and light bulbs [14]. Moreover, it has also been shown that even simple off-the-shelf

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 118–132, 2011.

statistical tools can be used to extract complex usage patterns from high-resolution consumption data [15]. This extracted information can be used to profile and monitor users for various purposes, creating serious privacy risks and concerns. As data recorded by smart meters is lowering in resolution, and inductive algorithms are quickly improving, it is urgent to develop privacy-preserving smart metering systems that provide strong and provable guarantees.

Contributions: We propose a privacy-preserving smart metering scheme that guarantees users' privacy while still preserving the benefits and promises of smart metering. Our contributions are many-fold and summarized as follows:

- We provide the first provably private and distributed solution for smart metering that optimizes utility without relying on a trusted third party (i.e., an aggregator). We were able to avoid the use of a trusted third party by proposing a new distributed Laplacian Perturbation Algorithm (DLPA).

  In our scheme, smart meters are grouped into clusters, where a cluster is a group of hundreds or thousands of smart meters corresponding, for example, to a quarter of a city. Each smart meter sends, at each sampling period, their measures to the supplier. These measures are noised and encrypted such that the supplier can compute the noised aggregated electricity consumption of the cluster, at each sampling period, without getting access to individual values. The aggregate is noised just enough to provide differential privacy to each participating user, while still providing high utility (i.e., low error). Our scheme is secure under the differential privacy model and therefore provides strong and provable privacy guarantees. In particular, we guarantee that the supplier can retrieve information about any user consumption only up to a predefined threshold, no matter what auxiliary information it knows about that user. Our scheme is simple, efficient and practical. It requires either one or two rounds of message exchanges between a meter and the supplier. Furthermore, processing cost is very limited: smart meters only have to add noise to their data and encrypt the results with an efficient stream cipher. Finally, our scheme is robust against smart meter failures and malicious nodes. More specifically, it is secure even if an α fraction of all nodes of a cluster collude with the supplier, where α is a security parameter.
- We implemented a new electricity trace generation tool based on [19] which generates realistic, one-minute resolution synthetic consumption data of different households. We used this simulator to evaluate the performance and privacy of our proposal.

Because of space constraint, the security analysis of our scheme is not included in this paper. This analysis is however included in the longer version of this paper [1]. This extended version also includes additional performance results.

2 Related Work

Several papers addressed the privacy problems of smart metering in the recent past [8,15,2,16,3,4,18,10]. However, only a few of them have proposed technical solutions to protect users' privacy. In [2,3], the authors discuss the different security aspects of

smart metering and the conflicting interests among stakeholders. The privacy of billing is considered in [18,15]. Seemingly, the privacy of monitoring the sum consumption of multiple users may be solved by simply anonymizing individual measurements like in [8] or using some mixnet. However, these "ad-hoc" techniques are dangerous and do not provide any real assurances of privacy. Several prominent examples in the history have shown that ad-hoc methods do not work [12]. Moreover, these techniques require an existing trusted third party who performs anonymization. The authors in [4] perturb the released aggregate with random noise and use a different model from ours to analyze the privacy of their scheme. However, they do not encrypt individual measurements which means that the added noise must be large enough to guarantee reasonable privacy. As individual noise shares sum up at the aggregation, the final noise makes the aggregate useless. In contrast to this, [10] uses homomorphic encryption to guarantee privacy for individual measurements. However, the aggregate is not perturbed which means that it is not differential private.

Three closely related works to ours are [17,20,6]. [6] describes protocols for generating shares of random noise which is secure against malicious participants. However, it requires communication between users and it uses expensive secret sharing techniques resulting in high overhead in case of large number of users. In [17], the authors propose a scheme to differential privately aggregate sums over multiple slots when the aggregator is untrusted. However, they use the threshold Paillier cryptosystem [9] for homomorphic encryption which is much more expensive compared to [5] that we use. They also use different noise distribution technique which requires several rounds of message exchanges between the users and the aggregator. By contrast, our solution is much more efficient and simple: it requires only a single message exchange if there are no node failures, otherwise, we only need one extra round. In addition, our solution does not rely on expensive public key cryptography during aggregation.

A recent paper [20] proposes another technique to privately aggregate time series data. This work differs from ours as follows: (1) they use a Diffie-Hellman-based encryption scheme, whereas our construction is based on a more efficient construction that only use modular additions. This approach is better adapted to resource constrained devices like smart meters. (2) Although [20] does not require the establishment (and storage) of pairwise keys between nodes as opposed to our approach, it is unclear how [20] can be extended to tolerate node and communication failures. By contrast, our scheme is more robust, as the encryption key of non-responding nodes is known to other nodes in the network that can help to recover the aggregate. (3) Finally, [20] uses a different noise generation method from ours, but this technique only satisfies the relaxed (ε, δ)-differential privacy definition. Indeed, in their scheme, each node adds noise probabilistically which means that none of the nodes add noise with some positive probability δ. Although δ can be arbitrarily small, this also decreases the utility. By contrast, in our scheme, $\delta = 0$ while ensuring nearly optimal utility.

3 The Model

3.1 Network Model

The network is composed of four major parts: the *supplier/aggregator*, the *electricty distribution network*, the *communication network*, and the *users* (customers). Every user

is equipped with an electricity smart meter, which measures the electricity consumption of the user in every T_p long period, and, using the communication network, sends the measurement to the aggregator at the end of every slot (in practice, T_p is around 1-30 minutes). Note that the communication and distribution network can be the same (e.g., when PLC technology is used to transfer data). The measurement of user i in slot t is denoted by X_t^i. The consumption profile of user i is described by the vector $(X_1^i, X_2^i, \ldots)$. Privacy directly correlates with T_p; finer-grained samples means more accurate profile, but also entails weaker privacy. The supplier is interested in the sum of all measurements in every slot (i.e., $\sum_{i=1}^{N} X_t^i \stackrel{\text{def}}{=} \mathbf{X}_t$).

As in [4], we also assume that smart meters are trusted devices (i.e., tamper-resistant) which can store key materials and perform crypto computations. This realistic assumption has also been confirmed in [3]. We assume that each node is configured with a private key and gets the corresponding certificate from a trusted third party. For example, each country might have a third party that generates these certificate and can additionally generate the "supplier" certificates to supplier companies [3]. As in [3], we also assume that public key operations are employed only for initial key establishment, probably when a meter is taken over by a new supplier. Messages exchanged between the supplier and the meters are authenticated using pairwise MACs[1]. Smart meters are assumed to have bidirectional communication channel (using some wireless or PLC technology) with the aggregator, but the meters cannot communicate with each other. We suppose that nodes may (randomly) fail, and in these cases, cannot send their measurements to the aggregator. However, nodes are supposed to use some reliable transport protocol to overcome the transient communication failures of the channel. Finally, we note that smart meters also allow the supplier to perform fine-grained billing based on time-dependant variable tariffs. Here, we are not concerned with the privacy and security problems of this service. Interested readers are referred to [18, 15].

3.2 Adversary Model

In general, the objective of the adversary is to infer detailed information about household activity (e.g, how many people are in home and what they are doing at a given time). In order to do that, it needs to extract complex usage patterns of appliances which include the level of power consumption, periodicity, and duration.

In this paper we consider a *dishonest-but-non-intrusive (DN) adversary*. A DN adversary may not follow the protocol correctly and is allowed to provide false information to manipulate the collected data. He may also collude with some (malicious) smart meters. However, he is not allowed to access or modify the distribution network to mount attacks. In particular, he is not allowed to install wiretapping devices to eavesdrop on the victim's consumption.

3.3 Privacy Model

We use differential privacy [7] that models the adversary described above. In particular, differential privacy guarantees that a user's privacy should not be threatened substantially more if he provides his measurement to the supplier.

[1] Please refer to [16] for a more detailed discussion about key management issues in smart metering systems.

Definition 1 (ε-differential privacy). *An algorithm $\mathcal{A}$ is ε-differential private, if for all data sets D_1 and D_2, where D_1 and D_2 differ in at most a single user, and for all subsets of possible answers $S \subseteq Range(\mathcal{A})$,*

$$P(\mathcal{A}(D_1) \in S) \leq e^{\varepsilon} \cdot P(\mathcal{A}(D_2) \in S)$$

Differential private algorithms produce indistinguishable outputs for similar inputs (more precisely, differing by a single entry), and thus, the modification of any single user's data in the dataset (including its removal or addition) changes the probability of any output only up to a multiplicative factor e^{ε}. The parameter ε allows us to control the level of privacy. Lower values of ε implies stronger privacy, as they restrict further the influence of a user's data on the output. Note that this model guarantees privacy for a user even if all other users' data is known to the adversary (e.g., it knows all measurements comprising the aggregate except the target user's), like when $N-1$ out of N users are malicious and cooperate with the supplier. The definition of differential privacy also maintains a *composability property*: the composition of differential private algorithms remains differential private and their ε parameters are accumulated. In particular, a protocol having t rounds, where each round is individually ε differential private, is itself $t \cdot \varepsilon$ differential private.

3.4 Output Perturbation: Achieving Differential Privacy

Let's say that we want to publish in a differentially private way the output of a function f. The following theorem says that this goal can be achieved by perturbing the output of f; simply adding a random noise to the value of f, where the noise distribution is carefully calibrated to the global sensitivity of f, results in ε-differential privacy. The global sensitivity of a function is the maximum "change" in the value of the function when its input differs in a single entry. For instance, if f is the sum of all its inputs, the sensitivity is the maximum value that an input can take.

Theorem 1 ([7]). *For all $f : \mathbb{D} \rightarrow \mathbb{R}^r$, the following mechanism $\mathcal{A}$ is ε-differential private: $\mathcal{A}(D) = f(D) + \mathcal{L}(S(f)/\varepsilon)$, where $\mathcal{L}(S(f)/\varepsilon)$ is an independently generated random variable following the Laplace distribution and $S(f)$ denotes the global sensitivity of f*[2].

Example 1. To illustrate these definitions, consider a mini smart metering application, where users U_1, U_2, and U_3 need to send the sum of their measurements in two consecutive slots. The measurements of U_1, U_2 and U_3 are $(X_1^1 = 300, X_2^1 = 300)$, $(X_1^2 = 100, X_2^2 = 400)$, and $(X_1^3 = 50, X_2^3 = 150)$, resp. The nodes want differential privacy for the released sums with at least a $\varepsilon = 0.5$. Based on Theorem 1, they need to add $\mathcal{L}(\lambda = \max_i \sum_t X_t^i/0.5 = 1200)$ noise to the released sum in **each** slot. This noise ensures $\varepsilon = \sum_t X_t^1/\lambda = 0.5$ individual indistinguishability for U_1, $\varepsilon = 0.42$ for U_2, and $\varepsilon = 0.17$ for U_3. Hence, the global $\varepsilon = 0.5$ bound is guaranteed to all. Another interpretation is that U_1 has $\varepsilon_1 = X_1^1/\lambda = 0.25$, $\varepsilon_2 = X_2^1/\lambda = 0.25$ privacy in each individual slot, and $\varepsilon = \varepsilon_1 + \varepsilon_2 = 0.5$ considering all two slots following from the composition property of differential privacy.

[2] Formally, let $f : \mathbb{D} \rightarrow \mathbb{R}^r$, then the global sensitivity of f is $S(f) = \max ||f(D_1) - f(D_2)||_1$, where D_1 and D_2 differ in a single entry and $|| \cdot ||_1$ denotes the L_1 distance.

3.5 Utility Definition

Let $f : \mathbb{D} \rightarrow \mathbb{R}$. In order to measure the utility, we quantify the difference between $f(D)$ and its perturbed value (i.e., $\hat{f}(D) = f(D) + \mathcal{L}(\lambda)$) which is the error introduced by LPA (Laplacian Perturbation Algorithm). A common scale-dependant error measure is the Mean Absolute Error (MAE), which is $\mathbb{E}|f(D) - \hat{f}(D)|$ in our case. However, the error should be dependent on the non-perturbed value of $f(D)$; if $f(D)$ is greater, the added noise becomes small compared to $f(D)$ which intuitively results in better utility. Hence, we rather use a slightly modified version of a scale-independent metric called Mean Absolute Percentage Error (MAPE), which shows the proportion of the error to the data, as follows.

Definition 2 (Error function). *Let $D_t \in \mathbb{D}$ denote a dataset in time-slot t. Furthermore, let $\delta_t = \frac{|f(D_t) - \hat{f}(D_t)|}{f(D_t)+1}$ (i.e., the value of the error in slot t). The error function is defined as $\mu(t) = \mathbb{E}(\delta_t)$. The expectation is taken on the randomness of $\hat{f}(D_t)$. The standard deviation of the error is $\sigma(t) = \sqrt{Var(\delta_t)}$ in time t.*

In the rest of this paper, the terms "utility" and "error" are used interchangeably.

4 Secure Aggregation without Aggregator: An Overview

Our scheme enables the supplier to calculate the sum of maximum N measurements (i.e., $\sum_{i=1}^{N} X_t^i = \mathbf{X}_t$ in all t) coming from N different smart meters while ensuring ε-differential privacy for each user. This is guaranteed if the supplier can only access $\mathbf{X}_t + \mathcal{L}(\lambda(t))$, where $\mathcal{L}(\lambda(t))$[3] is the Laplacian noise calibrated to ε as it has been described in Section 3.4.

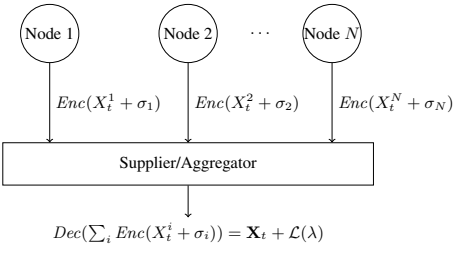


Fig. 1. Our approach: aggregation without trusted entity. If $\sigma_i = \mathcal{G}_1(N, \lambda) + \mathcal{G}_2(N, \lambda)$, where $\mathcal{G}_1, \mathcal{G}_2$ are i.i.d gamma noise, then $\sum_{i=1}^{N} \sigma_i = \mathcal{L}(\lambda)$.

A simple solution would be to rely on an aggregator that aggregates the N samples and adds Laplacian noise before forwarding the result to the supplier. Although this scheme would be differential private, it only works if the aggregator is trusted. In particular, the scheme will not be secure if the aggregator omits to add the noise.

[3] We will use the notation λ instead of $\lambda(t)$ if the dependency on time is obvious in the context.

Our scheme, instead, does not rely on any centralized aggregator. The noise is added by each smart meter on their individual data and encrypted in such a way that the aggregator can only compute the (noisy) aggregate. Note that with our approach the aggregator and the supplier do need to be separate entities. The supplier can even play the role of the aggregator, as the encryption prevents it to access individual measurements, and the distributed generation of the noise ensures that it cannot manipulate the noise.

Our proposal is composed of 2 main steps: distributed generation of the Laplacian noise and encryption of individual measurements. These 2 steps are described in the remainder of this section.

4.1 Distributed Noise Generation: A New Approach

In our proposal, the Laplacian noise is generated in a fully distributed way as is illustrated in Figure 4. We use the following lemma that states that the Laplace distribution is divisible and be constructed as the sum of i.i.d. gamma distributions. As this divisibility is infinite, it works for arbitrary number of users.

Lemma 1 (Divisibility of Laplace distribution [13]). *Let $\mathcal{L}(\lambda)$ denote a random variable which has a Laplace distribution with PDF $f(x,\lambda) = \frac{1}{2\lambda}e^{\frac{|x|}{\lambda}}$. Then the distribution of $\mathcal{L}(\lambda)$ is infinitely divisible. Furthermore, for every integer $n \geq 1$, $\mathcal{L}(\lambda) = \sum_{i=1}^{n}[\mathcal{G}_1(n,\lambda) - \mathcal{G}_2(n,\lambda)]$, where $\mathcal{G}_1(n,\lambda)$ and $\mathcal{G}_2(n,\lambda)$ are i.i.d. random variables having gamma distribution with PDF $g(x,n,\lambda) = \frac{(1/\lambda)^{1/n}}{\Gamma(1/n)}x^{\frac{1}{n}-1}e^{-x/\lambda}$ where $x \geq 0$.*

The lemma comes from the fact that $\mathcal{L}(\lambda)$ can be represented as the difference of two i.i.d exponential random variables with rate parameter $1/\lambda$. Moreover, $\sum_{i=1}^{n}\mathcal{G}_1(n,\lambda) - \sum_{i=1}^{n}\mathcal{G}_2(n,\lambda) = \mathcal{G}_1(1/\sum_{i=1}^{n}\frac{1}{n},\lambda) - \mathcal{G}_2(1/\sum_{i=1}^{n}\frac{1}{n},\lambda) = \mathcal{G}_1(1,\lambda) - \mathcal{G}_2(1,\lambda)$ due to the summation property of the gamma distribution[4]. Here, $\mathcal{G}_1(1,\lambda)$ and $\mathcal{G}_2(1,\lambda)$ are i.i.d exponential random variable with rate parameter $1/\lambda$ which completes the argument.

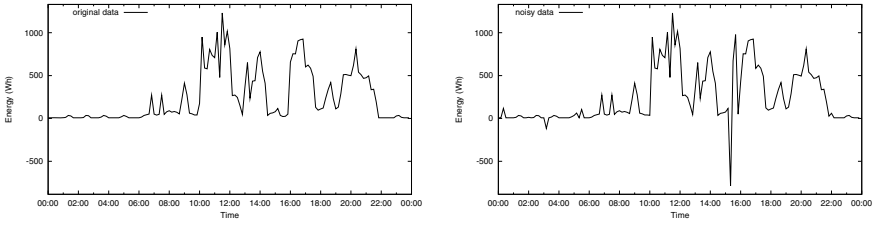
Our distributed sanitization algorithm is simple; user i calculates value $\hat{X}_t^i = X_t^i + \mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)$ in slot t and sends it to the aggregator, where $\mathcal{G}_1(N,\lambda)$ and $\mathcal{G}_2(N,\lambda)$ denote two random values independently drawn from the same gamma distribution. Now, if the aggregator sums up all values received from the N users of a cluster, then $\sum_{i=1}^{N}\hat{X}_t^i = \sum_{i=1}^{N}X_t^i + \sum_{i=1}^{N}[\mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)] = \mathbf{X}_t + \mathcal{L}(\lambda)$ based on Lemma 1.

The utility of our distributed scheme is defined as $\mu(t) = \frac{1}{\mathbf{X}_t+1}\mathbb{E}|\mathbf{X}_t - \mathbf{X}_t + \sum_{i=1}^{n}[\mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)]| = \frac{\mathbb{E}|\mathcal{L}(\lambda)|}{\mathbf{X}_t+1} = \frac{\lambda}{\mathbf{X}_t+1}$, and $\delta(t) = \frac{\lambda}{\mathbf{X}_t+1}$.

4.2 Encryption

The previous step is not enough to guarantee privacy as only the sum of the measurements (i.e., $\hat{\mathbf{X}}_t$) is differential private but not the individual measurements. In particular, the aggregator has access to $\hat{X}_t^i$, and even if $\hat{X}_t^i$ is noisy, $\mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)$ is usually insufficient to provide reasonable privacy for individual users if $N \gg 1$. This is

[4] The sum of i.i.d. gamma random variables follows gamma distribution (i.e., $\sum_{i=1}^{n}\mathcal{G}(k_i,\lambda) = \mathcal{G}(1/\sum_{i=1}^{n}\frac{1}{k_i},\lambda)$).



(a) X_t^i (b) $X_t^i + \mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)$

Fig. 2. The original and noisy measurements of user i, where the added noise is $\mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)$ ($N = 100$, T_p is 10 min).

illustrated in Figure 2, where an individual's noisy and original measurements slightly differ.

To address this problem, each contribution is encrypted using a modulo addition-based encryption scheme, inspired by [5], such that the aggregator can only decrypt the sum of the individual values, and cannot access any of them. In particular, let k_i denote a random key generated by user i inside a cluster such that $\sum_{i=1}^{N} k_i = 0$, and k_i is not known to the aggregator. Furthermore, $Enc()$ denotes a probabilistic encryption scheme such that $Enc(p,k,m) = p + k \mod m$, where p is the plaintext, k is the encryption key, and m is a large integer. The adversary cannot decrypt any $Enc(\hat{X}_t^i, k_i, m)$, since it does not know k_i, but it can easily retrieve the noisy sum by adding the encrypted noisy measurements of all users; $\sum_{i=1}^{N} Enc(\hat{X}_t^i, k_i, m) = \sum_{i=1}^{N} \hat{X}_t^i + \sum_{i=1}^{N} k_i = \sum_{i=1}^{N} \hat{X}_t^i \mod m$. If $z = \max_{i,t}(\hat{X}_t^i)$ then m should be selected as $m = 2^{\lceil \log_2(z \cdot N) \rceil}$ [5]. The generation of k_i is described in Section 5.2.

5 Protocol Description

5.1 System Setup

In our scheme, nodes are grouped into clusters of size N, where N is a parameter. The protocol requires the establishment of pairwise keys between each pair of nodes inside a cluster that can be done by using traditional Diffie-Hellman key exchange as follows. When a node v_i is installed, it provides a self-signed DH component and its certificate to the supplier. Once all the nodes of a cluster are installed, or a new node is deployed, the supplier broadcasts the certificates and public DH components of all nodes. Finally, each node v_i of the cluster can compute a pairwise key $K_{i,j}$ shared with any other node v_j in the networks.

5.2 Smart Meter Processing

Each node v_i sends at time t its periodic measurement, X_t^i, to the supplier as follows:

Phase 1 (Data sanitization): Node v_i calculates value $\hat{X}_t^i = X_t^i + \mathcal{G}_1(N,\lambda) - \mathcal{G}_2(N,\lambda)$, where $\mathcal{G}_1(N,\lambda)$ and $\mathcal{G}_2(N,\lambda)$ denote two random values independently drawn from the same gamma distribution and N is the cluster size.

Phase 2 (Data encryption): Each noisy data $\hat{X}_t^i$ is then encrypted into $Enc(\hat{X}_t^i)$ using the modulo addition-based encryption scheme detailed in Section 4.2. The following extension is then applied to generate the encryption keys: Each node, v_i, selects ℓ other nodes randomly, such that if v_i selects v_j, then v_j also selects v_i. Afterwards, both nodes generate a common dummy key k from their pairwise key $K_{i,j}$; v_i adds k to $Enc(\hat{X}_t^i)$ and v_j adds $-k$ to $Enc(\hat{X}_t^j)$. As a result, the aggregator cannot decrypt the individual ciphertexts (it does not know the dummy key k). However, it adds all the ciphertexts of a given cluster, the dummy keys cancel out and it retrieves the encrypted sum of the (noisy) contributions. The more formal description is as follows:

1. node v_i selects some nodes of the cluster randomly (we call them participating nodes) using a secure pseudo random function (PRF) such that if v_i selects v_j, then v_j also selects v_i. In particular, v_i selects v_j if mapping $PRF(K_{i,j}, r_1)$ to a value between 0 and 1 is less or equal than $\frac{w}{N-1}$, where r_1 is a public value changing in each slot. We denote by ℓ the number of selected participating nodes, and $\mathsf{ind}_i[j]$ (for $j = 1, \ldots, \ell$) denotes the index of the ℓ nodes selected by node v_i. Note that, for the supplier, the probability that v_i selects v_j is $\frac{w}{N-1}$ as it does not know $K_{i,j}$. The expected value of ℓ is w.
2. v_i computes for each of its ℓ participating nodes a *dummy key*. A dummy key between v_i and v_j is defined as $\mathsf{dkey}_{i,j} = (i-j)/|i-j| \cdot PRF(K_{i,j}, r_2)$, where $K_{i,j}$ is the key shared by v_i and v_j, and $r_2 \neq r_1$ is public value changing in each slot. Note that $\mathsf{dkey}_{i,j} = -\mathsf{dkey}_{j,i}$.
3. v_i then computes $Enc(\hat{X}_t^i) = \hat{X}_t^i + K_i' + \sum_{j=1}^{\ell} \mathsf{dkey}_{i,\mathsf{ind}_i[j]} \pmod m$, where K_i' is the keystream shared by v_i and the aggregator which can be established using the DH protocol as above, and m is a large integer (see [5]). Note that m must be larger than the sum of all contributions (i.e., final aggregate) plus the Laplacian noise.[5]

   Note that $\hat{X}_t^i$ is encrypted multiple times: it is first encrypted with the keystream K_i' and then with several dummy keys. K_i' is needed to ensure confidentiality between a user and the aggregator. The dummy keys are needed to prevent the aggregator (supplier) from retrieving $\hat{X}_t^i$.
4. $Enc(\hat{X}_t^i)$ is sent to the aggregator (supplier).

5.3 Supplier Processing

Phase 1 (Data Aggregation): At each slot, the supplier aggregates the N measurements received from the cluster smart meters by summing them, and obtains $\sum_{i=1}^{N} Enc(X_t^i)$. In particular, $Enc(\hat{\mathbf{X}}_t) = \sum_{i=1}^{N}(\hat{X}_t^i + K_i') + \sum_{i=1}^{N}\sum_{j=1}^{\ell} \mathsf{dkey}_{i,\mathsf{ind}_i[j]} \pmod m$, where $\sum_{i=1}^{N}\sum_{j=1}^{\ell} \mathsf{dkey}_{i,\mathsf{ind}_i[j]} = 0$ because $\mathsf{dkey}_{i,j} = -\mathsf{dkey}_{j,i}$. Hence, $Enc(\hat{\mathbf{X}}_t) = \sum_{i=1}^{N}(\hat{X}_t^i + K_i') = \sum_{i=1}^{N} Enc(\hat{X}_t^i)$.

[5] Note that the noise is a random value from an infinite domain and this sum might be larger than m. However, choosing sufficiently large m, the probability that the sum exceeds m can be made arbitrary small due to the exponential tail of the Laplace distribution.

Phase 2 (Data decryption): The aggregator then decrypts the aggregated value by subtracting the sum of the node's keystream, and retrieves the sum of the noisy measures: $\sum_{i=1}^{N} Enc(\hat{X}_t^i) - \sum_{i=1}^{N} K_i' = \sum_{i=1}^{N} \hat{X}_t^i \pmod m$ where $\sum_{i=1}^{N} \hat{X}_t^i = \sum_{i=1}^{N} X_t^i + \sum_{i=1}^{N} \mathcal{G}_1(N,\lambda) - \sum_{i=1}^{N} \mathcal{G}_2(N,\lambda) = \sum_{i=1}^{N} X_t^i + \mathcal{L}(\lambda)$ based on Lemma 1.

The main idea of the scheme is that the aggregator is not able to decrypt the individual encrypted values because it does not know the dummy keys. However, by adding the different encrypted contributions, dummy keys cancel each other and the aggregator can retrieve the sum of the plaintext. The resulting plaintext is then the perturbed sums of the measurements, where the noise ensures the differential privacy of each user.

Complexity: Let b denote the size of the pairwise keys (i.e., $K_{i,j}$). Our scheme has $O(N \cdot b)$ storage complexity, as each node needs to store $\ell \leq N$ pairwise keys. The computational overhead is dominated by the encryption and the key generation complexity. The encryption is composed of $\ell \leq N$ modular addition of $\log_2 m$ bits long integers, while the key generation needs the same number of PRF executions. This results in a complexity of $O(N \cdot (\log_2 m + c(b)))$, where $c(b)$ is the complexity of the applied PRF function.[6]

6 Adding Robustness

We have assumed so far that all the N nodes of a cluster participated in the protocol. However, it might happen that, for several different reasons (e.g., node or communication failures) some nodes are not able to participate in each epoch. This would have two effects: first, security will be reduced since the sum of the noise added by each node will not be equivalent to $\mathcal{L}(\lambda)$. Hence, differential privacy may not be guaranteed. Second, the aggregator will not be able to decrypt the aggregated value since the sum of the dummy keys will not cancel out.

In this section, we extend our scheme to resist node failures. We propose a scheme which resists the failure of up to M out of N nodes, where M is a configuration parameter. We will study later the impact of the value M on the scheme performance.

Sanitization Phase Extension. In order to resist the failure of M nodes, each node should add the following noise to their individual measurement: $\mathcal{G}_1(N - M, \lambda) - \mathcal{G}_2(N - M, \lambda)$. Note that $\sum_{i=1}^{N-M} [\mathcal{G}_1(N - M, \lambda) - \mathcal{G}_2(N - M, \lambda)] = \mathcal{L}(\lambda)$. Therefore, this sanitization algorithm remains differential private, if at least $N - M$ nodes participate in the protocol. Note that in that case each node adds extra noise to the aggregate in order to ensure differential privacy even if fewer than M nodes fail to send their noise share to the aggregator.

[6] For instance, if $\log_2 m = 32$ bits (which should be sufficient in our application), $b = 128$, and $N = 1000$, a node needs to store 16 Kb of key data and perform maximum 1000 additions along with 1000 subtractions (for modular reduction) on 32 bits long integers, and maximum 1000 PRF executions. This overhead should be negligible even on constrained embedded devices.

Encryption Phase Extension. The encryption phase consists of two rounds. In the first round, each node adds a secret random value to its encrypted value before releasing it. In the second round, every node reveals its random value along with the missing dummy keys that it knows:

1. Each node v_i sends $Enc(\hat{X}_t^i) = \hat{X}_t^i + K_i' + \sum_{j=1}^{\ell} \mathsf{dkey}_{i,\mathsf{ind}_i[j]} + C_i \pmod m$ where C_i is the secret random key of v_i generated randomly in each round.
2. After receiving all measurements, the aggregator asks all nodes for their random keys and the missing dummy keys through broadcasting the id of the non-responding nodes.
3. Each node v_i verifies whether any ids in this broadcast message are in its participating node list, where the set of the corresponding participating nodes is denoted by S. Then, v_i replies with $\sum_{j\in S} \mathsf{dkey}_{i,\mathsf{ind}_i[j]} + C_i \pmod m$.
4. The aggregator subtracts all received values from $\sum_{i=1}^{N} Enc(\hat{X}_t^i)$ which results in $\sum_{i=1}^{N}(\hat{X}_t^i + K_i')$, as the random keys as well as the dummy keys cancel out.

The main idea of this scheme is that C_i prevents the supplier to recover $\hat{X}_t^i$ by combining the messages of nodes. Indeed, if v_i did not add C_i to its messages in Step 1 and 3, the supplier could easily get $\hat{X}_t^i$ by subtracting the responses of v_i's participating nodes (and K_i' that it knows), received in Step 3, from $Enc(\hat{X}_t^i)$, which is received in Step 1. However, since the supplier does not know the random keys, it cannot remove them from any messages but only from the final aggregate; subtracting the response of each node, received in Step 3, from the aggregate, all the dummy keys and secret random keys cancel out and the supplier obtains $\tilde{\mathbf{X}}_t$. Although the supplier can still recover $\hat{X}_t^i$ if it knows v_i's participating nodes (the supplier simply asks for all the dummy keys of v_i in Step 2 and subtracts v_i's response in Step 4 from $Enc(\hat{X}_t^i)$), this probability can be made practically small by adjusting w and N correctly (see [1] for details).

Utility Evaluation. If all N nodes participate in the protocol, the added noise will be larger than $\mathcal{L}(\lambda)$ which is needed to ensure differential privacy. In particular, $\sum_{i=1}^{N}[\mathcal{G}_1(N-M,\lambda)-\mathcal{G}_2(N-M,\lambda)] = \mathcal{L}(\lambda)+\sum_{i=1}^{M}[\mathcal{G}_1(N-M,\lambda)-\mathcal{G}_2(N-M,\lambda)]$, where the last summand is the extra noise needed to tolerate the failure of maximum M nodes. Clearly, this extra noise increases the error if all N nodes operate correctly and add their noise shares faithfully. In what follows, we calculate the error and its standard deviation if we add this extra noise to the aggregate.

Theorem 2. *Let $\alpha = M/N$ and $\alpha < 1$. Then, $\mu(t) \leq \frac{2}{B(1/2,\frac{1}{1-\alpha})} \cdot \frac{\lambda(t)}{\mathbf{X}_t+1}$ and $\sigma(t) \leq \sqrt{\left(\frac{2}{1-\alpha} - \frac{4}{B(1/2,\frac{1}{1-\alpha})^2}\right)} \cdot \frac{\lambda(t)}{\mathbf{X}_t+1}$, where $B(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$ is the beta function.*

The derivation can be found in the full version of this paper [1]. Based on Theorem 2, $\sigma(t) = \mu(t) \cdot \left(\frac{2}{B(1/2,\frac{1}{1-\alpha})}\right)^{-1} \cdot \sqrt{\left(\frac{2}{1-\alpha} - \frac{4}{B(1/2,\frac{1}{1-\alpha})^2}\right)}$. It is easy to check that $\sigma(t)$ is always less or equal than $\mu(t)$. In particular, if $\alpha = 0$ (there are no malicious nodes and node failures), then $\sigma(t) = \mu(t)$. If $\alpha > 0$ then $\sigma(t) < \mu(t)$ but $\sigma(t) \approx \mu(t)$.

7 Simulation Results

7.1 Electricity Trace Simulator

Due to the lack of high-resolution real world data, we implemented an electricity trace simulator that can generate realistic one-minute resolution synthetic consumption traces. It is an extended version of the simulator developed in [19]. The simulator includes 33 different appliances. A trace is associated to a household and generated as follows: (1) A number of active persons is selected according to some distribution derived from real statistics. This number may vary as some members can enter or leave the house. (2) A set of appliances is then selected and activated at different time of the day according to an other distribution, which was also derived from real statistics.

Using this simulator, we generated 3000 electricity traces corresponding to different households, where the number of residents in each household was randomly selected between 1 and 5. Each trace was then sanitized according to our scheme. The noise added in each slot (i.e., $\lambda(t)$) was set to the maximum consumption in the slot (i.e., $\lambda(t) = \max_{1 \leq i \leq N} X_t^i$ where the maximum is taken on all users in the cluster). This amount of noise ensures $\varepsilon = 1$ indistinguishability for individual measurements in all slots. Although one can increase $\lambda(t)$ to get better privacy, the error will also increase. Note that the error $\mu_{\varepsilon'}(t)$ for other $\varepsilon' \neq \varepsilon$ values if $\mu_\varepsilon(t)$ is given is $\mu_{\varepsilon'}(t) = \frac{\varepsilon}{\varepsilon'} \cdot \mu_\varepsilon(t)$. We assume that $\lambda(t) = \max_i X_t^i$ is known a priori.

7.2 Performance Analysis: Error According to the Cluster Size

The error introduced by our scheme depends on the cluster size N. In this section, we present how the error varies according to N. Table 1 shows the average error value and its standard deviation, resp., depending on the size of the cluster in case of different values of α. The average error of a given cluster size N is the average of $\mathsf{mean}_t(\mu(t))$ of all N-sized clusters[7]. Obviously, higher N causes smaller error. Furthermore, a high α results in larger noise added by each meters, as described in Section 6, which also implies larger error. Interestingly, increasing the sampling period (i.e., T_p) results in slight error decrease[8], hence, we only considered 10 min sampling period. Otherwise noted explicitly, we assume 10 min sampling period in the sequel.

7.3 Privacy Evaluation

Privacy over Multiple Slots. So far, we have considered the privacy of individual slots, i.e. added noise to guarantee $\varepsilon = 1$ privacy in each slot of size 10 minutes. However, a trace is composed of several slots. For instance, if a user watches TV during multiple slots, we have guaranteed that an adversary cannot tell if the TV is watched in any particular slot (up to $\varepsilon = 1$). However, by analysing s consecutive slots corresponding

[7] In fact, the average error is approximated in Table 1: we picked up 200 different clusters for each N, and plotted the average of their $\mathsf{mean}_t(\mu(t))$. 200 is chosen according to experimental analysis. Above 200, the average error does not change significantly.

[8] This increase is less than 0.01 even if N is small when the sampling period is changed from 5 min to 15 min.

Table 1. The error depending on N and α using random clustering. The sampling period is 10 min.

N	$\alpha = 0$		$\alpha = 0.1$		$\alpha = 0.3$		$\alpha = 0.5$	
	mean	*dev*	*mean*	*dev*	*mean*	*dev*	*mean*	*dev*
100	0.118	0.021	0.135	0.023	0.150	0.026	0.177	0.032
300	0.047	0.004	0.050	0.005	0.054	0.006	0.070	0.007
500	0.029	0.002	0.031	0.002	0.036	0.002	0.044	0.003
800	0.019	0.001	0.020	0.001	0.023	0.001	0.028	0.001
1000	0.015	0.0008	0.016	0.0008	0.019	0.001	0.023	0.001

to a given period, it may be able to tell whether the TV was watched during that period (the privacy bound of this is $\varepsilon_s = \varepsilon \cdot s$ due to the composition property of differential privacy). Based on Theorem 1, we need to add noise $\lambda(t) = \sum_{i=1}^{s} \max_i X_t^i$ to *each* aggregate to guarantee $\varepsilon_s = 1$ bound in consecutive s slots, which, of course, results in higher error than in the case of $s = 1$ that we have assumed so far. Obviously, using the LPA technique, we cannot guarantee reasonably low error if s increases, as the necessary noise $\lambda(t) = \sum_{i=1}^{s} \max_i X_t^i$ can be large. In order to keep the error $\lambda(t)/\sum_{i=1}^{N} X_t^i$ low while ensuring better privacy than $\varepsilon_s = s \cdot \varepsilon$, one can increase the number of users inside each cluster (i.e., N).

Let's say that we want to compute the privacy of a user i between 14:00 and 18:00. If $\varepsilon(t) = X_t^i/\lambda(t)$ denotes the bound in a single slot t, then, based on the composition property of differential privacy, the bound ε_s for the $s = 24$ slots between 14:00 (84th slot) and 18:00 (108th slot) is $\sum_{t=84}^{108} \varepsilon(t)$. In general, $\varepsilon_s(t) = \sum_{i=t}^{t+s} \varepsilon(i)$.

Table 2 shows what average privacy of a user, in our dataset, as a function of the cluster size and value s. As the cluster size increases, the privacy bound decreases (i.e. privacy increases). The reason is that when the cluster size increases, the maximum consumption also increases with high probability. Since the noise is calibrated according to the maximum consumption within the cluster, it will be larger. This results in better privacy.

Table 2. ε_s of users considering all appliances depending on N and s. T_p is 10 min.

N	$s = 3$ (30 min)		$s = 24$ (4 h)		$s = 48$ (8 h)		$s = 144$ (24 h)	
	mean	*dev*	*mean*	*dev*	*mean*	*dev*	*mean*	*dev*
100	2.34	0.40	9.05	2.59	14.18	3.94	26.24	4.52
300	2.02	0.44	7.60	2.69	11.81	4.14	20.95	4.62
500	1.87	0.45	7.04	2.76	10.90	4.25	19.01	4.85
800	1.76	0.45	6.64	2.79	10.27	4.34	17.56	5.10
1000	1.67	0.47	6.35	2.87	9.83	4.47	16.55	5.40

Privacy of Appliances. In the previous section, we analysed how a user's privacy varies over time. In this section, we consider the privacy of the different appliances. For example, we aim at answering the following question: *what was my privacy when I was watching TV last evening between 18:00 and 20:00?* In order to

compute the corresponding privacy (i.e. ε_s), we compute $\sum_{t=108}^{120} \varepsilon(t)$, where $\varepsilon(t) = \{\text{TV's consumption in } t\}/\lambda(t)$.

We summarized some of the appliance privacy[9] in Table 3. Each value is computed by averaging the privacy provided in our 3000 traces.

The appliances can be divided into two major groups: the usage of active appliances indicate that the user is at home and uses the appliance (their consumption significantly changes during their active usage such as iron, vacuum, kettle, etc.), whereas passive appliances (like fridge, freezers, storage heater, etc.) have more or less identical consumption regardless the user is at home or not.

Table 3. ε_s of different appliances in case of different s. $N = 100$ and T_p is 10 min. The name of active devices are in bold.

	$s = 3$ (30 min)		$s = 24$ (4 h)		$s = 48$ (8 h)		$s = 144$ (24 h)	
	mean	*dev*	*mean*	*dev*	*mean*	*dev*	*mean*	*dev*
Lighting	0.91	1.28	2.68	1.82	3.63	2.29	4.89	2.97
Cassette / CD Player	0.02	0.04	0.05	0.05	0.07	0.05	0.09	0.07
Vacuum	1.67	7.59	1.82	7.58	1.90	7.60	1.94	7.63
Personal computer	0.21	0.32	0.83	0.49	1.09	0.58	1.42	0.83
TV	0.15	0.47	0.37	0.52	0.45	0.58	0.50	0.63
Microwave	1.13	4.23	1.26	4.24	1.29	4.27	1.31	4.29
Kettle	0.55	2.71	0.72	2.73	0.83	2.76	1.02	2.79
Washing machine	1.23	1.43	1.96	1.63	2.55	1.76	3.07	2.07
DESWH	3.34	14.01	6.13	14.06	7.83	14.23	10.85	14.57
Storage heaters	3.22	0.32	20.20	1.99	30.45	4.23	30.45	4.23
Refrigerator	0.44	0.22	1.06	0.49	1.40	0.64	1.92	0.80

Previous tables show two different, and conflicting, results. Table 2 shows that it may actually be difficult to hide the presence of activities in a household. In fact, computed ε values are quite high, even for large clusters. However, results presented in Table 3 are more encouraging. They show that, although, it might be difficult to hide a user's presence, it is still possible to hide his actual activity. In fact, appliances privacy bounds (ε values) are quite small, which indicates that an adversary will have difficulty telling whether the user is, for example, using his computer or watching TV during a given period of time. Furthermore, results show that it is even more difficult for an adversary to tell when a given activity actually started. Finally, we recall that in order to keep the error $\lambda(t)/\sum_{i=1}^{N} X_t^i$ low while ensuring better privacy one can always increase the number of users inside each cluster. For instance, doubling N from 100 to 200 allows to double the noise while keeping approximately the same error value (0.118 in Table 1 if $\alpha = 0$). This results in much better privacy, since, on average, doubling the noise halves the privacy parameter ε_s.

Although more work and research is needed, we believe this is a encouraging result for privacy. Protecting users' privacy against smart metering systems might not be a dream after all!

[9] Because of space constraint, we are only able to display a small sample of our results. A larger table can be found in [1]

References

1. Acs, G., Castelluccia, C.: I have a DREAM! (DIffeRentially PrivatE smart Metering). Technical Report (2011), http://planete.inrialpes.fr/~ccastel/PAPERS/IH_TR.pdf
2. Anderson, R., Fuloria, S.: On the security economics of electricity metering. In: Proceedings of the WEIS (June 2010)
3. Anderson, R., Fuloria, S.: Who controls the off switch? In: Proceedings of the IEEE SmartGridComm (June 2010)
4. Bohli, J.-M., Sorge, C., Ugus, O.: A Privacy Model for Smart Metering. In: Proceedings of IEEE ICC (2010)
5. Castelluccia, C., Mykletun, E., Tsudik, G.: Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. In: ACM/IEEE Mobiquitous Conference (2005)
6. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our Data, Ourselves: Privacy Via Distributed Noise Generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006)
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating Noise to Sensitivity in Private Data Analysis. In: Proceedings of the 3rd IACR TCC (2006)
8. Efthymiou, C., Kalogridis, G.: Smart Grid Privacy via Anonymization of Smart Metering Data. In: Proceedings of IEEE SmartGridComm (October 2010)
9. Fouque, P.A., Poupard, G., Stern, J.: Sharing decryption in the context of voting or lotteries. In: Proceedings of FC, pp. 90–104 (2001)
10. Garcia, F.D., Jacobs, B.: Privacy-friendly Energy-metering via Homomorphic Encryption. In: Proceedings of the STM (2010)
11. Hart, G.: Nonintrusive appliance load monitoring. Proceedings of the IEEE 80(12), 1870–1891 (1992)
12. Korolova, A., Kenthapadi, K., Mishra, N., Ntoulas, A.: Releasing Search Queries and Clicks Privately. In: Proceedings of WWW 2009 (2009)
13. Kotz, S., Kozubowski, T.J., Podgorski, K.: The Laplace distribution and generalizations. Birkhäuser, Basel (2001)
14. Lam, H., Fung, G., Lee, W.K.: A novel method to construct taxonomy electrical appliances based on load signatures. IEEE Transactions on Consumer Electronics 53(2), 653–660 (2007)
15. Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., Irwin, D.: Private memoirs of a smart meter. In: Proceedings of ACM Buildsys (2010)
16. Anderson, R., Fuloria, S., Alvarez, F., McGrath, K.: Key Management for Substations: Symmetric Keys, Public Keys or No Keys? In: IEEE PSCE (2011)
17. Rastogi, V., Nath, S.: Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption. In: Proceedings of the ACM SIGMOD (June 2010)
18. Rial, A., Danezis, G.: Privacy-Preserving Smart Metering. In: Technical Report, MSR-TR-2010-150. Microsoft Research (2010)
19. Richardson, I., Thomson, M., Infield, D., Clifford, C.: Domestic electricity use: A high-resolution energy demand model. Energy and Buildings 42, 1878–1887 (2010)
20. Shi, E., Chan, T., Rieffel, E., Chow, R., Song, D.: Privacy-Preserving Aggregation of Time-Series Data. In: Proceedings of NDSS (February 2011)

Anonymity Attacks on Mix Systems: A Formal Analysis

Sami Zhioua

Information and Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

Abstract. Information theory turned out to be very useful in analyzing anonymity attacks in general. The concept of channel information leak is a good indicator of how successful an attack can be. While different information leak measures exist in the literature, the problem of representing anonymity systems using noisy channels has not been well studied. The main goal of this paper is to show how anonymity attacks on mix systems can be formally represented as noisy channels in the information-theoretic sense. This formal representation provides a deeper understanding of mix systems and prepares the field for a more rigorous and accurate analysis of possible attacks. We performed empirical analysis using three information leak measures (mutual information, KLSD, and Min-entropy) which revealed interesting findings about some mix variants. This paper tries to bridge the gap between theory and practice in the field of anonymous communication systems.

1 Introduction

Cryptography alone is not enough to guarantee anonymity. Encrypting a message can help protect its content from being revealed to an undesired observer but the identities of the sender as well as the receiver remain generally known. Some anonymity techniques should be used in order to confuse an observer and conceal the communication relationship between the sender and the receiver. Since the first and most influential work on anonymity systems where Chaum introduced the concept of mix [1], several systems for anonymous communications have been proposed. These can be divided into two categories: high-latency and low-latency systems. High latency systems try to maximize anonymity at the cost of relatively large delays. These systems are more appropriate for anonymous remailers and include Mixmaster [2] and Mixminion [3]. Low latency systems try to anonymize real-time network communications like web browsing, instant chat, SSH communications, etc. Web MIXes [4] and Tor [5] fall in this category. Attacks on anonymous systems aim to reduce anonymity of users by linking senders to receivers, messages to senders and/or receivers, or messages with one another. They can be divided into two categories passive and active attacks. In this paper we consider only passive attacks. A passive attacker observes the outputs of the system and try to make deductions from these outputs.

Nearly all anonymity systems use randomness to conceal the link between messages, senders and receivers. While the use of randomness makes the task of the passive attacker more difficult, it also make the analysis of such systems more challenging. Information theory turns out to be very useful to analyze anonymity protocols [6,7,8,9].

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 133–147, 2011.

Typically, an anonymity protocol under a passive attack can be represented as a noisy channel where the secret information represent the channel's input and the attacker observations represent the channel's output. The channel is represented as a matrix of conditional probabilities of the form $Pr(\text{output}|\text{input})$. Since a passive attacker tries to make deductions about the secret information (input) based on what he observes (output), the concept of channel information leak is a good indicator of how successful the attack can be. In order to compute the information leakage and hence the degree of protection of the protocol, the corresponding channel's matrix has to be defined (set of secrets and set of observations) and computed (conditional probabilities). While several information leakage measures exist (e.g. mutual information [8], min-entropy [10], and KLSD [11]), the problem of how to compute the conditional probability matrix has not been deeply studied in the literature. Only few works [12,13,14,15,8] provided some details about computing the conditional probability matrix for a few number of protocols, mainly Crowds [16].

This paper gives a detailed account on how the conditional probabilities matrix can be generated for different types of mixes. We show how to define the set of secret information as well as the set of attacker observations and most importantly how to compute the channel's conditional probabilities values. Then, for every mix type we carry out empirical analysis to observe how the information leak behave as the parameters of the mix change. Section 2 describes the analogy between an anonymity protocol under passive attack and a noisy channel. Section 3 presents the three information leak measures. Then, Sections 4, 5, 6, and 7 detail the matrix generation an analysis for simple mixes, pool mixes, binomial mixes, and Stop-And-Go mixes respectively. Section 8 concludes.

2 Anonymity Protocol Representation through Noisy Channel

An anonymity protocol can be represented as a memoryless noisy channel where the input is the information to be kept secret and the output is the observed events. The attacker's challenge is then to guess the secret information based on the observed event. The set of observations depends on the capabilities of the attacker. A channel is a tuple $(A, O, p(\cdot|\cdot))$ where A is a random variable representing the inputs with n values $\{a_1, \ldots, a_n\}$, O is a random variable representing the outputs (observations) with m values $\{o_1, \ldots, o_m\}$, and $p(o|a)$ is a conditional probability of observing $o \in O$ given that $a \in A$ is the input. Intuitively, events in A represent the information to hide from a potential attacker while events in O are the ones that the attacker actually observes. The channel is noisy because an input might lead to different outputs with different probabilities. The probability values $p(o|a)$ for every input/output pair constitutes the channel matrix. Typically, the inputs are arranged by rows and the outputs by columns. The probability distribution $p(\cdot)$ over A is called the a priori distribution and is generally not known in advance. When an output o is observed, the probability that the input is a certain a is given by the a posteriori probability of a given o ($p(a|o)$). The a priori and the a posteriori probabilities are related by the Bayes theorem:

$$p(a|o) = \frac{p(o|a)\ p(a)}{p(o)}$$

3 Information Leakage Measures

A good anonymity protocol should make it hard to the attacker to guess the anonymous event given the observable event. The extreme case is when the distributions A and O are completely independent. This is called *noninterference* and achieving it, unfortunately, is often not possible because in most of the cases the protocol needs to reveal information about A. For example, in an election protocol, the individual votes should be secret but ultimately, the result of the votes must be made public which reveals information about individual votes. Hence the degree of anonymity of a protocol is tightly related to the amount of information leaked about the anonymous event when an observation is observed. In particular, more information leakage means less anonymity to the users of the system and vice versa. In the remaining of this section, $p(a)$ denotes the a priori distribution on the secret information.

3.1 Mutual Information

In Shannon information theory, the information leaked by a noisy channel is given by the notion of mutual information. Mutual Information of A and O, noted $I(A;O)$, represents the correlation of information between A and O and is defined as: $I(A;O) = H(A) - H(A|O)$ where $H(A)$ is the Shannon entropy of A and $H(A|O)$ is the conditional entropy of A given O:

$$H(A) = -\sum_{a \in A} p(a)\ \log(p(a)) \tag{1}$$

$$H(A|O) = \sum_{a \in A} \sum_{o \in O} p(a,o) \log(p(a|o)) \tag{2}$$

So mutual information is the difference between the uncertainty of the a priori distribution and the uncertainty of the a posteriori distribution.

3.2 Min-entropy

As an alternative to Shannon entropy, one can use the concept of probability of error of an adversary [15]. In an anonymity protocol, the attacker tries to guess the secret information based on the information he observes. His goal is to use a decision function so that to minimize the probability of error (probability of guessing wrong).

It is well known that the best decision function is based on the MAP rule [17] and the corresponding probability of error is called Bayes risk: $1 - \sum_{o \in O} \max_{a \in A}(p(o|a)\ p(a))$.

The probability of error is not a measure of information leakage. Instead, it can be used to measure the attacker's initial capability (based on the a priori distribution) and also the attacker capability after observing the output (based on the a posteriori distribution). A notion of "difference" between these probabilities of error can give rise to an information leakage measure. Smith [10] introduced an information leakage measure along this idea and used Rényi min-entropy:

$$\begin{aligned} \text{Min-entropy} &= H_\infty(A) - H_\infty(A|O) \\ &= \log \frac{1}{\max_{a \in A} p(a)} - \log \frac{1}{\sum_{o \in O} \max_{a \in A} p(o|a) p(a)} \end{aligned} \tag{3}$$

Smith showed through an interesting example that when an adversary tries to guess the value of the input in a single try, min-entropy information leak is more suitable than mutual information. The example features two systems with the same mutual information, the same a priori uncertainty, but with very different MAP probabilities of error.

3.3 KL Standard Deviation (KLSD)

Zhioua introduced a family of information leak measures based on how much the rows of the channel's matrix are different from each others [11]. Zhioua considers every row of the matrix as a point in the $m-$dimensional space and he interprets the scattering of these points as the degree of leakage of the channel. If the rows are different, then the associated points will be very scattered in the space and if they are similar they will be close to each others. The information leak notion formula is based on standard deviation statistical dispersion measure and the Kullback-Leibler divergence between probability distributions.

$$KLSD = \sqrt{\sum_{a \in A} p(a)\, D_{KL}(\overrightarrow{R_a} \,||\, \overrightarrow{Mean_p})^2} \tag{4}$$

where D_{KL} is the Kullback-Leibler divergence (a.k.a. relative entropy), $\overrightarrow{R_a}$ denotes the matrix row associated to input a, and $\overrightarrow{Mean_p}$ is the mean distribution with respect to the prior distribution p. $Mean_p(o) = \sum_a p(a)\, p(o|a)$.

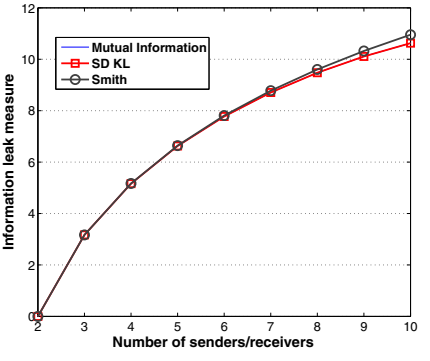
4 Simple Mix Systems

Since Chaum [1] introduced the concept of mixes, various mix designs have been proposed. The simplest ones are threshold and timed mixes. Threshold mix waits until a given number (threshold) of messages accumulates in the mix and then flushes them all. Timed mix flushes all the messages it contains every fixed period of time regardless of how many messages received since the last flushing. The operation of both mixes can be divided into rounds. A round is defined as the period between two flushings. In every round r, the mix receives a set of messages from a set of senders $S_r \subseteq \mathcal{S}$ and forwards them (after mixing) to a set of receivers $R_r \subseteq \mathcal{R}$ where $\mathcal{S}$ and $\mathcal{R}$ are the sets of all possible senders and receivers respectively. The primary role of a mix is to hide the link between senders and receivers. The focus of an attacker is then on unveiling sender-receiver linkability rather than the sender and/or receiver identity. Assuming that in round r the mix receives messages from t senders and flushes them to t receivers ($|S_r| = |R_r| = t$), the set of anonymous events can be the set of t-tuples of pairs (s_i, r_j) where $s_i \in S_r$ and $r_j \in R_r$ $\forall 1 \leq i, j \leq t$. The set of attacker observations can very well be the set of possible couples (S,R) where S is a set of senders and R is a set of receivers and such that $|S| = |R| = t$. Let $senders(a)$ and $receivers(a)$ be the set of senders, respectively the set of receivers, in the anonymous event a. For example, if $a = \{(s_2, r_3), (s_1, r_2), (s_3, r_1)\}$, then $senders(a) = \{s_1, s_2, s_3\}$. Let $senders(o)$ and $receivers(o)$ be the set of senders, respectively set of receivers, in observation o. The conditional probability $p(o|a)$ is defined as follows:

$$p(o|a) = \begin{cases} 1 & \text{if} \quad senders(o) = senders(a) \text{ and} \\ & \quad\ \ receivers(o) = receivers(a) \\ 0 & \text{otherwise} \end{cases}$$

	{1,2},{1,2}	{1,2},{1,3}	{1,2},{2,3}	{1,3},{1,2}	{1,3},{1,3}	{1,3},{2,3}	{2,3},{1,2}	{2,3},{1,3}	{2,3},{2,3}
{(1,1),(2,2)}	1	0	0	0	0	0	0	0	0
{(1,1),(2,3)}	0	1	0	0	0	0	0	0	0
{(1,2),(2,1)}	1	0	0	0	0	0	0	0	0
{(1,2),(2,3)}	0	0	1	0	0	0	0	0	0
{(1,3),(2,1)}	0	1	0	0	0	0	0	0	0
{(1,3),(2,2)}	0	0	1	0	0	0	0	0	0
{(1,1),(3,2)}	0	0	0	1	0	0	0	0	0
{(1,1),(3,3)}	0	0	0	0	1	0	0	0	0
{(1,2),(3,1)}	0	0	0	1	0	0	0	0	0
{(1,2),(3,3)}	0	0	0	0	0	1	0	0	0
{(1,3),(3,1)}	0	0	0	0	1	0	0	0	0
{(1,3),(3,2)}	0	0	0	0	0	1	0	0	0
{(2,1),(3,2)}	0	0	0	0	0	0	1	0	0
{(2,1),(3,3)}	0	0	0	0	0	0	0	1	0
{(2,2),(3,1)}	0	0	0	0	0	0	1	0	0
{(2,2),(3,3)}	0	0	0	0	0	0	0	0	1
{(2,3),(3,1)}	0	0	0	0	0	0	0	1	0
{(2,3),(3,2)}	0	0	0	0	0	0	0	0	1

(a)



(b)

Fig. 1. (a) The probabilities matrix of a threshold mix with threshold 2, 3 possible senders and 3 possible receivers. **(b)** Measuring the anonymity of a threshold mix system ($t = 2$) while increasing the number of senders/receivers from 2 to 10.

As example, Fig. 1(a) shows the conditional probabilities matrix of threshold mix with 3 senders and 3 receivers assuming the threshold t is 2[1]. Having the matrix hand, it is possible to apply the information leak measures of Section 3 to assess the anonymity of this simple mix system. The experiment we carried out consists in measuring the information leak of the system as we increase the number of senders/receivers. Intuitively, as the number of possible senders and receivers increases, the anonymity of the system should be improved. Interestingly, Fig. 1(b), which shows graphically the result of the experiment, reveals the contrary. By increasing the number of senders/receivers from 2 to 10 the information leak, according to all three measures (mutual information, SDKL and Min-entropy), increases as well. This means that the system gets less and less anonymous. The explanation is that the attacker's observation (the couple (S,R)) reveals more information about the secret events when the number of senders/receivers is higher. Consider for instance the case where there are only two senders and two receivers. Since the threshold t is 2, there is only one possible observation which is $(\{1,2\},\{1,2\})$. This observation will provide no clue to the attacker about the sender-receiver mapping because all the possible combinations are equally likely. This corresponds to a perfectly anonymous system with no information leak as depicted by the far left point of Fig. 1(b). Adding one sender and one receiver coincides exactly with the system in Fig. 1(a) with 9 possible observations and 18 possible secret events. By observing the sets of senders and receivers in a particular round, the attacker will be able to narrow down the set of potential secret events from 18 to only 2 which is considered as an important information leak. This is reflected by the three measures as the information leak increases from 0 to 3 in Fig. 1(b).

[1] A timed mix is represented in the same way. However, the number of messages t is not the same from one round to the other.

Finally, it is important to mention that this matrix representation is not compact because the number of secret events and observations grow exponentially as the number of senders and receivers increase. If the total number of possible senders $|\mathcal{S}|$ is n and the total number of possible receivers $|\mathcal{R}|$ is m, the number of secret events is $|A| = C_t^n P_t^m = \frac{n!}{t!(n-t)!} \frac{m!}{(m-t)!}$ while the number of observations is $|O| = C_t^n C_t^m = \frac{n!}{t!\,(n-t)!} \frac{m!}{t!\,(m-t)!}$.

5 Pool Mixes

Clearly, in threshold and timed mixes, every message will stay only one round in the mix. Pool mixes are improved versions of the above simple mixes. In a pool mix, a message can stay more than one round in the mix. A threshold pool mix has two parameters: a size of the pool $fmin$ and a threshold N. When $fmin + N$ messages accumulate in the mix, N randomly selected messages among them are flushed. The other $fmin$ messages are retained in the mix. They constitute the pool. A timed pool mix flushes periodically every t elapsed time units (generally seconds). If the number of messages is smaller or equal than $fmin$, no message is flushed. Otherwise, $fmin$ messages are retained in the pool and the others are flushed. Threshold-or-Timed mix and Threshold-and-Timed mix are two additional pool mixes obtained by combining the ideas of threshold and timed mixes. The former flushes every t seconds or when N messages accumulated in the mix while the latter flushes every t seconds but only when at least N messages have accumulated in the mix. Hence, in all these mix systems, a message can stay more than one round in the mix which improves the anonymity of the system but comes with a cost which is message delay.

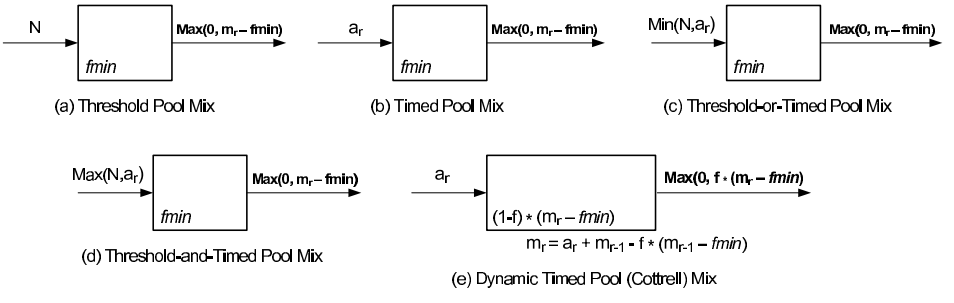


Fig. 2. The different pool mix types

The above pool mixes can be seen as constant pool mixes because the number of messages that stays in the mix after a flushing is constant ($fmin$). Dynamic pool mixes deviate from constant pool mixes in that the messages left after every flushing is not constant. Timed dynamic pool mix, called also Cottrell mix, has been relatively popular and was used in anonymity systems like Mixmaster [2] and Mixminion [3] protocols. It has three parameters: a period t, a minimum size of the pool $fmin$, and a fraction f. The mix flushes every t time units. However, instead of sending out exactly $m_r - fmin$ messages, where m_r is the total number of messages in the mix at round r, only a fraction of that quantity: $f * (m_r - fmin)$ is sent. Fig. 2 lists the different pool mix types

and indicates the number of messages entering, staying and leaving the mix in a given round r. Note that a_r refers to the number of messages entering the mix during the time interval t of a timed mix whereas m_r refers to the total number of messages accumulated in the mix at round r before flushing.

It is easy to see that a message can stay several rounds in a pool mix without being flushed. Hence, if the attacker tries to guess the sender-receiver linkability in a given round r, the conditional probabilities matrix can be constructed in a similar way as for a simple mix system (Section 4). However, the set of possible senders will not only contain the senders of messages received in round r but the senders of messages received in all previous rounds $r-1, r-2, \ldots, 2, 1$. Indeed, theoretically any of those senders can be the originator of a message flushed in round r. This will make the conditional probabilities matrix much bigger. However, it is possible to keep the same size as a simple threshold or timed mix but the analysis will be less accurate.

In another scenario, assume that an attacker targets a message flushed by the mix in round r and has an objective to guess the round in which that message entered the mix. By doing so, the attacker will narrow down the identity of the sender of that message. Recall that it is assumed that the attacker is a permanent global passive observer. He can observe the number of messages that arrive to the mix in every round (noted a_r) and the number of messages sent by the mix in every round. In addition, he knows the whole history and the parameters ($fmin$ and f) of the mix. The next paragraphs illustrate how the matrix of conditional probabilities matrix can be constructed given this particular scenario[2]. In this scenario, the set of secret events is the set of rounds from 1 to r: $\{1, 2, \ldots, r\}$[3]. The set of observables is less obvious. From all the observations on which the attacker has access, the numbers of messages entering the mix in each round give some hint about the actual secret event. Therefore we choose the set of observables to be the set of possible tuples of the form $(a_1, a_2, \ldots, a_r)$ where a_i denotes the number of messages arriving to the mix in round i. For example, an observation might be: $o = (15, 6, 36, 29, 10)$ which means 15 messages entered the mix in round 1, 6 messages in round 2, etc. To avoid confusion between different observations, we use the following convention: $o_1 = (a_1^1, a_2^1, \ldots, a_r^1)$, $o_2 = (a_1^2, a_2^2, \ldots, a_r^2)$, etc. More generally, $o_j = (a_1^j, a_2^j, \ldots, a_r^j)$.

The next step is to compute the conditional probabilities of the channel matrix. The computations are inspired by the work of Diaz and Preneel [18]. The matrix is composed of conditional probabilities of the form $p(o|i)$ where i refers to a round and $o = (a_1, a_2, \ldots, a_r)$ refers to an observation. However, in this particular setting, it is easier to compute first the a posteriori probabilities: $p(i|o)$. Indeed, $p(i|o)$ is the probability that the target message (flushed in round r) has entered the mix in round i given the observation o. If the message arrived in round r, it is certain that it is in the mix at flushing time. So,

$$p(r|o_j) = a_r^j * \frac{1}{m_r^j} \tag{5}$$

[2] The illustration is generic and can apply to any pool mix type in Fig. 2.

[3] In the rest of this section, we use i to denote some secret event.

where m_r^j is the total number of messages in the mix at round r given observation o_j. m_r^j is the only unknown in Equation (5) and round r and it is equal to the number of messages kept in the pool from the previous round plus the number of messages arrived in round j, that is, a_r^j. Consequently, it is defined recursively as follows:

$$m_r^j = \begin{cases} a_r^j + m_{r-1}^j - flushed(r-1) & \text{if} \quad r > 1 \\ a_r^j & \text{if} \quad r = 1. \end{cases} \tag{6}$$

where $flushed(i)$ refers to the number of messages flushed in round i. More formally,

$$flushed(i) = \begin{cases} \max(0, m_i - fmin) & \text{for a threshold pool mix} \\ \max(0, m_i - fmin) & \text{for a timed pool mix} \\ \max(0, m_i - fmin) & \text{for a threshold-or-timed pool mix} \\ \max(0, m_i - fmin) & \text{for a threshold-and-timed pool mix} \\ \max(0, f * (m_i - fmin)) & \text{for a dynamic timed pool mix} \end{cases}$$

If the target message arrived to the mix in round $r-1$, it might have already been flushed by the mix in the same round $r-1$. The probability that the message stayed in the mix in round $r-1$ is:

$$\frac{m_{r-1}^j - flushed(r-1)}{m_{r-1}^j}.$$

Hence, the probability that the target message arrived in round $r-1$ given observation o_j is:

$$p(r-1|o_j) = a_{r-1}^j * \frac{m_{r-1}^j - flushed(r-1)}{m_{r-1}^j} * \frac{1}{m_r^j}$$

More generally, the probability that the target message arrived in round i given observation o_j is:

$$p(i|o_j) = a_i^j * \prod_{k=i}^{r-1} \frac{m_k^j - flushed(k)}{m_k^j} * \frac{1}{m_r^j} \tag{7}$$

Bayes theorem is then used to turn the conditional a posteriori probabilities of Equation (7) into memoryless channel's conditional probabilities $p(o_j|i)$:

$$p(o_j|i) = \frac{p(i|o_j)\, p(o_j)}{p(i)}$$

where we assume that all observations might happen with equal probabilities, that is,

$$p(o_j) = p(o_k)\ \forall j,k$$

and hence,

$$p(i) = \sum_j p(i|o_j) p(o_j).$$

As an example, the cottrell mix with parameters $f = 0.8$, $fmin = 15$, and number of rounds 3 yields the channel matrix[4]:

[4] For simplicity of illustration, we assumed that the numbers of messages entering the mix in every round is either 10 or 20. Obviously the analysis can be generalized to any number of messages.

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8
1	0.146	0.105	0.101	0.074	0.187	0.138	0.140	0.105
2	0.130	0.094	0.182	0.134	0.105	0.077	0.157	0.118
3	0.108	0.156	0.100	0.148	0.100	0.148	0.094	0.141

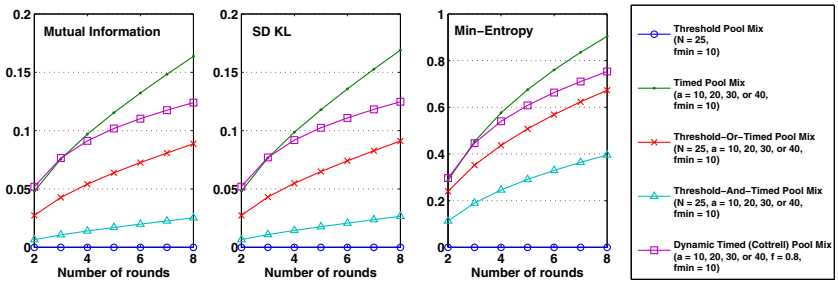


Fig. 3. Information leak of the five pool mix types as the number of rounds increases

For all pool mix types, the anonymity of the system depends on the number of rounds realized. Intuitively, the more rounds are completed, the more difficult for the attacker to guess the secret information (the round on which the message entered the mix). Surprisingly, the analysis we performed showed the opposite. The experiment is simple and consists in measuring the information leakage of all pool mix types for an increasing number of rounds (from 2 to 8 rounds). Fig. 3 shows that according to all information leakage measures (mutual information, SDKL, and Min-entropy), all pool mixes in our setting leak more as the number of rounds (x-axis) increases. This means that, when an attacker tries to guess the round a target message entered the mix and when the same attacker knows the number of messages entering the mix in every round, then the more rounds are performed the more chances he has to guess the correct round. The only exception in Fig. 3 is the threshold pool mix which is constant at 0 as the number of rounds grows. An information leak of 0 means the attacker has no clue about the secret event given what he observed. Indeed, since the number of messages entering the mix in every round is constant (N messages), there is only one observation the attacker can observe which is a tuple of the form $(N,N,\ldots,N)$. The same explanation can help to understand the fact that threshold-or-timed and threshold-and-timed pool mixes leak less information than timed and cottrell pool mixes in Fig. 3. In particular, for threshold-or-timed and threshold-and-timed pool mixes, the number of messages entering the mix in some rounds is exactly N. This limits the information the attacker can utilize to guess the correct secret event. Note also that in the initial rounds (2 and 3) timed and Cottrell pool mixes have similar information leakage values. Then, as more rounds are completed, Cottrell pool mix manifests less information leakage. It is important to mention finally that the "ranking" suggested by Fig. 3 holds only in the described setting. That is, if the objective is to protect the anonymity system from an

attacker who can only track the number of messages entering the mix in every round, then the best pool mix type would be a simple threshold pool mix. This holds for any order of message entering. However, in presence of attackers with different capabilities (e.g. $n-1$ attacks [19]), most probably the Cottrell pool mix will be the best option.

In the next experiment, we focused on the Cottrell mix. In particular, the goal of the experiment is to analyze the impact of f and $fmin$ parameters on the information leakage of Cottrell mix. In this analysis, the number of rounds is fixed to 6 and the number of messages entering the mix in every round is assumed to be 5, 10, 15, or 20. Fig. 4 shows a summary of this analysis. Every 3D diagram is obtained using every one of the three information leak measures (mutual information, SDKL, and Min-Entropy). In the x-axis, the threshold $fmin$ grows from 0 to 60. In the y-axis, the fraction f decreases from 1 to 0. According to all measures, information leakage is maximized when $f = 0.98$ and $fmin = 0$. Note that, with those values Cottrell mix is almost equivalent to a simple timed mix where no messages are kept in the mix from round to round. So the attacker knows with certainty that the target message flushed in round r arrived in the same round r. On the other hand, Cottrell mix in the described setting preserves best the identity of the target message's sender when $f = 0.78$ and $fmin = 4$ for mutual information and SDKL and when $f = 0.42$ and $fmin = 8$ for min-entropy. At a first glance, the result looks counter intuitive. Indeed, one would expect the mix to be the most anonymous when all the messages received by the mix stayed until round r. This happens when $fmin$ is very large and the fraction f is 0. However, since the number of messages arriving to the mix might be different from round to round, the attacker can use this information to narrow down the identity of the round where the target message arrived to the mix. For example, if 20 messages entered the mix in round 1, 5 in rounds 2, 3, 4, 5, and 6, and all the messages stayed in the mix until round 6, the attacker having observed these numbers, will suspect round 1 with probability $\frac{4}{9}$, and everyone of rounds 2, 3, 4, 5, and 6 with probability $\frac{1}{9}$. This is clearly not the most secure situation. The best situation in that setting would be, for instance, to end-up with 10 messages lasting from every round. The attacker will suspect the six rounds with equal probability $\frac{1}{6}$.

6 Binomial Mix

All mix types described so far fit into the generalized framework of Diaz and Serjantov [20]. In that framework, a mix is characterized by a function $P(m)$ from the number of messages in the mix m to the fraction of messages to be flushed. For instance, simple threshold and timed pool mixes can be represented by a function $P(m) = 1$ since all messages are flushed in every round. For pool mixes, the function is simply:

$$P(m_r) = flushed(r)/m_r$$

where m_r is the number of messages in the mix at round r. This framework is more precisely called generalized deterministic mixes because the number of flushed messages is exactly determined by the values of n and the function $P(n)$. Generalized binomial mixes [20], on the other hand, are expressed in terms of the same function $P(n)$ but this function is not considered as a fraction but as a probability. That is, for every message

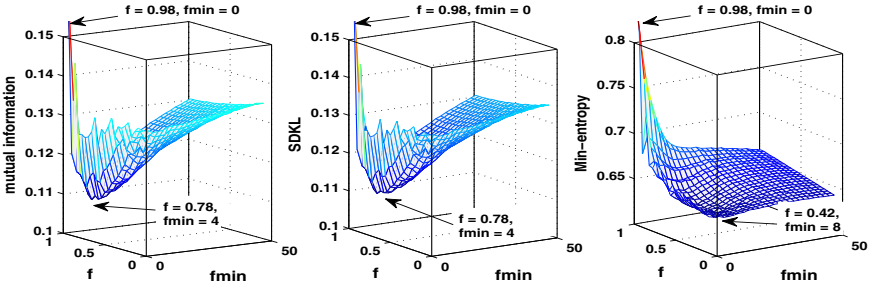


Fig. 4. Information leakage of Cottrell mix measured using mutual information, SDKL, and Min-Entropy for different values of f (x-axis) and $fmin$ (y-axis)

in the mix, a biased coin is tossed and depending on the result, the message is kept in the mix or flushed.

As in the previous section, assume that the attacker is a global passive observer that tries to guess the round in which a message flushed in round r entered the mix. The set of secret events is the same, that is, the number of rounds from 1 to r. Since the number of messages flushed in every round is not deterministic, that number is useful for the attacker to make a more accurate guess. Hence, the set of observables of the attacker is the set of all possible r tuples of the form $((a_1,s_1),(a_2,r_2),\ldots,(a_r,s_r))$ where a_i and s_i refer to the number of messages received and flushed at round i respectively. Similarly to the computation of the previous section, it is easier to compute first the a posteriori probabilities: $p(i|o)$ where i refers to a round (secret event) and $o = ((a_1,s_1),(a_2,r_2),\ldots,(a_r,s_r))$ refers to an observation. If the message arrived in round r, it is certain that it is in the mix at flushing time. So,

$$p(r|o) = \frac{a_r * P(\overline{m_r})}{\overline{m_r} * P(\overline{m_r})} = a_r * \frac{1}{\overline{m_r}} \tag{8}$$

where $\overline{m_r}$ is the average number of messages in the mix at round r given s_r messages are flushed in that round. According to Diaz and Serjantov [20],

$$\overline{m_r} = \sum_{i=s_r}^{M_{max}} i * p(i|s_r) \tag{9}$$

$$= \sum_{i=s_r}^{M_{max}} i * \frac{p(s_r|i)}{\sum_{j=s_r}^{M_{max}} p(j|i)} \tag{10}$$

$$= \sum_{i=s_r}^{M_{max}} i * \frac{\frac{i!}{s_r!\,(i-s_r)!}\,P(i)^{s_r}\,(1-P(i))^{i-s_r}}{\sum_{j=s_r}^{M_{max}} \frac{i!}{j!\,(i-j)!}\,P(i)^{j}\,(1-P(i))^{i-j}} \tag{11}$$

where M_{max} is the maximum capacity of the mix and $P(i)$ is the probability of flushing given i messages are in the mix.

If the target message arrived to the mix in round $r-1$, it might have already been flushed by the mix in the same round $r-1$. The probability that the message stayed in the mix in round $r-1$ is simply $1-P(\overline{m_{r-1}})$ and hence the probability that the target message arrived in round $r-1$ given observation o is:

$$p(r-1|o) = a_{r-1}\,(1-P(\overline{m_{r-1}}))\,\frac{1}{P(\overline{m_{r-1}})}$$

Generalizing that probability to a previous round i gives:

$$p(i|o) = a_i \prod_{k=i}^{r-1}(1-P(\overline{m_{r-1}}))\,\frac{1}{\overline{m_r}} \tag{12}$$

Finally, the conditional a posteriori probabilities are turned into memoryless channel's conditional probability using the bayes rule as in the previous section.

7 Stop-and-Go Mix

All mix systems described so far operate in rounds. Every round ends by the flushing of a group of messages collectively. Stop-And-Go-Mix (SG-Mix) [21] on the other hand processes every message individually. In particular, every message is delayed for a different amount of time while it passes through the mix. A sender of a message starts by selecting a sequence of nodes through a network of SG-Mixes. For every node i of the path the sender sets a waiting time t_i sampled from an exponential distribution with parameter μ. This information is embedded in the message and encrypted. At reception of the message, a node (SG-Mix) i extracts t_i information and makes the message wait t_i units of time before forwarding it to the next node $i+1$ and so on. In order to protect the system from $n-1$ attacks[5], the sender determines for every node i a time interval $[TS^{min}, TS^{max}]_i$ where the message is expected to arrive. If the arriving time of the message to node i is outside the interval (earlier than TS^{min} or later than TS^{min}), the message is discarded.

The functioning of an SG-Mix can be seen as an alternation of idle and busy periods. In an idle period, the SG-Mix is empty and no message arrives. With the arrival of a message, the busy period starts and will last until the SG-Mix is empty again. In the busy period, the SG-Mix receives a sequence of messages individually, delays them according to a duration t sampled from the exponential distribution with rate μ and then flushes every message as soon as its delay time elapses. As for simple mixes (Section 4), a passive attacker tries to correlate input messages to output messages to unveil who is communicating with whom. In this regard, the arrival-departure order of messages to and from the SG-Mix is very helpful for the attacker to narrow down the sender-receiver correlation. For instance, if the attacker observes the scenario $(s_2, s_4, r_1, s_1, r_4, s_3, r_3, r_2)$ where s_i refers to the arrival of a message from sender i and r_j refers to the departure of a message to receiver j, then he can deduce that sender 1 is not communicating

[5] $n-1$ attack consists in shaping the traffic to isolate a target message. For instance, blocking a target message until the mix is empty and then forwards it with a set of identifiable dummy messages.

with receiver 1, sender 3 is not communicating with receiver 4, etc. Assuming that in a busy period, the SG-Mix receives in total n messages from n senders $S = \{s_1, s_2, \ldots s_n\}$ and forwards them to n receivers $R = \{r_1, r_2, \ldots, r_n\}$, the set of secret events is defined as the set of all possible n-tuples of pairs (s_i, r_j) where $s_i \in S$ and $r_j \in R$ for $1 \leq i, j \leq n$. The set of observations is a subset of permutations[6] of the set $S \cup R$. Let S and R be two sets of size n ($|S| = |R| = n$). A permutation of the elements of $S \cup R$ is a bijection from $\{1, 2, \ldots, 2n\}$ to $S \cup R$. Let p be a permutation of $S \cup R$ and let Q be a subset of $\{1, 2, \ldots, 2n\}$. $p_{|Q}$ refers to the restriction of p to Q. For instance, if $p = (s_2, s_4, r_1, s_1, r_4, s_3, r_3, r_2)$ then $p_{|\{1,2,\ldots,5\}} = (s_2, s_4, r_1, s_1, r_4)$. Let $\sharp_S(p)$ and $\sharp_R(p)$ denote the number of elements of permutation p whose image is respectively in the set S and R. More formally, $\sharp_S(p) = |img(p) \cap S|$ and $\sharp_R(p) = |img(p) \cap R|$. The set of observations is the set of permutations satisfying the following condition:

$$\left\{ p : \{1, \ldots, 2n\} \mapsto S \cup R \mid \forall i, 1 \leq i < 2n,\ \sharp_S(|p_{\{1\ldots i\}}) > \sharp_R(p_{|\{1\ldots i\}}) \right\} \tag{13}$$

Intuitively, Equation 13 ensures that for a permutation to be a valid observation, at any moment except for the last step ($i = 2n$), the number of flushed messages should not equate or exceed the number of arrived messages. Indeed, the mix cannot flush more messages than what arrived and equality means that the SG-Mix is empty and consequently the busy period is over.

Let $senders(a), receivers(a), senders(o)$, and $receivers(o)$ defined as in Section 4. Let $position(o, s)$ be the position of sender s in observation o. For instance, if $o = (s_2, s_4, r_1, s_1, r_4, s_3, r_3, r_2)$ then $position(o, s_1) = 4$. The conditional probability $p(o|a)$ is defined as:

$$p(o|a) = \begin{cases} 1 & \text{if } \begin{array}{l} senders(o) = senders(a) \text{ and} \\ receivers(o) = receivers(a) \text{ and} \\ \forall (s,r) \in a, position(o,s) < position(o,r) \end{array} \\ 0 & \text{otherwise} \end{cases}$$

Using this formulation, the number of secret events is $|A| = n!$ and the number of observations is $|O| = \frac{(2n)!}{2(n-1)}$.

The above formulation of SG-Mix uses the same set of secret events as the simple mix formulation in Section 4. The sets of observations however are different. In particular, observations in the SG-Mix formulation are more informative than the observations in the simple mix formulation. This gives an attacker of the SG-Mix more information to guess the correct sender-receiver correlation. The common set of secret events in the SG-Mix as well as the simple mix makes it possible to compare their information leakage. Figure 5 shows the information leakage of simple threshold mix as well as Stop-And-Go mix as the number of senders and receivers gets larger.

In the above formulations, the goal of the attacker is to guess the correlation between senders and receivers. In this regard, Figure 5 shows that simple threshold mix and SG-Mix have very similar information leak values. This means that if the goal of the protocol is protect users from an passive attacker whose sole purpose is to correlate senders to receivers, both mix systems are equivalent. However, if the attacker is able to

[6] The order of elements is taken into consideration.

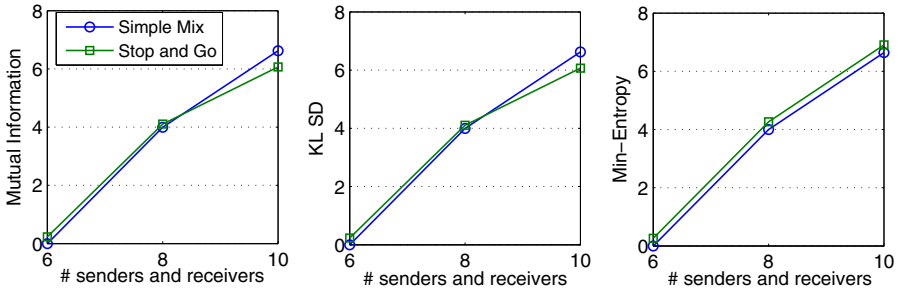


Fig. 5. Comparison of the information leakage of simple threshold mix with Stop-And-Go mix based on three measures

mount $n-1$ attacks, SG-Mix would be more secure because it can detect the blocking of messages thanks to the time window associated with every message. On the other hand, if messages delay time is an important element to consider, SG-Mix would again be preferred to the simple threshold mix.

8 Conclusion

Anonymity protocols in presence of a passive attacker can very well be represented using a noisy channel. This formal representation opens the way for a more rigorous and more accurate analysis of anonymous communication systems in general. In this paper, we applied this approach on several types of mixes. In particular, we illustrated how to define the set of secret information as well as the set of attacker observations and most importantly how to compute the channel's conditional probabilities values. Then, using three information leak measures (mutual information, min-entropy, and KLSD) we analyzed the anonymity provided by these mixes. Note, however, that other representations of mix systems are possible using noisy channels and these might lead other insights and other findings. For instance, Newman et al. [13] represented timed mix and focused on the sender anonymity of a target malicious sender. In [8], Zhu and Bettati used *MI* to measure several mix systems.

Our plans for future work include the study of the scalability of this approach since the channel matrix size for some anonymity systems might be very large. A possible alternative could be to compute only an estimation of information leak measures and provide tight guarantees on those values [9]. The long term objective of this research is to bridge the gap between theoretical results in the field of anonymous communications and deployed anonymity protocols. This paper prepares the field for the analysis of deployed systems, in particular Tor.

Acknowledgement. This research is funded by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum and Minerals (KFUPM) under Junior Faculty Grant JF100007.

References

1. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24(2), 84–90 (1981)
2. Möller, U., Cottrell, L., Palfrader, P., Sassaman, L.: Mixmaster Protocol — Version 2. IETF Internet Draft (July 2003)
3. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp. 2–15 (May 2003)
4. Berthold, O., Federrath, H., Köpsell, S.: Web MIXes: A System for Anonymous and Unobservable Internet Access. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 115–129. Springer, Heidelberg (2001)
5. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: Proceedings of the 13th Usenix Security Symposium (August 2004)
6. Diaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Dingledine, R., Syverson, P. (eds.) PET 2002. LNCS, vol. 2482, pp. 54–68. Springer, Heidelberg (2003)
7. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Anonymity protocols as noisy channels. Information and Computation 206(2-4), 378–401 (2008)
8. Zhu, Y., Bettati, R.: Anonymity vs. information leakage in anonymity systems. In: Proceedings of ICDCS 2005, Columbus, Ohio, pp. 514–524 (2005)
9. Chatzikokolakis, K., Chothia, T., Guha, A.: Statistical measurement of information leakage. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 390–404. Springer, Heidelberg (2010)
10. Smith, G.: On the foundations of quantitative information flow. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 288–302. Springer, Heidelberg (2009)
11. Zhioua, S.: A new information leakage measure for anonymity protocols. In: Jajodia, S., Zhou, J. (eds.) SecureComm 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 50, pp. 398–414. Springer, Heidelberg (2010)
12. Chatzikokolakis, K.: Probabilistic and Information-Theoretic Approaches to Anonymity. PhD thesis, Laboratoire d'Informatique (LIX), École Polytechnique, Paris (October 2007)
13. Newman, R.E., Nalla, V.R., Moskowitz, I.S.: Anonymity and covert channels in simple timed mix-firewalls. In: Martin, D., Serjantov, A. (eds.) PET 2004. LNCS, vol. 3424, pp. 1–16. Springer, Heidelberg (2005)
14. Chen, H., Malacaria, P.: Quantifying maximal loss of anonymity in protocols. In: Proceedings of ASIACCS 2009, pp. 206–217. ACM, New York (2009)
15. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: On the bayes risk in information-hiding protocols. Journal of Computer Security 16(5), 531–571 (2008)
16. Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security 1(1), 66–92 (1998)
17. DeGroot, M.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
18. Díaz, C., Preneel, B.: Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 535–543. Springer, Heidelberg (2004)
19. Serjantov, A., Dingledine, R., Syverson, P.: From a trickle to a flood: Active attacks on several mix types. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 36–52. Springer, Heidelberg (2003)
20. Díaz, C., Serjantov, A.: Generalising mixes. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 18–31. Springer, Heidelberg (2003)
21. Kesdogan, D., Egner, J., Büschkes, R.: Stop-and-go-mIXes providing probabilistic anonymity in an open system. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 83–98. Springer, Heidelberg (1998)

Differentially Private Billing with Rebates

George Danezis[1], Markulf Kohlweiss[1], and Alfredo Rial[2]

[1] Microsoft Research, Cambridge, UK
{gdane,markulf}@microsoft.com
[2] ESAT-COSIC / IBBT, KU Leuven, Belgium
alfredo.rial@esat.kuleuven.be

Abstract. A number of established and novel business models are based on fine grained billing, including pay-per-view, mobile messaging, voice calls, pay-as-you-drive insurance, smart metering for utility provision, private computing clouds and hosted services. These models apply fine-grained tariffs dependent on time-of-use or place of-use to readings to compute a bill.

We extend previously proposed billing protocols to strengthen their privacy in two key ways. First, we study the monetary amount a customer should add to their bill in order to provably hide their activities, within the differential privacy framework. Second, we propose a cryptographic protocol for oblivious billing that ensures any additional expenditure, aimed at protecting privacy, can be tracked and reclaimed in the future, thus minimising its cost. Our proposals can be used together or separately and are backed by provable guarantees of security.

1 Introduction

A number of business models are based on billing customers for fine grained use of a system or resource: mobile network providers charge per call length and type, pay-per-view TV providers charge for the actual requested content. Newer businesses rely heavily on fine grained recordings of activity for billing. Pay-as-you-drive automotive insurance bills drivers per mile depending on the type of road and time of travel. Electronic tolling and congestion charging schemes have been proposed on similar lines. Smart-metering for electricity and gas is being rolled out in the EU and the US in the next few years. Finally, private cloud provision as well as hosted on-line service provision might rely on fine-grained measurements of CPU time usage, memory allocation, disk storage, peak bandwidth, or even the demand and network congestion at the time of day.

The downside of fine-grained metering and billing is the potential threat to privacy [1,2]. A common privacy-invasive architecture to support such billing consists of providers collecting all usage information in order to apply the appropriate tariffs. Privacy-friendly protocols have also been developed: it is possible to cryptographically combine certified readings with a tariff policy to produce a certified bill that leaks no additional information about the detailed readings [3,4,5]. Yet, even the final bill, which is for instance aggregated over a period of usage, may leak information or be used to leak specific readings.

This work makes two contributions to the field of privacy-friendly metering and billing. First, we discuss how to eliminate incidental, accidental or deliberate leakages

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 148–162, 2011.

of information resulting from disclosing the final bill. We show that by adding some, in the long run small, amount of noise it is possible to offer strong privacy guarantees on what an adversary can infer from the final bill. This problem is similar to covert channel minimization [6], and we use techniques from differential privacy that could be more widely applicable. Second, we attempt to minimise the cost of privacy through a cryptographic oblivious billing mechanism. The true cost of service provision is tracked across billing periods, but not revealed to the service provider, which can only verify the deposited funds cover costs. This allows customers to determine the levels of privacy they require and even get a rebate for the additional funds they used to protect their privacy.

Throughout this work we motivate our protocols through the example of a leased private computation cloud. A service provider installs a cloud of 10000 CPUs in the premisses of a large government intelligence agency. In our example, billing is performed on the basis of the compute hours actually used at a fixed rate of $0.12 per CPU instance / hour[1]. A more complex tariff scheme where each hour in the year is costed differently is also supported. The government agency needs to settle the bill each month, but is worried that the amount of computation on particular days is leaked to its adversaries. We will show how our protocols can be used to reduce any leakage below a desired level.

Discussion of the state-of-the-art. Deployed systems for fine grained billing usually employ procedural access control mechanisms to protect privacy: usage data is gathered, and often stored centrally for the purposes of billing. Access control allows only designated parties and processes to access the data, and encryption technology might be used to protect storage and communications. Despite those protections, the fact that personal information is under the control of a service provider raises privacy concerns. A pilot deployment of a pay-as-you-drive insurance scheme by Norwich Union failed, stating privacy concerns as a leading reason for low uptake.[2]

Two types of privacy preserving metering and billing have been proposed in the literature. First, a meter can be entrusted with applying a fine grained tariff to the usage data and only communicating to the service provider a final total fee. In this setting the meter has to be trusted by the users and the service providers both for privacy and correctness. This is usually achieved through trusted hardware and certification. In the automotive setting, where meters record positions of cars for tolling, spot checks have also been proposed to verify the correctness of the meter operation [3]. The second architecture requires meters to cryptographically certify readings and securely hand them over to a user device or service. Cryptographic operations can then be used to apply a tariff scheme, and output a bill along with the necessary cryptographic proofs that certify its correctness. Meters are simpler, and any device can be used to compute bills [5]. Both architectures achieve the same goal: the bill and other necessary information are made available to the service provider, but further information on detailed readings is hidden from it and only available to the consumer.

[1] The value of a standard compute instance / hour on Amazon EC2 and Microsoft Azure in December 2010.

[2] Insurer stops ‘pay as you drive’, BBC Radio 4’s Money Box `http://news.bbc.co.uk/2/hi/programmes/moneybox/7453546.stm`

In this work we are concerned with the remaining information leakage from privacy-preserving billing systems. The value revealed by the protocols, namely the value of the bill, could leak information or be used as a covert channel.

To illustrate the threat, consider a resource consumed in a number of i_{max} distinct time periods i, for $i \in [0, i_{max}]$. Some consumption takes place at each time period i denoted by $c_i \in [0, c_{max}]$, that should be billed at a tariff of p_i per unit. Thus the final bill for all periods should be $B = \sum_{i=0}^{i_{max}} c_i \cdot p_i$. Without making any assumptions on the consumption patterns, as they are out of the system designer's control, it is difficult to estimate what information may be leaking from the final value B. For example an adversary may know, through some side information, that the user consumed only in a single time period T. In such a case the exact value of c_T can be inferred straightforwardly by computing $c_T = B/p_T$. This example threat illustrates that a solution to this problem should make no assumptions about the consumption pattern, assume that arbitrary side-information is available to the adversary, and work for arbitrary (but known) tariff schemes.

We will use a trivial solution as a benchmark to evaluate our own proposals: the user could always pay an amount equivalent to the maximum possible consumption. In the example used so far, this would be: $\max B = c_{max} \cdot \sum_{i=0}^{i_{max}} p_i$. While this is an adequate solution from a privacy perspective, it nullifies the benefits of fine-grained billing as users end up paying a fixed premium irrespective of their consumption. Furthermore it is very wasteful, if the objective is to hide usage of the private cluster at the granularity of an hour or a day.

Outline. Our techniques provide guarantees of privacy depending on the level of protection required by the customers, as well as a cryptographic scheme to amortise the cost of such privacy provision. In Section 2 we study how much noise one needs to add to a bill to ensure specific consumption windows are protected. In Section 3 we propose a cryptographic rebate protocol that keeps a hidden track of the actual amounts due accross multiple billing periods, allowing users to reclaim some of the extra payments made. The rebate protocols also support deposits, anonymous payments using e-cash, and negative bill noise, and prevent abuse by ensuring the funds paid cover the costs of consumption.

2 Differential Privacy for Billing

We start from the premise that customers can add some "noise" to their bill in order to hide their exact usage at specific times. Of course this billing noise represents real money, so they wish to minimise it for a given level of protection required. The first problem we tackle is to determine how much more a customer should pay to hide their pattern of activity for a particular time frame.

Differential privacy was developed as a framework for hiding personal records within databases [7]. A statistic extracted from a database is differentially private if it is nearly as likely as if it was extracted from a database with an arbitrary record removed. This definition encapsulates the intuition that a single individual's record does not overwhelmingly affect the statistic in a way that information about the record might leak.

We have to modify this definition as well as its precise mathematical counterpart to make it applicable to the billing setting. We consider as our database the set of all readings from a meter. In the case of billing private cloud usage each record represents the number of CPUs used for each hour of the billing period. The customer then has to specify its privacy goal: for example they may wish to hide their activity at any arbitrary hour or any arbitrary day of computing. Then they should determine the quality of the protection provided, in terms of how much information the bill reveals about any particular period. Using those parameters we can calculate the additional amount to bill in order to achieve the desired privacy goals.

2.1 Privacy Definitions

For simplicity we consider fixed size databases corresponding to a fixed term billing period. For our application this is sufficient, as we are primarily interested in the number of CPU instances used during each hour of the pricing period. For this reason the domain of all possible data sets is described as the Cartesian product: $\mathcal{D} = \{0, \ldots, c_{max}\}^{i_{max}}$. For our private cloud scenario c_{max} is the number of instances in the private cloud, and i_{max} is the number of records per billing period. In our concrete example $c_{max} = 10000$ and i_{max} is the number of hours in a month or a year.

First we define the "distance" between two sets of readings, and repeat some key definitions and results from differential privacy [7], upon which we will be building.

Definition 1. *The record distance $RDist(D_1, D_2)$ between two data sets $D_1, D_2 \in \mathcal{D}$ corresponds to the number of elements (records) in which D_1 and D_2 differ.*

Definition 2. *A randomized function K gives ϵ-differential privacy if for all data sets $D_1, D_2 \in \mathcal{D}$ with $RDist(D_1, D_2) \leq 1$, and all $S \in \Sigma_{Image(K)}$,*[3]

$$Pr[K(D) \in S | D = D_1] \leq \exp(\epsilon) \times Pr[K(D) \in S | D = D_2] \ .$$

The probability is taken over the randomness of K.

Intuitively, mechanisms fulfilling this definition address concerns that an individual might have about filling in one record truthfully, rather than arbitrarily. Differential privacy guarantees that no output (and thus consequences of outputs) becomes significantly more or less likely. In our case the randomized function K will be the billing amount increased by some random value.

A further observation about hiding multiple records k from a database will also prove useful:

Definition 3. *A randomized function K gives (k, ϵ)-differential privacy if for all data sets $D_1, D_2 \in \mathcal{D}$ with $RDist(D_1, D_2) \leq k$, and all $S \in \Sigma_{Image(K)}$,*

$$Pr[K(D) \in S | D = D_1] \leq \exp(\epsilon \cdot k) \times Pr[K(D) \in S | D = D_2] \ .$$

The probability is taken over the randomness of K.

[3] A σ-algebra over a set X is a set $\Sigma_X \subset 2^X$ such that $\emptyset \in \Sigma_X$; $S \in \Sigma_X \Rightarrow (X \setminus S) \in \Sigma_X$; and for any $(S_i)_{i \in \mathbb{N}}$, $S_i \in \Sigma_X$, $\bigcap S_i \in \Sigma_X$.

Lemma 1. *A ϵ-differentially private privacy mechanism K is also (k, ϵ)-differentially private.*

Lemma 1 follows from Definition 3, and shows that the same privacy mechanism K can obstruct inferences on multiple records. In such cases it provides a lower amount of privacy (i.e. $\epsilon' = \epsilon \cdot k$). Hence if a mechanism is to be used to protect multiple records suitable security margins should be provided.

Differentially private mechanisms. The classical differential privacy mechanism by Dwork [7] adds Laplacian noise to the outcome of a query, parametrised by the "sensitivity" of the function f.

Definition 4. *The sensitivity of a function $f : \mathcal{D} \rightarrow R^n$ is the maximum distance between output values for which the domain differs in at most one record:*

$$\Delta_f = \max_{\substack{D_1, D_2 \in \mathcal{D} \\ RDist(D_1, D_2) \leq 1}} \|f(D_1) - f(D_2)\|_1$$

For $n = 1$ the sensitivity of f is the maximum difference $|f(D_1) - f(D_2)|$ between pairs of databases D_1, D_2 that differ in only one element. It is shown in [7] that if $f : \mathcal{D} \rightarrow R$ is a function with sensitivity Δ_f, then $K(D) = Lap(f(D), \Delta_f/\epsilon)$ is differentially private.

Our adaptations of the differential privacy definitions. Instead of bounding the ratio between output probabilities of actual vs. arbitrary information for a single hourly record, we want to give customers the option of hiding an arbitrary period of time. For example we may want to hide specifics of daily (chunks of 24 records) or weekly (chunks of 168 records) consumption. We call the period length a user is concerned with the *privacy unit*. Furthermore we need to achieve this for statistics in discrete domains (not continuous function), that can only make the bills bigger, never smaller.

Definition 5. *The u-distance $Dist_u(D_1, D_2)$, e.g., $u \in \{hourly, dayly, weekly\}$ between two data sets $D_1, D_2 \in \mathcal{D}$ corresponds to the number of u-units (collection of records) in which D_1 and D_2 differ.*

Our pricing scheme maps each $D \in \mathcal{D}$, $D = (c_1, \ldots, c_{i_{max}})$ to a discrete price: $price(D) = \sum_{i=1}^{i_{max}} c_i \cdot p_I$, where i_{max} is the number of records per billing period, and p_I is the price per hour per instance. Rather than having continuous positive and negative noise as in the original Laplacian differential privacy mechanism, we want to only add discrete positive noise.

If we consider only privacy mechanisms with discrete outputs, we can simplify the differential privacy definition. For discrete distributions, $\Sigma_{Image(K)} = 2^{Image(K)}$, and $Pr[K(D) \in S] = \sum_{r \in S} Pr[K(D) = r]$. Definition 2 can thus be restated as the following equation: $\sum_{r \in S} Pr[K(D) = r | D = D_1] \leq \exp(\epsilon) \cdot \sum_{r \in S} Pr[K(D) = r | D = D_2]$. From this we derive an alternative definition for differential privacy for discrete distributions:

Definition 6. *A randomized function K gives ϵ-differential u-privacy if for all data sets $D_1, D_2 \in \mathcal{D}$ with $Dist_u(D_1, D_2) \leq 1$, and all $r \in Image(K)$,*

$$Pr[K(D) = r | D = D_1] \leq \exp(\epsilon) \times Pr[K(D) = r | D = D_2] \ .$$

The probability is taken over the randomness of K.

Lemma 2. *Definition 2, Definition 3, and Lemma 1 apply to u-privacy:*

1. *For discrete privacy mechanisms Definition 2 and Definition 6 for u = hourly are equivalent.*
2. *Let n_u be the number of records in a u-unit. If K is (n_u, ϵ)-differential hourly-private, then K is also $(n_u \cdot \epsilon)$-differential u-private.*

Dwork [8] notes that, because of the multiplicative nature of the definition, an output whose probability is zero on a given database must also have probability zero on any neighboring database, and therefore, by repeated application of the definition, on any other database.

Handling privacy mechanisms that result in distributions for which the support of $K(D_1)$ and $K(D_2)$ may differ requires extra care. Such a situation arises, e.g., when K adds only positive noise. If for instance $price(D_1) < price(D_2)$ to which K adds positive noise. Let ν_{min} be the minimum amount of noise that is added, then the value $r = price(D_1) + \nu_{min}$ is in the support of $K(D_1)$ but has 0 probability for $K(D_2)$. It follows that such a mechanism can never be differentially private.

To overcome this problem, we define partial differential privacy. A statistic offers partially differential u-privacy if it is differentially private for all outputs in the overlapping support of any two databases D_1 and D_2 with $Dist_u(D_1, D_2) \leq 1$. Furthermore we require the probability that the output of the statistic is not in the overlapping domains to be bound by a small probability δ. This means that the function is differentially private most of the time (or with probability at least $1 - \delta$).

Definition 7. *A randomized function K gives δ-partially ϵ-differential u-privacy if the following two properties hold:*

1. *For all $D_1, D_2 \in \mathcal{D}$ with $Dist_u(D_1, D_2) \leq 1$, and all $r \in Supp(K(D_1)) \cap Supp(K(D_2))$,*

$$Pr[K(D_1) = r] \leq \exp(\epsilon) \times Pr[K(D_2) = r] \ .$$

2. *For all data sets $D_1, D_2 \in \mathcal{D}$ with $Dist_u(D_1, D_2) \leq 1$,*

$$Pr[r \leftarrow K(D_1) : r \notin Supp(K(D_2))] < \delta \ .$$

For both properties, the probability is taken over the randomness of K.

As for the traditional differential privacy definitions, longer periods of privacy can be guaranteed with lower security parameters:

Lemma 3. *Let n_u be the number of records in a u-unit. If K is δ-partially (n_u, ϵ)-differential hourly-private, then K is also $(n_u \cdot \delta)$-partially $(n_u \cdot \epsilon)$-differential u-private*

Proof. Consider the joint distribution of K for all D_1 and D_2 with $RDist(D_1, D_2) \leq n_u$. The probability of drawing a value r not in the domain of at least one of $K(D_i)$ is $\delta' \leq 1 - (1 - \delta)^{n_u} \leq n_u \cdot \delta$. This proves Property 2 for partial differential privacy. If r is in the domain, Property 1 is proved as in Lemma 2. □

Given the above definition for privacy we propose a concrete mechanism to obscure readings. We simply add to the bill $f(D)$ for consumption D an amount of noise drawn from a Geometric distribution with parameter $p = \epsilon/\Delta_{f,u}$.[4] The sensitivity $\Delta_{f,u}$ is the

[4] Two-sided Geometric noise was also proposed in [9] as a differential privacy mechanism.

maximum difference of a bill between two databases D_1 and D_2 differing in at most 1 u-unit (e.g. an hour, a day, or a week). Similarly, ϵ is a security parameter expressing information leakage.

Theorem 1. *Let $f : \mathcal{D} \rightarrow R$ be a function with sensitivity $\Delta_{f,u}$, then $K(D) = f(D) + Geo(\epsilon/\Delta_{f,u})$ is $(2 \cdot \epsilon)$-partially ϵ-differentially u-private.* (See [10] for the proof.)

As also noted by [11], the application of a public function on the outputs of a differentially private statistic does not leak any additional information. We can modify the billing function to only charge up to the maximum possible consumption: $K'(D) = \min(f(D) + Geo(\epsilon/\Delta_{f,u}), \max_{D'} f(D'))$. Intuitively we use geometric noise, as this adds the maximal uncertainty for a given mean. The variant of the geometric distribution with support for negative and positive integers defined as $\Pr[k] = \frac{1}{2}(1-p)^{|k|}p$ is the discrete equivalent of the Laplace distribution, and would also provide differentially private guarantees. We limit ourselves to the proposed noise distribution to ensure users only add positive noise to their bills.

Interpretation of differential privacy in terms of inference. From the attackers perspective the goal of collecting statistics about the output of the privacy mechanism K is to infer something about the underlying database. For instance, the attacker might want to distinguish between two databases D_1 and D_2, in the sense of semantic security.

Differential privacy does not guarantee anything about the probability ratio (likelihood ratio) between databases D_1 and D_2 with $Dist_u(D_1, D_2) \leq 1$ given an observed outcome of K; it merely says that this ratio will differ only by a small factor from the ratio of the prior. Note that because D_1 and D_2 are interchangeable, the new ratio is also bounded from below.

Lemma 4. *Given an observed outcome of a differentially private K the probability ratio (likelihood ratio) between databases D_1 and D_2 with $Dist_u(D_1, D_2) \leq 1$ differs by less than a factor $\exp(\epsilon)$ from the ratio of the prior.*

$$\frac{Pr[D = D_1|K(D) = r]}{Pr[D = D_2|K(D) = r]} \leq \exp(\epsilon) \times \frac{Pr[D = D_1]}{Pr[D = D_2]}.$$

Proof. From Bayes theorem we can write:

$$Pr[D = D_i|K(D) = r] = \frac{Pr[K(D) = r|D = D_i] \times Pr[D = D_i]}{Pr[K(D) = r]}$$

whence, since K is differentially private, we can write:

$$\begin{aligned}\frac{Pr[D = D_1|K(D) = r]}{Pr[D = D_2|K(D) = r]} &= \frac{Pr[K(D) = r|D = D_1]}{Pr[K(D) = r|D = D_2]} \times \frac{Pr[D = D_1]}{Pr[D = D_2]} \\ &\leq \exp(\epsilon) \times \frac{Pr[D = D_1]}{Pr[D = D_2]}\,.\end{aligned}$$

□

Table 1. Yearly average bill after the application of the privacy mechanism K' compared with the fixed rate privacy mechanism. Different values of the security parameter (ϵ), different privacy units (hourly, daily and weekly) as well as the options of paying monthly or yearly are presented. Amounts in parenthesis indicate that the expected cost is higher than paying for the maximum consumption.

Privacy units	Security (ϵ)	Pay Monthly	Pay Yearly	Fixed Rate
Hourly	0.1	$\beta + \$144,000$	$\beta + \$12,000$	$\$10,512,000$
(units = 1)	0.01	$\beta + \$1,440,000$	$\beta + \$120,000$	$\$10,512,000$
Daily	0.1	$\beta + \$3,456,000$	$\beta + \$288,000$	$\$10,512,000$
(units = 24)	0.01	$(\$10,512,000)$	$\beta + \$2,880,000$	$\$10,512,000$
Weekly	0.1	$(\$10,512,000)$	$\beta + \$2,016,000$	$\$10,512,000$
(units = 168)	0.01	$(\$10,512,000)$	$(\$10,512,000)$	$\$10,512,000$

The cost of privacy. Obscuring bills by adding noise may lead to paying extra for a service. Customers have incentives to minimise their costs for a desired level of privacy protection. We provide a few illustrative examples of the average extra cost involved in settling a bill for different privacy units of an hour, a day or a week. In our usual example we consider a private cloud of 10K CPUs, billed as \$0.12 a CPU / hour. We denote as $\beta = f(D)$ the actual service cost associated with the use of the service for a year.

It is clear from Table 1 that providing a differentially private bill for more than a single hourly period is an expensive business. The proposed mechanism allows for lower overheads for yearly bills when customers wish to protect arbitrary hours or days in the year. When it comes to protecting arbitrary weeks this protection is only offered with a low security parameter ($\epsilon = 0.1$). Why is the cost so high? It is because the privacy guarantee offered is very strong: no matter what side information the adversary has, including the detailed readings for other periods, they should not be able to infer information about an arbitrary privacy unit. For example if the adversary knows the exact consumption for the other 364 days they should still not learn more than permitted about the last day. This is a very strong guarantee and as a result it comes at a high cost, when applied directly.

Table 1 also contains the cost of paying bills monthly, which incur a 12 fold overhead for the same level of protection. It is clear that there are advantages in paying in batches if in fact the desired property is to hide any fixed period of time within the billing period (an hour, a day, a week). We will see in the next section how we can do better than this: we can aggregate the true cost of service provision, and use cryptographic methods to reclaim most of the additional cost of privacy in the long term without sacrificing any security.

Longer guarantees. Degradation of privacy in our framework is graceful, since some privacy guarantees are provided for periods longer than what is strictly defined by the chosen u-units. For example a user may choose a partially ϵ-differential function $K_{\epsilon,24}$ providing u-privacy for a day (i.e. 1 u-unit = 24 hourly periods) with $\epsilon = 0.01$. In our standing example this means he should add an extra amount to his bill drawn from a Geometric distribution with parameter $\$2,880,000$. What does that guarantee? Let's assume the adversary knows the exact consumption about all days except for one.

Furthermore the adversary knows that the consumption on the target day could only have taken one out of two values with equal probability: this means that the ratio of priors $\frac{Pr[D=D_1]}{Pr[D=D_2]} = 1$. Then after receiving information about the bill the adversary would at best know that $0.99 \approx 1/(1+\epsilon) \approx 1/e^{\epsilon} \leq \frac{Pr[D=D_1|K'(D)]}{Pr[D=D_2|K'(D)]} \leq e^{\epsilon} \approx 1+\epsilon = 1.01$. This is a small amount of information.

Now let's consider an adversary that tries to infer something over a longer period, e.g., a week. The adversary knows all user consumption outside this target week, and furthermore knows that user consumption within the week could only have been one of two possibilities D'_1 or D'_2 with equal probability as before. Due to Lemma 3 we know that the $K'(D)$ scheme is also partially ϵ-differentially private for a longer u-unit of a week (1 weakly-unit = 7×24 hourly-units), with a new security parameter $\epsilon' = \epsilon \cdot 7$. This means that the new posterior ratio of probabilities over the two only possible outcomes is $0.93 \approx 1/(1+\epsilon') \approx 1/e^{\epsilon'} \leq \frac{Pr[D=D_1|K'(D)]}{Pr[D=D_2|K'(D)]} \leq e^{\epsilon'} \approx 1+\epsilon' = 1.07$. Despite the lower degree of privacy, some quantifiable protection is still available against longer-term profiling.

Limitations. Our variant of differential privacy relies on only introducing positive noise. This is desirable as it guarantees that the bill at least covers the cost of service provision. At the same time this provides a one sided security property: a final bill can always be confused with a lower bill, but not always with a higher bill. For example there is a positive probability that a sensitive day passes with no consumption and then no noise is added to the bill. If an adversary knows all other consumptions in the year, they can infer that indeed no consumption took place on the unknown day. Our mechanism thus assumes that the baseline of no consumption is not as sensitive as high consumption.

While information leakage about low levels of consumption is possible, it is not very likely for high levels of security as characterised by the security parameter δ.

Summary. We have shown that adding noise to the bill can provide high levels of security parametrised by a parameter ϵ and a privacy unit. This security holds even against adversaries with knowledge of many readings. At the same time this comes with a high overhead. In the next section we show that the bulk of the cost of providing privacy can be recuperated in the long run. We achieve this by keeping hidden accounts of what is actually due for service provision, versus what has been paid. In the long run users can only add the necessary noise to keep their accounts positive, including negative noise – while ensuring that their funds cover their consumption.

3 Private Billing with Rebates

We have seen that one way of protecting privacy involves adding 'noise' to the bill to be payed for a certain period. Yet, the amount of noise can become significant particularly to achieve a high quality of privacy or privacy for longer periods within the billing time frame. For this reason we develop a complementary oblivious billing protocol that can be used to alleviate those concerns. Its key features include:

– The ability to maintain a hidden bill of actual consumption that can be used to reclaim any excess used for protecting privacy at a later time.

- A mechanism for proving that the amount payed to the utility provider exceeds the bill for actual consumption without revealing the actual bill.
- Support for an initial deposit to support later use of positive as well as negative noise for the bills.
- Compatibility with anonymous e-cash schemes allowing bills to be settled anonymously, as well as advanced privacy friendly payment mechanisms that allow users to hide the amounts actually payed to the utilities.

We discuss in detail and prove the correctness of the billing protocols, and the mechanisms to ensure payments exceed the amount consumed. The specifics of optional e-cash protocols that allow hidden payments are beyond the scope of this work, and we leave their detailed description to future work.

Our oblivious payment protocols can be used to reclaim in the long run an excess payed as a result of a differentially private billing mechanism as presented in the previous sections. With the deposit facility, adding negative noise is possible, as long as the overall balance of payments stays positive. The protocols can also be used to support the naive mechanism where a bill for maximal consumption is payed, and allow parties to later reclaim some of it back. Finally given anonymous e-cash they can be used to provide full oblivious payments without the need to add any noise to the bills, as they never need to be revealed (technically: $noise = -fee$). Which variant to use therefore depends on the infrastructure available and the degree of complexity parties are ready to accept.

3.1 The PSM Protocol

We will be building upon PSM (Privacy-Preserving Smart Metering), a cryptographic protocol for privacy-friendly billing [5]. PSM mediates interactions among three parties: a meter M that outputs consumption data $cons$ and related information $other$; a service provider P that establishes a pricing policy Υ and a user U that receives consumption readings from meter M and at each billing period pays a fee to provider P. The pricing policy Υ is a public function that takes consumption data $cons$ together with other information $other$ (e.g., the time of consumption) and computes a price. The overall price $price(D) = \sum_{i=1}^{|D|} price_i$ is computed by adding the prices corresponding to the individual consumptions in a billing period. For our running private cloud example, $\Upsilon(cons, other) = cons \cdot 0.12$ and does not depend on $other$. As in the original protocols we assume a tamper resistant meter is used to provide accurate and appropriately cryptographically packaged readings. These can be processed by the user to prove their bill in zero-knowledge to the provider. At this point users may also choose to add some noise to ensure differential privacy.

The security of PSM is shown in the simulation-based security paradigm [12,13]. In the real world, the protocol $\mathsf{PSM}(\mathrm{M}, \mathrm{P}, \mathrm{U})$ is run in an adversarial environment that may corrupt some of the protocol parties, indicated by $\tilde{\mathrm{M}}$, $\tilde{\mathrm{P}}$, $\tilde{\mathrm{U}}$. Corrupted parties just forward messages between the environment and honest protocol participants. In the ideal world, dummy protocol parties D_{M}, D_{P}, D_{U} run an ideal protocol $\mathsf{Ideal}(\mathcal{F}_{\mathsf{PSM}}, D_{\mathrm{M}}, D_{\mathrm{P}}, D_{\mathrm{U}})$ by just forwarding messages to an ideal functionality $\mathcal{F}_{\mathsf{PSM}}$. Uncorrupted D_x, $x \in \{\mathrm{M}, \mathrm{P}, \mathrm{U}\}$ interact with the environment while corrupted dummy parties $\tilde{D}_x$ interact with a simulator Sim.

We consider w.l.o.g. a corrupted provider $\tilde{\mathrm{P}}$ and say that a protocol is secure against $\tilde{\mathrm{P}}$, if there exists a simulator Sim such that no environment Env can tell whether it is interacting with $\mathsf{PSM}(\mathrm{M}, \tilde{\mathrm{P}}, \mathrm{U})$ or with $\mathsf{Sim}\|\mathsf{Ideal}(\mathcal{F}_{\mathsf{PSM}}, D_{\mathrm{M}}, \tilde{D}_{\mathrm{P}}, D_{\mathrm{U}})$. Conceptually Sim translates influence that Env has through $\tilde{\mathrm{P}}$ on the protocol into influence on $\mathcal{F}_{\mathsf{PSM}}$ through $\tilde{D}_{\mathrm{P}}$, and leakage that $\tilde{D}_{\mathrm{P}}$ receives from $\mathcal{F}_{\mathsf{PSM}}$ into leakage that Env could learn from $\tilde{\mathrm{P}}$. Similarly, PSM is proven secure against a corrupted user $\tilde{\mathrm{U}}$.

Listing 1. Functionality $\mathcal{F}_{\mathsf{PBR}}$

$\mathcal{F}_{\mathsf{PBR}}$ is parameterized by deposit relation R and a policy set Y and interacts with dummy parties D_M, D_P and D_U. Initially $T = \emptyset$, $d = 0$, $account = 0$.

On $(\texttt{Policy}, \Upsilon)$ from D_P where $\Upsilon \in Y$
- store Υ; send $(\texttt{Policy}, \Upsilon)$ to D_U

On $(\texttt{Consume}, cons, other)$ from D_M
- increment counter d; add $(d, cons, other)$ to T; send $(\texttt{Consume}, cons, other)$ to D_U

On $(\texttt{Deposit}, (inc, wit), instance)$ from D_U where $balance + inc \geq 0$
- if $((inc, wit), instance) \in R$, let $balance$ += inc, send $(\texttt{Deposit}, instance)$ to D_P

On $(\texttt{Payment}, from, until, noise)$ from D_U where $0 \leq from \leq until \leq d$ and $balance + noise \geq 0$
- for $i = from$ to $until$, calculate $price_i = \Upsilon(cons_i, other_i)$
- let $fee = \sum_{i=from}^{until} price_i + noise$ and $balance$ += $noise$
- send $(\texttt{Pay}, from, until, fee)$ to D_P

3.2 Rebate Ideal Functionality

We propose a new ideal functionality $\mathcal{F}_{\mathsf{PBR}}$ (see Listing 1) that extends the functionality $\mathcal{F}_{\mathsf{PSM}}$. The functionality keeps track of the user's consumptions in a set T containing tuples $(i, cons, other)$. During a payment, the policy Υ is applied to all $(cons, other)$ in the interval $[from, until]$ to compute the price $price_i = \Upsilon(cons_i, other_i)$ per consumption. The overall fee that the user has to pay is computed as $fee = \sum_{i=from}^{until} price_i + noise$. The value $noise$ is added to the fee to improve the user's privacy. The ideal functionality also maintains a balance that corresponds to the sum of all the $noise$ added to payments. Note that the user can get rebates by using negative noise, but that the balance is never allowed to be negative.

The ideal functionality also allows to increase the balance through a deposit. The user has to provide input $((inc, wit), instance) \in R$. The parameterization by relation R allows to support both standard deposit mechanisms that reveal the deposited amount inc as well as advanced deposit mechanisms that hide this value from the provider. In the simple mechanism the user reveals how much he wants to deposit: $wit = \epsilon$ and R corresponds to simple equality, i.e. $R = \{((inc, \epsilon), inc) | inc \in \mathbb{Z}\}$.

To obtain a more advanced privacy-friendly deposit mechanism, the witness could correspond to a one-show anonymous credential $cred$. The relation requires that $cred$ is a one-show credential with an increment value inc and serial number s issued under public key pk_B, i.e, $R = \{((inc, cred), (s, pk_B)) | \mathsf{Verify}(pk_B, cred, (inc, s)) = \mathsf{accept}\}$. The real protocol cryptographically enforces this using a zero-knowledge

proof of signature possession [14].[5] To obtain such a one-show credential without revealing the value of inc to the provider, additional infrastructure is needed. In particular such a mechanism seems to require some form of anonymous payment, either physical cash or anonymous e-cash. Given such a payment mechanism, the provider's bank, after receiving an anonymous payment of value inc and depositing this amount on the provider's bank account, could blindly issue the signature $\mathsf{Sign}(pk_B, (inc, s))$ using a partially-blind issuing protocol [14]. The issue protocol guarantees that the bank does not learn s, and thus even if the provider and his bank collude they cannot link the issuing of $cred$ to its use.

Listing 2. Protocol PBR(M, P, U)

Parties M, P, U are parameterized by R and Y and interact over secure channels. All participants have registered public keys generated by Mkeygen, Pkeygen, Ukeygen with a key registration authority $\mathcal{F}_{\mathsf{REG}}$ and keep their private keys secret. P also registers commitment parameters par_c.

On $(\texttt{Policy}, \Upsilon)$ from Env
- P runs $\Upsilon_s \leftarrow \mathsf{SignPolicy}(sk_{\mathrm{P}}, \Upsilon)$ and sends Υ_s to U
- Upon receiving Υ_s, U extracts Υ; if $\Upsilon \notin Y$, he aborts
- if $\mathsf{VerifyPolicy}(pk_{\mathrm{P}}, \Upsilon_s) = 1$, U stores Υ_s, and sends $(\texttt{Policy}, \Upsilon)$ to Env

On $(\texttt{Consume}, cons, other)$ from Env
- M increments d_{M}, runs SC $\leftarrow \mathsf{SignConsumption}(sk_{\mathrm{M}}, par_c, cons, other, d_{\mathrm{M}})$ and sends (SC) to U
- Upon receiving (SC), U runs $b \leftarrow \mathsf{VerifyConsumption}(pk_{\mathrm{M}}, par_c, \mathrm{SC}, d_{\mathrm{U}} + 1)$
- if $b = 1$, U increments d_{U}, adds SC to T_{U}, parses SC as $(d_{\mathrm{M}}, cons, open_{cons}, c_{cons}, other, open_{other}, c_{other}, \mathrm{sc})$, and sends $(\texttt{Consume}, cons, other)$ to Env

On $(\texttt{Deposit}, (inc, wit), instance)$ from Env where $balance + inc \geq 0$ and $((inc, wit), instance) \in R$
- U runs $(aux', D) \leftarrow \mathsf{Deposit}(par_c, (inc, wit), instance, aux, R)$
- U sets $balance$ += inc and $aux = aux'$ and sends $(D, instance)$ to P
- Upon receiving $(D, instance)$, P runs $(c'_{balance}, b) \leftarrow \mathsf{VerifyDeposit}(par_c, D, c_{balance}, instance, R)$
- if $b = 1$, he sets $c_{balance} = c'_{balance}$ and sends $(\texttt{Deposit}, instance)$ to Env

On $(\texttt{Payment}, from, until, noise)$ from Env where $0 \leq from \leq until \leq d_{\mathrm{U}}$ and $balance + noise \geq 0$
- U runs $(aux', \mathrm{Q}) \leftarrow \mathsf{Pay}(sk_{\mathrm{U}}, par_c, \Upsilon_s, T_{\mathrm{U}}[from : until], noise, aux)$
- U sets $aux = aux'$ and $balance$ += $noise$; U sends $(\mathrm{Q}, from, until)$ to P
- Upon receiving $(\mathrm{Q}, from, until)$, P runs $(fee, c'_{balance}, b) \leftarrow \mathsf{VerifyPayment}(pk_{\mathrm{M}}, pk_{\mathrm{U}}, pk_{\mathrm{P}}, par_c, \mathrm{Q}, c_{balance}, from, until)$
- if b=1, he sets $c_{balance} = c'_{balance}$ and sends $(\mathsf{Pay}, from, until, fee)$ to Env

[5] A zero-knowledge proof of knowledge is a two-party protocol between a prover and a verifier. The prover convinces the verifier, who knows only a public proof $instance$, that he knows a secret input (called $witness$) that allows him to prove that the public and the secret value together fulfill some relational statement $(witness, instance) \in R$ without disclosing the secret input to the verifier.

3.3 Rebate Protocol

We propose a new protocol for privacy-preserving billing with rebates (PBR) (see Listing 2) that extends PSM with a mechanism for adding noise, keeping a hidden balance, and making deposits. Like PSM, our protocol operates in the $\mathcal{F}_{\mathsf{REG}}$ hybrid-model [12] where parties register their public keys at a trusted registration entity. As in the original scheme the user receives signed policies from the utility provider P and signed readings from the meter M. The payment transaction only reveals the overall fee, which now can be subject to additional noise.

We extend this protocol with a novel oblivious rebate system that allows the user to get rebates (in the amount of his noise) in future payments. The rebate is implemented using a homomorphic update c_{noise} to a balance commitment $c_{balance}$ that commits the user to his balance towards the provider but keeps the $balance$ itself secret. Our protocol supports an optional Deposit mechanism that allows the user to add or withdraw funds from the rebate $balance$. Value aux contains the opening for a commitment $c_{balance}$ to $balance$. Through the use of zero-knowledge proofs the provider is guaranteed that the value committed to in $c_{balance}$ is updated correctly and never becomes negative.

The protocol parties P, U, and M interact with each other using algorithms Pkeygen, Ukeygen, Mkeygen (for key generation); SignPolicy, SignConsumption, Deposit, and Pay (for generation of input); and VerifyPolicy, VerifyConsumption, VerifyDeposit, and VerifyPayment (for verification of input). The functionality of the meter as well as SignPolicy, SignConsumption, VerifyPolicy, and VerifyConsumption are unchanged from the original scheme.[6] We describe the new Deposit and VerifyDeposit algorithms and the changes to Pay and VerifyPayment:

Listing 3. Algorithms

- $\mathsf{Deposit}(par_c, (inc, wit), instance, aux, R)$. Parse aux as $(balance, open_{balance}, c_{balance})$. Compute commitment $(c_{inc}, open_{inc}) = \mathsf{Commit}(par_c, inc)$ and a non-interactive proof π_{inc}:[7]

$$\begin{aligned}\pi_{inc} \leftarrow \mathsf{NIPK}\{\ & (inc, open_{inc}, wit, balance, open_{balance}) : \\ & (c_{balance}, open_{balance}) = \mathsf{Commit}(par_c, balance) \wedge \\ & (c_{inc}, open_{inc}) = \mathsf{Commit}(par_c, inc) \wedge \\ & ((inc, wit), instance) \in R \ \wedge \ balance + inc \geq 0\}\ .\end{aligned}$$

  Let $D = (\pi_{inc}, c_{inc})$ and $aux' = (balance + inc, open_{balance} + open_{inc}, c_{balance} \otimes c_{inc})$. Output (aux', D).
- $\mathsf{VerifyDeposit}(par_c, D, c_{balance}, instance, R)$. Parse D as $(\pi_{balance}, c_{inc})$. Verify π_{inc}. If verification succeeds, set $b = 1$ and $c'_{balance} = c_{balance} \otimes c_{inc}$, otherwise set $b = 0$. Output $(c'_{balance}, b)$.

[6] For the sake of brevity we omit the Reveal mechanism of PSM. It would add little new and could be implemented in a straight forward manner using trapdoor commitments.

[7] If R corresponds to equality, the protocol can be simplified to avoid computing c_{inc}.

- $\mathsf{Pay}(sk_{\mathrm{U}}, par_c, \Upsilon_s, T, noise, aux)$. Parse aux as $(balance, open_{balance}, c_{balance})$. Compute commitment $(c_{noise}, open_{noise}) = \mathsf{Commit}(par_c, noise)$ and a non-interactive proof π_{noise}:

$$\pi_{noise} \leftarrow \mathsf{NIPK}\{\ (noise, open_{noise}, balance, open_{balance}) : (c_{balance}, open_{balance}) = \mathsf{Commit}(par_c, balance) \wedge (c_{inc}, open_{inc}) = \mathsf{Commit}(par_c, inc)\ \wedge\ balance + noise \geq 0\}\ .$$

  Let $aux' = (balance + noise, open_{balance} + open_{noise}, c_{balance} \otimes c_{noise})$.

  The rest of the algorithm follows [5]: For each $(i, cons, open_{cons}, c_{cons}, other, open_{other}, c_{other}, \mathrm{sc}) \in T$ where $from \leq i \leq until$, calculate $price_i = \Upsilon(cons, other)$, commitment $(c_{price_i}, open_{price_i}) = \mathsf{Commit}(par_c, price)$, and a proof π_i that $price_i$ was computed correctly according to the policy and the commitments c_{cons}, c_{other}. The proof π_i depends on the policy Υ and can use auxiliary values in Υ_s, see [5] on how to implement different pricing policies.
  Computing $fee = noise + \sum_{i=from}^{until} price_i$ and $open_{fee} = open_{noise} + \sum_{i=from}^{until} open_{price_i}$ gives an opening to a commitment to fee. Let $\mathrm{Q} = (fee, open_{fee}, c_{noise}, \pi_{balance}, \{\mathrm{sc}_i, i, c_{cons_i}, c_{other_i}, c_{price_i}, \pi_i\}_{i=1}^{N})$. Output (aux', Q).[8]
- $\mathsf{VerifyPayment}(pk_{\mathrm{M}}, pk_{\mathrm{U}}, pk_{\mathrm{P}}, par_c, \mathrm{Q}, c_{balance}, from, until)$. Parse Q as $(fee, open_{fee}, c_{noise}, \pi_{balance}, \{\mathrm{sc}_i, d_i, c_{cons_i}, c_{other_i}, c_{price_i}, \pi_i\}_{i=1}^{N})$. Verify π_{noise}. If verification fails, set $b = 0$. Otherwise set $c'_{balance} = c_{balance} \otimes c_{noise}$ and $b = 1$.

  The rest of the algorithm follows [5]: For $i = from$ to $until$, run $\mathrm{Mverify}(pk_{\mathrm{M}}, \mathrm{sc}_i, \langle i, c_{cons_i}, c_{other_i}\rangle)$ and verify π_i. Set $b = 0$ if any of the signatures or the proofs is not correct. Add the commitments to the prices $c'_{fee} = c_{noise} \otimes \left(\otimes_{i=1}^{N} c_{price_i}\right)$ and execute $\mathsf{Open}(par_c, c'_{fee}, fee, open_{fee})$. If the output is reject set $b = 0$. Output $(fee, c'_{balance}, b)$.

Theorem 2. *Given the security of its building blocks,* PBR *is secure against a corrupted provider* $\tilde{\mathrm{P}}$ *and a corrupted user* $\tilde{\mathrm{U}}$. (See [10] for the proof.)

Using PBR *for differential privacy.* Even an ideal cryptographic billing mechanism as described by the PSM or PBR ideal functionalities cannot protect the user's privacy against an adversary/environment that already knows enough about the user's behavior – possibly including all consumption or additional random noise – to infer privacy sensitive information from the final fee alone. For our privacy analysis we assume that the environment Env is divided into a part $\mathsf{Env}_{\mathrm{U}}$ that is controlled by the user, and a part $\mathsf{Env}_{\tilde{\mathrm{P}}}$ that is controlled by the adversary and that may have some influence on and knowledge about the user's behavior. In the original PSM protocol all the final fee is only the result of the individual consumptions of $\mathsf{Env}_{\mathrm{U}}$ for which the provider may make inferences or gain side information. The PBR protocol gives $\mathsf{Env}_{\mathrm{U}}$ the possibility to obscure the fee with random noise, which is easier to conceal from $\mathsf{Env}_{\tilde{\mathrm{P}}}$.

4 Conclusions

Our PBR protocol allows the user to add random noise to the final bill, to hide usage patterns that could otherwise be deduced from the fee. The rebate protocol supports

[8] In [5], the user keys sk_{U} and pk_{U} are used to create and verify a signature on the payment message. This intuitively facilitates non-repudiation and non-exculpability properties, but is not modeled by the ideal functionality. This carries over to our adaptations.

deposits, anonymous payments using e-cash, and negative bill noise, while ensuring that the funds paid always cover the cost of consumption. The use of noise, however, comes at a cost, as it is money that the user has to pay upfront as a deposit and cannot invest elsewhere. Consequently, we adapt the differential privacy framework to study how much noise is needed to protect specific consumption windows at different security levels. The differential privacy framework protects users against worse case outcomes – we leave as an open problem crafting more economical noise regimes to protect privacy by making further assumption about the users' typical behavior.

Acknowledgment. We would like to thank Claudia Diaz, Carmela Troncoso, Cedric Fournet, and Jorn Lapon for discussions that were most helpful in the preparation of this work. This work was supported in part by GOA TENSE (GOA/11/007), and by the IAP Programme P6/26 BCRYPT (Belgian Science Policy). Alfredo Rial is a Research Foundation - Flanders (FWO) doctoral researcher.

References

1. Anderson, R., Fuloria, S.: On the security economics of electricity metering. In: The Ninth Workshop on the Economics of Information Security (2010)
2. Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., Irwin, D.: Private memoirs of a smart meter. In: 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland (November 2010)
3. Balasch, J., Rial, A., Troncoso, C., Preneel, B., Verbauwhede, I., Geuens, C.: Pretp: Privacy-preserving electronic toll pricing. In: USENIX Security Symposium, pp. 63–78. USENIX Association (2010)
4. Bohli, J.M., Sorge, C., Ugus, O.: A privacy model for smart metering. In: 2010 IEEE International Conference on Communications Workshops (ICC), pp. 1–5 (May 2010)
5. Rial, A., Danezis, G.: Privacy-preserving smart metering. Technical Report MSR-TR-2010-150, Microsoft Research (November 2010)
6. Lipner, S.B.: A comment on the confinement problem. In: Proceedings of the Fifth ACM Symposium on Operating Systems Principles, SOSP 1975, pp. 192–196. ACM, New York (1975)
7. Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
8. Dwork, C.: Differential privacy in new settings. In: Charikar, M. (ed.) SODA, pp. 174–183. SIAM, Philadelphia (2010)
9. Ghosh, A., Roughgarden, T., Sundararajan, M.: Universally utility-maximizing privacy mechanisms. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, pp. 351–360. ACM, New York (2009)
10. Danezis, G., Kohlweiss, M., Rial, A.: Differentially private billing with rebates. Cryptology ePrint Archive, Report 2011/134 (2011), http://eprint.iacr.org/
11. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In: Libkin, L. (ed.) PODS, pp. 273–282. ACM, New York (2007)
12. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS, pp. 136–145 (2001)
13. Kusters, R.: Simulation-based security with inexhaustible interactive turing machines. In: 19th IEEE Computer Security Foundations Workshop, pp. 12–320. IEEE, Los Alamitos (2006)
14. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)

Statistical Decision Methods in Hidden Information Detection

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

ICD - LM2S - Université de Technologie de Troyes - UMR STMR CNRS 6279
12, rue Marie Curie - B.P. 2060 - 10010 Troyes cedex - France
name.surname@utt.fr

Abstract. The goal of this paper is to show how the statistical decision theory based on the parametric statistical model of the cover media can be useful in theory and practice of hidden information detection.

1 Introduction and Contribution

It is an important and useful challenge for security forces to reliably detect in a huge set of files (image, audio, and video) which of these files contain the hidden information (like a text, an image, or an audio or a video record). An efficient statistical test should be able to detect the presence of hidden information embedded in the cover media. It is assumed that the embedding scheme is a priori unknown but it belongs to a commonly used family of steganographic LSB replacement based algorithms. Certainly, such steganographic algorithms are not extremely efficient but they are simple, popular, and downloadable on the Internet and can be easily applied by any person.

In such an operational context, the most important challenge is to get the hidden information detection algorithms with analytically predictable probabilities of false alarm and non detection. These algorithms should be immediately applicable without any supervised learning methods using sets of training examples (SVM-based algorithms). On the contrary, the capacity of a hidden information detection algorithm to detect a very sophisticated but not frequently used embedding algorithm with a low embedding rate is not very important in the framework of the above mentioned scenario.

The recently proposed steganalysers [7,8,9] are certainly very interesting and efficient but these *ad hoc* algorithms have been designed with a limited exploitation of cover media statistical model and hypothesis testing theory. Moreover, the only solution to get the statistical properties of these *ad hoc* algorithms is the statistical simulation by using large databases of cover media.

An alternative approach is to use the hypothesis testing theory with a parametric model of cover media. The first step in the direction of hypothesis testing has been done in [14].

* This work is supported by French National Agency (ANR) through ANR-CSOSG Program (Project ANR-07-SECU-004).

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 163–177, 2011.

In the actual paper, the direction started in [14] is extended to take into account two new phenomena :

- an impact of data quantization on the statistical decision;
- benefits from using a parametric statistical model of cover media.

This paper is mainly, but not exclusively, devoted to the situation when the cover media is represented by a natural image produced by a digital camera. This fact defines the above mentioned points of our interest. The advantages of a parametric statistical model are well known. The hypothesis testing theory is relatively well developed for such models. We are especially interested in the asymptotic decision theory and in dealing with non informative (nuisance) parameters of the cover media model. Both directions are interesting because the number of bytes (or pixels) is typically very large for modern cover media and the nuisance parameters of statistical model are only partially known.

Natural images are obtained by using a digital camera which obligatory includes a quantization. The parameters of statistical model are related to several factors (the scene, the amount of light, the focus, the exposure, the objective lens, CCD,...). Physically these factors define a continuous state space model but the decision should be done by using the quantized output of digital camera. More profound discussion of a parametric statistical model of natural images is in the companion paper [2].

The goal of this paper is threefold :

1. define the statistical framework of hidden information detection based on a parametric model of cover media by using the quantized observations;
2. design optimal statistical tests and to study their statistical properties and the impact of quantized observations on the quality of these tests;
3. theoretically compare the (almost) optimal statistical tests with the WS steganalysers, recently developed and commonly used in hidden information detection.

The paper is organized as follows. The problem of statistical decision based on quantized observations is stated in Section 2. The case of a known embedding rate is discussed. Section 3 is devoted to the problem of quantization. Its impact on the quality of statistical tests (steganalysers) is also studied here. A more general and realistic case of unknown embedding rate is discussed in Section 4. A solution to this case based on the local asymptotic approach is presented in Section 5. Finally, the proposed (almost optimal) detection algorithm is theoretically compared with the WS steganalysers, commonly used in hidden information detection in Section 6. Some conclusions are drawn in Section 7.

2 Statistical Decision Based on Quantized Observations

2.1 Model of Quantized Cover Media

Let us assume that the observation vector $C_n = (c_1, \ldots, c_n)^T$ which characterizes a cover media is defined in the following manner :

$$C_n = Q_1[Y_n], \quad Y_n \sim P_\theta, \tag{1}$$

where $Q_1[y_i] = \lfloor y_i \rfloor$ is the operation of uniform quantization (integer part of y_i) and the vector $Y_n = (y_1, \ldots, y_n)^T$ follows the distribution P_θ parameterized by the parametric vector θ. The binary representation of c (the index is omitted to seek simplicity) is

$$c = Q_1[y] = \sum_{i=0}^{q-1} b_i 2^i, \quad \text{where } b_i \in \{0,1\}, \quad c \in \{0,1,2,\ldots,2^q-1\}. \tag{2}$$

A simplified model of quantization (1) is used in this paper. It is assumed that the saturation is absent, i.e. the probability of the excess over the boundary 0 or $2^q - 1$ for the observation y is negligible.

2.2 Problem Statement: Test between Two Hypotheses

First, let us define two alternative hypotheses for one quantized observation z (seeking simplicity) :

$$\mathcal{H}_0 \ : \ z = c = Q_1[y] \sim Q_{Q_1} = [q_0, \ldots, q_{2^q-1}] \tag{3}$$

and

$$\mathcal{H}_1 \ : z = \begin{cases} Q_2[y] + z_s \,\text{with probability}\, R \\ c = Q_1[y] \,\text{with probability}\, 1-R, \end{cases} \tag{4}$$

where R is the embedding rate, $Q_2[y] = \sum_{i=1}^{q-1} b_i 2^i$, is a uniform quantization by using 2^{q-1} thresholds, $Q_2[y] \sim Q_{Q_2}$, $z_s \sim Q_s = \mathrm{B}(1,p)$ is the Bernoulli distribution which defines the hidden information (usually $p = 0.5$). In other words, to get the double quantization $Q_2[y]$ from $z = Q_1[y]$ the LSB is deleted, i.e. $b_0 \equiv 0$. Hence, under hypothesis $\mathcal{H}_1$, the LSB is used as a container of hidden information. In the rest of the paper it is assumed that $Q_2[z] = Q_2[y]$.

2.3 A Known Embedding Rate. Two Simple Hypotheses: Likelihood Ratio Test

Let us suppose that the distributions $Q_s(z_s) = 1/2$, $z_s \in \{0,1\}$, Q_{Q_1}, Q_{Q_2} and the embedding rate R are exactly known. In this case the LR for one observation is written as follows :

$$\Lambda_R(z) = R\Lambda_1(z) + (1-R), \quad \Lambda_1(z) = \frac{Q_s(b_0) Q_{Q_2}(Q_2[z])}{Q_{Q_1}(z)} = \frac{Q_{Q_2}(Q_2[z])}{2 Q_{Q_1}(z)}. \tag{5}$$

where b_0 is the LSB of z. The most powerful (MP) Neyman-Pearson test over the class

$$\mathcal{K}_{\alpha_0} = \{\delta : \mathbb{P}_0(\delta(Z_n) = \mathcal{H}_1) \le \alpha_0\}, \tag{6}$$

where $\mathbb{P}_i(\ldots)$ denotes the probability under hypothesis $\mathcal{H}_i$, $i = 0,1$, is given by the following decision rule :

$$\delta_R(Z_n) = \begin{cases} \mathcal{H}_0 \text{ if } \Lambda_R(Z_n) = \prod_{i=1}^{n} \Lambda_R(z_i) < h \\ \mathcal{H}_1 \text{ if } \Lambda_R(Z_n) = \prod_{i=1}^{n} \Lambda_R(z_i) \ge h \end{cases}, \tag{7}$$

where the threshold h is the solution of the following equation $\mathbb{P}_0(\Lambda_R(Z_n) \geq h) = \alpha_0$. The MP test $\delta_R(Z_n)$ maximizes the power

$$\beta_{\delta_R} = 1 - \mathbb{P}_1(\delta_R(Z_n) = \mathcal{H}_0) = 1 - \alpha_1 \tag{8}$$

over the class $\mathcal{K}_{\alpha_0}$.

3 Simple Model of Cover Media

3.1 Exact and Approximate Likelihood Ratio

Let us assume an independent random sequence $y_1, \ldots, y_n$, $y_i \sim \mathcal{N}(\theta, \sigma^2)$. The quantized variable z_i follows a "discrete" normal distribution i.e. :

$$z_i = Q_1[y_i] \sim Q_{Q_1} = [q_0, \ldots, q_{2^q-1}], \quad z \in [0, 1, 2, \ldots, 2^q - 1], \tag{9}$$

where the coefficients q_i are computed in the following manner

$$q_i = \int_i^{i+1} \varphi(x)dx = \Phi(i+1) - \Phi(i), \quad \varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\theta)^2}{2\sigma^2}\right\}, \tag{10}$$

where $\Phi(x) = \int_{-\infty}^{x} \varphi(u)du$. It is easy to see that for any R the LR given by equation (5) depends on the observations through the LR ratio $\Lambda_1(z)$ computed under assumption that $R = 1$. The exact equation of this log LR is given by :

$$\begin{aligned} \log \Lambda_1(Z_n) &= n \log \frac{1}{2} + \sum_{i=1}^{n} \log Q_{Q_2}\left(Q_2[z_i]\right) - \sum_{i=1}^{n} \log Q_{Q_1}\left(z_i\right) \\ &= \sum_{i=1}^{n} \frac{1}{2\sigma^2}\left[-\left(Q_2[z_i] + 1 + \delta_{2,i} - \theta\right)^2 + \left(z_i + 0.5 + \delta_{1,i} - \theta\right)^2\right]. \end{aligned} \tag{11}$$

The approximate equation of the log LR is

$$\log \Lambda_1(Z_n) \simeq \log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n} \frac{1}{2\sigma^2}\left[-\left(Q_2[z_i] + 1 - \theta\right)^2 + \left(z_i + 0.5 - \theta\right)^2\right]. \tag{12}$$

The corrective terms due to quantization $\delta_{1,i}$ and $\delta_{2,i}$ are omitted in the last equation and in the rest of this section.

3.2 The Moments of Approximate Log Likelihood Ratio

It follows from the central limit theorem [17] that the fraction

$$\frac{\log \widetilde{\Lambda}_1(Z_n) - n\mathbb{E}\left(\log \widetilde{\Lambda}_1(z)\right)}{\sigma\sqrt{n}} \underset{n\to\infty}{\rightsquigarrow} \mathcal{N}(0, 1), \tag{13}$$

where $\sigma^2 = \mathrm{Var}\left(\log \widetilde{\Lambda}_1(z)\right)$, $\rightsquigarrow$ is the weak convergence and $\log \widetilde{\Lambda}_1(Z_n)$ is the approximate log LR given by (12), will converge in distribution to the standard normal distribution as n goes to infinity. The expectation and variance

are denoted by $\mathbb{E}(\ldots)$ and $\mathrm{Var}(\ldots)$ respectively. Hence, to compute the error probabilities it is necessary to get the expectations and variances of the approximate log LR. Under hypothesis $\mathcal{H}_0$, the approximate log LR can be re-written as follows

$$\log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n} \left[\frac{\zeta_i(b_{0,i} - 0.5)}{\sigma^2} - \frac{(b_{0,i} - 0.5)^2}{2\sigma^2} \right] = \sum_{i=1}^{n} \left[\frac{\zeta_i(b_{0,i} - 0.5)}{\sigma^2} - \frac{1}{8\sigma^2} \right], \quad (14)$$

where $\zeta_i = z_i + 0.5 - \theta$, $b_{0,i} = \mathrm{LSB}(z_i)$ and under hypothesis $\mathcal{H}_1$ is

$$\log \widetilde{\Lambda}_1(Z_n) = \sum_{i=1}^{n} \left[\frac{\xi_i(b_{0,i} - 0.5)}{\sigma^2} + \frac{1}{8\sigma^2} \right], \quad (15)$$

where $\xi_i = Q_2[z_i] + 1 - \theta$ and $b_{0,i} = z_{s,i}$. Under hypothesis $\mathcal{H}_0$, the expectation of the approximate log LR is given by the following expression

$$m_0 = \mathbb{E}_0 \left[\log \widetilde{\Lambda}_1(z) \right] = -\frac{1}{8\sigma^2} + \frac{\varepsilon}{\sigma^2}, \quad (16)$$

where the coefficient ε defines the impact of the quantization. This coefficient is given by

$$\varepsilon = \mathbb{E}_0 \left[\zeta(b_0 - 0.5) \right] = \sum_{m=-\infty}^{\infty} \left[\Phi\left(\frac{2m+2-\theta}{\sigma} \right) - \Phi\left(\frac{2m+1-\theta}{\sigma} \right) \right] \frac{(2m+1.5-\theta)}{2}$$
$$- \sum_{m=-\infty}^{\infty} \left[\Phi\left(\frac{2m+1-\theta}{\sigma} \right) - \Phi\left(\frac{2m-\theta}{\sigma} \right) \right] \frac{(2m+0.5-\theta)}{2}. \quad (17)$$

Finally, the variance is given by

$$\sigma_0^2 = \mathrm{Var}_0 \left[\log \widetilde{\Lambda}_1(z) \right] = \frac{1}{\sigma^4} \left\{ \mathbb{E}_0 \left[\zeta^2 (b_0 - 0.5)^2 \right] - \left[\mathbb{E}_0 \left(\zeta(b_0 - 0.5) \right) \right]^2 \right\} = \frac{\mathbb{E}_0 \left[\zeta^2 \right] - 4\varepsilon^2}{4\sigma^4}, \quad (18)$$

where

$$\mathbb{E}_0 \left[\zeta^2 \right] = \sum_{m=-\infty}^{\infty} \left[\Phi\left(\frac{m+1-\theta}{\sigma} \right) - \Phi\left(\frac{m-\theta}{\sigma} \right) \right] (m+0.5-\theta)^2. \quad (19)$$

Under hypothesis $\mathcal{H}_1$, the expectation and variance of the approximate log LR are given by the following expressions

$$m_1 = \mathbb{E}_1 \left[\log \widetilde{\Lambda}_1(z) \right] = \frac{1}{8\sigma^2}, \quad (20)$$

$$\sigma_1^2 = \mathrm{Var}_1 \left[\log \widetilde{\Lambda}_1(z) \right] = \mathrm{Var}_1 \left[\frac{\xi(b_0 - 0.5)}{\sigma^2} \right] = \frac{1}{4\sigma^4} \mathbb{E}_1 \left[\xi^2 \right], \quad (21)$$

where

$$\mathbb{E}_1 \left[\xi^2 \right] = \sum_{m=-\infty}^{\infty} \left[\Phi\left(\frac{2m+2-\theta}{\sigma} \right) - \Phi\left(\frac{2m-\theta}{\sigma} \right) \right] (m+1-\theta)^2, \quad (22)$$

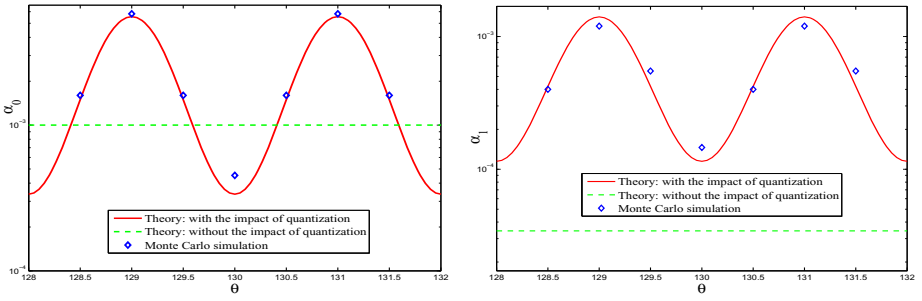


Fig. 1. The impact of the quantization on the probability of false alarm α_0 (left figure) and missed detection α_1 (right figure)

respectively. The following simplified equations can be proposed for the expectation and variance of the approximate log LR given by (12) without taking into account the impact of quantization

$$m_i = (-1)^{i+1}\frac{1}{8\sigma^2}, \quad \sigma_i^2 = \frac{1}{4\sigma^2}, \quad i = 0, 1. \tag{23}$$

Proposition 1. *Let us assume that the true embedding rate takes an arbitrary value* $\widetilde{R} \ : \ 0 < \widetilde{R} \leq 1$. *The power* β_{δ_1} *of the MP test (7) with the log LR* $\log \widetilde{\Lambda}_1(Z_n)$ *given by (12) can be approximated by*

$$\beta_{\delta_1} \simeq 1 - \Phi\left(\Phi^{-1}(1-\alpha_0)\frac{\sigma_0}{\sigma_{\widetilde{R}}} - \frac{(m_1 - m_0)\widetilde{R}\sqrt{n}}{\sigma_{\widetilde{R}}}\right) \tag{24}$$

for large n. *The expectations* m_i *and variance* σ_0^2 *are computed by using equations (16) - (22) (resp. (23)) with (resp. without) taking into account the impact of quantization. The variance* $\sigma_{\widetilde{R}}^2$ *is also computed with taking into account the impact of quantization*

$$\sigma_{\widetilde{R}}^2 = \frac{1}{4\sigma^2}\left[\left(\mathbb{E}_1\left[\xi^2\right] + \frac{1}{16}\right)\widetilde{R} + \left(\mathbb{E}_0\left[\zeta^2\right] + \frac{1}{16} - \varepsilon\right)(1-\widetilde{R})\right] - \left[m_1\widetilde{R} + m_0(1-\widetilde{R})\right]^2 \tag{25}$$

or without taking into account the impact of quantization

$$\sigma_{\widetilde{R}}^2 = \frac{1 + \widetilde{R} - \widetilde{R}^2}{4\sigma^2}. \tag{26}$$

It is worth noting that the explicit for form of the power function β_{δ_1} given in Proposition 1 conforms with the fact established in [10] that the "secure" steganographic capacity is proportional only to the square root of the number of covers n.

To illustrate the impact of the quantization, let us assume the following parameters of the Gaussian cover media model : $\widetilde{R} = 1$, $\theta \in [128; 132]$, $\sigma = 1$ and

$n = 200$. The comparison of theoretical equations for α_0 and α_1 with the Monte Carlo simulation (10^6 repetitions) are presented in Figure 1. The left figure shows the probability of false alarm α_0 calculated with (solid line) and without (dashed line) taking into account the impact of quantization. Here, the required probability of false alarm is $\alpha_0 = 10^{-3}$. First, the threshold h for the MP test $\delta_1(Z_n)$ given by (7) is computed by using equations (23) and (24). Next, the probability of false alarm $\alpha_0 = \alpha_0(h)$ is computed as a function of this threshold by using the corrected equations for the expectations and variances of the log LR, i.e. (16) - (21) and (24) (with taking into account the impact of quantization). The right figure shows the probability of missed detection α_1 calculated with (solid line) and without (dashed line) taking into account the impact of quantization for the prescribed significance level $\alpha_0 = 10^{-3}$. As it follows from Figure 1, the impact of quantization on the probability of false alarm α_0 and missed detection α_1 is significant.

4 An Unknown Embedding Rate: Two Composite Hypotheses

Let us assume that the previously defined distributions are known, but the embedding rate R is unknown. The following alternative composite hypotheses have to be tested by using n observations Z_n representing the cover media :

$$\mathcal{H}_0 = \{R \le r^*\} \text{ against } \mathcal{H}_1 = \{R > r^*\}, \tag{27}$$

where r^* denotes the "frontier" value of embedding rate separating $\mathcal{H}_0$ and $\mathcal{H}_1$. Hence, the LR (5) becomes

$$\Lambda_{R_0,R_1}(Z_n) = \prod_{i=1}^{n} \frac{R_1\Lambda_1(z_i) + (1-R_1)}{R_0\Lambda_1(z_i) + (1-R_0)}, \quad \Lambda_1(z_i) = \frac{Q_{Q_2}\left(Q_2[z_i]\right)}{2Q_{Q_1}\left(z_i\right)} \tag{28}$$

where $R_0 \le r^* < R_1$. The main difficulty is that the values of acceptable R_0 and unacceptable R_1 embedding rates are unknown. The ultimate challenge for anyone in the case of two composite hypotheses is to get a uniformly MP (UMP) test δ which maximises the power function

$$\beta(R) = 1 - \mathbb{P}_R(\delta(Z_n) = \mathcal{H}_0) \tag{29}$$

for any $R > r^*$ over the class

$$\mathcal{K}_{\alpha_0} = \left\{\delta : \sup_{R \le r^*} \mathbb{P}_R(\delta(Z_n) = \mathcal{H}_1) \le \alpha_0\right\} \tag{30}$$

The above mentioned hypothesis testing problem can be efficiently solved by a UMP test only if for any $R_0 < R_1$ the LR given by (28) is a monotonic function of a certain statistics $T = T(Z_n)$, see detailed description of UMP tests in [1,13]. Unfortunately, this is not the case for the LR given by (28) and, hence, the existence of a UMP test is compromised.

5 Local Asymptotic Approach

Let us continue the discussion of the case of random embedding. An efficient solution is based on the asymptotic local approach proposed by L. Le Cam [1,12,11,15]. The idea of this approach is that the "distance" between alternative hypotheses depends on the sample size n in such a way that the two hypotheses get closer to each other when n tends to infinity. By using an asymptotic expansion of the log LR, a particular hypothesis testing problem can be locally reduced to a relatively simple UMP hypothesis testing problem between two Gaussian scalar means [1,11,12,15]. This approach is applied to the following model

$$Z_n \sim Q_R = \prod_{i=1}^{n} R\frac{1}{2}Q_{Q_2}\left(Q_2[z_i]\right) + (1-R)Q_{Q_1}\left(z_i\right). \tag{31}$$

Let us consider two converging sequences of hypotheses $\mathcal{H}_j(n) = \{R \in \mathbb{R}_j(n)\}$ $(j = 0, 1)$. The sets $\mathbb{R}_j(n)$ are of the form $\mathbb{R}_j(n) = r^* + \frac{1}{\sqrt{n}}\delta_r$. If the Fisher information $\mathcal{F}(r)$ for the observation z_i is bounded and positively defined for any $R \in]0;1[$, the log LR

$$\log \Lambda_{r^*}\left(Z_n; \frac{\delta_r}{\sqrt{n}}\right) \stackrel{\text{def.}}{=} \log Q_{r^*+\frac{1}{\sqrt{n}}\delta_r}\left(Z_n\right) - \log Q_{r^*}\left(Z_n\right) \tag{32}$$

possesses the following asymptotic expansion (see details in [1,11,12,15]) :

$$\log\Lambda_{r^*}\left(Z_n; \frac{\delta_r}{\sqrt{n}}\right) \simeq \frac{\delta_r}{\sqrt{n}}\zeta_n(Z_n; r^*) - \frac{\delta_r^2\mathcal{F}(r^*)}{2},\ \zeta_n(Z_n; r^*) = \sum_{i=1}^{n} \left.\frac{\partial \log Q_R(z_i)}{\partial R}\right|_{R=r^*} \tag{33}$$

Moreover, the distribution of the efficient score weakly converges to the normal law

$$\mathcal{L}\left(\frac{1}{\sqrt{n}}\zeta_n(Z_n; r^*)\right) \underset{n\to\infty}{\rightsquigarrow} \begin{cases} \mathcal{N}(0, \mathcal{F}(r^*)) & \text{under } z_i \sim Q_{r^*} \\ \mathcal{N}(\mathcal{F}(r^*)\delta_r, \mathcal{F}(r^*)) & \text{under } z_i \sim Q_{r^*+\frac{\delta_r}{\sqrt{n}}} \end{cases} \tag{34}$$

It can be shown that the efficient score is given by

$$\zeta_n(Z_n; r^*) = \sum_{i=1}^{n} \zeta(z_i; r^*) = \sum_{i=1}^{n} \frac{\Lambda_1(z_i) - 1}{r^*\Lambda_1(z_i) + (1 - r^*)} \tag{35}$$

and the Fisher information $\mathcal{F}(R)$ is

$$\mathcal{F}(R) = \mathbb{E}_R\left[\frac{\Lambda_1(z) - 1}{R\Lambda_1(z) + (1-R)}\right]^2. \tag{36}$$

Therefore, the following decision rule

$$\delta_{r^*}(Z_n) = \begin{cases} \mathcal{H}_0 \text{ if } \zeta_n(Z_n; r^*) < h \\ \mathcal{H}_1 \text{ if } \zeta_n(Z_n; r^*) \geq h \end{cases}. \tag{37}$$

defines a local MP test designed to choose between two alternatives (27). The threshold h is the solution of the equation $\sup_{R \leq r^*} \mathbb{P}_R(\zeta_n(Z_n; r^*) \geq h) = \alpha_0$.

6 Tractable Algorithm and Its Relation with Known Steganalysers

6.1 Tractable Likelihood Ratio

As it follows from the previous sections, in the case of arbitrary embedding rate R, an optimal solution is based on the log LR given by (28) if R_0 and R_1 are known or on the efficient score given by (35) if they are unknown but the value r^* is known. It is easy to see that in both cases the useful information obtained from observations Z_n of cover media (with or without a secret message) is concentrated in $\Lambda_1(z)$ or equivalently in $\log \Lambda_1(z)$. Let us denote $y \stackrel{\text{def.}}{=} \zeta(z; r^*)$, hence

$$y = f(x; r^*) \stackrel{\text{def.}}{=} \frac{e^x - 1}{r^* e^x + 1 - r^*} \quad \text{with} \quad x \stackrel{\text{def.}}{=} \log \Lambda_1(z). \tag{38}$$

This function is represented in Figure 2 for different values of r^*. Typical densities of $x = \log \Lambda_1(z)$ under alternative hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are also shown in Figure 2 for the case of $\sigma = 1.5$. The asymptotic normality of $\zeta_n(Z_n; r^*) = \sum_{i=1}^n \zeta(z_i; r^*)$ is warranted due to Le Cam expansion (see equation (34)). Hence, it is sufficient to compute the expectations and variances of $f(\Lambda_1(z); r^*)$ under alternative hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. It follows from (14) that the efficient score for one observation is

$$y = g(\zeta_i, (b_{0,i} - 0.5)) \stackrel{\text{def.}}{=} f(x; r^*) \quad \text{with} \quad x = \frac{\zeta_i (b_{0,i} - 0.5)}{\sigma^2} - \frac{1}{8\sigma^2} \tag{39}$$

under $\mathcal{H}_0$ and, hence, two first moments ($k = 1, 2$) are given by

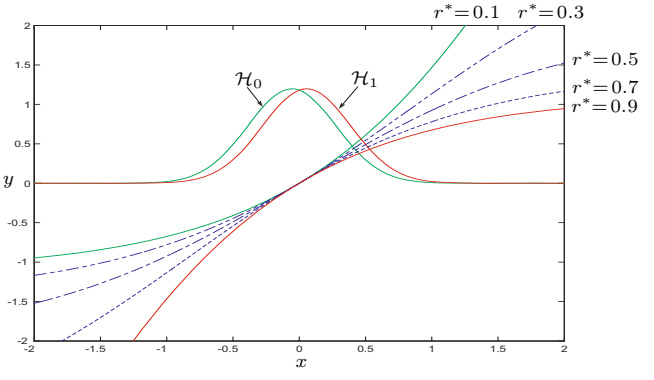


Fig. 2. The efficient score $y = f(x; r^*)$ as a function of $x = \log \Lambda_1(z)$ for $r^* = 0.1, 0.3, 0.5, 0.7, 0.9$

$$\mathbb{E}_0\left[g^k(\zeta_i,(b_{0,i}-0.5))\right]=\sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{2m+2-\theta}{\sigma}\right)-\Phi\left(\frac{2m+1-\theta}{\sigma}\right)\right]g^k\left(2m+\frac{3}{2}-\theta,\frac{1}{2}\right)$$
$$+\sum_{m=-\infty}^{\infty}\left[\Phi\left(\frac{2m+1-\theta}{\sigma}\right)-\Phi\left(\frac{2m-\theta}{\sigma}\right)\right]g^k\left(2m+\frac{1}{2}-\theta,-\frac{1}{2}\right). \quad (40)$$

It follows from (15) that the efficient score for one observation is

$$y=g(\xi_i,(b_{0,i}-0.5))\overset{\text{def.}}{=}f(x;r^*)\ \text{ with }\ x=\frac{\xi_i(b_{0,i}-0.5)}{\sigma^2}+\frac{1}{8\sigma^2} \quad (41)$$

under $\mathcal{H}_1$ and, hence, two first moments are given by

$$\mathbb{E}_1\left[g^k(\xi_i,(b_{0,i}-0.5))\right]=\frac{1}{2}\mathbb{E}_1\left[g^k(\xi_i,(b_{0,i}-0.5))|b_{0,i}=1\right]+$$
$$\frac{1}{2}\mathbb{E}_1\left[g^k(\xi_i,(b_{0,i}-0.5))|b_{0,i}=0\right]. \quad (42)$$

To compute the loss of optimality of the MP test based on $\log\Lambda_1(Z_n)$ given by (7) and designed for $R=1$ against the local MP test given by (37) with $r^*=0.05$ and the MP test based on $\log\Lambda_{\widetilde{R}}(Z_n)$ when the true embedding rate is $\widetilde{R}=0.1$, let us consider the following Gaussian cover media model : $\theta=129$,

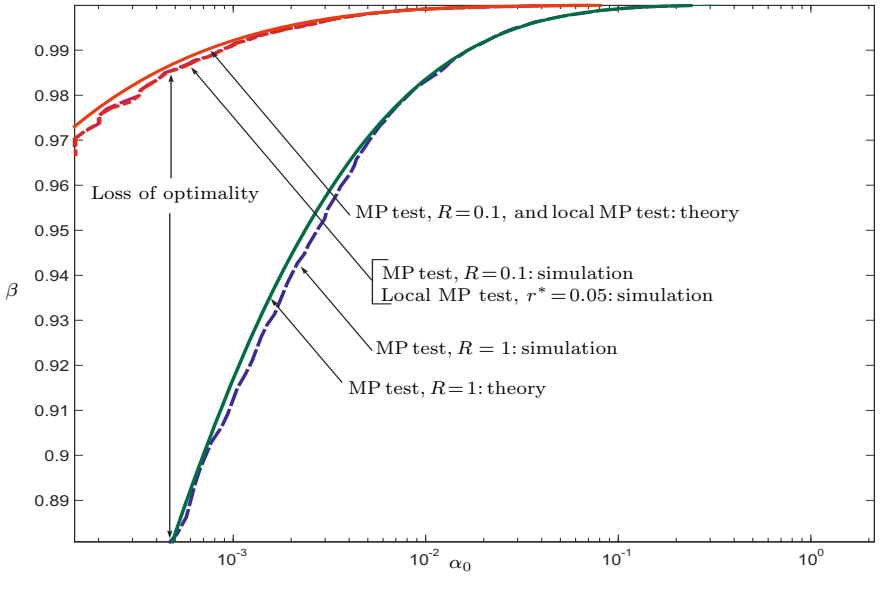


Fig. 3. The power function $\beta=\beta(\alpha_0)$ for the MP tests designed for $R=1$ and $R=0.1$ and for the local MP test designed for $r^*=0.05$: theory and simulation

$\sigma = 1$ and $n = 10^4$. The comparison of the theoretical power $\beta = \beta(\alpha_0)$ as a function of the false alarm rate α_0 for these tests with the Monte Carlo simulation (10^5 repetitions) are presented in Figure 3. These curves reflet the worst case situation, i.e. the augmentation of σ or $\widetilde{R}$ leads to the smaller difference between the above mentioned tests.

6.2 A More Realistic Model of Cover Media

As it follows from equation (24), the power β of an optimal steganalyser depends on the standard deviation σ of cover media pixels for a given rate of false alarm α_0. Hence, to increase the power β, someone has to reduce the standard deviation σ by using a parametric model of cover media. As it is motivated in the companion paper [2], the observation vector (pixels) extracted from the cover media file (digital image, for instance) by using a specially chosen segment or mask is characterized "block by block" by a regression model. Let us split the observation vector C in M statistically independent n dimensional sub-vectors C_j, i.e. $C^T = (C_1^T, \ldots, C_M^T)$. It is assumed that each segment C_j is approximated by :

$$C_j = Q_1[Y_j], \quad Y_j = Hx_j + \xi \sim \mathcal{N}(Hx_j, \sigma_j^2 I_n), \quad j = 1, \ldots, M, \tag{43}$$

where H is a known $[n \times l]$ full rank matrix, $n > l$, $x_j \in \mathbb{R}^l$ is a nuisance parameter (content of the image), I_n is an $(n \times n)$ identity matrix and σ_j^2 is the residual variance. The l columns of H span a column subspace $R(H)$ of the observation space $Y_j \in \mathbb{R}^n$. It is assumed that one column of H is obligatory formed of ones. Such a parametric model is an efficient method to reduce the standard deviation σ [2]. The new hypothesis testing problem with a parametric model of cover media consists in deciding between

$$\mathcal{H}_0 \ : \ Z = C = Q_1[Y], \tag{44}$$

and

$$\mathcal{H}_1\colon z_i = \begin{cases} Q_2[y_i] + z_{s,i} \text{ with probability } R \\ c_i = Q_1[y_i] \ \text{ with probability } 1-R \end{cases}, \ i = 1, \ldots, Mn, \tag{45}$$

where $Y^T = (Y_1^T, \ldots, Y_M^T)$, $Y_j \sim \mathcal{N}(Hx_j, \sigma_j^2 I_n)$. It follows from the previous subsection that the tractable log LR $\log \Lambda_1(Z_j)$ in the case parametric model can be re-written as follows :

$$\log \Lambda_1(Z_j) = -\frac{1}{2\sigma_j^2} \left\| Q_2[Z_j] - Hx_j + \mathbf{1}_n + \Delta_2 \right\|_2^2 + \frac{1}{2\sigma_j^2} \left\| Z_j - Hx_j + 0.5 \cdot \mathbf{1}_n + \Delta_1 \right\|_2^2, \tag{46}$$

where $\mathbf{1}_n$ is an n-dimensional vector composed of ones, Δ_j is an n-dimensional vector composed of corrective terms due to quantization $\delta_{j,i}$, $j = 1, 2$. The "approximate" log LR is given by

$$\log \Lambda_1(Z_j) \simeq -\frac{1}{2\sigma_j^2} \left\| Q_2[Z_j] - Hx_j + \mathbf{1}_n \right\|_2^2 + \frac{1}{2\sigma_j^2} \left\| Z_j - Hx_j + 0.5 \cdot \mathbf{1}_n \right\|_2^2. \tag{47}$$

In practice, x_j and σ_j^2 are unknown. The theoretical aspects of dealing with nuisance parameters in the framework of statistical decision theory is discussed in [1,13]. An efficient approach to this problem is based on the theory of invariance in statistics. The optimal invariant tests and their properties in the context of image processing have been designed and studied in [4,5,6,16].

Let us first assume that σ_j^2 is known. The nuisance parameter x_j can be estimated (or more exactly rejected) by using $Q_2[Z_j] = Q_2[Y_j]$ which is free from the embedded information. To reject the nuisance parameters, the theory of invariance is usually used in the case non-quantized observations. The detailed description of theoretical and practical aspects (together with all necessary proofs) how to use the invariance principle in the case of regression model can be found in [4,5,6,16]. The idea of the invariant hypotheses testing approach is based on the existence of the natural invariance of the detection problem with respect to a certain group of transformation. Let us note that the above mentioned hypotheses testing problem given by (44) - (45) remains "almost" invariant under the group of translations $G = \{g : g(Y) = Y + Hx\}$, $x \in \mathbb{R}^l$. The word "almost" is due to the quantization $Q_j[y]$, $j = 1, 2$. Without the quantization, the invariance will be exact. In such a case, the statistical decision should be based on a maximal invariant to the group of translations G, i.e. all invariant tests with respect to G are functions of a maximal invariant statistics (see the definition in [3]). It is shown that the projection $\varepsilon = W^T Y$ of Y onto the left null space $R(H)^{\perp}$ of the matrix H is a maximal invariant. The matrix $W = (w_1, \ldots, w_{n-l})$ of size $n \times (n - l)$ is composed of eigenvectors $w_1, \ldots, w_{n-l}$ of the projection matrix $P_H^{\perp} = I_n - H(H^T H)^{-1} H^T$ corresponding to eigenvalue 1. The matrix W satisfies the following conditions: $W^T H = 0$, $WW^T = P_H^{\perp}$ and $W^T W = I_{n-l}$. In practice, the nuisance parameter rejection is usually done by using the matrix $P_H^{\perp}$, because $P_H^{\perp} H = 0$. Moreover, if the matrix H is full rank, then the invariant test is equivalent to the generalized LR (or GLR) test. The "approximate" log GLR (or "almost" invariant) is given by

$$\begin{aligned} \log \widehat{\Lambda}_1(Z_j) &\simeq -\frac{1}{2\sigma_j^2} \left\| Q_2[Z_j] - H\widehat{x} + \mathbf{1}_n \right\|_2^2 + \frac{1}{2\sigma_j^2} \left\| Z_j - H\widehat{x} + 0.5 \cdot \mathbf{1}_n \right\|_2^2 \\ &= \frac{1}{\sigma_j^2} \left[P_H^{\perp} Q_2[Z_j] \right]^T [B_0 - 0.5 \cdot \mathbf{1}_n] + \frac{n}{8\sigma_j^2}, \end{aligned} \tag{48}$$

where $B_0 = (b_{0,1}, \ldots, b_{0,n})^T$ and $\widehat{x} = (H^T H)^{-1} H^T Q_2[Z_j]$ is the ML estimate of the nuisance parameter x.

Under hypothesis $\mathcal{H}_0$, the expectation and variance of the "approximate" log GLR for the total observation vector Y are given by the following expressions :

$$m_0 = \mathbb{E}_0 \left[\sum_{j=1}^{M} \log \widehat{\Lambda}_1(Z_j) \right] \simeq \frac{M(2l - n)}{8\overline{\sigma}^2} \quad \text{with} \quad \frac{1}{\overline{\sigma}^2} = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{\sigma_j^2} \tag{49}$$

and

$$\sigma_0^2 = \mathrm{Var}_0\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right] \simeq M(n-l)\left[\frac{1}{4\overline{\sigma}^2}+\frac{1}{16\overline{\sigma}^4}\right] \text{ with } \frac{1}{\overline{\sigma}^4}=\frac{1}{M}\sum_{j=1}^{M}\frac{1}{\sigma_j^4}. \quad (50)$$

Let us assume that the true embedding rate takes an arbitrary value $\widetilde{R}$: $0 < \widetilde{R} \leq 1$. Under hypothesis $\mathcal{H}_1$ with the true embedding rate $\widetilde{R}$, the expectation and variance of the "approximate" log GLR for the total observation vector Y are given by the following expressions :

$$m_{\widetilde{R}} = \mathbb{E}_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right] \simeq \frac{M(2l-n+2\widetilde{R}(n-l))}{8\overline{\sigma}^2} \quad (51)$$

and

$$\sigma_{\widetilde{R}}^2 = \mathrm{Var}_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right] \simeq \frac{M(n-l)}{4\overline{\sigma}^2}+\frac{M(n-l)(1-R)^2}{16\overline{\sigma}^4} \quad (52)$$

Proposition 2. *Let us assume that the Lindeberg's condition imposed on the log LR* $\log\widehat{\Lambda}_1(Z_j)$ *is satisfied. It follows from the central limit theorem that the following fraction*

$$\frac{\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)-\mathbb{E}_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right]}{\sqrt{\mathrm{Var}_{\widetilde{R}}\left[\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)\right]}} \underset{M\to\infty}{\rightsquigarrow} \mathcal{N}(0,1) \quad (53)$$

weakly converges to the standard normal distribution [17]. For large M*, the power* β_{δ_1} *of the test (7) with the log LR* $\sum_{j=1}^{M}\log\widehat{\Lambda}_1(Z_j)$ *given by (48) can be approximated*

$$\beta_{\delta_1} \simeq 1-\Phi\left(\Phi^{-1}(1-\alpha_0)\frac{\sigma_0}{\sigma_{\widetilde{R}}}-\frac{(m_{\widetilde{R}}-m_0)}{\sigma_{\widetilde{R}}}\right) \quad (54)$$

where m_0*,* $m_{\widetilde{R}}$*,* σ_0 *and* $\sigma_{\widetilde{R}}$ *are calculated by using equations (49) - (52).*

If the residual variance σ_j^2 is unknown, then the following GLR is used

$$\log\widehat{\Lambda}_1(Z_j) \simeq \frac{1}{\widehat{\sigma}_j^2}\left[P_H^{\perp}Q_2[Z_j]\right]^T\left[B_0-0.5\cdot\mathbf{1}_n\right]+\frac{n}{8\widehat{\sigma}_j^2}, \quad (55)$$

where $\widehat{\sigma}_j^2 = \frac{1}{n-l}\left\|P_H^{\perp}Q_2[Z_j]\right\|_2^2$.

The first right hand side term in equation (48) defines the sensitivity of the test because the second right hand side term $\frac{n}{8\widehat{\sigma}^2}$ does not depend on the embedded secret message. Nevertheless, the second right hand side term $\frac{n}{8\widehat{\sigma}^2}$ of (48) is also necessary to correctly calculate the threshold h in (7) by using the equation

$\mathbb{P}_0(\sum_{j=1}^{M} \log \widehat{\Lambda}_1(Z_j) \geq h) = \alpha_0$. The first right hand side term in equation (48) represents an inner product of the vector of "residuals" $\varepsilon = P_H^{\perp} Q_2[Z_j]$, i.e. the vector of projection of $Q_2[Z_j]$ on the orthogonal complement $R(H)^{\perp}$ of the column space $R(H)$, and the vector $[B_0 - 0.5 \cdot \mathbf{1}_n]$ composed of $\mathrm{LSB}(z_i) - 0.5$:

$$\frac{1}{\widehat{\sigma}^2}\left[P_H^{\perp} Q_2[Z_j]\right]^T [B_0 - 0.5 \cdot \mathbf{1}_n] = \sum_{i=1}^{n} \overbrace{\widehat{\sigma}^{-2}}^{=\text{"weight"}} \cdot \overbrace{(Q_2[z_i] - (H\widehat{x}_j)_i + 1)}^{=\text{"residual" } \varepsilon_i} \cdot \overbrace{(b_{0,i} - 0.5)}^{=\mathrm{LSB}(z_i)-0.5}, \quad (56)$$

where $(H\widehat{x}_j)_i$ is the i-th row of the vector $H\widehat{x}_j$. Let us now compare the last equation with the recently developed steganalysers [7,8,9]. These steganalysers are based on the following statistics [9] :

$$\sum_{i=1}^{n} \overbrace{w_i}^{=\text{"weight"}} \cdot \overbrace{(z_i - \mathcal{F}(z)_i)}^{=\text{"residual" } \varepsilon_i} \cdot \overbrace{(z_i - \overline{z}_i)}^{=2\cdot(\mathrm{LSB}(z_i)-0.5)}, \quad (57)$$

where $\mathcal{F}(s)$ denotes a "filter" dedicated to estimate the cover-image by filtering the stego-image, the weight w_i is chosen as $\frac{1}{1+\sigma_i^2}$, σ_i^2 is the "local" variance and $\overline{z}_i$ denotes the nonnegative integer z_i with the LSB flipped. As it follows from equations (56) - (57), the steganalysers developed in [7,8,9] coincide with the first term of the tractable log GLR (48).

7 Conclusions

The problem of hidden information detection has been addressed from a statistical point of view. Two new phenomena have been studied : *i)* the impact of observation quantization on the probabilities of false alarm and non detection; *ii)* the benefits from using a parametric statistical model of cover media. Some (almost) optimal statistical solutions have been designed and studied to solve the problem of hidden information detection. These solutions have been theoretically compared with the WS steganalysers algorithm recently developed [7,8,9]. Based on these theoretical findings, an efficient parametric model and hidden information detection algorithms have been developed and tested in the companion paper [2].

References

1. Borovkov, A.A.: Mathematical Statistics. Gordon and Breach Sciences Publishers, Amsterdam (1998)
2. Cogranne, R., Zitzmann, C., Fillatre, L., Retraint, F., Nikiforov, I., Cornu, P.: A cover image model for reliable steganalysis. In: Filler, T., et al. (eds.) IH 2011. LNCS, vol. 6958, pp. 178–192. Springer, Heidelberg (2011)
3. Ferguson, T.: Mathematical Statistics: A Decision Theoretic Approach. Academic Press, London (1967)

4. Fillatre, L., Nikiforov, I.: A statistical detection of an anomaly from a few noisy tomographic projections. Journal of Applied Signal Processing, Special issue on Advances in Intelligent Vision Systems: Methods and Applications-Part II 14, 2215–2228 (2005)
5. Fillatre, L., Nikiforov, I.: Non-bayesian detection and detectability of anomalies from a few noisy tomographic projections. IEEE Trans. Signal Processing 55(2), 401–413 (2007)
6. Fillatre, L., Nikiforov, I., Retraint, F.: ε-optimal non-bayesian anomaly detection for parametric tomography. IEEE Transactions on Image Processing 17(11), 1985–1999 (2008)
7. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Proc. of SPIE, pp. 23–34. Addison-Wesley, Reading (2004)
8. Ker, A.D.: Locating steganographic payload via WS residuals. In: Proceedings of the 10th ACM Workshop on Multimedia and Security, Oxford, September 22-23, pp. 27–31 (2008)
9. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Delp, E.J., Wong, P.W. (eds.) Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA, January 27-31, vol. 6819, pp. 51 – 517 (2008)
10. Ker, A.D., Pevný, T., Kodovský, J., Fridrich, J.: The square root law of steganographic capacity. In: MM&Sec 2008: Proceedings of the 10th ACM Workshop on Multimedia and Security, pp. 107–116. ACM, New York (2008), http://doi.acm.org/10.1145/1411328.1411349
11. Le Cam, L.: Asymptotic Methods in Statistical Decision Theory. Series in Statistics. Springer, New York (1986)
12. Le Cam, L., Yang, G.L.: Asymptotics in Statistics. Springer, Heidelberg (1990)
13. Lehmann, E.: Testing Statistical Hypotheses, 2nd edn. Chapman & Hall, Boca Raton (1986)
14. Dabeer, O., Sullivan, K., Madhow, U., Chandrasekaran, S.: Detection of hiding in the least significant bit. IEEE Trans. Signal Processing 52(10), 3047–3058 (2004)
15. Roussas, G.G.: Contiguity of Probability Measures, Some Applications in Statistics. Cambridge University Press, Mass (1972)
16. Scharf, L., Friedlander, B.: Matched subspace detectors. IEEE Trans. Signal Processing 42(8), 2146–2157 (1994)
17. Shiryaev, A.N.: Probability, 2nd edn. Springer, New York (1996)

A Cover Image Model For Reliable Steganalysis

Rémi Cogranne, Cathel Zitzmann, Lionel Fillatre, Florent Retraint,
Igor Nikiforov, 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 reliable steganalysis of natural cover images using a local non-linear parametric model. In the framework of hypothesis testing theory, the use of the model permits to warrant predictable results under false alarm constraint.

1 Introduction

Information hiding has received an increasing interest in the last decades driven by the numerous possible applications such as watermark identification and tampering detection. Unfortunately malicious usage of information hiding, like steganography, have also emerged [8]. Steganography and steganalysis are a cat and mouse game : steganographers embed a secret message in a cover medium while steganalysts try to detect the presence of this hidden message. This paper focuses on the simple but popular LSB replacement. Surely, much better algorithms are nowadays available. However, the proposed methodology can be applied to other schemes providing that a statistical model of steganographic impact is available.

With many tools available in the public domain, steganography is within reach of anyone, for legitimate or malicious usage. It is thus crucial for security forces to reliably detect steganographic content among a set of media ; many methods have been proposed for this purpose, see [2,5]. Even though some steganalyzers are very efficient (the BOSS contest [4] is a good example), detection rate is not the only performance criterion is some circumstances. For instance, when carrying an investigation, steganalysis results will hardly be accepted without an analytically predictable and warranted false alarm rate. In this situation, supervised learning based method can hardly be used. This justifies the statistical study of a steganography detection scheme to which this paper is devoted.

The study of steganalysis as a hypothesis test requires an accurate image model ; only few works in the literature explicitly use such models. In [16,19], the distribution of Discrete Cosine Transform (DCT) coefficients is used to detect steganography in JPG images. In a similar fashion, the distribution of Discrete Fourier Transform (DFT) coefficients is used in [20]. An independent and identically distributed (i.i.d) pixels model is exploited in [6] to derive a statistical

* This work is supported by French National Agency (ANR) through ANR-CSOSG Program (Project ANR-07-SECU-004).

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 178–192, 2011.

hypothesis test. None of the previously cited image models used for steganalysis offers a precise statistical description of image noise and an accurate model of image content. This observation highlights the fundamental motivation of current paper: filling the gap between the physical model of cover-image and steganalysis. In the framework of hypothesis testing, an accurate image model is fundamental to design a test which can warrant a predictable false alarm rate.

Unfortunately, modelling such a complex signal as image remains an open problem as well as designing an optimal steganalysis, even in ideal context of a known image(see the companion paper [22]). The goal of this paper is threefold:

- to locally model the content of a cover image by describing the optical system which gives birth to a natural image;
- to exploit, as simply as possible, this model of natural image in the design of an almost optimal test.
- to numerically compare the proposed detection scheme with other steganalysis methods.

2 Overview of Proposed Methodology

This section presents the main contributions of the paper and describes the organization of the paper. The goal is to give a complete overview of the proposed work and to relate it to the well-known WS detector investigated successively by Fridrich [7] and Ker [12,11].

2.1 Main Contributions

This paper assumes that a cover image is a matrix $\mathbf{C}=\{c_{l,m}\}$ of $L \times M$ grayscale value pixels and that the corresponding stego-image $\mathbf{Z}=\{z_{l,m}\}$ is created by replacing the LSBs of proportion R of the cover pixels. The set of grayscale levels is denoted $\mathcal{Y}=\{0,\ldots,2^b-1\}$; b bits are used to quantize each pixel. As explained in the companion paper [22], a cover pixel $c_{l,m}$ satisfies

$$c_{l,m} = Q_1[y_{l,m}], \tag{1}$$

where $y_{l,m}$ denotes the real value recorded by the digital camera before the quantization and $Q_1[y_i] = \lfloor y_i \rfloor$ is the operation of uniform quantization (integer part of y_i). Some important details on the quantization are given in the companion paper [22]. It is assumed (see Section 4) that

$$y_{l,m} = \theta_{l,m} + \xi_{l,m}$$

where $\theta_{l,m}$ is the mathematical expectation of $y_{l,m}$ and $\xi_{l,m}$ is a zero mean Gaussian random variable. The variance $\sigma^2_{l,m}$ of $\xi_{l,m}$ is assumed to be known for all (l,m). The matrix of parameters $\theta_{l,m}$ is denoted $\boldsymbol{\theta}$. It is supposed that $\boldsymbol{\theta} \in \Theta$ where Θ is a known compact set.

In the following, the notation $\overline{z}$ indicates the integer z with LSB flipped [7], i.e., $\overline{z} = z + 1 - 2\,\mathrm{lsb}(z)$ where $\mathrm{lsb}(z)$ is the LSB of z. The first step of the

proposed detection algorithm is to estimate the cover image parameters $\theta_{l,m}$; the notation $\widehat{\boldsymbol{\theta}}=\{\widehat{\theta}_{l,m}\}$ is used for the estimate of $\boldsymbol{\theta} = \{\theta_{l,m}\}$ obtained from the analyzed image $\mathbf{Z}$. The second step consists in calculating the matrix of residuals

$$\widehat{r}_{l,m} = (z_{l,m} - \widehat{\theta}_{l,m})(z_{l,m} - \overline{z}_{l,m}) \tag{2}$$

which indicate the difference between stego-image and estimated cover, with the sign adjusted to take into account the asymmetry in LSB replacement (even pixels could only be incremented, and odd pixels decremented).Then the proposed test is based on a decision function with the following form:

$$\Lambda(\mathbf{Z}) = \sum_{l=1}^{L} \sum_{m=1}^{M} w_{l,m} \widehat{r}_{l,m} \tag{3}$$

where $w_{l,m}$ is a weight so that the influence of pixels depends on their noise level. Noisy areas, for which estimation of the cover is more difficult, are given less weight than those in flatter areas.

This detector is quite similar to the Weighted Stego-image (WS) analysis initially proposed by [7] to estimate the payload size and deeply studied by [12]. Used as a detector, which is the focus of this paper, the WS is known to have good performance. Contrary to the approach followed by [7,12], this paper proposes two major novelties. First, the test is derived from the statistical theory of hypotheses testing (see Section 3). Hence, the weights $w_{l,m}$ are theoretically established and not empirically chosen. Second, the estimates $\widehat{c}_{l,m}$ of the cover pixels initially used by [7,12] are replaced with the estimates $\widehat{\theta}_{l,m}$, see (2), of the physical parameters describing the cover image content. From this way, it is expected to reach a higher level of performance.

The proposed methodology is summed up in Fig. 1. This approach leads to a reliable steganalysis because of two main advantages. First, the decision is made independently from the image content as it is explicitly taken into account as a "nuisance parameter". Second, the performance of the test is clearly established; this allows to meet a false alarm rate constraint by fixing a precalculated decision threshold and to know in advance the power detection of the test with respect to the insertion rate.

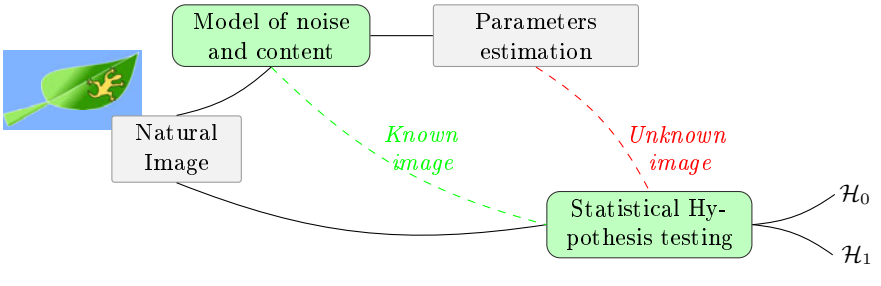


Fig. 1. Working diagram of proposed applied test

2.2 Organization of the Paper

Section 3 recalls the main theoretical results of the companion paper [22]. This section presents the design of the optimal test, namely the Likelihood Ratio Test (LRT), solving the problem of LSB replacement detection whatever the message length (or insertion rate) is, when quantization corrective term is negligible. The performance of the optimal LRT is asymptotically given in a closed form as the number of pixel grows to infinity. This analytic expression of optimal test power function can be used as an optimal upper bound for any practical test in agreement with the square root law of steganographic capacity, see [13].

In [12], the author uses an empirical weighted autoregressive model to estimate pixels'variance and value; the weighted coefficients are theoretically calculated in [22]. To get good detection performance, it is underlined by [12] that cover image is fundamental. Hence, this paper proposes an accurate model of natural images which takes account of the content $\theta_{l,m}$, including specifically non-stationarities and non-linearities. This content is usually not affected by the steganographic content and hence can be estimated in a stego-image as well. The general idea is to model the redundancies which locally exist between pixels. To build such an image model, the physical process that gives birth to a digital image is examined, and a generic model of digital imaging system along with a model of the scene are presented in Section 4.

The non-linear model of the cover image is used in Section 5 to design a statistical test for the practical case of unknown cover image content parameters $\theta_{l,m}$. The image non-linear model is "linearized" to allow a simple but yet efficient estimation. The effects of both estimation and linearization on the test performance are analyzed and the loss of optimality of the test, with respect to the ideal LRT, is bounded.

Finally, Section 6 presents numerical results. The proposed test is applied to some natural images. It is shown that for small false alarm rate, the test outperforms the five detectors used for comparison. On the contrary, the revisited WS exhibits slightly better performance for higher false alarm rate. Numerical results are presented to explain and discuss this point.

3 Optimal Statistical Test for Known Cover Image

The main results of the theoretical analysis proposed in the companion paper [22] is the design of the optimal LRT solving the problem of LSB replacement detection when the parameters $\theta_{l,m}$ of the cover image are known. Neglecting the quantization impact, this test is optimal whatever the message length (or insertion rate) is. The performance of the optimal LRT is asymptotically given in a closed form as the number of pixels grows to infinity.

3.1 Statistical Analysis of LSB Replacement Steganography

The probability mass function (pmf) of the pixel $z_{l,m}$ from a natural cover image is given as:

$$Q_{Q_1}(\theta_{l,m}) = [q_0(\theta_{l,m}), \ldots, q_{2^b-1}(\theta_{l,m})]$$

where, $\forall k \in \mathcal{Y}$,

$$q_k(\theta_{l,m}) = \frac{1}{\sigma_{l,m}} \int_{(k-\frac{1}{2})}^{(k+\frac{1}{2})} \phi\left(\frac{x-\theta_{l,m}}{\sigma_{l,m}}\right) dx \quad \text{with} \quad \phi(u)=\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right). \tag{4}$$

Let the insertion rate R be defined as the number of hidden bits per pixel. The steganographic process modifies in a known way the distribution $Q_{Q_1}(\theta_{l,m})$. In these conditions, a short calculation [9,6,22] shows that the pmf of pixel $z_{l,m}$ after insertion is given by $Q^R_{Q_1}(\theta_{l,m}) = [q^R_0(\theta_{l,m}), \cdots, q^R_{2^b-1}(\theta_{l,m})]$ where

$$\forall k \in \mathcal{Y}\,,\; q^R_k(\theta_{l,m}) = \left(1-\frac{R}{2}\right) q_k(\theta_{l,m}) + \frac{R}{2} q_{\overline{k}}(\theta_{l,m}). \tag{5}$$

As explained in [22, Eq.(27)], the hypothesis testing problem of steganalysis consists in choosing between $\mathcal{H}_0 = \{R \leq r^*\}$ vs $\mathcal{H}_1 = \{R > r^*\}$ or equivalently:

$$\begin{cases} \mathcal{H}_0 = \{z_{l,m} \sim Q^R_{Q_1}(\theta_{l,m})\,,\; l \in 1,\ldots,L, \forall m = 1\ldots,M,\; \forall R \leq r^*\} \\ \mathcal{H}_1 = \{z_{l,m} \sim Q^R_{Q_1}(\theta_{l,m})\,,\; l \in 1,\ldots,L, \forall m = 1\ldots,M,\; \forall R > r^*\} \end{cases} \tag{6}$$

where r^* is a (reasonable) minimal insertion rate. The goal is to find a test $\delta : \mathcal{Y}^{L\cdot M} \mapsto \{\mathcal{H}_0; \mathcal{H}_1\}$ such that hypothesis $\mathcal{H}_i$ is accepted if $\delta(\mathbf{Z}) = \mathcal{H}_i$ (see [14] for complete information). Let

$$\mathcal{K}_{\alpha_0} = \left\{ \delta : \sup_{\boldsymbol{\theta}\in\Theta,\; R<r^*} \mathbb{P}_{\boldsymbol{\theta},R}(\delta(\mathbf{Z}) = \mathcal{H}_1) \leq \alpha_0 \right\}$$

be the class of tests with an upper-bounded false alarm probability α_0. Here $\mathbb{P}_{\boldsymbol{\theta},R}(A)$ stands for the probability of the event A when $z_{l,m}$ is generated by $Q^R_{Q_1}(\theta_{l,m})$ for all (l,m). The power function β_δ of the test δ is defined by the probability of hidden bits detection

$$\beta_\delta(\boldsymbol{\theta}, R) = \mathbb{P}_{\boldsymbol{\theta},R}(\delta(\mathbf{Z}) = \mathcal{H}_1).$$

3.2 Optimal Theoretical LRT

For theoretical convenience, let the mean variance $\bar{\sigma}$ be defined by

$$\frac{1}{\bar{\sigma}^2} = \frac{1}{L\,M} \sum_{l=1}^{L} \sum_{m=1}^{M} \frac{1}{\sigma^2_{l,m}}. \tag{7}$$

When $\theta_{l,m}$ is known for all (l,m), the optimal solution, namely the LRT, is given in the companion paper (cf. [22, section 6]). For large $\bar{\sigma}$, this test is given as:

$$\delta(\mathbf{Z}) = \begin{cases} \mathcal{H}_0 & \text{if } \; \Lambda(\mathbf{Z}) < \tau_{\alpha_0}, \\ \mathcal{H}_1 & \text{if } \; \Lambda(\mathbf{Z}) \geq \tau_{\alpha_0}, \end{cases} \tag{8}$$

where

$$\Lambda(\mathbf{Z}) = \sum_{l=1}^{L} \sum_{m=1}^{M} w_{l,m}(z_{l,m} - \theta_{l,m})(z_{l,m} - \overline{z}_{l,m})\,, \quad w_{l,m} = \frac{\bar{\sigma}}{\sigma^2_{l,m}\sqrt{L\,M}}. \tag{9}$$

The threshold τ_{α_0} is chosen such that $\delta \in \mathcal{K}_{\alpha_0}$. The following theorem is easily derived from the central limit theorem applied to $\Lambda(\mathbf{Z})$ for $r^\star \approx 0$.

Theorem 1. *In virtue of the Lindeberg's central limit theorem [14]:*

$$\begin{cases} \Lambda(\mathbf{Z}) \xrightarrow{d} \mathcal{N}(0\,;\,1) & \text{under } \mathcal{H}_0 \\ \Lambda(\mathbf{Z}) \xrightarrow{d} \mathcal{N}\left(\frac{R}{2\bar{\sigma}}\sqrt{LM}\,;\,1\right) & \text{under } \mathcal{H}_1 \end{cases} \tag{10}$$

with $\xrightarrow{d}$ *the convergence in distribution as* $L\,M \to \infty$. *Choosing* $\tau_{\alpha_0} = \Phi^{-1}(1-\alpha_0)$, *it follows that* $\delta(\mathbf{Z}) \in \mathcal{K}_{\alpha_0}$ *and*

$$\beta_\delta(\boldsymbol{\theta}, R) = 1 - \Phi\left(\tau_{\alpha_0} - \frac{R\sqrt{LM}}{2\bar{\sigma}}\right). \tag{11}$$

In the companion paper [22], the quantization impact lead us to design a local most powerful (LMP) test for R in the neighborhood of r^*. On the contrary, the decision $\Lambda(\mathbf{Z})$ defined in (9), does not depend on R. Hence, the main conclusion of the above results it that in $\mathcal{K}_{\alpha_0}$, the test $\delta(\mathbf{Z})$ is uniformly most powerful (UMP) with respect to R, provided that the quantization is negligible. The power function $\beta_\delta(\boldsymbol{\theta}, R)$ has been established only for small R, but is shown to be meaningful in practice for higher insertion rate, see Fig. 3 and [22, Fig. 3].Finally, Theorem 1 asymptotically gives an explicit form of $\beta_\delta(\boldsymbol{\theta}, R)$ in agreement with the square root law of steganographic capacity [13] and independently from the cover content parameters $\boldsymbol{\theta}$. The function $\beta_\delta(\boldsymbol{\theta}, R)$ can be used as an upper bound for the power of any test.

4 Natural Cover Image Model

In practice, the cover image parameters $\theta_{l,m}$ are not known. Estimating these parameters is crucial for any detection algorithm. To this end a physical local model of raw images (*i.e.* without in-camera processing) content is proposed. This model will be used to simply and efficiently estimate image content.

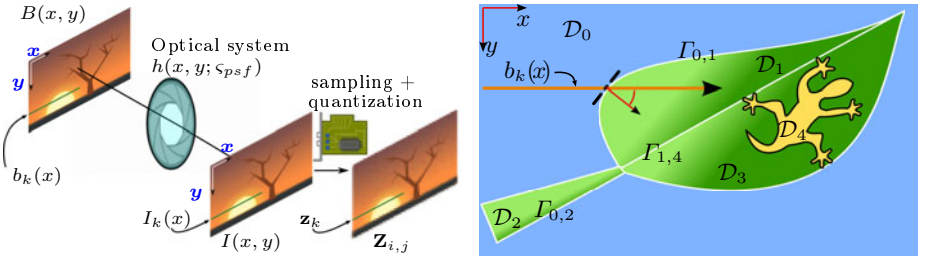
4.1 Model of the Imaged Scene

A scene is described from camera point of view by its emitted radiance for several color channels. Without any loss of generality, this section deals with grayscale image (colors are processed individually). Hence, the radiance of the scene is a function $B(x, y)$ where $(x, y) \in [0, x_{\max}] \times [0, y_{\max}] = \mathcal{D} \subset \mathbb{R}^2$ are the imaged scene coordinates scale to sensor.

As described in [15], see fig. 2b, a scene is made of various solid objects O_i associated each with the open domain $\mathcal{D}_i \subset \mathcal{D}$ and the radiance function $B_i(x, y)$ such that $(x, y) \notin \mathcal{D}_i \Rightarrow B_i(x, y) = 0$. The boundaries between domains $\mathcal{D}_i$ and $\mathcal{D}_j$ is a curve $\Gamma_{i,j} : [0, 1] \to \mathcal{D}$. The whole scene is consequently described as:

$$B(x, y) = \sum_i B_i(x, y). \tag{12}$$

From the properties of solid objects light emission, one can expect that [15]:



(a) A general image acquisition pipeline. (b) A 2D scene and its 1D model.

Fig. 2. Describing the scene and the image acquisition to model a natural image

1. for each object O_i, the radiance $B_i(x,y)$ is a smooth function over $\mathcal{D}_i$,
2. the radiance $B(x,y)$ is discontinuous across (most of) the boundaries $\Gamma_{i,j}$,
3. the objects are of regular shape in the sense that each curve $\Gamma_{i,j}(t)$ is twice continuously differentiable for almost all $t \in [0;1]$.

4.2 Raw Pixel Recorded Value

The fundamental model used in this paper relies on the above scene description. For a simpler yet efficient estimation of image content, it is proposed to adapt the model in one dimension (1D). Hence, the scene is divided in K segments $\mathcal{X}_k$ of negligible width, associated with the radiance $b_k(x), k \in \{1,\dots,K\}$, see fig. 2. The variable y is (almost) constant over $\mathcal{X}_k$, hence it is omitted to simplify the notation. According to the above mentioned properties, the univariate function b_k is continuous except at the boundaries and hence admits the following decomposition [1]:

$$\forall x \in \mathcal{X}_k,\ b_k(x) = b_k^{(c)}(x) + b_k^{(s)}(x)$$

where $b_k^{(c)}$ is continuous and the singular part, $b_k^{(s)}$, is piecewise constant and can be written:

$$\forall x \in \mathcal{X}_k\ ,\ b_k^{(s)}(x) = \sum_{d=1}^{r_k} u_{k,d}\, \mathbf{1}(x - t_{k,d}) \tag{13}$$

with r_k the number of discontinuities in the k-th segment $\mathcal{X}_k$, $\mathbf{1}(\cdot)$ the unitary step function defined as $\mathbf{1}(x)=1$ if $x>0$ and $\mathbf{1}(x)=0$ otherwise. The parameters $u_{k,d}$ and $t_{k,d}$ are respectively the intensity and the location of d-th discontinuity.

The optical system is modelled by the Point Spread Function (PSF) $h(x,y)$ which characterizes the optical system [10]. The PSF depends on many unknown elements (lens, focal length, atmosphere, ...) among which some are spatially variant (aberration, out-of-focus, ...). This paper is restricted to an isotropic Gaussian PSF:

$$h(x,y;\varsigma_{\text{psf}}) = \frac{1}{\varsigma_{\text{psf}}^2}\varphi\left(\frac{\rho}{\varsigma_{\text{psf}}}\right) \quad \text{where} \quad \varphi(\rho) = \frac{1}{2\pi}\exp\left(-\frac{\rho^2}{2}\right),$$

$\rho^2 = x^2 + y^2$ and $\varsigma_{\text{psf}} > 0$ is the blur parameter. Hence, the irradiance I_k reaching $\mathcal{X}_k$, the k-th sensor segment, can be written as:

$$I_k(x) = I_k^{(c)}(x) + I_k^{(s)}(x) \tag{14}$$

where $I_k^{(c)}$ and $I_k^{(s)}$ correspond respectively to the continuous and singular part of radiance b_k. They result from the convolution between $b_k(x)$ and $h(x, y; \varsigma_{\text{psf}})$ restricted to $\mathcal{X}_k$. The function $I_k^{(c)}$ is expected to be very smooth and hence is assumed to be well approximated by algebraic polynomial of degree $p-1$:

$$I_k^{(c)}(x) = \sum_{i=0}^{p-1} s_i^{(k)} x^i \ , \ \forall x \in \mathcal{X}_k \tag{15}$$

where the real coefficients $s_i^{(k)}$, which depend on $\mathcal{X}_k$, are unknown. The goal is here to highlight crucial importance of discontinuities and subsequent difficulties it raised. That is why, a rather simple algebraic polynomials was used to model for the continuous part $I_k^{(c)}$ which is not the main focus. The function $I_k^{(s)}$ accounts for the discontinuities and should obviously be accurately modeled to later accurately estimate the function $I_k(x)$ from a raw image. After some algebra (omitted due space limitations) the functions $I_k^{(s)}$ admits the decomposition:

$$I_k^{(s)}(x) = \sum_{d=1}^{r_k} u_{k,d}\, \Phi\left(\frac{x - t_{k,d}}{\varsigma_{k,d}}\right) \quad \text{where} \quad \Phi(u) = \int_{-\infty}^{u} \phi(\nu)\, d\nu \tag{16}$$

and $\phi(\cdot)$ is the Gaussian distribution function as defined in (4). Note that the local blur parameter $\varsigma_{k,d} > \varsigma_{\text{psf}}$ in (16) varies for each discontinuity due to the angle between the local 2D discontinuity orientation and the segment $\mathcal{X}_k$, see Fig.2b.

Finally, the irradiance $I_k(x)$ is integrated over sensor pixels of the Charge-Coupled Device (CCD) matrix. Hence, the intensity value $y_{l,m}$ recorded by the sensor at position (l, m) over the CCD matrix is given by

$$y_{l,m} = \theta_{l,m} + \xi_{l,m} \tag{17}$$

where $\xi_{l,m}$ is a Gaussian random noise representing all the noises corrupting the expected signal $\theta_{l,m}$ given by

$$\theta_{l,m} = \sum_{i=0}^{p-1} s_i^{(k)} x_{l,m}^i + \sum_{d=1}^{r_k} u_{k,d} \int_{\mathcal{X}_k \cap \mathcal{C}_{l,m}} \Phi\left(\frac{\nu - t_{k,d}}{\varsigma_{k,d}}\right) d\nu \tag{18}$$

where $x_{l,m}$ is the coordinate of the sensor's center at position (l, m), assuming that $x_{l,m} \in \mathcal{X}_k$ and that the sensor at position (l, m) records all the photons arriving on the square region $\mathcal{C}_{l,m} \subset \mathcal{D}$.

The methodology presented in this paper relies on the model (17)-(18) of a raw pixel intensity $y_{l,m}$. The image analyzed in Section 6.2 have been subjected to in-camera processing (demosaicing, gamma correction, white balance, etc. . .) which can not be modeled accurately as it remains partially unknown. However, one can expect that the content model is not strongly modified by post-acquisition processing as it is empirically shown in numerical results presented in section 6.

5 From Cover Image Estimation to Almost Optimal Steganalysis

In this section, it is proposed to use the optimal LRT, previously defined in section 3.2, by replacing the parameters $\theta_{l,m}$ by some estimates $\widehat{\theta}_{l,m}$. This estimation is based on the model (18) and the analyzed image grayscale values $z_{l,m}$. The main difficulty is that the unknown parameters $(t_{k,d}, \varsigma_{k,d})$ for all (k,d) appear in a non-linear way in (18).

5.1 Estimation of the Cover Image Parameters

The goal is to estimate the unknown parameters $\theta_{l,m}$. The model (18) shows that there exist some redundancies between neighbor pixels but these redundancies can only be exploited locally. To sum up this important point, the model (18) is rewritten in matrix form. Let $\boldsymbol{\eta}_k=(t_{k,1}, \varsigma_{k,1}, \ldots, t_{k,r_k}, \varsigma_{k,r_k})$ be the vector containing all the discontinuity parameters of $\mathcal{X}_k$. Let $\boldsymbol{\theta}_k$ be the vector containing all the values $\theta_{l,m}$ such that $x_{l,m} \in \mathcal{X}_k$. It is assumed that all vectors $\boldsymbol{\theta}_k$ have the same number N of pixels such that $N{=}L\,M/K$. From (18), the ensuing theoretical vector $\boldsymbol{\theta}_k$ can be written:

$$\boldsymbol{\theta}_k = \mathbf{H}\mathbf{s}_k + \mathbf{F}(\boldsymbol{\eta}_k)\mathbf{u}_k. \tag{19}$$

The polynomial coefficients $\mathbf{s}_k = \left(s_0^{(k)}, \ldots, s_{p-1}^{(k)}\right)^T$ characterize the continuous part spanned by the matrix $\mathbf{H}$ of size (N,p). For each value $\theta_{l,m}$ whose pixel has coordinate $x_{l,m} \in \mathcal{X}_k$, the corresponding row in $\mathbf{H}$ is given by

$$\left[x_{l,m}^0 \;\; x_{l,m}^1 \;\; \cdots \;\; x_{l,m}^{p-1}\right].$$

The coefficients $\mathbf{u}_k{=}(u_1, \ldots, u_{r_k})^T$ represent the intensity of discontinuities spanned by the matrix $\mathbf{F}(\boldsymbol{\eta}_k)$ of size (N, r_k). For each value $\theta_{l,m}$ whose pixel has coordinate $x_{l,m} \in \mathcal{X}_k$, the corresponding raw in $\mathbf{F}(\boldsymbol{\eta}_k)$ is given by

$$\left[\int_{\mathcal{X}_k \cap \mathcal{C}_{l,m}} \Phi\left(\frac{\nu - t_{k,1}}{\varsigma_{k,1}}\right) d\nu \quad \cdots \quad \int_{\mathcal{X}_k \cap \mathcal{C}_{l,m}} \Phi\left(\frac{\nu - t_{k,r_k}}{\varsigma_{k,r_k}}\right) d\nu\right].$$

Due to space limitations, it is assumed that each segment $\mathcal{X}_k$ has at most one discontinuity and that an estimate $\widehat{\boldsymbol{\eta}}_k{=}(\hat{t}_{k,1}, \widehat{\varsigma}_{k,1})^T$ is available for each discontinuity (if present) such that $\|\boldsymbol{\eta}_k{-}\widehat{\boldsymbol{\eta}}_k\|_1{\leq}\vartheta$ where ϑ is a small constant. The literature proposes many methods giving such estimates (see for example [1]). Adapting the methodology from [18], the non-linearity is treated by writing (19) :

$$\boldsymbol{\theta}_k = \mathbf{H}\mathbf{s}_k + u_{k,1}\mathbf{F}(\widehat{\boldsymbol{\eta}}_k) + \dot{\mathbf{F}}(\widehat{\boldsymbol{\eta}}_k)u_{k,1}(\boldsymbol{\eta}_k{-}\hat{\boldsymbol{\eta}}_k) + o(\vartheta^2) \tag{20}$$

where $\dot{\mathbf{F}}(\widehat{\boldsymbol{\eta}}_k)$ is the jacobian $(N \times 2)$ matrix of $\mathbf{F}(\boldsymbol{\eta}_k)$. This yields to the locally-adapted linear model:

$$\boldsymbol{\theta}_k = \mathbf{G}(\widehat{\boldsymbol{\eta}}_k)\mathbf{v}_k + o(\vartheta^2) \quad \text{with} \quad \mathbf{G}(\widehat{\boldsymbol{\eta}}_k){=}\left(\mathbf{H} \,|\, \mathbf{F}(\widehat{\boldsymbol{\eta}}_k) \,|\, \dot{\mathbf{F}}(\widehat{\boldsymbol{\eta}}_k)\right)$$

and $\mathbf{v}_k{=}(\mathbf{s}_k, u_{k,1}, u_{k,1}(\boldsymbol{\eta}_k{-}\widehat{\boldsymbol{\eta}}_k))^T$.

When the analyzed image does not contain hidden information, $z_{l,m} = Q_1[y_{l,m}] = Q_1[\theta_{l,m} + \xi_{l,m}]$. Let $\mathbf{z}_k$ be the vector containing all the pixels $z_{l,m}$ corresponding to the k-th segment. Assuming that the noise variance is constant in each segment and that the quantization has negligible effects on the estimation, $\boldsymbol{\theta}_k$ can be estimated by using the linear estimate:

$$\widehat{\boldsymbol{\theta}}_k = \mathbf{G}(\widehat{\boldsymbol{\eta}}_k) \left(\mathbf{G}(\widehat{\boldsymbol{\eta}}_k)^T \mathbf{G}(\widehat{\boldsymbol{\eta}}_k)\right)^{-1} \mathbf{G}(\widehat{\boldsymbol{\eta}}_k)^T \mathbf{z}_k. \tag{21}$$

The estimates $\widehat{\theta}_{l,m}$ are obtained for all (l, m) by calculating the estimate (21) for all segments. Alternatively, for LSB replacement, the LSB plane can for instance be removed to have an estimation which is independent from steganography.

5.2 Almost Optimal Steganalysis

Let $r = \sum_{k=1}^{K} r_k$ be the total number of discontinuities over $\mathcal{D}$. Let $\widehat{\delta}(\mathbf{Z})$ be the test defined as in (8), associated with the threshold $\widehat{\tau}_{\alpha_0}$ and the following decision function $\widehat{\Lambda}(\mathbf{Z})$:

$$\widehat{\Lambda}(\mathbf{Z}) = \sum_{l=1}^{L} \sum_{m=1}^{M} w_{l,m}(z_{l,m} - \overline{z}_{l,m})(z_{l,m} - \widehat{\theta}_{l,m}). \tag{22}$$

where

$$w_{l,m} = \frac{\bar{\sigma}}{\sigma_{l,m}^2 \sqrt{K(N-p-3)}}.$$

The following theorem establishes the loss of optimality of the test $\widehat{\delta}(\mathbf{Z})$ with respect to the optimal LRT (when the parameters $\theta_{l,m}$ are known).

Theorem 2. *In virtue of the Lindeberg's central limit theorem:*

$$\begin{cases} \widehat{\Lambda}(\mathbf{Z}) \xrightarrow{d} \mathcal{N}(0\,;\, 1+b) & \text{under } \mathcal{H}_0 \\ \widehat{\Lambda}(\mathbf{Z}) \xrightarrow{d} \mathcal{N}\left(\frac{R}{2\bar{\sigma}}\sqrt{\kappa}\,;\, 1+b\right) & \text{under } \mathcal{H}_1 \end{cases} \tag{23}$$

where $\kappa = K(N-p)-3r$ *and* b *is an unknown bias:* $0 \leq b \leq \frac{\varepsilon}{\bar{\sigma}^2(N-p-3)} \stackrel{def.}{=} b_{\max}$ *with* ε *a known (little) positive constant. Choosing* $\widehat{\tau}_{\alpha_0} = \Phi^{-1}(1-\alpha_0)\sqrt{1+b_{\max}}$, *it follows that* $\widehat{\delta}(\mathbf{Z}) \in \mathcal{K}_{\alpha_0}$ *and*

$$1-\Phi\left(\frac{1}{1+b_{\max}}\left(\widehat{\tau}_{\alpha_0} - \frac{R}{2\bar{\sigma}}\sqrt{\kappa}\right)\right) \leq \widehat{\beta}(\boldsymbol{\theta}, R) \leq 1-\Phi\left(\widehat{\tau}_{\alpha_0} - \frac{R}{2\bar{\sigma}}\sqrt{\kappa}\right). \tag{24}$$

Proof. Omitted due to space limitations.

The comparison between $\beta(\boldsymbol{\theta}, R)$ and $\widehat{\beta}(\boldsymbol{\theta}, R)$ shows that the loss of optimality of the later is due to: 1) the reduction of the number of "free parameters" from $L\,M$ to κ and 2) the unknown bias $b_{\max}$ which is due to linearization of $\mathbf{F}(\boldsymbol{\eta}_k)$ around the estimation values of discontinuity parameter $\widehat{\boldsymbol{\eta}}_k$. Hence, provided

that r and $b_{\max}$ are sufficiently small, the test $\widehat{\delta}$ is almost optimal. Values r and ϑ were arbitrarily bounded to analytically calculate the power function $\widehat{\beta}(\boldsymbol{\theta}, R)$.

The loss of optimality highlights a more general problem inherent to the "calibration" process, used to estimate cover image. Indeed, a tradeoff has to be found between sparsity (to increase κ) and accuracy (to keep $b_{\max}$ low). Unfortunately, this problem remains open. This problem is inspected in section 6 through a comparison with the WS.

6 Numerical Results and Comparisons

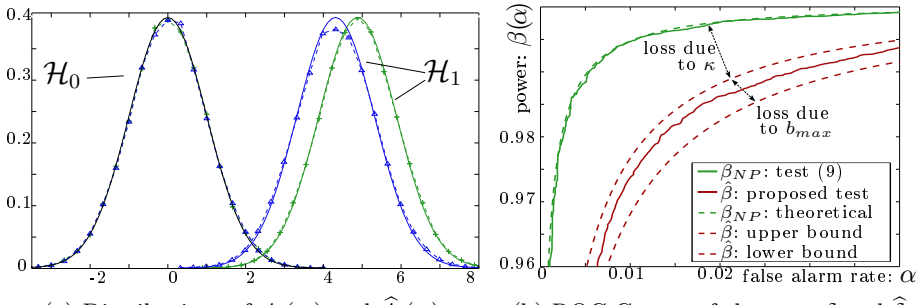
6.1 Theoretical Results on Simulated Data

Figs. 3a and 3b present the results of Theorems 1 and 2 through a numerical simulation. A Monte-Carlo simulation was repeated 25 000 times each with 400 segments of 32 pixels. Every segment has one discontinuity, whose location was uniformly distributed, with settings u_k=96, $\varsigma = 1.75$ and $\vartheta = 1$. An algebraic polynomial of degree 3 was used, the insertion rate was set to 0,47 and the additive noise was stationary with $\bar{\sigma}$=5.43.

6.2 Comparisons with Other Detectors on Real Images

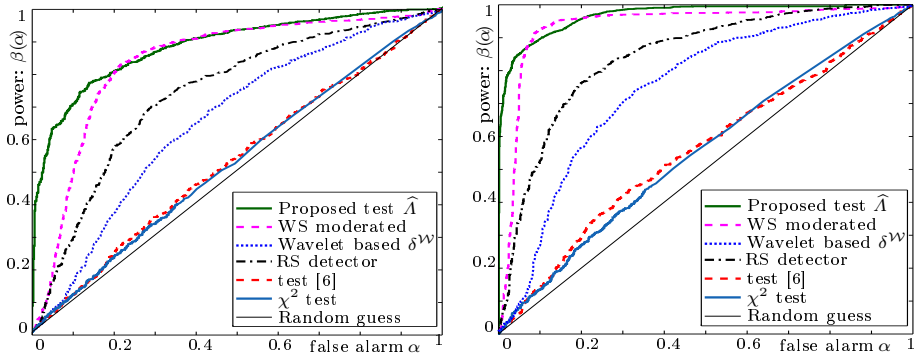
One of the main motivations of this paper was to define a reliable steganalysis in the sense that it explicitly takes into account image content and has an analytically predictable performance. Hence, it was chosen not to compare the proposed test with supervised learning based detectors because, as discussed in section 1, they cannot warrant any optimality of the decision rule.

The LSB replacement detectors compared in this section are : the proposed test $\widehat{\delta}$, the test proposed in [6], the χ^2 test from [21], the RS detector [9] with the original mask [0 1 1 0] and the WS [7] with moderated weights $w_{l,m} = (\sigma_{l,m}+5)^{-1}$ and the filter $\begin{pmatrix} -1/4 & 1/2 & -1/4 \\ 1/2 & 0 & 1/2 \\ -1/4 & 1/2 & -1/4 \end{pmatrix}$ as described in [12]. The key role of image model is



(a) Distributions of Λ (+) and $\widehat{\Lambda}$ (△). (b) ROC Curves of the tests δ and $\widehat{\delta}$.

Fig. 3. Theoretical (--) and empirical (—) results for simulated data



(a) ROC Curves for $R = 0.05$ (b) ROC Curves for $R = 0.1$

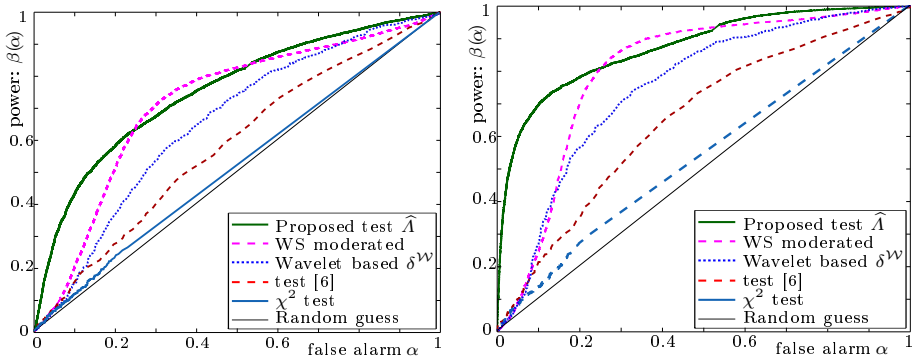
Fig. 4. Comparisons of detectors : ROC curves for UCID database

highlighted with a sixth detector $\delta^{\mathcal{W}}$ based on the LRT defined in section 3 and using a wavelet shrinkage to estimate the cover content $\boldsymbol{\theta}_k$. For a large scale comparison the 1338 image of UCID [17] and 9000 images, previously cropped to size 128× 128, from BOSSbase [4] were used.

On Fig. 4 and Fig. 5, the WS surprisingly exhibits higher power than the proposed test $\widehat{\delta}$ for intermediate false alarm rate α_0. But in a practical application, for instance when analysing the whole data of a suspect, it is obviously less serious to consider that one set a false alarm rate constraint of 0.3 or even 0.1. Indeed it is reasonable to think that thousands or millions of images will be inspected and a constraint of $\alpha_0 = 10^{-2}$ is much more realistic. For such low false alarm rate, Fig. 4 and Fig. 5 show that the proposed test outperforms the WS which is the most serious challenger. Note that the detection power is higher (for all detectors) on UCID images because their size is bigger that cropped BOSS images.

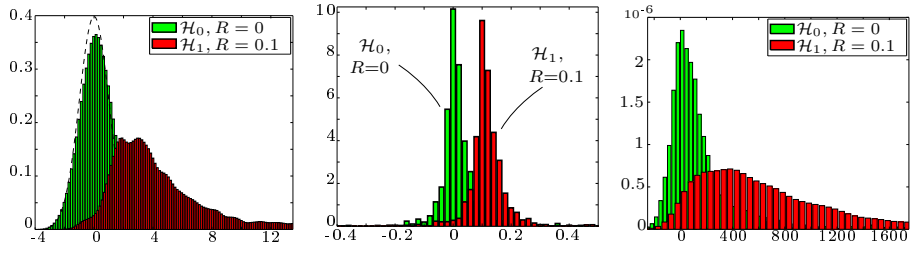
To understand the tests performances depicted on the ROC curves of Fig. 4 and Fig. 5, a thorough comparison of the statics used by the detectors is necessary. To this end, Fig. 6 shows the empirical distribution obtained on UCID image database for $R = 0$ and $R = 0.1$ with the proposed test $\widehat{\Lambda}$, the WS and the RS.

The results drawn in Fig. 6 permits understanding the importance of the image content model. The WS detector relies on a basic autoregressive model which fairly works for most images but fails for few. Hence, the distributions of WS residuals exhibit heavy tails and outlier values under both hypothesis of cover or stego images (see [3] for a thorough numerical analysis). These values explain why one can not warrant a very low false alarm rate and a high power. On the contrary, the proposed model of natural images permits an accurate estimation of the cover which later prevents occurrence of most outliers. The distribution of $\widehat{\Lambda}$ shown in Fig. 6 under null hypothesis $\mathcal{H}_0$ is close to the theoretically calculated standard Gaussian ; this known distribution permits to meet a false alarm rate



(a) ROC Curves for $R = 0.05$ (b) ROC Curves for $R = 0.1$

Fig. 5. Comparisons of detectors : ROC curves for BOSS database



(a) $\widehat{\Lambda}$ distribution (b) WS residuals Distribution (c) RS Distribution

Fig. 6. Comparisons of decision ROC curves for UCID database

constraint. Note that under (stego) hypothesis $\mathcal{H}_1$ the distribution of $\widehat{\Lambda}$ is not Gaussian anymore because as defined in (23), result depends on the noise power ϱ which varies for each inspected image.

The proposed test $\widehat{\Lambda}$ and the WS have a very similar expression but fundamentally differ on cover estimation. Highly textured images are typically difficult to analyse without an accurate model of image content. Importance of that point is illustrated in Fig. 7. Thirty highly textured images have been analysed 1000 times with an additive Gaussian stationary noise with standard deviation σ=0.5. Results are normalised to have the same theoretical mean. Fig. 8 shows that for all images, the standard deviation does not change much between the WS (with standard $w_{l,m}=(\sigma_{l,m}+1)^{-1}$ or moderated $w_{l,m}=(\sigma_{l,m}+5)^{-1}$ weights) and the proposed test $\widehat{\Lambda}$. However, the textured content of these images can not be accurately estimated with a rather simple model. This causes a textured error of content estimation which might result in a bias in the WS residual. On the contrary, the proposed image model allows an accurate estimation of image content and thus prevents the occurrence of most spurious values which later avoid a reliable decision.

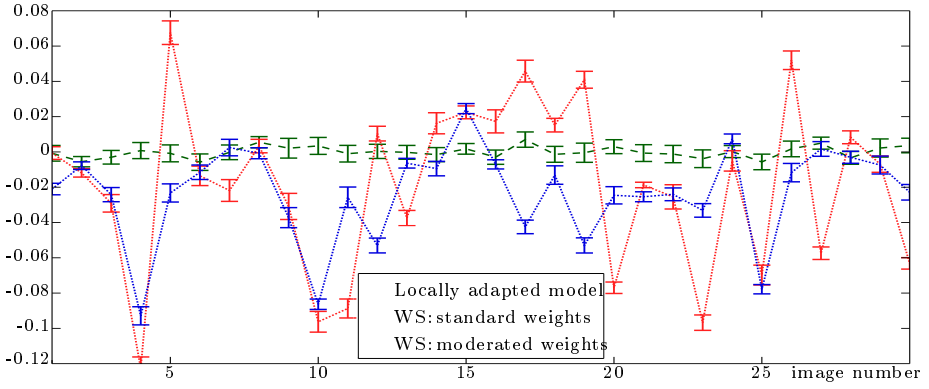


Fig. 7. Monte-Carlo analysis (Mean and standard deviation) of textured images

Fig. 8. Three of the thirty textured images used for Monte Carlo simulation

7 Conclusions

This paper made a first step to fill the gap between physical model of cover-image and steganalysis. A local non-linear parametric model of natural images is proposed based on the physical properties of acquisition. To estimate simply, yet efficiently, the cover image content it is proposed to linearized the model. The theoretical findings of the companion paper [22] are exploited to design an almost optimal test, *i.e.* with a bounded loss of optimality. This allows a reliable steganalysis as the proposed test permits to analytically predict and warrant a false alarm constraint.

Numerical results on two image databases show the relevance of the presented approach. Thanks to the accurate image model, the proposed test exhibits much better performance for small false alarm rate.

References

1. Bruni, C., De Santi, A., Koch, G., Sinisgalli, C.: Identification of discontinuities in blurred noisy signals. IEEE Trans. on Circuit and systems-I 44(5), 422–433 (1997)
2. Böhme, R.: Advanced Statistical Steganalysis, 1st edn. Springer Publishing Company, Incorporated, Heidelberg (2010)

3. 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)
4. BOSS Contest: Break Our Steganographic System (2010), http://boss.gipsa-lab.grenoble-inp.fr
5. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2nd edn. Morgan Kaufmann, San Francisco (2007)
6. 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)
7. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Proc.of SPIE Security, Steganography, and Watermarking of Multimedia Contents VI, vol. 5306, pp. 23–34 (2004)
8. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications, 1st edn. Cambridge University Press, Cambridge (2009)
9. Fridrich, J., Goljan, M., Du, R.: Reliable detection of LSB steganography in color and grayscale images. IEEE Multimedia 8, 22–28 (2001)
10. Goodman, J.W.: Introduction to Fourier Optics, 3rd edn. Roberts & Company Publishers, Englewood (2005)
11. Ker, A.D.: Locating steganographic payload via WS residuals. In: ACM Proceedings of 10th Multimedia and Security Workshop, pp. 27–31 (2008)
12. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Proc. of SPIE Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, vol. 6819, pp. 501–517 (2008)
13. Ker, A.D.: A Capacity Result for Batch Steganography. IEEE Signal Processing Letters 14(8), 525–528 (2007)
14. Lehman, E.: Testing Statistical Hypotheses, 2nd edn. Chapman & Hall, Boca Raton (1986)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics (5), 577–685 (1989)
16. Sallee, P.: Model-based methods for steganography and steganalysis. International Journal of Image and Graphics 5(1), 167–189 (2005)
17. Schaefer, G., Stich, M.: UCID - an uncompressed colour image database. In: SPIE Proceedings, vol. 5307, pp. 472–480, http://vision.cs.aston.ac.uk/datasets/UCID/
18. Seber, G., Wild, C.: Nonlinear Regression. Wiley, Chichester (1989)
19. Wang, Y., Moulin, P.: Steganalysis of block-DCT image steganography. In: 2003 IEEE Workshop on Statistical Signal Processing (2003)
20. Wang, Y., Moulin, P.: Statistical modelling and steganalysis of DFT-based image steganography, vol. 6072, p. 607202. SPIE, CA (2006)
21. Westfeld, A., Andreas, P.: Attacks on steganographic systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–76. Springer, Heidelberg (2000)
22. Zitzmann, C., Cogranne, R., Retraint, F., Nikiforov, I., Fillatre, L., Cornu, P.: Statistical Decision Methods in Hidden Information Detection. In: Filler, T., et al. (eds.) IH 2011. LNCS, vol. 6958, pp. 163–177. Springer-, Heidelberg (2011)

Video Steganography with Perturbed Motion Estimation

Yun Cao[1,2], Xianfeng Zhao[1], Dengguo Feng[1], and Rennong Sheng[3]

[1] State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.R.China
{caoyun,xfzhao,feng}@is.iscas.ac.cn
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049, P.R.China
[3] Beijing Institute of Electronics Technology and Application, Beijing 100091, P.R.China
rennongsheng@vip.sina.com

Abstract. In this paper, we propose an adaptive video steganography tightly bound to video compression. Unlike traditional approaches utilizing spatial/transformed domain of images or raw videos which are vulnerable to certain existing steganalyzers, our approach targets the internal dynamics of video compression. Inspired by Fridrich et al's perturbed quantization (PQ) steganography, a technique called perturbed motion estimation (PME) is introduced to perform motion estimation and message hiding in one step. Intending to minimize the embedding impacts, the perturbations are optimized with the hope that these perturbations will be confused with normal estimation deviations. Experimental results show that, satisfactory levels of visual quality and security are achieved with adequate payloads.

1 Introduction

Steganography is the art and science of hiding the very presence of communication by embedding secret messages into innocent-looking digital signals. Although with a huge capacity, the video content has been less exploited for steganography mainly due to processing complexities. Within these years, the advent of high performance graphics processing unit (GPU) has made video processing a much easier job, even with portable devices. What's more, high performance networking technologies have made networked multimedia applications increasingly popular such as video on demand, internet television, video telephony, etc. In order to achieve real-time covert communications with adequate payloads, it is obviously less suspicious to transmit video streams than a large number of individual images.

In this paper, a specific integral part of video compression, namely motion estimation is utilized for embedding purposes. We target this stage for the following three reasons: First, most existing steganalyzers (e.g., [4,10,12,17]) model the videos as successive still images and the embedding process as adding independent mean zero Gaussian noises. The reliability of the model is likely to

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 193–207, 2011.

deteriorate when embedding with motion information. Secondly, digital videos are usually highly compressed for economical storage and efficient transmission. The compression process can be modeled as an information-reducing process, and improved steganographic security can be achieved by using the cover object as side information to confine the embedding changes to those elements whose values are the most "uncertain"[8]. Finally, as we will discuss in details in 3.3, compared to regular compression, very limited distortion is introduced by embedding with motion information.

The idea of using motion vectors (MV) as the covert information carrier can be dated back to Kutter et al's work [11] in which they proposed a video watermark scheme by altering the MVs directly. In recent years, steganographic methods adopting improved strategies have been developed. Xu et al [15] suggested embedding message bits in the magnitudes of MVs and the control data in the intra frames, the LSBs of MVs' horizontal or vertical components are used for embedding. Fang and Chang [7] designed a steganography using MVs' phase angles. The MVs are arranged in pairs, and for each pair, if the phase angle difference does not satisfy the embedding condition, one in the pair has to be replaced by a new qualified MV. These two schemes select candidate motion vectors (CMV) according to their magnitudes with the assumption that modifications applied to MVs with larger magnitudes introduce less distortion. But Aly had pointed out in his latest work [2] that the magnitude-based selection rule cannot ensure minimum prediction errors. He hence designed a new selection rule directly associated with macro block (MB) prediction errors. MVs associated with high prediction errors are chosen, and secret bits are embedded in the LSBs of both their horizontal and vertical components.

We have realized that the methods outlined above share some features in common, i.e., each of them first selects a subset of MVs during motion estimation following a pre-defined selection rule, then makes direct modifications to them for data hiding. There are two issues of concern: First, these methods select CMVs according to their magnitudes or associated prediction errors, and these information are known to public and needed for extraction. In fact, if the adaptive selection rule is public, or only weakly dependent on a key, an attacker can apply the same rule and start building an attack. Secondly, CMVs are arbitrarily modified (e.g., LSB replacement) which violates the encoding principles a lot. Consequently unexpected distortions and detectable statistical changes would be invited which implies that even Aly's selection rule cannot eventually guarantee minimum distortion either.

In this paper, we propose an adaptive video steganography combined with MPEG-4 video compression. Inspired by Fridrich et al's PQ steganography [8], a technique called perturbed motion estimation (PME) is introduced for information hiding. The secret bits are embedded at the same time of motion estimation, and as no further modification is needed, the compressed data is affected in a natural fashion. With PME, by virtue of the wet paper code [8], the sender is free to use different criteria for non-shared selection rule designing. In terms of security evaluation, the Kullback-Leibler divergence between the MV

probability distributions of non-stego and stego compressed videos is leveraged as a preliminary benchmark of the inherent detectability. We subject our method to 2 image-oriented steganalytic algorithms (i.e.,[13,16]) to show their ineffectiveness in attacking MV-based methods. Moreover, one specific steganalysis against the MV-based steganography [18] is implemented for security tests. Experimental results show that compared to other MV-based methods, PME achieves a better visual quality and a higher security level.

The rest of the paper is structured as follows. In section 2, the basic concepts of motion estimation and wet paper code are introduced. In section 3, the PME technique is presented. We give detailed descriptions of the embedding and extracting procedures and make analysis of the introduced distortion. In section 4, comparative experiments are conducted to show the performance of our scheme with special attention paid to the security evaluation. Finally in section 5, concluding remarks are given with some future research directions.

2 Preliminaries and Notations

2.1 Motion Estimation

Due to the large amount of data involved, lossy compression is routinely employed for economically storing digital videos on storage constrained devices or efficiently transmitting them over bandwidth-limited networks. The raw video is essentially a series of highly correlated image frames, and the temporal redundancy that exists between frames can be greatly reduced by inter-frame coding. State-of-the-art compression standards perform inter-frame coding based on a local motion model of $b \times b$ pixels macro blocks (MB). Most MBs within an inter-frame are coded as inter-MBs, and to encode current inter-MB $\boldsymbol{C}$, the encoder typically uses a prior coded frame as its reference and search for a good matching MB within it. Block matching problem is generally formulated by quantifying the similarity between $\boldsymbol{C}$ and candidate MBs in the reference frame using a similarity metric. The candidate with the largest similarity is taken as $\boldsymbol{C}$'s best prediction, and denoted as $\boldsymbol{R}$. In this paper, without loss of generality, we use mean square error (MSE) as the matching criterion for its good theoretical significance.

$$\mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R}) = \frac{1}{b^2} \sum_{1 \le i,j \le b} (c_{i,j} - r_{i,j})^2, \tag{1}$$

where $c_{i,j}$ and $r_{i,j}$ represent the luminance values of $\boldsymbol{C}$ and $\boldsymbol{R}$ respectively. Once $\boldsymbol{R}$ is found, $\boldsymbol{C}$'s MV will be calculated as

$$\boldsymbol{mv} = (h, v) = (H_r - H_c, V_r - V_c) \tag{2}$$

where h, v are the horizontal and vertical components, (H_r, V_r) and (H_c, V_c) denote the coordinates of $\boldsymbol{R}$ and $\boldsymbol{C}$ respectively. Schematic diagram of MV calculation is shown in Fig. 1. Consequently only $\boldsymbol{mv}$ representing the motion of $\boldsymbol{C}$ and the differential block $\boldsymbol{D} = \boldsymbol{C} - \boldsymbol{R}$ need to be further coded and transmitted. A generic structure of inter-MB coding is depicted in Fig. 2.

Later in Section 3, an optimized perturbation is introduced into regular motion estimation for data hiding.

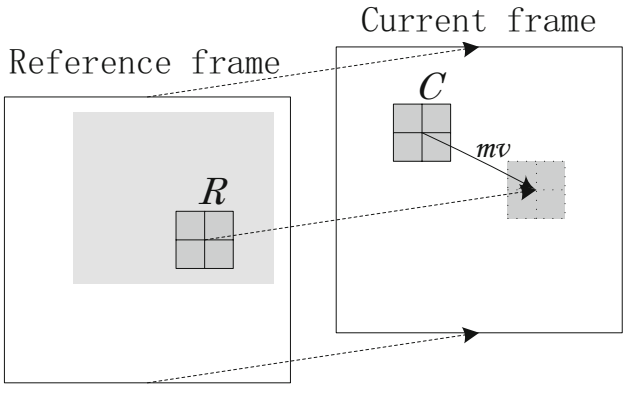


Fig. 1. Motion estimation applied to $\boldsymbol{C}$

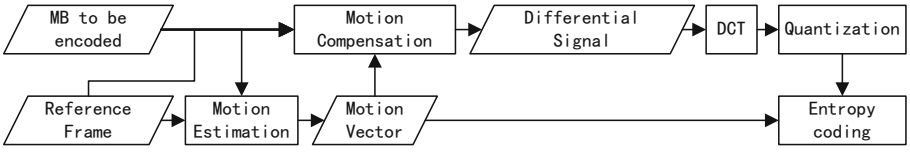


Fig. 2. Generic structure of inter-MB coding

2.2 Wet Paper Code

Fridrich et al suggested that one possible measure to improve the steganographic security is to embed message using adaptively selected components of the cover object, such as noisy areas or segments with a complex texture. However, if the adaptive selection rule is public, or only weakly dependent on a key, the attacker can apply the same rule and start building an attack. As a countermeasure, they designed the wet paper code, a simple variable-rate random linear code, to improve the performance of their PQ steganography [8] where the encoding process is modeled as writing in memory with defective cells. Wet paper code enables the sender to embed a message into arbitrary selected components of the cover object without sharing the selection rule with the recipient.

In the embedding scenario, with a cover object $\mathcal{X}$, the sender wants to send out a q-bit message $\boldsymbol{m} = (m_1, \ldots, m_q)^{\mathrm{T}}$. $\mathcal{X}$ consists of n elements $\{x_i\}_{i=1}^{n}$, $x_i \in \mathcal{J}$, where $\mathcal{J}$ is the range of discrete values for x_i. Then an arbitrary selection rule is used to pick up a k-element subset $\mathcal{S} \subset \mathcal{X}$ as the information channel. Any element in $\mathcal{S}$ is allowed to be modified if necessary for embedding, and the remaining $n-k$ ones are kept untouched. Assuming that the sender and the recipient agree on a secret key K and a public parity function P, which is a

mapping $\mathrm{P}: \mathcal{J} \rightarrow \{0,1\}$[1], the sender's job is to modify some $x_i \in \mathcal{S}$ to create the stego object $\mathcal{Y} = \{y_i\}_{i=1}^n$ with the purpose that the newly obtained binary column vector $\boldsymbol{v}' = \{\mathrm{P}(y_i)\}_{i=1}^n$ satisfies

$$\boldsymbol{M}\boldsymbol{v}' = \boldsymbol{m} \tag{3}$$

where $\boldsymbol{M}$ is a $q \times n$ pseudo-random binary matrix generated by K. For detailed encoding process, please refer to [8].

As to the recipient, with the stego object $\mathcal{Y}$, he first calculates the column vector $\boldsymbol{v}' = \{\mathrm{P}(y_i)\}_{i=1}^n$, then extract the message $\boldsymbol{m} = \boldsymbol{M}\boldsymbol{v}'$ using the matrix $\boldsymbol{M}$ generated by the shared key K.

3 The Proposed Video Steganography

3.1 Perturbed Motion Estimation

We call our method Perturbed Motion Estimation (PME) because during inter-frame coding we slightly perturb the encoder (the process of motion estimation) for certain MBs to embed message bits. The sender can arbitrarily design a non-shared selection rule for different considerations. In this paper, we take a MSE based selection rule for example, and the selected MBs are called applicable MBs defined by Definition 1.

Definition 1. *(applicable MB). When searching for current MB $\boldsymbol{C}$'s prediction, with a preset scaling parameter α, we call $\boldsymbol{C}$ an applicable MB if other than its best prediction $\boldsymbol{R}$, there is at least one candidate $\boldsymbol{R}'$ which satisfies*

1.
$$\mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R}') \leq (1+\alpha)\mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R}) \tag{4}$$

2.
$$\mathrm{P}(\boldsymbol{mv}) \bigoplus \mathrm{P}(\boldsymbol{mv}') = 1 \tag{5}$$

where $\boldsymbol{mv}'$ corresponds to the MV pointed to $\boldsymbol{R}'$, P is employed as the parity function defined as $\mathrm{P}(\boldsymbol{mv}) = \mathrm{P}((h,v)) = \mathrm{LSB}(h+v)$ and $\bigoplus$ denotes the XOR operator.

Among all qualified candidates of the applicable MB, the one with minimum $\mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R}')$ is called $\boldsymbol{C}$'s suboptimal prediction and denoted as $\hat{\boldsymbol{R}}$. Then the MV $\hat{\boldsymbol{mv}}$ pointed to $\hat{\boldsymbol{R}}$ is calculated as sketched in Fig. 3.

In the embedding scenario, we model one single inter-frame as the cover object $\mathcal{X} = \{\boldsymbol{X}_i\}_{i=1}^n$ where $\boldsymbol{X}_i$ denotes the i^{th} inter-MB of $\mathcal{X}$ and n the total number. Assuming that the sender wants to communicate a q-bit message $\boldsymbol{m} = (m_1, \ldots, m_q)^{\mathrm{T}}$ where q is less than $\mathcal{X}$'s capacity (i.e., the total number of its applicable MBs), and he agrees with the recipient on a secret key K used to

[1] $\mathrm{P}(\cdot)$ could be any function defined on $\mathcal{J}$ with the range {0,1}, if x_i is a single integer, $\mathrm{LSB}(x_i)$ could be one good example.

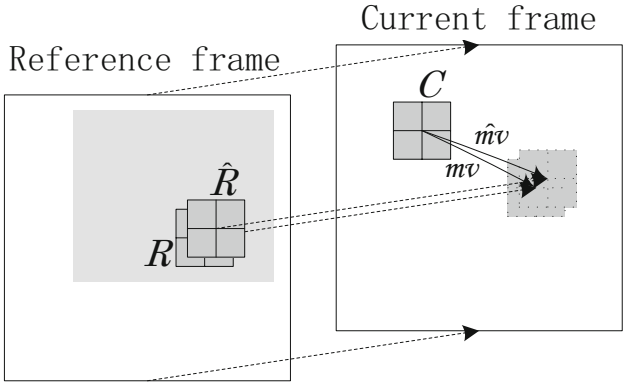


Fig. 3. Applicable MB $\boldsymbol{C}$

generate a pseudo-random binary matrix $\boldsymbol{M}$ of dimensions $q \times n$, the procedure of PME is described below.

Channel Building: With a preset scaling parameter α, this process is applied to all inter-MBs to determine whether they are applicable. As a result, k applicable MBs are selected with their indices recorded as $\boldsymbol{w} = (w_1, w_2, \ldots, w_k)$. These MBs constitute the information channel $\mathcal{S} \subset \mathcal{X}$.

Wet Paper Coding: First, the sender calculates the column vector $\boldsymbol{v} = \{\mathrm{P}(\boldsymbol{mv}_i)\}_{i=1}^{n} = \{v_i\}_{i=1}^{n}$, where $\boldsymbol{mv}_i$ is the MV of $\boldsymbol{X}_i$. Secondly, a new $q \times k$ matrix $\boldsymbol{M'}$ is formed using the $\{w_i^{th}\}_{i=1}^{k}$ columns of $\boldsymbol{M}$, and the sender solves a system of linear equations to get a k-bit binary column vector $\boldsymbol{u}_1$ which satisfies

$$\boldsymbol{M}'\boldsymbol{u}_1 = \boldsymbol{m} \oplus \boldsymbol{M}\boldsymbol{v}. \tag{6}$$

Finally, a k-bit column vector $\boldsymbol{u}_2 = \{v_{w_i}\}_{i=1}^{k}$ is formed, and $\boldsymbol{u} = \{u_i\}_{i=1}^{k}$ is obtained as

$$\boldsymbol{u} = \boldsymbol{u}_1 \oplus \boldsymbol{u}_2. \tag{7}$$

Perturbation: For $\boldsymbol{X}_i (i = 1, 2, \ldots, n)$, if $i = w_j$ and $\mathrm{P}(\boldsymbol{mv}_i) \neq u_j$, $\hat{\boldsymbol{mv}}_i$ is taken as its MV. Otherwise, $\boldsymbol{mv}_i$ is used as usual.

After PME, further encoding processes will continue to generate the compressed frame $\mathcal{Y}$.

3.2 Extraction

Compared to the embedding process, the message extraction is much simpler since most of the job has been done by the sender. When decoding the received frame $\mathcal{Y}$, the recipient first calculates the column vector $\boldsymbol{v}' = \{\mathrm{P}(\boldsymbol{mv}'_i)\}_{i=1}^{n}$, where $\boldsymbol{mv}'_i$ is the MV of the i^{th} inter-MB of $\mathcal{Y}$. Then the agreed key K is used to generate the $q \times n$ matrix $\boldsymbol{M}$ and message $\boldsymbol{m}$ is extracted as

$$\boldsymbol{m} = \boldsymbol{M}\boldsymbol{v}'. \tag{8}$$

3.3 Embedding Distortion

As described in Fig. 2, after motion compensation, the differences signal $\boldsymbol{D} = \boldsymbol{C} - \boldsymbol{R}$ will be subject to DCT coding, and the coefficients in $\boldsymbol{T} = \mathrm{DCT}(\boldsymbol{D})$ will be quantized and entropy coded before transmission. The main distortion is introduced by the quantization step where a uniform quantizer is commonly considered. The quantizer simply divides the sample $t_{i,j}$ by integer Q and rounds to the nearest integer. Bellifemine et al's research [3] looked into the distributions of $\boldsymbol{T}$'s coefficients, and pointed out that if the motion compensation technique is used, the 2D-DCT coefficients of the differential signal tend to be less correlated. Thus the distribution of a sample $t \in \boldsymbol{T}$ can be well modeled with the Laplacian probability density function given by

$$f_t(t, \sigma) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}}{\sigma}|t|}, \tag{9}$$

where σ^2 is the variance of $\boldsymbol{D}$ [9]. The coefficients in $\boldsymbol{D}$ are usually modeled as signals with zero mean, and we can take $d^2 = \mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R})$ as an approximation of σ^2, i.e.,

$$f_t(t, d) = \frac{1}{\sqrt{2}d} e^{-\frac{\sqrt{2}}{d}|t|}. \tag{10}$$

The probability that a sample will be quantized to Qi is simply the probability that the sample is between $Q(i - 1/2)$ and $Q(i + 1/2)$ which is

$$p_i = \int_{Q(i-1/2)}^{Q(i+1/2)} f_t(t, d)dt. \tag{11}$$

It is possible to compute the expected distortion as a function of Q and d:

$$\begin{aligned} \mathrm{D}(Q, d) &= \textstyle\sum_{i=-\infty}^{+\infty} \int_{Q(i-1/2)}^{Q(i+1/2)} (t - Qi)^2 f_t(t, d)dt \\ &= d^2 + \frac{dQ}{\sqrt{2}} e^{\frac{Q}{\sqrt{2}d}} - \frac{\sqrt{2}dQ\cosh(Q/\sqrt{2}d)}{1 - e^{-(\sqrt{2}Q/d)}}. \end{aligned} \tag{12}$$

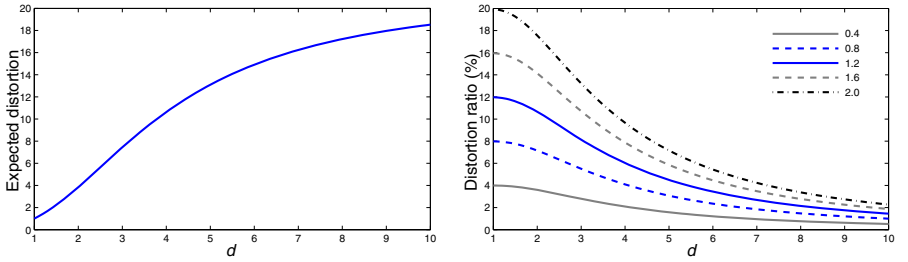
A commonly used quantization matrix sets $Q = 16$, and the values of $\mathrm{D}(16, d)$ are plotted in Fig. 4(a) with d increased from 1 to 10. Practically for a given Q, $\mathrm{D}(Q, d)$ is an increasing function with respect to common values of d. During PME, if $\boldsymbol{C}$ happens to be an applicable MB and $\hat{\boldsymbol{m}}\hat{\boldsymbol{v}}$ is used for substitution, the differential signal will be calculated based on $\hat{\boldsymbol{R}}$, i.e., $\hat{\boldsymbol{D}} = \boldsymbol{C} - \hat{\boldsymbol{R}}$. Similarly we use $\hat{d}^2 = \mathrm{MSE}(\boldsymbol{C}, \hat{\boldsymbol{R}})$ to estimate $\hat{\boldsymbol{D}}$'s variance $\hat{\sigma}^2$. A ratio defined as

$$\gamma = \frac{\mathrm{D}(Q, \hat{d}) - \mathrm{D}(Q, d)}{\mathrm{D}(Q, d)} \times 100\% \tag{13}$$

is leveraged to measure the degree of embedding distortion introduced to $\boldsymbol{C}$. Since

$$\begin{aligned} \hat{d}^2 &= \mathrm{MSE}(\boldsymbol{C}, \hat{\boldsymbol{R}}) \\ &\leq (1 + \alpha)\mathrm{MSE}(\boldsymbol{C}, \boldsymbol{R}) \\ &= (1 + \alpha)d^2, \end{aligned} \tag{14}$$

we have



(a) Expected distortion (b) Upper bounds of distortion ratios with different α

Fig. 4. Expected distortions and upper bounds of distortion ratios in typical settings

$$\begin{aligned}\gamma &\leq \frac{\mathrm{D}(Q,\sqrt{1+\alpha}d)-\mathrm{D}(Q,d)}{\mathrm{D}(Q,d)} \times 100\% \\ &= \lceil \gamma \rceil,\end{aligned} \tag{15}$$

and the values of $\lceil \gamma \rceil$ with some different values of α are plotted in Fig. 4(b). With typical settings, if perturbation is applied to one MB, little distortion would be introduced. Furthermore, as to the entire inter-frame, since only a subset of its inter-MBs is likely to be affected, the overall impact is limited.

3.4 Practical Implementation for Video Applications

One dominant advantage of video data as the cover object is its huge capacity. Although each single inter-frame offers a capacity far less than a comparative still image since it is highly compressed, the payload can be shared, and the embedding impact on the individual frame can be limited to a low level. Practically the message to be sent is divided into small pieces, and the embedding can be performed as follows.

We use a fixed l-bit binary vector $\boldsymbol{q}_i$ to denote the binary-stored capacity of the i^{th} frame. At the beginning, $\boldsymbol{q}_2$ is estimated and embedded into the first frame without any message bits using PME. Then every time before the i^{th} frame is used for embedding, $\boldsymbol{q}_{i+1}$ is acquired and embedded with $\boldsymbol{q}_i - l$ message bits. This continues until all message bits have been embedded, and in the last frame, an agreed l-bit binary vector $\boldsymbol{q}_e$ is embedded with the last message bits.

As for the recipient, first, $\boldsymbol{q}_2$ is extracted since the recipient knows that the first frame always carries l bits. Then every time before extracting from the i^{th} frame, the size information $\boldsymbol{q}_i$ is used to determine the dimension of the matrix. This continues until $\boldsymbol{q}_i = \boldsymbol{q}_e$ which indicates that there will be no more message bits in the coming frames.

4 Performance Experiments

In this paper, the proposed steganography is implemented using a well-known MPEG-4 video codec Xvid [1]. Besides, Fang and Chang's [7], Xu et al's [15] and

Fig. 5. Sequences used in experiments

Aly's [2] methods are also implemented for comparison, and are referred to as ALG1, ALG2 and ALG3 respectively. As the message bits are embedded using MVs, the embedding strength is measured by the average embedded bits per inter-frame (bpf). As shown in Fig. 5, 22 standard test sequences in the 4:2:0 YUV format are used in our experiments, and they each have a frame size of 352×288 which corresponds to 396 MBs per frame.

4.1 Impacts on Visual Quality and Computational Efficiency

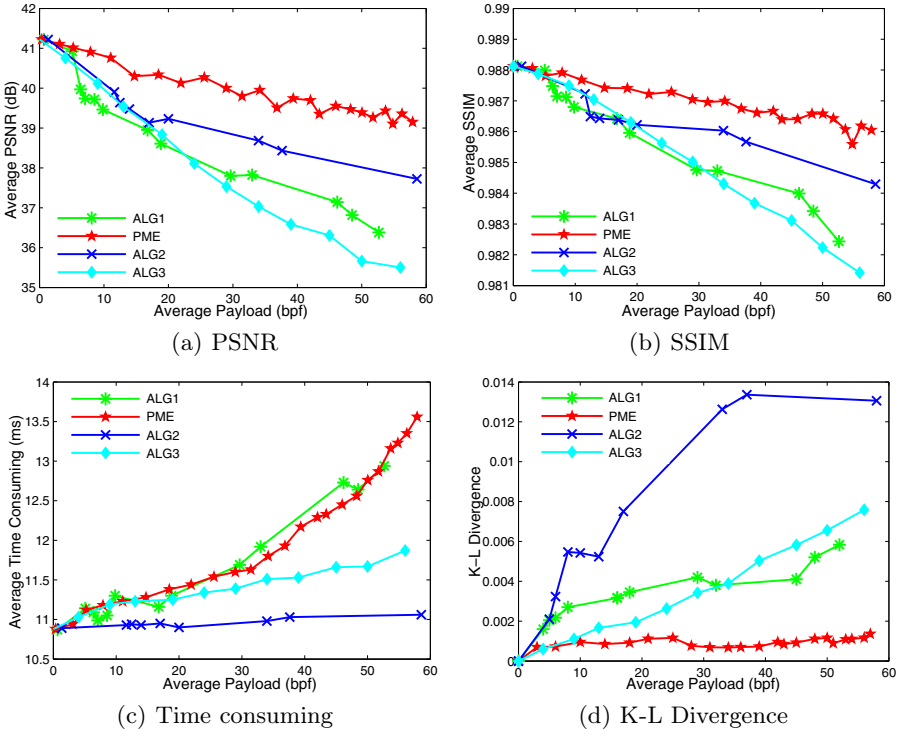
The visual quality is measured by the PSNR (peak signal-to-noise rate) and the SSIM (structural similarity) [14] values with respect to the human visual system. First, we repeat embedding processes over one 300-frame sequence "coastguard" with increasing embedding strengths. For PME, the scaling factor α is increased from 0 to 0.25 with the average payload increases from 0 to 57.9 bpf. For ALG1, ALG2 and ALG3 different values of thresholds associated with MV magnitude or prediction error are assigned to achieve comparative embedding strengths. As illustrated in Fig. 6(a) and (b), the average values of PSNR and SSIS decrease with embedding strength increases. Meanwhile, there are increases in the average encoding time (ms per frame) as shown in Fig. 6(c).

For a more thorough investigation into how visual quality is affected, the dynamic changes of PSNR and SSIM values along adjacent frames are calculated and plotted in Fig. 7(a) and (b) respectively. The embedding strengths of the four methods are set at a similar level, i.e., PME at 34.19 bpf, ALG1 at 32.99 bpf, ALG2 at 33.98 bpf and ALG3 at 33.95 bpf. In addition, the standard Xvid MPEG-4 encoder is also employed as a reference and is referred to as STD. It is observed that the values of PME decrease in a slighter and steadier manner along frames compared to its competitors.

Tests on some different sequences are also conducted. For each sequence, with comparative embedding strengths, the performances of the four methods are evaluated. Compared to the standard Xvid MPEG-4 encoder, the computational overheads, the decreases in PSNR and SSIM values are calculated and recorded in Table 1.

4.2 Impacts on MV Statistical Characteristics

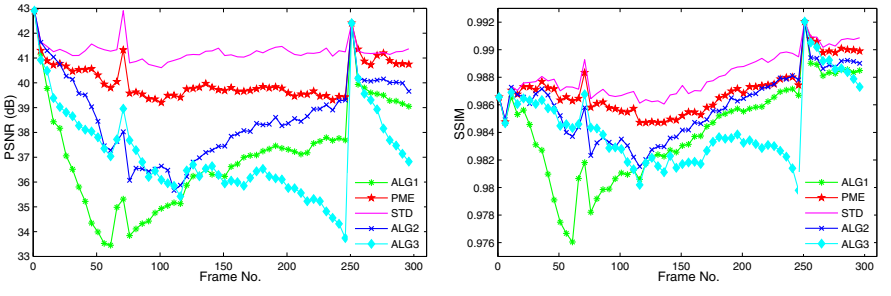
A natural approach to steganalysis is to model a non-stego object as a realization of a random process and leverage detection theory to determine optimal solutions and estimate performance. Since MV-based steganographic methods are discussed here, we are interested in their ability to preserve MV statistical



(a) PSNR (b) SSIM

(c) Time consuming (d) K-L Divergence

Fig. 6. Tests on "coastguard"



(a) PSNR (b) SSIM

Fig. 7. Impact on visual quality along frames

characteristics. The K-L divergence is a measure of "closeness" of histograms in a way that is compatible with optimal hypothesis testing. Because of this property, Cachin [5] suggested using the K-L divergence as a benchmark of the inherent detectability of a steganographic system. In this paper, K-L divergence is utilized as a basic means of gauging how easy it is to discriminate between

Table 1. Test results of some different sequences. ES (Embedding Strength (bpf)), CO (Computational Overhead (%)), DP (Decrease in PSNR(dB)), DS (Decrease in SSIS (10^{-3})), K-L (K-L Divergence (10^{-3})).

Sequence	Method	ES	CO	DP	DS	K-L
bus	PME	32.54	5.3	1.56	1.23	3.4
	ALG1	31.81	6.1	2.75	2.11	6.1
	ALG2	32.03	0.9	2.63	2.24	4.5
	ALG3	29.05	1.1	3.67	3.50	5.0
coastguard	PME	39.38	7.8	1.48	1.51	0.7
	ALG1	32.99	7.6	3.45	3.59	3.7
	ALG2	37.35	0.9	2.67	2.47	12.7
	ALG3	34.39	1.2	4.19	3.81	3.9
foreman	PME	33.20	5.9	0.49	0.25	8.3
	ALG1	30.51	5.2	2.29	1.23	13.3
	ALG2	32.38	0.9	0.84	0.45	13.1
	ALG3	31.15	1.2	5.48	5.69	11.3
stefan	PME	24.01	4.6	1.56	1.23	1.6
	ALG1	22.08	4.5	1.84	1.62	5.6
	ALG2	23.57	0.9	1.55	1.29	3.3
	ALG3	21.08	1.1	2.72	2.45	3.1
tempete	PME	15.43	2.5	0.95	0.41	3.1
	ALG1	15.70	2.8	4.3	3.5	8.2
	ALG2	16.27	1.1	0.64	0.26	10.3
	ALG3	15.26	1.3	2.28	1.1	10.3
walk	PME	51.52	8.6	0.63	0.35	9.4
	ALG1	46.94	8.2	4.15	5.7	16.3
	ALG2	51.38	1.1	1.56	0.89	17.1
	ALG3	47.42	1.3	2.60	1.86	14.2

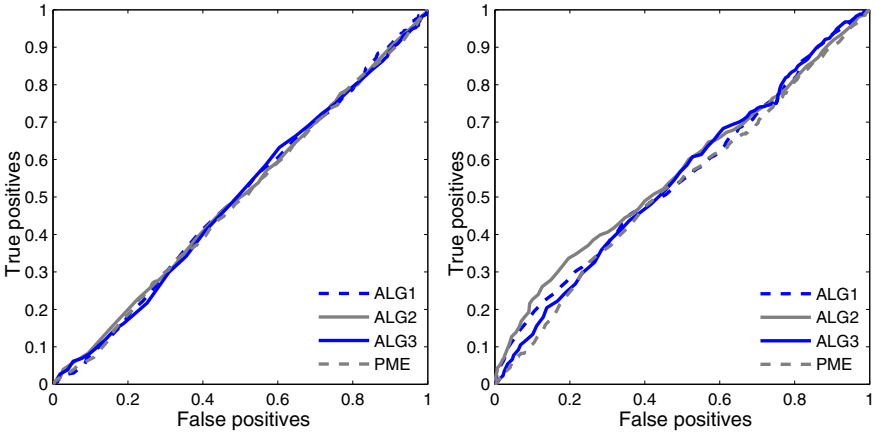
Table 2. Performance comparison among different steganalytic features (in the unit of %)

	Xuan's			Pevny's			Zhang's		
	TN	TP	AR	TN	TP	AR	TN	TP	AR
PME	59.7	39.2	49.5	48.3	53.5	50.9	50.5	51.8	51.2
ALG1	46.8	53.3	50.1	51.3	52.9	52.1	57.0	47.8	52.4
ALG2	48.6	50.3	49.5	48.9	56.4	52.6	56.5	53.1	54.8
ALG3	45.5	54.4	50.0	49.1	53.3	51.2	60.3	56.1	58.2

non-stego and stego compressed videos. Let P_X and P_Y be the MV probability distributions of non-stego and stego videos, the K-L divergence between the two is calculated as

$$\mathrm{D_{K-L}}(P_X||P_Y) = \sum P_X(e)\log\frac{P_X(e)}{P_Y(e)}. \tag{16}$$

Using the same settings as in 4.1, tests on "coastguard" with increasing embedding strengths are performed and the results of the four methods are plotted in



(a) Xuan's (b) Pevny's

Fig. 8. ROC curves of the steganalyzers using Xuan's and Pevny's features

Fig. 6(d). K-L values of some different sequences are also calculated and recorded in the last column of Table 1.

4.3 Steganalysis

In our steganalytic work, 20 CIF video sequences are trimmed to equal length of 240 frames ("bus" and "stefan" are excluded due to their small sizes). The considered embedding strength for each steganography is 50 bpf. For a given steganography, each 240-frame sequence is compressed with random message embedded to represent the class of stego videos. The other class comprises of the compressed videos of the same sequences with no embedding involved.

In literature, most existing steganalyzers (e.g., [4,10,12,17]) model the videos as successive still images and the embedding procedure as adding independent mean zero Gaussian noises. However, if only a small portion of MVs are slightly altered, the spatial/frequent coefficients will not be directly affected, thus the accuracy of the model will be compromised. To support this conclusion, we have subjected ALG1, ALG2, ALG3 and PME to steganalyzers utilizing Xuan et al's [16] [2] and Pevny et al's [13] [3] features to test their spacial domain detectabilities. In each attack, 16 non-stego and stego compressed video pairs are randomly selected and decompressed to still images to train the classifier and the remaining 4 pairs to test the trained classifier. The classifier is implemented using Chang's support vector machine (SVM) [6] with the polynomial kernel.

To the best of our knowledge, the only specific steganalysis against MV-based steganography was proposed by Zhang et al [18] using features derived from the

[2] A 39-d feature vector formed by statistical moments of wavelet characteristic functions [16].

[3] A 686-d feature vector derived from the second-order subtractive pixel adjacency matrix (SPAM) [13].

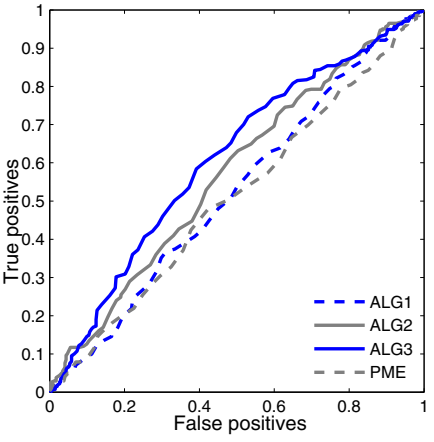


Fig. 9. ROC curves of the steganalyzer using Zhang's features

changes in MV statistical characteristics. With the same SVM classifier mentioned above, in each attack, 16 non-stego and stego compressed video pairs are randomly selected and divided into none-overlapping 6-frame units as described in [18] for training purposes and the rest 4 pairs for testing.

The true negative (TN) rates, true positive (TP) rates and their average accuracy rates (AR) are computed by counting the number of detections in the test sets. The results shown in Table 2 are the arithmetic averages of 20 random attacks. Fig. 8 and Fig. 9 show ROC curves after testing on different data sets. It can be seen that with the considered embedding strength, the steganalyzers with Xuan et al's and Pevny et al's features can not reliably detect MV-based steganography, and PME outperforms its competitors when attacked by the specific steganalyzer.

5 Conclusion and Future Work

In this paper, a novel adaptive video steganography combined with MPEG-4 video compression is proposed. Optimized perturbations are introduced to motion estimation for data hiding. Since our approach targets the internal dynamics of video compression, it is immune to most existing blind steganalyzers. In addition, the PME method shows a good ability in preserving MV statistical characteristics which makes it less detectable to the specific steganalyzer against MV-based steganography. Experimental results show that, satisfactory levels of visual quality and security are achieved with adequate payloads. With PME, steady covert communication can be carried out without incurring much suspicion.

In our future work, the PME would be further optimized by testing on different parity functions and selection rule designing criteria. Besides, methods to improve the computational efficiency and simplify the implementation are also

to be explored. Meanwhile, attempts of further steganalysis are to be carried out using a larger and more diversified database to ensure steganalytic security.

Acknowledgment. This work is supported by the Beijing Natural Science Foundation under the Grant No. 4112063.

References

1. Xvid Codec 1.1.3 (2009), http://www.xvid.org/
2. Aly, H.: Data hiding in motion vectors of compressed video based on their associated prediction error. IEEE Transactions on Information Forensics and Security 6(1), 14–18 (2011)
3. Bellifemine, F., Capellino, A., Chimienti, A., Picco, R., Ponti, R.: Statistical analysis of the 2d-dct coefficients of the differential signal for images. Signal Processing: Image Communication 4(6), 477–488 (1992)
4. Budhia, U., Kundur, D., Zourntos, T.: Digital video steganalysis exploiting statistical visibility in the temporal domain. IEEE Transactions on Information Forensics and Security 1(4), 502–516 (2006)
5. Cachin, C.: An information-theoretic model for steganography. Cryptology ePrint Archive, Report 2000/028 (2000)
6. Chang, C., Lin, C.: Libsvm: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
7. Fang, D.Y., Chang, L.W.: Data hiding for digital video with phase of motion vector. In: Proceedings of IEEE International Symposium on Circuits and Systems, pp. 1422–1425 (2006)
8. Fridrich, J., Goljan, M., Soukal, D.: Perturbed quantization steganography with wet paper codes. In: Proceedings of the 2004 Workshop on Multimedia & Security, MM&Sec 2004, pp. 4–15. ACM, New York (2004)
9. Gormish, M.J., Gill, J.T.: Computation-rate-distortion in transform coders for image compression. In: SPIE Visual Communications and Image Processing, pp. 146–152 (1993)
10. Jainsky, J.S., Kundur, D., Halverson, D.R.: Towards digital video steganalysis using asymptotic memoryless detection. In: Proceedings of the 9th Workshop on Multimedia & Security, MM&Sec 2007, pp. 161–168. ACM, New York (2007)
11. Kutter, M., Jordan, F., Ebrahimi, T.: Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video. Technical report M2281, ISO/IEC document, JTC1/SC29/WG11 (1997)
12. Pankajakshan, V., Doerr, G., Bora, P.K.: Detection of motion-incoherent components in video streams. IEEE Transactions on Information Forensics and Security 4(1), 49–58 (2009)
13. Pevny, T., Bas, P., Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security 5(2), 215–224 (2010)
14. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)
15. Xu, C., Ping, X., Zhang, T.: Steganography in compressed video stream. In: Proceedings of the First International Conference on Innovative Computing, Information and Control, ICICIC 2006, pp. 269–272. IEEE Computer Society, Washington, DC, USA (2006)

16. Xuan, G., Shi, Y.Q., Gao, J., Zou, D., Yang, C., Zhang, Z., Chai, P., Chen, C., 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)
17. Zhang, C., Su, Y.: Video steganalysis based on aliasing detection. Electronics Letters 44(13), 801–803 (2008)
18. Zhang, C., Su, Y., Zhang, C.: A new video steganalysis algorithm against motion vector steganography. In: Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4 (2008)

Soft-SCS: Improving the Security and Robustness of the Scalar-Costa-Scheme by Optimal Distribution Matching

Patrick Bas

CNRS - LAGIS, Ecole centrale de Lille, Villeneuve D'ascq Cedex, France
Patrick.Bas@ec-lille.fr

Abstract. In this paper we propose an extension of the Scalar-Costa-Scheme (SCS), called Soft-SCS, which offers better or equal achievable rates than SCS for the AWGN channel. After recalling the principle of SCS we highlight its secure implementations regarding the Watermarked contents Only Attack, and we also describe the relations between the alphabet size and the secure embedding parameters. Since the gap between the achievable rates of secure-SCS and SCS is important for low Watermark to Noise Ratios (WNR) regimes, we introduce Soft-SCS, a scheme which enables to achieve security by matching a given distribution of watermarked content while minimizing the embedding distortion. The embedding is given by the optimal transport and the distortion is computed using the transportation theory. Contrary to SCS, the distribution of watermarked contents is not piecewise uniform of width (1-α)Δ, but contains affine portions parametrized by a new embedding parameter β used to maximize the robusness of Soft-SCS. As a consequence, the achievable rates of Soft-SCS for low WNR regimes for both its secure and robust implementations are higher than for SCS. Our conclusions are that (1) the loss of performance between the secure and robust implementations of Soft-SCS for WNR regimes smaller than 0 dB is negligible and (2) the robust implementation of Soft-SCS is equal to SCS for WNR regimes over 0 dB.

1 Introduction

Watermarking can be used to convey sensitive information in a secure and robust way. The security of symmetric watermarking techniques relies on the usage of a secret key by both the embedding and decoding schemes. One way to increase the security of the system is to use a different watermarking key for each content to be watermarked, however this solution is practically difficult to implement. For example, if one wants to watermark a database of images, he cannot use different keys for each images because the watermark decoder would have to know the mapping between the images and the keys. Another example is given by the watermarking of digital sequences where the watermark is embedded periodically and has to be decoded all along the sequence. In this practical

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 208–222, 2011.

scenario, the key has to be repeated from time to time in order to enable fast synchronization.

The assumption that a watermarking scheme uses the same key to watermark a set of N_o contents has given birth to a set of security attacks and counter-attacks. The goal of these security attacks is to try to estimate the secret key used to generate the watermark signal, they use Blind Source Separation techniques such as ICA [5,2] and PCA [7,3] or clustering techniques such as K-means [1] and feasible sets [14]. Counter-attacks are however possible through the development of secure watermarking schemes such as Natural Watermarking or its adaptations for Gaussian host [4], or the Scalar-Costa-Scheme (SCS) using specific parameters for uniform hosts. Those different schemes have been proved to be secure under the Watermarked contents Only Attack (WOA) assumption (e.g. the adversary only owns watermarked contents) and for i.i.d. embedded message. In this context the watermarking system can achieve *perfect secrecy* [14] aka *stego-security* [4] which means that the distributions of originals and watermarked contents are identical and that there is no information leakage about the secret key.

The goal of this paper is design a new robust watermarking scheme for uniform host which can be secure under the WOA setup. Section 2 presents SCS, its robust implementations (e.g. enabling to maximize the transmission rate) and its secure implementations (guarantying *perfect secrecy*). The maximum achievable rate for secure implementations is also analyzed for different Watermark to Noise Ratios (WNRs).

Section 3 proposes and extension of SCS called the Soft-Scalar-Costa-Scheme (Soft-SCS) and the embedding and computation of the distortion are detailed. Finally section 4 compares the achievable rates of SCS and Soft-SCS for both their secure and robust versions.

2 Scalar Costa Scheme

2.1 Notations

WCR and WNR denote respectively the Watermark to Content Ratio and the Watermark to Noise Ratio and are expressed in dB. y represents a sample of the watermarked signal, x of the host sample and w of the watermark sample with $y = x + w$. d is the symbol to embed over an alphabet $\mathcal{D}$ and $D = |\mathcal{D}|$. Sample y suffers a AWGN n to produce to attacked sample $z = y + n$.

The subscript $._r$ denotes a *robust* implementation or parameter, e.g. the one maximizing the achievable rates and the subscript $._s$ denotes the *secure* implementation or parameter, e.g. satisfying the constraint of perfect secrecy. Hence SCS_r and SCS_s denote respectively robust and secure implementations of SCS which use respectively parameters α_r and α_s.

2.2 SCS Embedding and Decoding

SCS [9] is built under the hypothesis called the *flat host assumption.* In this setting the distribution of the host signal x is considered as piecewise uniform,

additionally the embedding distortion is very small regarding the host signal, e.g. $\sigma_w^2 \ll \sigma_x^2$. The method uses uniform quantizers of step Δ during the embedding, this means that the distribution of the watermarked contents can be considered as periodical. As in the seminal paper, we will restrict our analysis on one period, e.g for $x \in (-\Delta/2; \Delta/2]$. We denote by $p_x(x)$, $p_y(y)$ and $p_z(z)$ the PDFs of respectively x, y and z, $\otimes$ represents the circular convolution.

To embed a symbol $d \in \mathcal{D}$, SCS extracts the quantization noise q obtained by applying one scalar uniform quantizer Q_Δ of width Δ translated according to d:

$$q(d) = Q_\Delta\left(x - \Delta\left(\frac{d}{D} + k\right)\right) - \left(x - \Delta\left(\frac{d}{D} + k\right)\right), \tag{1}$$

where k denotes the secret key. The watermark signal is given by:

$$w = \alpha q(d), \tag{2}$$

where α is a parameter that is used to maximize the achievable rate. In the sequel, we will assume that we are in the WOA setup and consequently that the secret key is constant. Without loss of generality, we set $k = 0$. The distortion of the embedding is given by

$$\sigma_w^2 = \frac{\alpha^2 \Delta^2}{12}, \tag{3}$$

and the authors have derived an approximation of the embedding parameter maximizing the achievable rate R for a given WNR. The approximation is given by:

$$\alpha_r = \sqrt{\frac{1}{1 + 2.71.10^{-WNR/10}}}. \tag{4}$$

Using the flat host assumption, the rate R is given by the mutual information between the attacked signal and the embedded symbol:

$$R = I(z, d) = -\int_\Delta p_z(z) \log_2 p_z(z) dz + \frac{1}{D} \sum_{d \in \mathcal{D}} \int_\Delta p_z(z|d) \log_2 p_z(z|d) dz. \tag{5}$$

Since the expressions of $p_z(z) = p_y(y) \otimes p_n(n)$ and $p_z(z|d) = p_y(y|d) \otimes p_n(n)$ have no closed-form solutions due to the periodicity of the PDF, they are computed as in [8] by working in Fourier domain using the convolution theorem[1]. The integral term are also thereafter numerically computed.

The decoding is performed by computing the distance $|z - c(d)|$ where $c(d)$ is the closest quantization cell for each of the D quantizers:

$$\hat{d} = \arg\min_d |z - c(d)|. \tag{6}$$

This tantamount to performing a maximum likelihood decoding:

$$\hat{d} = \arg\max_d p(z|d). \tag{7}$$

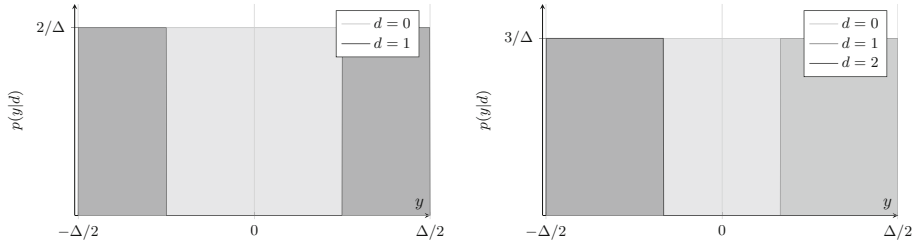
[1] In [13] authors have considered a similar approach in order to compute the achievable rate for Gaussian hosts.

2.3 SCS Secure Modes

As it is mentioned in [14,10], SCS achieves perfect secrecy under the WOA setup for an embedding parameter

$$\alpha_s = \frac{D-1}{D}. \tag{8}$$

Indeed in this case we have $p_y(y) = p_x(x)$ and there is no information leakage about the location of the quantization cells. Additionally, the adversary is unable to distinguish watermarked samples from original ones. Two examples for $D = 2$ and $D = 3$ are illustrated on Fig. 1.



(a) $D = 2$, $\alpha = \frac{1}{2}$ (b) $D = 3$, $\alpha = \frac{2}{3}$

Fig. 1. Distributions of the watermarked contents for the two first secure modes of SCS

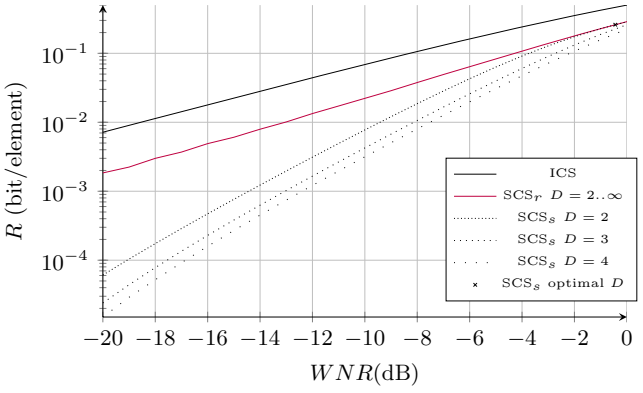
Eq. (8) and (4) imply that one can maximize robustness while assuring perfect secrecy only if $\alpha_s = \alpha_r$, e.g. for a set of "secure" WNR_s equal to

$$WNR_s = -10\log_{10}\left[\frac{1}{2.71}\left(\left(\frac{D}{D-1}\right)^2 - 1\right)\right]. \tag{9}$$

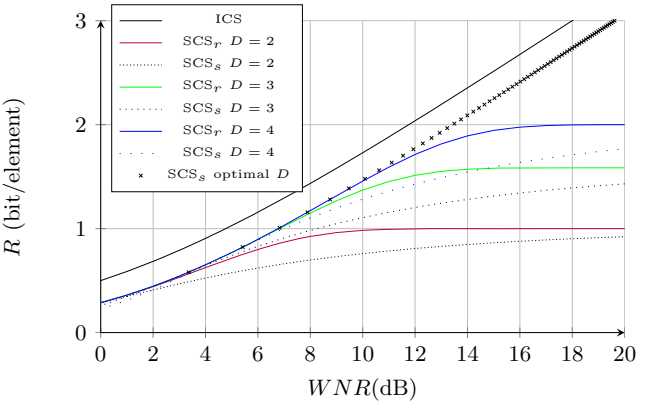
The range of WNR_s starts at $-0.44dB$ for $D = 2$ and $\alpha_s = 1/2$, consequently one way to perform both secure and robust watermarking is to select the alphabet size D which gives a WNR_s which is the closest to the targeted WNR. However SCS doesn't offer efficient solutions for low WNR (e.g. $< -1dB$).

In order to compare the performance of SCS_s and SCS_r we have computed the achievable rates using respectively α_r and α_s for a wide range of WNR and different alphabet size. The comparison is depicted on Fig. 2. All the rates are upper bounded by the Capacity of the Ideal Costa Scheme (ICS) $C_{ICS} = 0.5\log_2(1+10^{WNR/10})$ [6,9]. We can notice (Fig. 2(a)) that the performance gap between SCS_r and SCS_s is important for low WNR and it becomes negligible for high WNR (Fig. 2(b)), provided that the adequate alphabet size is selected. Note also that for a given D the gap between the secure and robust implementations grows with respect with the distance between the used WNR and WNR_s.

The inability of SCS_s to achieve efficient embedding for low WNR is due to the fact that SCS_r select a small embedding parameter α_r whereas SCS_s is lower bounded by $\alpha = 0.5$. The goal of the scheme presented in the next section is to



(a) Low WNR



(b) High WNR

Fig. 2. Achievable rates for secure and robust SCS. The capacity of the Ideal Costa Scheme is also represented.

modify SCS in such a way that the secure embedding provide better rates for low WNR.

3 Soft Scalar-Costa-Scheme

Contrary to classical watermarking embedding schemes, Soft-SCS is based on the principle of *optimal distribution matching*. In this context, the computation of the embedding can be seen as a two stages process. Firstly we set-up the distribution $p_Y(y|d)$ of the watermarked contents, this first step is mandatory if one wants to create an embedding that achieves perfect secrecy. Secondly we compute the embedding that enables to match $p_Y(y|d)$ from the host signal of distribution $p_X(x)$ while minimizing the average distortion. This second step is performed using optimal transport theory (see 3.2).

Because the performances of SCS_s for low WNR are maximized for $D = 2$, the proposed scheme will be studied for binary embedding but could without loss of generality be extended rato D-ary versions.

3.1 Shaping the Distributions of the Watermarked Contents

The rationale of Soft SCS is to mimic the behavior of SCS for $\alpha < 0.5$ while still granting the possibility to have perfect secrecy. This is done by keeping the α parameter (we call it $\tilde{\alpha}$ in order to avoid confusion with the parameter used in SCS) and by adding a second parameter, called β, that will enable to have linear portions in the PDF of watermarked contents. β (respectively $-\beta$) are defined as the slope of the first (respectively the second) linear portions. The cases $\beta = +\infty$ is equivalent to SCS embedding. The differences between the distributions of watermarked contents for SCS and Soft-SCS are depicted on Fig. 3.

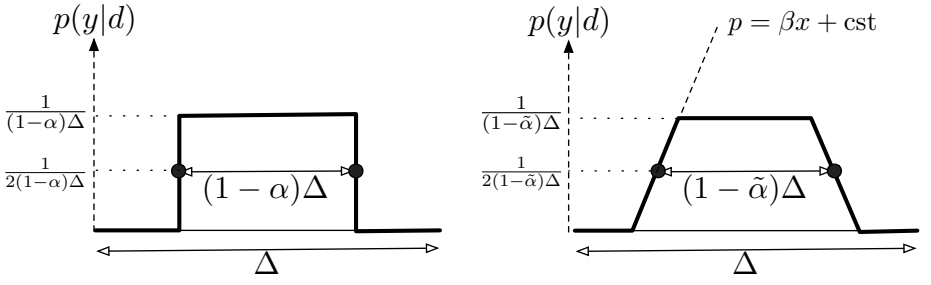


Fig. 3. Comparison between the distributions of SCS and Soft-SCS

In order to fulfill the constraint that $\int_{\Delta} p_Y(y|d, y \in [0; \Delta])dy = 1$, the equation of the first affine portion on $[0; \Delta]$ is given by:

$$p_Y(y|d = 1, y \in [0; \Delta]) = \beta y + \frac{1 - \tilde{\alpha}(1 - \tilde{\alpha})\beta\Delta^2}{2(1 - \tilde{\alpha})\Delta} = \beta y + A, \tag{10}$$

with $A = (1 - \tilde{\alpha}(1 - \tilde{\alpha})\beta\Delta^2)/(2(1 - \tilde{\alpha})\Delta)$ and by symmetry the second affine portion is gives $p_Y(y|d) = \beta(\Delta - y) + A$.

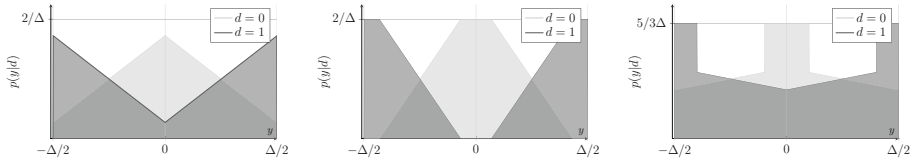
Depending of the values of $\tilde{\alpha}$ and β the distributions of $p_Y(y|d = 1, y \in [0; \Delta])$ for Soft-SCS can have three different shapes and the distributions will either look like a *big-top*, a *canyon* or a *plateau*. For illustration purpose, the 3 configurations are depicted on Fig. 4.

The intervals of the first linear portion (the second being computed by symmetry) and the type of shape are summarized on Table 1, they depend on a limit value of β called β_l which is different for $\tilde{\alpha} < 1/2$ or for $\tilde{\alpha} \geq 1/2$. For canyon and plateau shapes, the uniform portion of the PDF is equal to the one of SCS:

$$p_Y(y|d, y \in [0; \Delta]) = 1/((1 - \tilde{\alpha})\Delta). \tag{11}$$

Table 1. The different shapes of the distributions according to $\tilde{\alpha}$ and β

	$\tilde{\alpha} < 1/2,\ \beta_l = \frac{1}{\tilde{\alpha}(1-\tilde{\alpha})\Delta^2}$	$\tilde{\alpha} \geq 1/2,\ \beta_l = \frac{1}{(1-\tilde{\alpha}^2)\Delta^2}$
$\beta \leq \beta_l$	**Canyon** shape	**Big Top** shape
Domain of the affine portion	$[0; \tilde{\alpha}\Delta]$	$[(2\tilde{\alpha}-1)\Delta/2; \Delta/2]$
$\beta > \beta_l$	**Plateau** shape	
Domain of the affine portion	$\left[\frac{\tilde{\alpha}\Delta}{2} - \frac{1}{2(1-\tilde{\alpha})\beta\Delta}; \frac{\tilde{\alpha}\Delta}{2} + \frac{1}{2(1-\tilde{\alpha})\beta\Delta}\right]$	



(a) *Big Top*, $\tilde{\alpha} = \frac{1}{2}$, $\beta' = 0.4$ (b) *Plateau*, $\tilde{\alpha} = \frac{1}{2}$, $\beta' = 0.6$ (c) *Canyon*, $\tilde{\alpha} = \frac{2}{5}$, $\beta' = 0.1$

Fig. 4. Distributions of the watermarked contents for the 3 different configurations of Soft-SCS

3.2 Embedding Computation and Decoding

The optimal way for computing the embedding that match the distribution of watermarked contents while minimizing the average distortion is to use the transportation theory [15,11]. Given $F_Y(y|d)$ the CDF associated with $p_Y(y|d)$ and $F_X(x)$ the CDF associated with $p_X(x)$, the optimal transport minimizing the average L^2 distance is given by:

$$T(x) = F_Y^{-1} \circ F_X(x), \tag{12}$$

and the distortion by:

$$\sigma_w^2 = \int_0^1 (F_Y^{-1}(x|d) - F_X^{-1}(x))^2 dx. \tag{13}$$

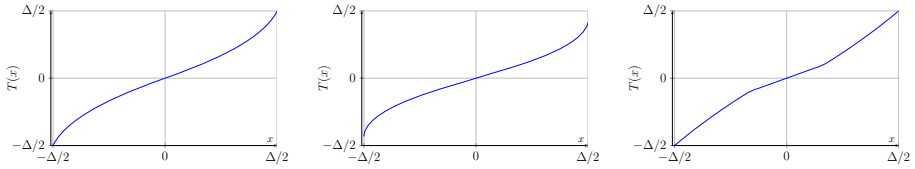
The embedding function $T(.)$ for the different configurations and $d = 1$ are given in Appendix A. Depending of the value of x, the transport is either non-linear affine:

$$T(x) = \frac{\nu_1 + \sqrt{\nu_2 + 2\beta(x - \nu_3)}}{\beta}, \tag{14}$$

or affine:

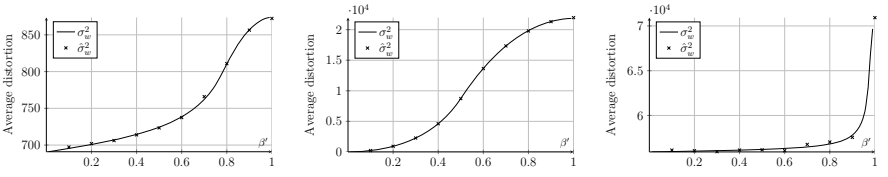
$$T(x) = (1 - \alpha)x + \frac{\alpha\Delta}{2}, \tag{15}$$

where ν_1, ν_2 and ν_3 are constants formulated in Table 2 of appendix A.



(a) $\tilde{\alpha} = \frac{1}{2}$, $\beta' = 0.4$ (b) $\tilde{\alpha} = \frac{1}{2}$, $\beta' = 0.6$ (c) $\tilde{\alpha} = \frac{2}{5}$, $\beta' = 0.1$

Fig. 5. Optimal transport for different configurations of Soft-SCS ($d = 0$)



(a) $\tilde{\alpha} = 0.1$ (b) $\tilde{\alpha} = 0.5$ (c) $\tilde{\alpha} = 0.9$

Fig. 6. Empirical distortions ($\hat{\sigma}_w^2$) computed by Monte-Carlo simulations with 10^6 trials, and closed-form distortions (σ_w^2) for $\Delta = 1024$, and 1024 bins used to compute the distributions

For visualization and parametrization purposes, since β ranges on $\mathbb{R}^+$and depends on Δ, we prefer to use $\beta^{'}$ such that:

$$\beta = 4 \tan\left(\pi\beta^{'}/2\right)/\Delta^2, \tag{16}$$

where $\beta^{'} \in [0, 1($. The shape of the distribution becomes independent of Δ and the couple $\beta^{'} = 0.5$ and $\tilde{\alpha} = 0.5$ corresponds to the case where the distribution $p_Y(y|d)$ is at the junction between the big-top and the plateau. The cases $\beta^{'} = 0$ and $\beta^{'} \to 1$ correspond respectively to $\beta = 0$ and $\beta \to +\infty$.

Figure 5 illustrates different embeddings for $d = 0$ and different configurations of $(\tilde{\alpha}, \beta^{'})$. Note that the embedding for $d \neq 0$ can be easily computed by translating both the host signal and the watermarked one by $\Delta/2$.

The embedding distortion is computed using eq. (13) and contains 2 terms related respectively to the affine and non-linear portions of the embedding. Its close-form is detailed in appendix B. Fig. 6 illustrates the fit between the closed-form formulae and Monte-Carlo simulations.

As for SCS, the decoding is performed using maximum likelihood decoding (7).

4 Performance Analysis

4.1 Secure Embedding

It is easy to show that for $\tilde{\alpha} = \tilde{\alpha}_s = 0.5$ and $D = 2$, Soft-SCS achieves perfect secrecy, the distributions can only have two shapes in this case which are the

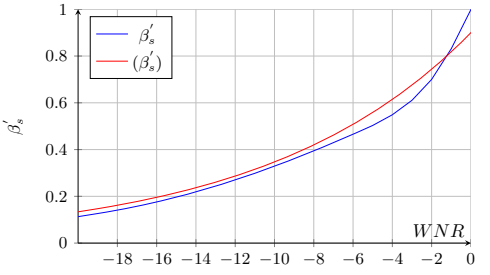


Fig. 7. β_s' and its approximation

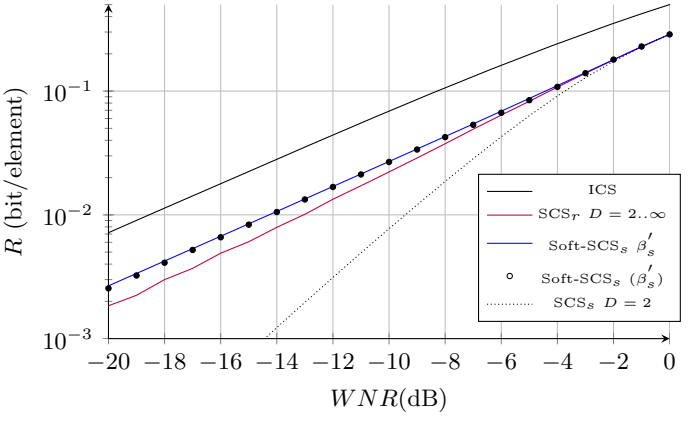
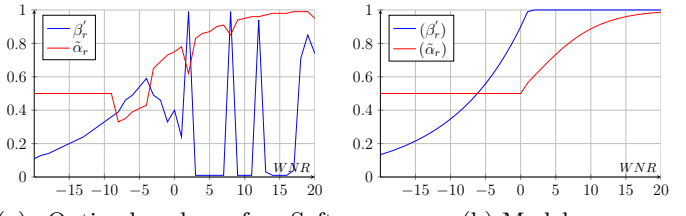


Fig. 8. Achievable rate of Secure Soft-SCS

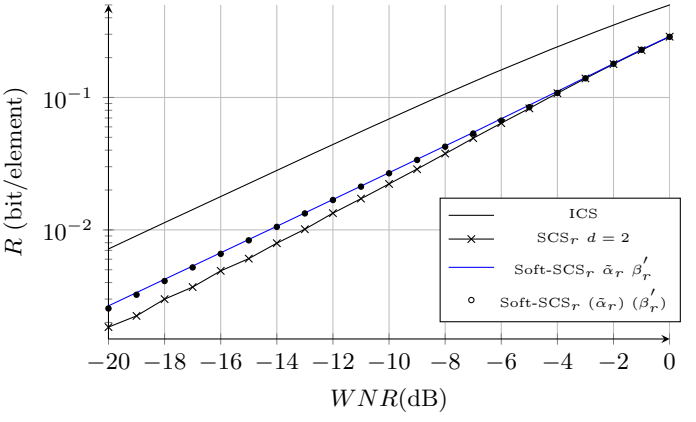


(a) Optimal values for Soft-SCS$_r$.

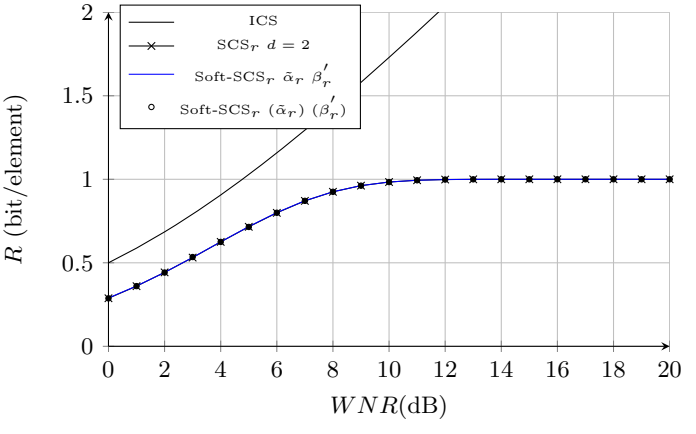
(b) Model

Fig. 9. Approximation of $\tilde{\alpha}_r$ and β_t'

big-top and the *plateau* illustrated on Fig. 4(a) and Fig. 4(b) respectively. Using numerical optimization, we compute for a given WNR the value of β' which enables to maximize the achievable rate (5) and obtain β_s'. The result of this optimization, and its approximation using least square regression is given on Fig. 7. The approximation gives



(a) Low WNR



(b) High WNR

Fig. 10. Achievable rates for Soft-SCS$_r$

$$\begin{cases} (\beta_s^{'}) = 0.9 \times 1.1^{WNR} & , WNR < 0\,dB \\ (\beta_s^{'}) = 1 & , WNR \geq 0\,dB. \end{cases} \tag{17}$$

which means that Soft-SCS$_s$ and SCS$_s$ differ only for $WNR < 0\,dB$.

The achievable rates of Soft-SCS$_s$ are depicted on Fig. 8and are compared with SCS$_r$ and SCS$_s$. We notice that Soft-SCS$_s$ not only outperforms the secure version of SCS but also the robust one. The gap between Soft-SCS$_s$ and SCS increases with respect to the noise power and is null for $WNR = -0.44\,dB$. The figure shows also that the gap between the implementation for the optimal value of $\beta_s^{'}$ and the approximation given in (17) is negligible.

4.2 Robust Embedding

The same methodology is applied without the security constraint in order to obtain the robust configuration of Soft-SCS. This time the rate has to be maximized according to $\tilde{\alpha}$ and $\beta^{'}$ and their values after the numerical optimization are depicted on Fig. 9. For $WNR > -0\,dB$, the values of $\beta_r^{'}$ oscillate between $\beta^{'} = 0$ and $\beta^{'} = 1$ which are two variations of SCS (the slope being null with a *big top* configuration or the slope being infinite *plateau* configuration.

Surprisingly we notice that there is no difference between Soft-SCS$_r$ and Soft-SCS$_s$ for $WNR < -9\,dB$, the common optimal value being $\tilde{\alpha} = 0.5$ and the difference between the two schemes is negligible for $WNR < -0\,dB$. For high WNR however, the approximation is identical to SCS$_r$ with $(\tilde{\alpha}_r) = \alpha_r$ (eq . 4) and $(\beta_r^{'}) = 1$. We can conclude that the implementation Soft-SCS$_r$ behaves as Soft-SCS$_w$ for low WNR and as SCS$_r$ for high WNR.

5 Conclusion and Perspectives

We have proposed in this paper an adaptation of the Scalar Costa Scheme based on the principle of optimal distribution matching. The computation of the embedding needs (1) to choose the distribution of the watermarked contents and (2) to compute the optimal mapping from the host to the watermarked contents. This method enables to outperform SCS both for its secure and robust implementations for $WNR \leq 0\,dB$.

Contrary to a spread idea that robustness and security are antagonist constraints in watermarking, we have shown in this study that there exists watermarking schemes that can guaranty perfect secrecy while maximizing the achievable rate. SCS$_s$ can be used for high WNR with appropriate dictionary size, $\alpha_s = (D-1)/D$; and Soft-SCS$_s$ can be used for low WNR , $\tilde{\alpha}_s$ and β_s and provide negligible loss of rate.

However, one can argue that for low WNR regimes the rates is rather small and that one system involving redundancy or error correction should be used in order to increase the reliability of the decoded symbols. This solution has to be employed in a very cautious way since the redundancy might compromise the security of the whole system [12]. Future works will investigate this direction if there is a way to perform secure coding.

A Embedding Formulas for Soft-SCS

Here, for the shake of simplicity the $\tilde{\alpha}$ parameter of Soft-SCS is written α.

A.1 Plateau Shape ($\beta \geq \beta_l$),

The CDF is given by, for $\left[\frac{\alpha\Delta}{2} - \frac{1}{2(1-\alpha)\beta\Delta}; \frac{\alpha\Delta}{2} + \frac{1}{2(1-\alpha)\beta\Delta}\right]$ by:

$$F_Y(x) = \frac{\beta}{2}\left(x + \frac{A}{\beta}\right)^2,$$

and the inverse function on $[0; y_1]$ is given by:

$$F_Y^{-1}(x) = \frac{-A + \sqrt{2\beta x}}{\beta}.$$

with

$$F_Y\left(\frac{\alpha\Delta}{2} + \frac{1}{2(1-\alpha)\beta\Delta}\right) = \frac{1}{2(1-\alpha)^2\beta\Delta^2} = y_1.$$

- The optimal transport on $[0; y_1\Delta]$ is given by ($y_1\Delta$ corresponds to the point were $F_X(x) = y_1$):

$$T(x) = F_Y^{-1} \circ F_X(x) = \frac{-A + \sqrt{2\beta x/\Delta}}{\beta}.$$

On $x \in \left[\frac{\alpha\Delta}{2} + \frac{1}{2(1-\alpha)\beta\Delta}, \frac{\Delta}{2}\right]$, we now have:

$$F_Y(x) = \frac{1}{(1-\alpha)\Delta}x - \frac{\alpha}{2(1-\alpha)},$$

The optimal transport on $[y_1\Delta, \frac{\Delta}{2}]$ is given by:

$$T(x) = F_Y^{-1} \circ F_X(x) = (1-\alpha)x + \frac{\alpha\Delta}{2}.$$

A.2 Canyon Shape ($\alpha < 1/2, \beta < \beta_l$)

for $x \in [0; \alpha\Delta]$ and $\alpha < 0.5$, the **CDF** is given by:

$$F_Y(x) = \frac{\beta}{2}x^2 + Ax$$

The inverse function is given by for $x \in [0; y_2]$, with $y_2 = F_Y(\alpha\Delta) = \beta\alpha^2\Delta^2/2 + \alpha\Delta A$:

$$F_Y^{-1}(x) = \frac{-A + \sqrt{A^2 + 2\beta x}}{\beta}.$$

- The optimal transport is given on $[0; y_2\Delta]$ by ($y_2\Delta$ corresponds to the point were $F_X(x) = y_2$):

$$T(x) = F_Y^{-1} \circ F_X(x) = \frac{-A + \sqrt{A^2 + 2\beta x/\Delta}}{\beta}.$$

On $[\alpha\Delta; \Delta/2]$, we now have:

$$F_Y(x) = \frac{1}{(1-\alpha)\Delta}x - \frac{\alpha}{2(1-\alpha)},$$

The optimal transport on $[y_2\Delta, \frac{\Delta}{2}]$ is given by:

$$T(x) = F_Y^{-1} \circ F_X(x) = (1-\alpha)x + \frac{\alpha\Delta}{2}.$$

A.3 Big Top Shape ($\alpha > 1/2, \beta < \beta_l$)

for $x \in [(2\alpha-1)\Delta/2; \Delta/2]$ and $\alpha > 0.5$, the **CDF** is given by:

$$F_Y(x) = \frac{\beta}{2}x^2 + Ax - (2\alpha-1)^2\beta\Delta^2/8 - A(2\alpha-1)\Delta/2 = \frac{\beta}{2}x^2 + Ax + C,$$

with $C = -(2\alpha-1)^2\beta\Delta^2/8 - A(2\alpha-1)\Delta/2$. The inverse function is given by for $x \in [0; 1/2]$:

$$F_Y^{-1}(x) = \frac{-A + \sqrt{A^2 + 2\beta(x-C)}}{\beta}.$$

The optimal transport is given on $[0; \Delta/2]$ by:

$$T(x) = F_Y^{-1} \circ F_X(x) = \frac{-A + \sqrt{A^2 + 2\beta(x/\Delta - C)}}{\beta}.$$

B Distortions Formulas for Soft-SCS

$$\sigma_w^2 = 2\int_0^{1/2} (F_Y^{-1}(x) - F_X^{-1}(x))^2 dx$$

$$\begin{aligned}\sigma_w^2 &= 2\int_{x_0}^{x_1} \left(\frac{\nu_1 + \sqrt{\nu_2 + 2\beta(x-\nu_3)}}{\beta} - \Delta x\right)^2 dx \\ &\quad +2\int_{x_1}^{x_2} \left((1-\alpha)\Delta x + \frac{\alpha\Delta}{2} - \Delta x\right)^2 dx \\ &= I_1 + I_2.\end{aligned}$$

The values of x_1 and x_2 depend of the configuration of the PDF and their closed-form are given in Table 2.

Table 2. The different configurations for the computation of the distortion

	$\alpha < 1/2$	$\alpha \geq 1/2$
$\beta < \beta_l$	**Canyon** shape	**Big Top** shape
(x_0, x_1, x_2)	$(0\,;\, \beta\alpha^2\Delta^2/2 + \alpha\Delta A\,;\, 1/2)$	$(0\,;\, 1/2\,;\, 1/2)$
(ν_1, ν_2, ν_3)	$(-A, A^2, 0)$	$(-A, A^2, \nu_3)$
$\beta > \beta_l$	**Plateau** shape	**Plateau** shape
(x_0, x_1, x_2)	$(0\,;\, 1/\left(2(1-\alpha)^2\beta\Delta^2\right)\,;\, 1/2)$	$(0\,;\, 1/\left(2(1-\alpha)^2\beta\Delta^2\right)\,;\, 1/2)$
(ν_1, ν_2, ν_3)	$(-A, 0, 0)$	$(-A, 0, 0)$
β_l	$\frac{1}{\alpha(1-\alpha)\Delta^2}$	$\frac{1}{(1-\alpha^2)\Delta^2}$

I_1 and I_2 are given by:

$$I_1 = 2(\Delta^2 \left[\frac{x^3}{3}\right]_{x_0}^{x_1} + \frac{2-2\Delta\nu_1}{\beta}\left[\frac{x^2}{2}\right]_{x_0}^{x_1} + \frac{2\nu_1}{3\beta^3}\left[(\nu_2 - 2\beta\nu_3 + 2\beta x)^{3/2}\right]_{x_0}^{x_1} + I_3 + \frac{\nu_1^2 + \nu_2 - 2\beta\nu_3}{\beta^2}(x_1 - x_0))$$

with

$$I_3 = -\frac{2\Delta}{3\beta^2}\left[x\,(\nu_2 - 2\beta\nu_3 + 2\beta x)^{3/2}\right]_{x_0}^{x_1} + \frac{2\Delta}{15\beta^3}\left[(\nu_2 - 2\beta\nu_3 + 2\beta x)^{5/2}\right]_{x_0}^{x_1},$$

and

$$I_2 = \frac{2\alpha^2\Delta^2}{3}\left[\left(x_2 - \frac{1}{2}\right)^3 - \left(x_1 - \frac{1}{2}\right)^3\right].$$

References

1. Bas, P., Doërr, G.: Evaluation of an optimal watermark tampering attack against dirty paper trellis schemes. In: ACM Multimedia and Security Workshop, Oxford, UK (September 2008)
2. Bas, P., Hurri, J.: Vulnerability of dm watermarking of non-iid host signals to attacks utilising the statistics of independent components. IEE Proceeding of Transaction on Information Security 153, 127–139 (2006)
3. Bas, P., Westfeld, A.: Two key estimation techniques for the Broken Arrows watermarking scheme. In: MM&Sec 2009: Proceedings of the 11th ACM Workshop on Multimedia and Security, pp. 1–8. ACM, New York (2009)
4. Cayre, F., Bas, P.: Kerckhoffs-based embedding security classes for WOA data-hiding. IEEE Transactions on Information Forensics and Security 3(1) (March 2008)
5. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing Special Issue "Supplement on Secure Media II" (2005)
6. Costa, M.: Writing on dirty paper. IEEE Trans. on Information Theory 29(3), 439–441 (1983)
7. Doërr, G.J., Dugelay, J.L.: Danger of low-dimensional watermarking subspaces. In: ICASSP 2004, 29th IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, May 17-21 (May 2004)
8. Eggers, J., Su, J., Girod, B.: A blind watermarking scheme based on structured codebooks. In: IEE Colloquium Secure Images and Image Authentication, London, UK, pp. 4/1–4/6 (April 2000)
9. Eggers, J.J., Buml, R., Tzschoppe, R., Girod, B.: Scalar costa scheme for information embedding. IEEE Trans. on Signal Processing 51(4), 1003–1019 (2003)
10. Guillon, P., Furon, T., Duhamel, P.: Applied public-key steganography. In: Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV, San Jose, CA, USA, vol. 4675 (January 2002)

11. Mathon, B., Bas, P., Cayre, F., Macq, B.: Optimization of natural watermarking using transportation theory. In: MM&Sec 2009: Proceedings of the 11th ACM Workshop on Multimedia and Security, pp. 33–38. ACM, New York (2009)
12. Pérez-Freire, L., Pérez-González, F., Furon, T., Comesaña, P.: Security of lattice-based data hiding against the Known Message Attack. IEEE Transactions on Information Forensics and Security 1(4), 421–439 (2006)
13. Pérez-Freire, L., Pérez-González, F., Voloshynovskiy, S.: Revealing the true achievable rates of scalar costa scheme. In: 2004 IEEE 6th Workshop on Multimedia Signal Processing, pp. 203–206. IEEE, Los Alamitos (2004)
14. Pérez-Freire, L., Pérez-González, F.: Spread spectrum watermarking security. IEEE Transactions on Information Forensics and Security 4(1), 2–24 (2009)
15. Villani, C.: Topics in Optimal Transportation. American Mathematical Society, Providence (2003)

Improving Tonality Measures for Audio Watermarking

Michael Arnold, Xiao-Ming Chen, Peter G. Baum, and Gwenaël Doërr

Technicolor – Security & Content Protection Labs
firstname.lastname@technicolor.com

Abstract. Psychoacoustic models are routinely used in audio watermarking algorithms to adjust the changes induced by the watermarking process to the sensitivity of the ear. The performances of such models in audio watermarking applications are tightly related to the determination of tonal and noise-like components. In this paper, we present an improved tonality estimation and its integration into a psychoacoustic model. Instead of conventional binary classification, we exploit bi-modal prediction for more precise tonality estimation. Experimental results show improved robustness of the considered audio watermarking algorithm integrating the new tonality estimation, while preserving the high quality of the audio track.

1 Introduction

A major requirement in audio watermarking is to guarantee the quality of a watermarked copy. An audible watermark would essentially make the audio track useless in most applications. For this reason, exploiting psychoacoustic models is one of the main cornerstones while designing watermarking system for audio so as to preserve high quality. Due to the interplay between fidelity and robustness constraints, higher watermark energy at a fixed quality setting naturally results in increased robustness.

Psychoacoustic models share a number of elementary modules e.g. time/frequency mapping via the short-time Fourier transform (STFT) [1], power spectrum estimation, tonal component detector and calculation of individual masking thresholds aka. perceptual slacks. Almost every psychoacoustic model includes a module that classifies frequency bins as tone- or noise-like, or that scores the tonality of a frequency bin. The performances of the overarching psychoacoustic model are intimately related to the reliability of this module. It is therefore important in many applications e.g. audio indexing which attempts at extracting perceptually significant components from the audio signal, or audio coding which exploits psychoacoustic criteria to discard insignificant components of the signals.

Existing tonality estimation methods fall into two categories: hard-decision classifiers with binary detection results regarding the tonality vs. soft-decision classifiers which assign a tonality score in $[0, 1]$ to a frequency bin. A well-known binary classifier is the tonal component detector used in the psychoacoustic model 1 of ISO-MPEG [2]. It is a so-called *intra-frame* method, i.e. each audio frame is processed individually, that operates in the frequency domain and inspects the relative magnitudes of neighbor frequency bins. Another intra-frame tonal detector is the Spectral Flatness Measure (SFM) introduced by Johnston in his perceptual transform coder [3]. It estimates the

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 223–237, 2011.

flatness of the spectrum (see [4]) by reducing spectrum information contained in a selected critical band to a single continuous tonality score in [0, 1]. In contrast, *inter-frame* methods consider the time evolution of the tonality over several audio frames. An example is the Unpredictability Measure (UM) defined in the psychoacoustic model 2 of ISO-MPEG [2] which measures the predictability error of each frequency bin based on the information of the two previous frames.

Reliable tonality estimation leads to more accurate masking threshold, which can considerably improve the performances of the psychoacoustic model. Subsequently, the overall embedded watermark energy may be increased, naturally resulting in improved robustness. In this paper, we present an improved approach for tonality estimation together with its integration within a baseline psychoacoustic model, which improves the robustness of the considered audio watermarking system without sacrificing perceptual fidelity. Section 2 briefly describes the baseline watermark embedding algorithm used in this paper. Section 3 then fully details the proposed tonality estimation method as well as its integration in the psychoacoustic model. The resulting modified psychoacoustic model is evaluated thoroughly in Section 4. It includes subjective evaluation of the audio quality with listening tests and extensive benchmarking of the watermark robustness. Section 5 summarizes the findings of the paper and provide insight for future research.

2 Audio Watermarking Embedding

2.1 Analysis-Synthesis Framework for Audio Signals

Audio signals are quasi-stationary within a short time period of e.g. 2-50 ms. Additionally, the human auditory system somehow performs a time-frequency analysis of acoustic signals (see [5]). As a result, it is common practice in audio processing to apply a short-time Fourier transform to obtain a time-frequency representation of the signal so as to 'mimic' the behavior of the ear.

The STFT consists in (i) segmenting the input signal $\mathbf{x}$ in B-samples long frames $\mathbf{x}_n$ using a sliding window with a hop-size of R samples, and (ii) applying the discrete Fourier transform (DFT) to each frame after multiplication with an analysis window. This *analysis phase* results in a collection of DFT-transformed windowed frames $\tilde{\mathbf{X}}_n$ which can then be input to the subsequent audio processing primitive, should it be lossy audio coding or watermarking.

At the other end, the modified DFT-transformed frames $\tilde{\mathbf{Y}}_n$ output by the audio processing application are used to reconstruct the audio signal during the so-called *synthesis phase*. In a nutshell, the frames are inverse transformed, multiplied by a synthesis window that suppresses audible artifacts by fading out spectral modifications at frame boundaries, and add the resulting frames together with the appropriate time offset.

The combination of (i) the segmenting-windowing-DFT in the analysis phase and (ii) the IDFT-windowing overlap-add in the synthesis phase is the so-called *weighted overlap add* (WOLA) technique which is summarized in Figure 1. Note that, in the remainder of the article, we will refer to embedding the watermark in between the analysis and synthesis phases as watermarking in the WOLA domain.

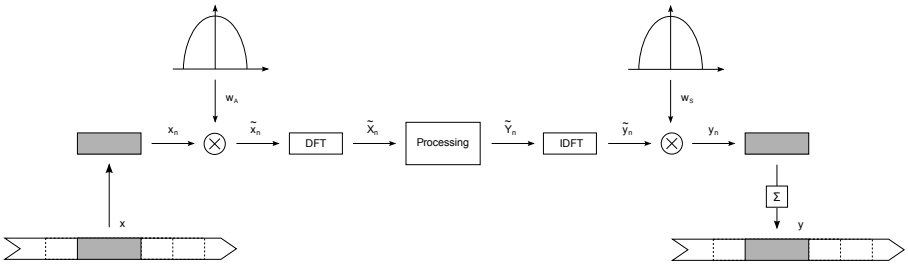


Fig. 1. Analysis-Synthesis processing framework

2.2 Watermarking in the WOLA Domain

Over the last decade, a number of audio watermarking techniques exploiting the analysis-synthesis framework have been proposed [6,7]. Hereafter, we will focus on a watermarking system [8] which basically introduces a watermark by quantizing the phase of WOLA coefficients while maintaining psychoacoustic fidelity. A correlation-based detector in the time domain is then exploited to retrieve the watermark.

Phase Quantization-Based Embedding. The baseline audio watermarking system detailed in [8] quantizes the phase of WOLA coefficients. In other words, the embedding process consists in (i) extracting the phase φ_n of WOLA coefficients from incoming transformed frames $\tilde{\mathbf{X}}_n$ and arranging them sequentially in a 1-D signal φ[1], (ii) applying the quantization based embedding algorithm to obtain the watermarked phases ψ, and (iii) segmenting this signal in B-samples long frames ψ_n to reconstruct the watermarked transformed frames $\tilde{\mathbf{Y}}_n$ which can be subsequently inverse transformed back to the time domain.

Assuming that the system embeds symbols taken from an A-ary alphabet $\mathcal{A}$, the embedding process can be written:

$$\psi[i] = \boldsymbol{\theta}_{\mathbf{a},K}[i], \quad \mathbf{a} \in \mathcal{A}, \quad i \in S.B.\mathbb{N} + [0 : S.B[, \tag{1}$$

where $\boldsymbol{\theta}_{\mathbf{a},K}$ is a sequence of reference angles in $[-\pi, \pi)$ associated to the symbol $\mathbf{a}$, pseudo-randomly generated from a secret key K [9]. The parameter S indicates that each symbol may be spread across several WOLA frames to guarantee robustness. This is a pure replacement watermarking strategy: the output angles ψ are independent of the input angles φ derived from the host signal.

Psycho-Acoustic Adaptation. The straightforward strategy given in Equation (1) has a major shortcoming: it introduces very audible artifacts. It is therefore necessary to slightly adjust the embedding protocol so as to accommodate for the sensitivity of human auditory system.

[1] Within this paper, all angles or angle differences are assumed to lie in the interval $[-\pi, \pi)$ after appropriate modulo-2π operations, if not otherwise stated.

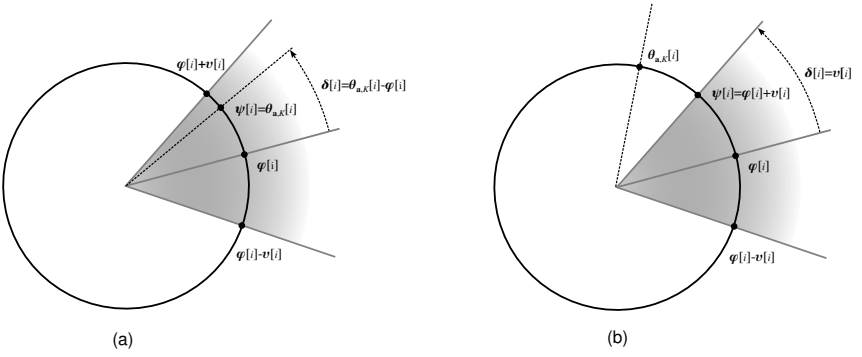


Fig. 2. Geometrical illustration of the embedding process. The phase $\boldsymbol{\varphi}[i]$ is moved towards the intended target value $\boldsymbol{\theta}_{\mathbf{a},K}[i]$ while making sure that the embedding distortion $\boldsymbol{\delta}[i]$ never exceeds the threshold $\boldsymbol{\upsilon}[i]$ recommended by the perceptual model. In case (a), the target angle is close enough to be reached whereas, in case (b), the embedding process is bridled by the perceptual constraint.

First, it is common practice to exclude samples outside a specified frequency band from the watermarking process i.e.

$$\boldsymbol{\psi}[i] = \boldsymbol{\varphi}[i], \quad i \in B.\mathbb{N} + \Big\{[0 : \xi_l[\cup]\xi_h : B/2]\Big\}. \tag{2}$$

Angles below frequency tap ξ_l are discarded due to their high audibility whereas angles above frequency tap ξ_h are ignored because of their high variability. For the remaining angles, the embedding process is modified so that the embedding distortion, as measured by the angle difference $\mathring{\boldsymbol{\delta}}[i] = |\boldsymbol{\psi}[i] - \boldsymbol{\varphi}[i]|$, remains below psycho-acoustic slacks $\boldsymbol{\upsilon}[i] \in [0, \pi]$ obtained after spectra analysis [8]. This can be formally written as:

$$\begin{aligned} &\boldsymbol{\psi}[i] = \boldsymbol{\varphi}[i] + \frac{\mathring{\boldsymbol{\delta}}[i]}{|\mathring{\boldsymbol{\delta}}[i]|} \min\Big\{|\mathring{\boldsymbol{\delta}}[i]|, \boldsymbol{\upsilon}[i]\Big\}, \quad i \in B.\mathbb{N} + [\xi_l : \xi_h] \\ &\text{with} \quad \mathring{\boldsymbol{\delta}}[i] = \boldsymbol{\theta}_{\mathbf{a},K}[i] - \boldsymbol{\varphi}[i]. \end{aligned} \tag{3}$$

Controlling distortion this way allows to guarantee that the introduced changes are strictly inaudible. For clarity, Figure 2 provides a geometrical interpretation of the embedding process. In a nutshell, the embedding algorithm first identifies the shortest path to reach the target angle $\boldsymbol{\theta}_{\mathbf{a},K}[i]$ and moves towards it until (i) it reaches the specified target value, or (ii) it reaches the psycho-acoustic threshold $\boldsymbol{\upsilon}[i]$ specified by the enforced perceptual model.

Correlation-Based Detection in the Time Domain. At the receiver side, an audio signal $\mathbf{z}$ is presented to the detector that may, or may not contain a watermark. For each potential symbol $\mathbf{a}$, an array of correlation scores is computed:

$$\boldsymbol{\rho}_{\mathbf{z},\mathbf{a}}[l] = \frac{1}{S.B} \sum_{i=0}^{S.B-1} \check{\mathbf{z}}[i]\check{\mathbf{r}}_{\mathbf{a},K}[i + l], \quad l \in [-(S.B - 1) : S.B - 1] \tag{4}$$

where $\check{\mathbf{z}}$ its the whitened version of the tested audio signal $\mathbf{z}$, $\check{\mathbf{r}}_{\mathbf{a},K}$ is the whitened version of a time-domain reference signal $\mathbf{r}_{\mathbf{a},K}$ associated to the reference angles $\boldsymbol{\theta}_{\mathbf{a},K}$, and l is the correlation lag. In the absence of watermark, this array of correlation scores should be normally distributed with zero mean. A statistical watermark detector is therefore set in place based on assess how much $\boldsymbol{\rho}_{\mathbf{z},\mathbf{a}}$ deviates from a Gaussian distribution [10].

3 Psychoacoustic Models and Tonality Estimation

3.1 Baseline Psychoacoustic Model

Psychoacoustic models conventionally determine the masking curve of an audio signal by means of (i) determining the sound pressure levels of maskers, (ii) calculating individual masking thresholds taking simultaneous and/or temporal masking effects into account, (iii) combining individual thresholds to get the global thresholds. In our baseline system [8], it is derived from the model 1 of ISO-MPEG [2] with three enhancements: an attack detection module, an altered tonal component detector, and a noise identification block. The perceptual slacks computation process is fully represented in Figure 3.

Attack Detection Module. For a time frame with a quiet section followed by an abrupt increase in audio energy, the calculated masking threshold would lead to smearing of the watermarking energy over the whole frame, resulting in so-called pre-echoes in the quiet portion. To circumvent this problem, the increase in average power between two adjacent frames is compared to a threshold. If the increase is larger than the threshold, no embedding will be carried out for the current frame [8].

Separation of Tonal, Noise and Non-tonal Components. The separation process is performed in the WOLA domain. The algorithm first detects local maxima of the magnitude spectrum as potential peaks of tonal components:

$$\mathcal{P} = \Big\{ i \in [0:B] \,\Big|\, \mathbf{A}_n[i-1] < \mathbf{A}_n[i] \;\wedge\; \mathbf{A}_n[i] \geq \mathbf{A}_n[i+1] \Big\}, \tag{5}$$

where $\mathbf{A}_n[i] = |\tilde{\mathbf{X}}_n[i]|$ is the amplitude of the spectrum $\tilde{\mathbf{X}}_n$ with $|.|$ denoting the magnitude of a complex number. When using the logarithmic scale to express a quantity in decibels (dB), the subscript notation $_{\mathrm{dB}}$ will be used.

In order to eliminate the influence from noise-like variations, local maxima are compared to the magnitudes of surrounding spectral lines. A local maximum $i^* \in \mathcal{P}$ is considered as tonal if

$$\begin{aligned} &\mathbf{A}_n[i^*]_{\mathrm{dB}} - \mathbf{A}_n[i^* - k]_{\mathrm{dB}} \geq 1, \quad \forall\, k \in \{\pm 1 \ldots \pm 4\} \\ \wedge\; &\mathbf{A}_n[i^*]_{\mathrm{dB}} - \mathbf{A}_n[i^* - k]_{\mathrm{dB}} \geq 7, \quad \exists\, k \in \{\pm 1 \ldots \pm 4\}. \end{aligned} \tag{6}$$

In comparison to the MPEG psychoacoustic model 1 this is a less stringent condition resulting in more peaks detected. Due to the asymmetry of the masking behavior, i.e. noise masks better than tonal components [11], the selection of additional tonal components is a more conservative approach leading to lower individual masking thresholds.

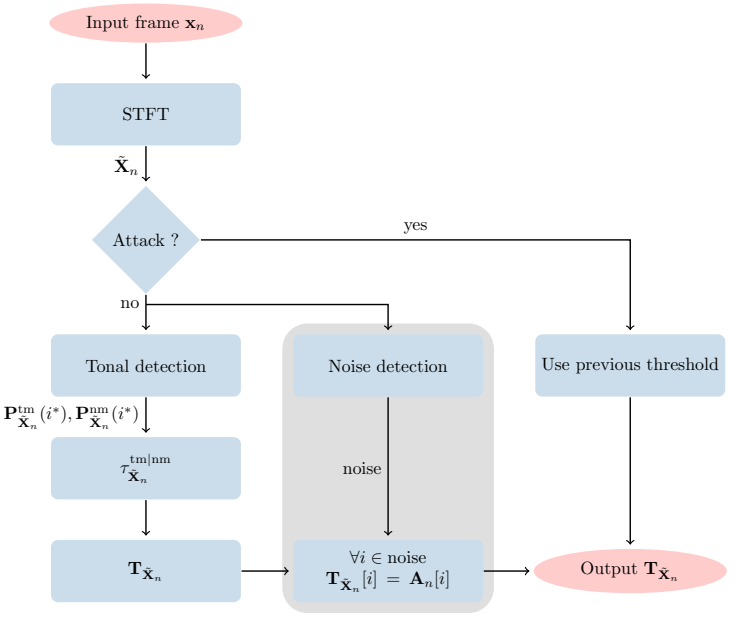


Fig. 3. Masking threshold computation with integrated noise detection

A tonal component is then identified as the group consisting of the peak i^* and its neighboring frequency bins to the next minima. It is characterized by the sound pressure level $\mathrm{P}^{\mathrm{tm}}_{\tilde{\mathbf{X}}_n}(i^*)$ of the tonal masker obtained by summarizing the energy of the bins belonging to the tonal component:

$$\mathrm{P}^{\mathrm{tm}}_{\tilde{\mathbf{X}}_n}(i^*) = \sum_{k=i^*_-}^{i^*_+} \mathbf{A}_n[k]^2, \tag{7}$$

where (i^*_-, i^*_+) denotes the bin indices of the next minima.

In a second step, the noise-like nature of the spectral bins not considered as tonal is further evaluated by investigating their evolution over time. If the relative magnitude slope variation for a frequency bin over successive frames is high compared to the mean value, the frequency bin is assumed to be noisy [8].

The last step operates by critical band[2]. Remaining spectral lines within each critical band compose a single non-tonal component. Its sound pressure level, denoted as $\mathrm{P}^{\mathrm{nm}}_{\tilde{\mathbf{X}}_n}(i^*)$, equals to the sum of the energy of left spectral lines within the critical band. The non-tonal component is attached to the center of its associated critical band.

Individual Masking Threshold Computation. The masking threshold at frequency bin index i due to a masker at the frequency bin index i^* is evaluated as

$$\tau^{\mathrm{tm|nm}}_{\tilde{\mathbf{X}}_n}(i, i^*)_{\mathrm{dB}} = \mathrm{P}^{\mathrm{tm|nm}}_{\tilde{\mathbf{X}}_n}(i^*)_{\mathrm{dB}} + \mathrm{f}^{\mathrm{tm|nm}}(\mathrm{z}(i^*)) + \mathrm{g}(\mathrm{z}(i), \mathrm{z}(i^*)), \tag{8}$$

[2] Bark defined 24 critical bands of hearing. Within each band, Bark's index z(.) operates a non-linear mapping between the Bark scale and the conventional frequency scale.

where z(i) denotes the Bark value for the i-th frequency bin. The functions $\mathrm{f}^{\mathrm{tm|nm}}(.)$ and g(., .) denote respectively the masking index and masking spread, defined in [2,12].

Global Masking Threshold Computation. The global masking threshold $\mathbf{T}_{\tilde{\mathbf{X}}_n}(i)$ for the i-th frequency bin is obtained by adding the individual masking thresholds due to tonal and non-tonal maskers

$$\mathbf{T}_{\tilde{\mathbf{X}}_n}[i] = \sum_{k=1}^{N_t} \tau^{\mathrm{tm}}_{\tilde{\mathbf{X}}_n}(i, i^*_k) + \sum_{k=1}^{N_c} \tau^{\mathrm{nm}}_{\tilde{\mathbf{X}}_n}(i, i^*_k), \tag{9}$$

where N_t denotes the number of tonal components and N_c the number of critical bands. The global masking threshold indicates how large the watermark energy is allowed for a specific frequency bin without introducing audible artifacts. The threshold in quiet segments is not taken into account in contrast to model 1 of ISO-MPEG, which prevents uncovering the structure of the watermark in silent segments of the audio stream. Finally, for all spectral bins identified as noisy the global threshold simply is set to the signal level $\mathbf{T}_{\tilde{\mathbf{X}}_n}[i] = \mathbf{A}_n[i]$. In other words, for noisy frequency bins the energy of watermark is maximized. The phase perceptual slacks $\boldsymbol{\upsilon}[i]$ used in Equation (3) can then be simply derived as follows [8]:

$$\boldsymbol{\upsilon}[i] = 2.\arcsin\left(\frac{\mathbf{T}_{\tilde{\mathbf{X}}_n}[i]}{\mathbf{A}_n[i]}\right). \tag{10}$$

3.2 Improved Tonality Estimation

The classification of frequency bins into tonal or noise components is critical for a psychoacoustic model. The tonal and noise classification method in Section 3.1 is a binary decision at each frequency bin. As a result, noise detection has to be performed very carefully, because mis-interpretation of a component as noise can result in audible artifacts. In contrast, assigning a tonality measure to all frequency bins – not only the identified noise components – could increase the overall embedded watermark energy and hence the robustness. Our new method replaces the noise detection module in Figure 3 with a tonality estimator combining intra- and inter-frame estimator for the magnitudes and phases of the spectrum. The new architecture is depicted in Figure 4.

Tonality Measure. Two tonality measures have been investigated in this work: the spectral flatness measure and the unpredictability measure.

Spectral Flatness Measure (SFM). The SFM (see [4]) is used to estimate the tonality of an entire frame $\mathbf{x}_n$ or a set of predefined frequency bands. It is an intra-frame technique which employs a criterion based on the spectrum's magnitude of a frame:

$$\mathrm{SFM}(\tilde{\mathbf{X}}_n) = \frac{\sqrt[B]{\prod_{i=0}^{B-1} \mathbf{S}_{\tilde{\mathbf{X}}_n}[i]}}{\frac{1}{B}\sum_{i=0}^{B-1} \mathbf{S}_{\tilde{\mathbf{X}}_n}[i]}, \qquad \mathbf{S}_{\tilde{\mathbf{X}}_n}[i] \triangleq \frac{\mathbf{A}_n[i]^2}{B}, \tag{11}$$

where $\mathbf{S}_{\tilde{\mathbf{X}}_n}$ is the short-time power-density spectrum of the WOLA frame $\tilde{\mathbf{X}}_n$. The SFM values are in $[0, 1]$, where 0 indicates a pure tonal signal and 1 a pure noise signal, respectively. It can be shown that the inverse of the SFM is a measure how well a signal can be predicted. A signal can be better predicted if it is more tone-like.

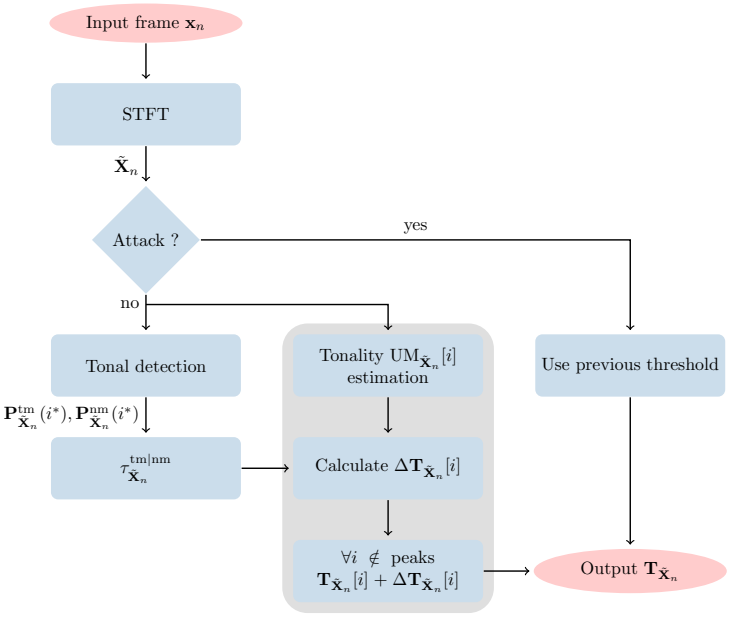


Fig. 4. New tonality estimation and integration in psychoacoustic model

Unpredictability Measure (UM). The UM – proposed in the psychoacoustic model 2 of ISO-MPEG –consists in predicting the amplitude $\mathbf{A}_n$ and the angle $\boldsymbol{\varphi}_n$ of the WOLA frame $\tilde{\mathbf{X}}_n$ as follows:

$$\hat{\mathbf{A}}_n[i] = 2\mathbf{A}_{n-1}[i] - \mathbf{A}_{n-2}[i], \quad \hat{\boldsymbol{\varphi}}_n[i] = 2\boldsymbol{\varphi}_{n-1}[i] - \boldsymbol{\varphi}_{n-2}[i], \tag{12}$$

and evaluating the resulting relative prediction error:

$$\mathrm{UM}_{\tilde{\mathbf{X}}_n}[i] = \frac{|\tilde{\mathbf{X}}_n[i] - \hat{\mathbf{X}}_n[i]|}{\mathbf{A}_n[i] + \hat{\mathbf{A}}_n[i]}, \quad \hat{\mathbf{X}}_n[i] \triangleq \hat{\mathbf{A}}_n[i]\mathrm{e}^{j\hat{\boldsymbol{\varphi}}_n[i]}. \tag{13}$$

It is an inter-frame method for a fixed frequency bin. A low UM value indicates that the frequency bin can be well predicted. This is typically the case of tonal components. Conversely, noise-like signals will have high UM values.

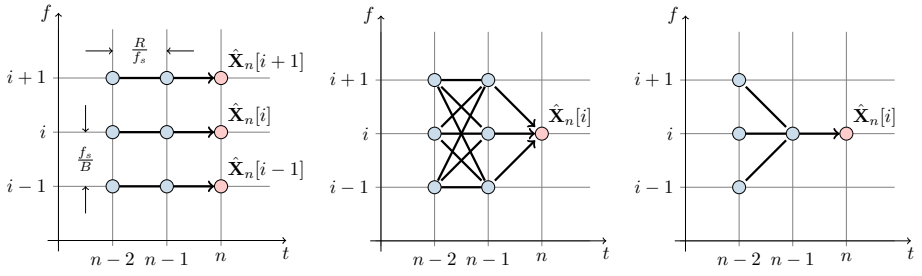
To obtain the predicted angle $\hat{\boldsymbol{\varphi}}_n[i]$, two angles $\boldsymbol{\varphi}_{n-1}[i], \boldsymbol{\varphi}_{n-2}[i]$ need to be extracted. Moreover, $\cos(\hat{\boldsymbol{\varphi}}_n[i])$ and $\sin(\hat{\boldsymbol{\varphi}}_n[i])$ need to be evaluated in order to compute the UM value. Such operations involve trigonometric operations which are computationally expensive. Most of this burden can be lifted by defining a couple of values[3]:

$$c_{n-1}[i] \triangleq \cos(\boldsymbol{\varphi}_{n-1}[i]) = \frac{\mathcal{R}\{\tilde{\mathbf{X}}_{n-1}[i]\}}{\mathbf{A}_{n-1}[i]}, \quad s_{n-1}[i] \triangleq \sin(\boldsymbol{\varphi}_{n-1}[i]) = \frac{\mathcal{I}\{\tilde{\mathbf{X}}_{n-1}[i]\}}{\mathbf{A}_{n-1}[i]}, \tag{14}$$

$$c_{n-2}[i] \triangleq \cos(\boldsymbol{\varphi}_{n-2}[i]) = \frac{\mathcal{R}\{\tilde{\mathbf{X}}_{n-2}[i]\}}{\mathbf{A}_{n-2}[i]}, \quad s_{n-2}[i] \triangleq \sin(\boldsymbol{\varphi}_{n-2}[i]) = \frac{\mathcal{I}\{\tilde{\mathbf{X}}_{n-2}[i]\}}{\mathbf{A}_{n-2}[i]}, \tag{15}$$

and by expressing $\cos(\hat{\boldsymbol{\varphi}}_n[i])$ and $\sin(\hat{\boldsymbol{\varphi}}_n[i])$ accordingly:

[3] $\mathcal{R}$ and $\mathcal{I}$ denote the real and imaginary parts of a complex number.



(a) Conventional mono-modal prediction ($M = 0$).

(b) Asymmetric bi-modal prediction ($M = 1$).

(c) Symmetric bi-modal prediction ($M = 1, m = 0$).

Fig. 5. Alternate prediction strategies in the time-frequency plane for the UM

$$\cos(\hat{\varphi}_n[i]) = \left(2c_{n-1}^2[i] - 1\right) c_{n-2}[i] + 2c_{n-1}[i]s_{n-1}[i]s_{n-2}[i], \tag{16}$$

$$\sin(\hat{\varphi}_n[i]) = 2c_{n-1}[i]s_{n-1}[i]c_{n-2}[i] - \left(2c_{n-1}^2[i] - 1\right) s_{n-2}[i]. \tag{17}$$

This alternate representation only uses multiplication and addition operations.

Combining intra-frame and inter-frame prediction. Compared to UM, the computational complexity of SFM is notably lower. However, as any global metric, it fails to grasp the local particularities of the spectrum. Let's assume that the input frame is made of a single tone. In the presence of noise, the SFM will be evaluated for the whole spectrum resulting in a relatively high value. As a result, spectral lines near the tone peak cause audible artifacts. Since the spectrum is processed globally, it is wrongly interpreted as more noise-like despite the more tonal nature of the signal.

According to our listening tests, the UM method works well for quasi-stationary signals with a long period. On the other hand, for audio signals with many short-period time-varying tones, e.g. speech signals, UM is not capable of tracking tones well, whereas SFM provides good results. A straightforward strategy to get better audio quality is then to use the minimum of the global SFM and UM, though it significantly hampers robustness performances as it is very conservative.

To solve this problem, we slightly modified the UM so as to combine the original inter-frame prediction with an intra-frame prediction method over the frequencies. This bi-modal prediction illustrated in Figure 5b should provide better tonal tracking capabilities. More specifically the prediction process is revised as follows:

$$(\hat{m}, \hat{m}') = \arg \min_{m,m' \in [-M:M]} \left| \tilde{\mathbf{X}}_n[i] - \hat{\mathbf{X}}_n[i, m, m'] \right|, \tag{18}$$

$$\hat{\mathbf{X}}_n[i, m, m'] \triangleq \hat{\mathbf{A}}_n[i, m, m'] \mathrm{e}^{j\hat{\varphi}_n[i,m,m']}$$

$$\hat{\mathbf{A}}_n[i, m, m'] = 2\mathbf{A}_{n-1}[i + m] - \mathbf{A}_{n-2}[i + m']$$

$$\hat{\varphi}_n[i, m, m'] = 2\varphi_{n-1}[i + m] - \varphi_{n-2}[i + m'].$$

The final $\hat{\mathbf{A}}_n[i, \hat{m}, \hat{m}']$ and $\hat{\varphi}_n[i, \hat{m}, \hat{m}']$ are subsequently used in Equation (13) for UM calculation. Increasing M improves the prediction by providing diversity, resulting in better audio quality but at the expense of watermarking power.

In order to mitigate this trade-off, it is possible to reduce the prediction diversity by imposing additional constraint e.g. the prediction necessarily use the same frequency bin in the previous frame:

$$\hat{m'} = \arg \min_{m' \in [-M:M]} \left| \tilde{\mathbf{X}}_n[i] - \hat{\mathbf{X}}_n[i, 0, m'] \right| . \tag{19}$$

This simplified bi-modal prediction depicted in Figure 5c maintains robustness performances and only requires a symmetric search compared to the asymmetric one of Equation (18). According to our experiments, UM evaluation using simplified bi-modal prediction with $M = 1$ already provides very good results, which will be used for quality and robustness evaluation.

Integration of UM into the Psychoacoustic Model. Given the UM values for individual frequency bins, the global masking thresholds are increased as follows:

$$\mathbf{T}_{\tilde{\mathbf{X}}_n}[i]_{\mathrm{dB}} \leftarrow \mathbf{T}_{\tilde{\mathbf{X}}_n}[i]_{\mathrm{dB}}(1 - \mathrm{UM}_{\tilde{\mathbf{X}}_n}[i]) + \mathbf{A}_n[i]_{\mathrm{dB}}.\mathrm{UM}_{\tilde{\mathbf{X}}_n}[i]. \tag{20}$$

It should be noted that the masking threshold $\mathbf{T}_{\tilde{\mathbf{X}}_n}[i]$ is always smaller than the spectral line energy $\mathbf{A}_n[i]$. Moreover, for a tonal component ($\mathrm{UM}_{\tilde{\mathbf{X}}_n}[i] = 0$), the masking threshold of the peak remains unchanged in order to avoid audible artifacts. Conversely, for pure noise components ($\mathrm{UM}_{\tilde{\mathbf{X}}_n}[i] = 1$), the masking threshold is set equal to the spectral line energy. Otherwise, the amount of threshold up-shift scales with the UM. In other words, the noisier is a particular frequency bin, the more its corresponding perceptual threshold can be lifted.

4 Comparison of Audio Quality and Robustness

To assess the impact of the psychoacoustic model on performances, we compared our proposed new method to the baseline algorithm using a heuristic noise detection approach. To guarantee fair a comparison in terms of robustness, special care has been taken to impose the same quality level in the two investigated setups.

4.1 Audio Quality Evaluation

Objective metrics are not reliable to quantify the quality of a audio track distorted by embedded watermarks (see [13] for a discussion). As a result, quality assessment has to be carried out via subjective listening tests.

The objective is to assess the quality of the new algorithm compared with the previous psychoacoustic model implementation. To do so, the ITU-R BS.1116 standard has been selected [14]. The recommendation BS.1116 has been designed to assess the degree of annoyance caused by any degradation of the audio quality to the listener. The use of a continuous grading scale allows comparing different watermarking embedders by rating their quality. This scale uses the fixed points derived from the ITU-R Subjective Difference Grade (SDG) scale [15] listed below:

Impairment	Grade	SDG
Imperceptible	5.0	0.0
Perceptible, but not annoying	4.0	-1.0
Slightly annoying	3.0	-2.0
Annoying	2.0	-3.0
Very annoying	1.0	-4.0

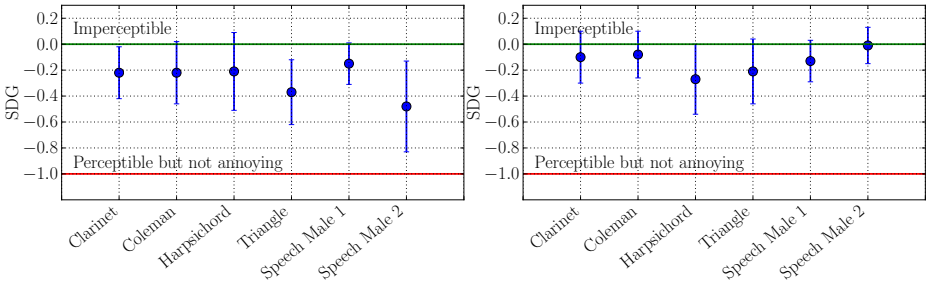
The test procedure is a so-called *double-blind A-B-C triple-stimulus* hidden reference comparison test. Stimuli A contains always the reference signal, whereas B and C are pseudo-randomly selected from the watermarked and the reference signal. The subject has to listen to all three items, select B or C as the reference – implicitly assigning a grade of 5.0 – and assign a grade to the other item. From the rating results, the SDG value is obtained by:

$$\mathrm{SDG} = \mathrm{Score}_{\mathrm{SignalUnderTest}} - \mathrm{Score}_{\mathrm{ReferenceSignal}}. \tag{21}$$

Test Design. The standard [14] specifies 20 subjects as an adequate size for the listening panel. Since expert listeners participated in the test, the number of listeners has been reduced to 13 for an informal test. A training session preceded the grading session where a trial was conducted for each signal. The tests were performed with headphones in a special cabin dedicated to listening tests. Six test signals – selected from the sound

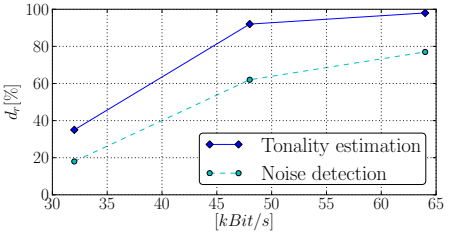
Table 1. Test signals used in the listening tests

Item	Attribute
Clarinet	Single instruments with many tonal components.
Harpsichord	Signal with attacks in time domain. Lot of harmonics misleading SFM.
Saxophon (Coleman)	Used in MPEG evaluation. Contains a lot of modulated components.
Speech Male Engl.(2)	A lot of interval times and modulated components.
Triangle	Small bandwidth signal with attacks in time domain.

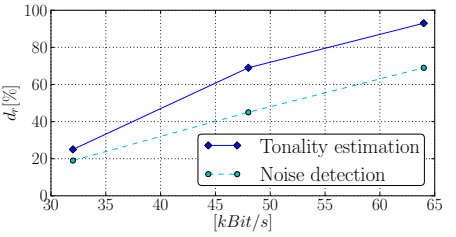


(a) SDG values using noise detection (b) SDG values using tonality estimation

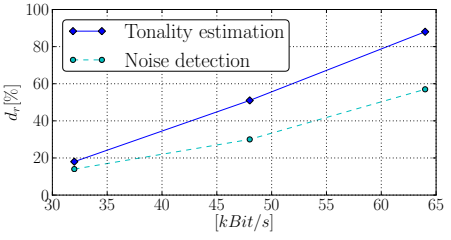
Fig. 6. Listening results for the BS.1116 test



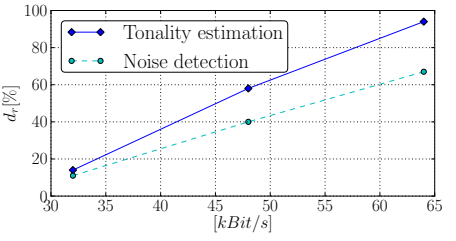
(a) mp3 compression.



(b) mp3Pro compression.

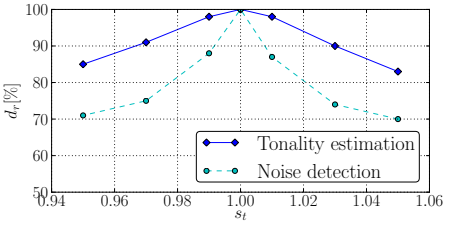


(c) AAC compression.

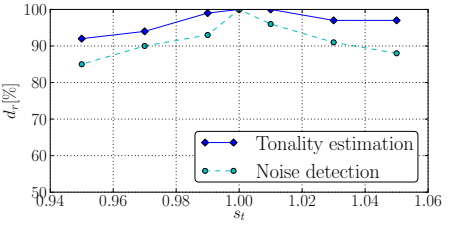


(d) AAC+ compression.

Fig. 7. Detection rates d_r[%] for lossy compression



(a) Time stretching.

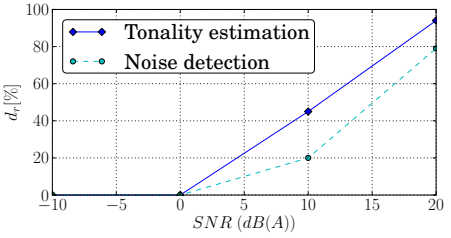


(b) Speed change.

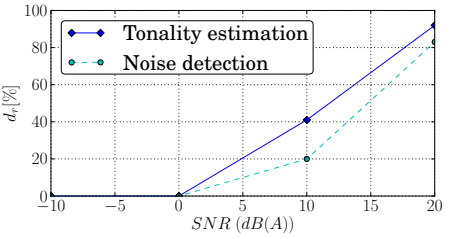
Fig. 8. Detection rates d_r[%] for time scaling with factor s_t

quality assessment material (SQAM) [16] – with a length of 10-20 s have been presented to the listeners. These signals were used in evaluating perceptual audio codecs and are chosen to reveal even small impairments to the listener. The six test signals can cause problems to the watermark embedder as detailed in Table 1.

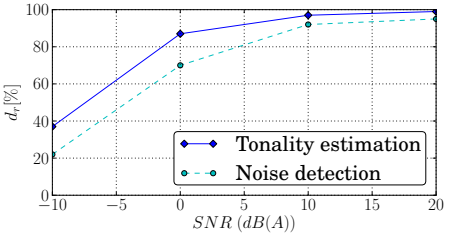
Analysis and Interpretation of Results. For the different audio files, the mean SDG value and the 95% confidence interval are plotted as a function of the different audio tracks to clearly reveal the distance to transparency (SDG = 0). The results in Figure 6a and Figure 6b show that the watermarked items from both algorithms have nearly the same, very good quality with SDG $\in [-1, 0]$.



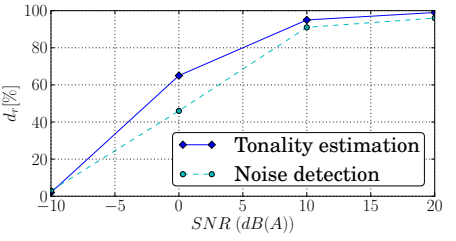
(a) White noise.



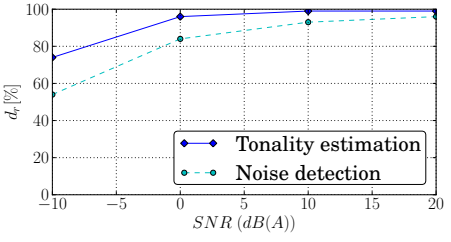
(b) Pink noise.



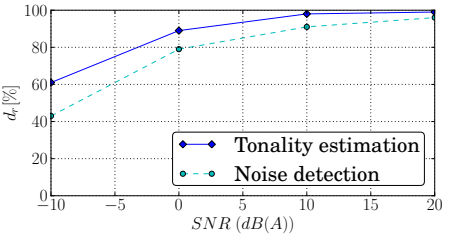
(c) Baby crying.



(d) Laughter.



(e) English female speech.



(f) German male speech.

Fig. 9. Detection rates $d_r[\%]$ for mixing signals

4.2 Robustness Tests

Having adjusted the quality to the same level, we evaluated the robustness of the algorithms with a payload rate of 2 bit/sec, where payload bits are protected by a rate $1/2$ tail-biting convolutional code. For robustness tests, a number of *digital attacks* have been applied after embedding the watermarks. To compare two audio watermarking embedders employing different psychoacoustic models, these attacks are sufficient, since attacks like the acoustic path can be simulated by combination of these signal processing operations. In all robustness tests 100 different sound files with a total play length of more than 7 hours were used.

Lossy Compression. Robustness against lossy compression was tested using a variety of audio codecs, as shown in Figures 7a-7d, where only bitrates lower than or equal to 64 kBits/sec were evaluated. According to the reported experimental results, the detection rate does not only depend on the bitrate, but also on the codec used. In all cases, the proposed system behaves better than the baseline algorithm.

Temporal Desynchronization. It is common practice in audio watermark benchmarks to distinguish between pitch-invariant time scaling, also referred to as time-stretching, and changing the playback speed of the audio track, which results in the modification of its pitch. Experimental results are given in Figures 8a and 8b and demonstrate reasonable robustness. Again, in all cases, the new algorithm proves better than the original system.

Mixing Signals. The robustness against mixing or overdubbing with different signals for different signal-to-noise ratios (SNRs) dB(A) has been tested to simulate the influence of environmental noise. The results in the Figures 9 indicate good robustness for signals other than white or pink noise even if the disturbing signal has the same energy (SNR = 0 dB(A)) as the watermarked one. In all cases, the proposed algorithm exhibits superior performances than the baseline system.

5 Conclusions

In this paper, we proposed an adaptation of the tonality measure in a psychoacoustic model for audio watermarking. The proposed algorithm revisits the prediction strategy of the previously proposed unpredictability measure so as to provide more accurate perceptual slacks. The resulting system is evaluated thoroughly with both early informal listening tests and extensive robustness evaluation.

The principal benefits that can be expected from the presented system are (i) the alterations of the psychoacoustic model can be re-used for other audio watermarking algorithms; (ii) an efficient implementation of the UM evaluation is presented providing the opportunity for a fixed-point implementation on embedded devices; (iii) an extensive robustness evaluation providing evidence of the improvement by the developed technique. Since tonal tracking and estimation is a research area on its own, further improvements can be expected by applying advancements in this field to the audio watermarking research.

References

1. Allen, J.B.: Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-25, 235–238 (1977)
2. ISO/IEC Joint Technical Committee 1 Subcommittee 29 Working Group 11: Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5Mbit/s Part 3: Audio. ISO/IEC 11172-3 (1993)
3. Johnston, J.D.: Transform Coding of Audio Signals Using Perceptual Noise Criteria. IEEE Journal on Selected Areas in Communications 6(2), 314–323 (1988)
4. Gray, A.H.J., Markel, J.D.: A Spectral-Flatness Measure for Studying the Autocorrelation Method of Linear Prediction of Speech Analysis. IEEE Transactions on Acoustics, Speech and Signal Processing 22(3), 207–217 (1974)
5. Zwicker, E., Fastl, H.: Psychoacoustics: Facts and Models, 2nd edn. Springer, Heidelberg (1999)

6. Kirovski, D., Malvar, H.S.: Spread-spectrum Watermarking of Audio Signals. IEEE Transactions on Signal Processing 51(4), 1020–1033 (2003)
7. Mansour, M., Tewfik, A.: Audio Watermarking by Time-Scale Modification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1353–1356 (May 2001)
8. Arnold, M., Baum, P.G., Voeßing, W.: A phase modulation audio watermarking technique. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 102–116. Springer, Heidelberg (2009)
9. Chen, X.M., Doërr, G., Arnold, M., Baum, P.G.: Efficient Coherent Phase Quantization for Audio Watermarking. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (May 2011)
10. Arnold, M., Chen, X.M., Baum, P.G.: Robust Detection of Audio Watermarks after Acoustic Path Transmission. In: Proceedings of the ACM Workshop on Multimedia and Security, pp. 117–126 (September 2010)
11. Painter, T., Spanias, A.: Perceptual Coding of Digital Audio. Proceedings of the IEEE 88(4), 451–515 (2000)
12. Arnold, M., Schmucker, M., Wolthusen, S.: Techniques and Applications of Digital Watermarking and Content Protection. Artech House, Boston (2003)
13. Arnold, M., Baum, P.G., Voeßing, W.: Subjective and Objective Quality Evaluation of Watermarked Audio. In: Cvejic, N., Seppänen, T. (eds.) Digital Audio Watermarking Techniques and Technologies, pp. 260–277. IGI Global, Hershey (2007)
14. ITU-R: Recommendation BS.1116-1, Methods for Subjective Assessement of Small Impairments in Audio Systems including Multichannel Sound Systems (1997)
15. ITU-R: Recommendation BS.1284-1, General Methods for the Subjective Assessement of Audio Quality (1997)
16. EBU: Sound Quality Assessment Material Recordings for Subjective Tests (April 1988)

Watermarking as a Means to Enhance Biometric Systems: A Critical Survey*

Jutta Hämmerle-Uhl, Karl Raab, and Andreas Uhl

Multimedia Signal Processing and Security Lab (WaveLab)
Department of Computer Sciences, University of Salzburg, Austria
uhl@cosy.sbg.ac.at

Abstract. Watermarking is discussed as possible means to enhance biometric systems. Application scenarios for the employment of watermarks as found in literature are discussed and analysed with respect to required watermark properties, possible attacks, and eventual (cryptographic) alternatives.

1 Introduction

Biometric recognition applications become more and more popular. Many institutions, governmental agencies and companies want to rely on this upcoming technology to secure their environment because standard authentication methods like PINs, passwords, smart-cards etc. have many disadvantages. Possession and knowledge based authentication techniques are prone to human errors since the former can be lost and the latter can be forgotten. Moreover, these technologies can be applied without actually guaranteeing a specific human presence. Biometric authentication systems can resolve most of these issues, since a biometric feature belongs only to one person and cannot be lost or forgotten. But eventually, biometric features can be stolen or adopted and there exist various other ways to circumvent the integrity of a biometric authentication system. Recent work systematically identifies security threats against biometric systems and possible countermeasures (among them watermarking) [40] and discusses man-in-the-middle attacks and BioPhishing against a web-based biometric authentication system [52].

In their classical paper [39] Ratha et al. identified and described several stages at which a biometric system may be attacked (denoted as attacked stage AS in the following) by an intruder or imposter:

1. Fake the biometric trait of a genuine user at the sensor (e.g. fake finger or printed face image)
2. The transmission between sensor and feature extractor may be intercepted and resubmitted by changed or replayed data
3. Override the feature extractor to produce predefined feature sets

* This work has been partially supported by the Austrian Science Fund, TRP project no. L554.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 238–254, 2011.

4. Intercept and replace the extracted feature sets by a synthetic or spoofed one
5. Override the matcher to always produce high matching scores
6. Modify, replace, remove stored ore add new templates at the database
7. Intercept the communication channel between template database and matcher
8. Override the final decision

Spoofing a physiological biometric feature at the sensor site can be seen as counterpart to exposing a password. If a fraudulent duplicate is accepted by the sensor, breaking a biometric system is at least as easy as spying out a password. However, the illegitimate acquisition of biometric features belonging to a target person does not necessarily involve complicated spoofing techniques. It is a fact that some biometric modalities, e.g. fingerprints, faces, irises, and hand-prints, can not be classified as being secret data. These data may be acquired quite easily: Fingerprint or even the full hand-print can be covertly lift off a glass, and current widespread digital cameras with telephoto lenses are able to take high resolution images of faces unseen by the photographed (using state of the art equipment may even provide enough resolution and sharpness to extract the iris out of a face image). Having acquired these "raw image data", a dedicated attack against the targeted person would be facilitated in case no further security mechanisms are employed. The acquired image data could be presented to the sensor (AS 1) or could be inserted into the transmission of data between sensor and feature extractor (AS 2). Since also the feature set extraction might be possible given the raw image data, the computed feature set could eventually be injected into the data before being submitted to the matcher (AS 4), which can also be applied to the communication channel between database and matcher (AS 7). The latter attack supposes that the feature extraction scheme is publicly known and is not protected by some template protection technique [19].

Of course, several strategies have been developed to cope with some these problems. Liveness detection helps to resolve the issue of fooling the sensor with prerecorded data (AS 1). Encryption techniques have been suggested to secure the above-mentioned communication channels (AS 2,4,7). However, the "public" availability of biometric data questions the necessity and appropriateness of encryption techniques for ensuring privacy of biometric data in the data transmissions in AS 2 and AS 4. Rather it is necessary to verify the sender (i.e. sensor and feature extractor) authenticity, as well as the integrity of the entire authentication mechanism.

In this paper, we provide a critical survey about work done if and how watermarking technology can either help to resolve some of the security issues or help to enhance biometric schemes in some other way. Given the significant amount of literature in the field, one might expect a more serious and systematic treatment of the topic. In many papers, the overall impression remains that two "buzzwords" have been combined without carefully analysing what this combination should actually achieve, why, and how this could be done. Section 2 covers fundamental issues related to biometrics and watermarking and

also shows watermarking application scenarios in the context of biometrics not aimed at improving biometric systems at all. In Section 3, we review several application scenarios for watermarking techniques found in literature which aim at improving current biometric schemes. For each scenario, we discuss the required properties of the watermarking system employed and sketch possible attacks against the approach. Additionally, also potential (cryptographic) alternatives to the application of watermarks are analysed. Finally, we discuss published proposals on watermarking and biometrics with respect to the required properties as identified before. Section 4 concludes the paper.

2 Watermarking in Biometric Systems

One of the first ideas to somehow combine biometric technologies and watermarking is "biometric watermarking". The aim of watermarking in this approach is not to improve any biometric system, but to employ biometric templates as "message" to be embedded in classical robust watermarking applications like copyright protection in order to enable biometric recognition after the extraction of the watermark (WM). As a consequence, the WM has to be capable of carrying the template data (capacity requirement) and should not be perceived. The robust WM has to resist against unintentional and malicious cover data manipulations.

Vielhauer et al. [46] introduce the general concept and notion of biometric watermarks, also discussed in [1]. One of the most interesting applications in the context is the "secure digital camera" [4], where an iris template of the photographer is embedded into digital images. Canon filed a corresponding patent recently (US Patent Application No. 2008/0025574). A similar idea also additionally addressing image integrity is proposed in [12]. Low et al. [29] suggest to embed offline signatures into digital images for copyright protection. Also for 3D graphics data, biometric watermarking has been suggested [32] by embedding an image of the copyright owners' biometric trait into a 3D mesh in a robust manner.

In order to motivate the use of watermarking in the biometric context with the aim of improving security, Jain et al. [17] suggest that if only traditional cryptographic techniques are used for the protection of biometric data, the data has to be decrypted somewhere along the way and therefore after decryption, security for the data is not maintained anymore – here watermarking comes in as a "second line of defence" similar to the DRM scenario since a watermark is still present after decryption. In this manner, information carried by the watermark can still be retrieved even if cryptographic tools have already been defeated.

There has been a lot of work done during the last years proposing watermarking techniques to enhance biometric systems in some way. Dong et al. [7] try to give a systematic view of the situation in the case of iris recognition by distinguishing whether biometric template / sample data are embedded into some host data ("template embedding"), or biometric sample data is watermarked by embedding some data into them ("sample watermarking"). In the latter case,

they distinguish between robust embedding techniques for database ownership protection and fragile techniques for sample tampering detection.

The impact of watermarking on the recognition performance of biometric systems has been investigated most thoroughly in the context of iris recognition also. While Dong et al. [7] do not report on performance degradations when investigating a single watermark embedding algorithm and one iris recognition technique only, Hämmerle et al. [10] find partially significant reductions in recognition accuracy (especially in case of high capacity) when assessing two iris recognition schemes and a couple of robust watermarking algorithms. Similar to the latter results, recognition impact has been observed as well for speech recognition [26] and fingerprint recognition [36] (the latter depending on the type of original which is employed in watermark extraction, thus, this is a non-blind approach).

For fingerprint recognition, watermarking techniques aiming at negligible impact on recognition performance have been designed. This is achieved for example by applying two blind robust spatial watermarking methods embedding a character bit string either sparing out fingerprint feature regions (i.e. close to minutiae data) or by maintaining the ridge gradient orientations [42, 9]. This approach has been followed by may other techniques (e.g. [5, 31]). On the other hand, recent work by Zebbiche et al. [50, 51] proposes two robust WM schemes for fingerprint images where WM data is embedded into the ridge area (region of interest RoI) only. The aim is to increase robustness of WM due to the concentration onto the RoI, while of course, some impact on recognition performance may be expected by using this idea.

In many papers covering the use of watermarking in biometric system, the authors remain rather vague about the actual aim, content, and required property of the employed watermarking system. A good example is the paper by Hong et al. [14] (although many more do exist) which discusses the application of robust watermarking and symmetric encryption techniques for the exchange of compressed biometric sample data, where they also investigate the impact on accuracy of a fingerprint recognition scheme. Additionally, energy consumption of different variants with respect to the applied compression in a distributed authentication scenario with mobile sensors is investigated. It is not discussed, which functionality the watermark is aimed to fulfil and how a successful attack against the used watermarking system would look like. Consequently, there is also no information about which data is embedded and which attacks against the biometric scheme should be prevented by the usage of watermarking. As a second example, we mention a quantisation-based robust watermarking of offline signatures [15] – while robustness against compression is investigated, the actual aim of robustly embedding a watermark is not described.

With this survey, we try to provide a systematic and critical view of the work done so far in this area. In the next section, we provide a discussion of several application scenarios described in literature using both watermarking and biometric technologies where the overall aim is to enhance a biometric system in some way. We follow the idea of distinguishing between template embedding and sample data watermarking schemes. For each scenario, we will explain the

overall aim of the WM technique, we will discuss required watermark properties, the types of attacks challenging the approach, and the possibility to replace the watermarking scheme by some alternative (cryptographic) approach.

3 Watermarking Application Scenarios for Enhancing Biometric Systems

3.1 Covert (Template) Communication

The aim of watermarking in this approach is to transmit biometric data (typically template data but also sample data is considered) hidden into arbitrary carrier / host data (in this manner AS 2 can be avoided an attacker does not realise that biometric data is transfered). For example, a fingerprint image may carry face data or any arbitrary image could include fingerprint minutiae. An attacker should be unaware of the concealed (real) data transfer. Therefore, this is a typical *steganographic* application scenario, which is based on template (or even sample data) embedding.

Attack. An attacker aims at *detecting* the WM in order to be able to intercept the template data transfer.

WM properties and content. As a consequence, the WM has to be capable of carrying the template / sample data (capacity requirement) and has to be undetectable. In the passive warden scenario, robustness of the WM is not an issue, however, robustness contradicts the requirement of a non-detectable WM. Blind extraction is required as it is a must for all steganographic application scenarios.

Crypto alternative. The is no cryptographic technique capable of replacing a steganographic approach.

A data hiding approach that targets this scenario is introduced by Jain et al. [16, 18] where fingerprint minutiae data are embedded into an arbitrary host image (scenario 1). A robust amplitude modulation-based watermarking method is used to embed a minutiae data set converted to a bitstream. Zebbiche et al. [49] introduce a robust wavelet based watermarking method hiding fingerprint minutiae data in fingerprint images (based on a method proposed by Kundur et al. [24]). They argue that an intruder is likely only to treat the fingerprint image instead the embedded data as well, so the scope is also a steganographic one. Khan et al. [20] uses a robust embedding technique to hide fingerprint templates into audio signals. Surprisingly, all proposals use robust embedding techniques which actually destroys the most important steganographic property as outlined above (non-detectability). In order to be able support non-detectability, steganographic WM needs to be applied instead of robust schemes. The remaining value of the proposed robust schemes is in communicating embedded templates in a way that the are not **perceived** by a human observer. When applying robust embedding as being proposed, embedded templates resist non-malicious cover data manipulations (which is an advantage over steganographic schemes in the case

of an active warden). Thus, the application context has to determine if robust WM or steganographic embedding serves the actual aim of the WM embedding (inperceptability vs. non-detectability).

When considering the application context, we have doubts that it is steganography which is actually most suited or required. When considering a classical biometric system, a biometric sensor is typically expected to transmit biometric authentication data over a dedicated channel to the feature extraction / matching module and no other types of data, so that it is not clear what is to be gained by steganography under these circumstances. It seems that in many papers, authors rather have confidentiality of embedded template data in mind, but this can only be achieved by using encryption (which is also inconvenient since decryption of the data is required before further processing). So in this scenario, many proposed robust WM schemes seem to represent an attempt to achieve a weak concealment of template content by embedding the data into some host material thereby avoiding encryption. Unfortunately, in this manner, neither the steganographic nor the confidentiality aim can be met.

In a distributed biometric system (as opposed to the classical case discussed before) where authentication data is transmitted over networks where also other type of data is communicated, the idea of applying steganography definitely makes more sense. The work described in [35] is explicitely focused to the application context – after a correlation analysis between host image and two images of different biometric modalities, residual data (sample images substracted from cover image) is embedded into the middle significant bit of the cover data. Since non-detectablity is not plausible considering the embedding stategy, this proposal fits better into the next category (two images are embedded thus enabling multibiometric recognition).

3.2 Multibiometric Recognition

The aim of watermarking in this scenario is to embed biometric data into a biometric sample in order to facilitate the employment of a multibiometric technique. The aim is an increased recognition performance due to the use of two different modalities. By using WM techniques, both informations are fused into a single data item which can be transmitted as a whole, however, the aim of using WM for this purpose is hardly ever motivated clearly.

There are two variants: First, biometric template data is embedded into the sample data of a different modality. These template data need to be generated at the sensor site which makes this approach somewhat unrealistic, especially in case of distributed biometric systems with low power sensor devices. The second approach is to embed sample data into the sample data of a different modality, which puts significant pressure to the capacity of the WM scheme. Therefore, in this scenario, sample watermarking as well as template / sample data embedding is used.

Attack. The resulting system is vulnerable in principle against all types of attacks endangering classical unimodal systems systems. In particular, an

attacker needs to *embed* sniffed biometric data of one modality into sniffed sample data of a second modality when targeting AS 2 (for example, face and iris biometric data can be extracted from a single high resolution image of the subject to be attacked).

WM properties and content. As a consequence, the WM has to be capable of carrying either template or sample data (capacity requirement) and extraction has to be blind, since otherwise, the advantage of transferring only a single data item would be lost by the required re-transmission of the original sample for extraction. It is of advantage if the WM resists unintentional image manipulations like compression or noise insertion, but robustness is not a required property here. In order to prevent an attacker to embed a stolen template, the embedding algorithm has to be dependent on a key. In this context the multibiometric approach also enhances the scheme with respect to resistance against targeting AS 2, since only samples which have correctly embedded WM data should be accepted by the system.

Crypto alternative. The benefit of embedding additional authentication data with WMs over classical cryptographic schemes is that this may be done in a way where "allowed" manipulations can be conducted on the data. Application of encryption to a concatenation of sample and template data results in slightly more data to be transmitted, but unauthorised embedding of stolen biometric data is prevented by this technique. Furthermore, this approach definitely has no impact on recognition performance as opposed to WM embedding.

Bartlow et al. [3] proposed a framework that encodes voice feature descriptors in raw iris images stored in a database. An asymmetric watermarking and cryptographic method using a public key infrastructure is used. The watermarking method is based on the robust technique by Kutter et al. [25] and is used to track data origin of data stored in centralised biometric databases. Hoang et al. [13] embed fingerprint minutiae in facial images (with fragile watermarks), while Jain et al. [18] embed face data into fingerprint images using a technique classified as being robust. Chung et al. [5, 31] use the same embedding technique as well to embed fingerprint templates into facial images and vice versa, and compare the recognition performance of the resulting systems. They also use this embedding technique as the fragile part of a dual watermarking approach [31, 22] so that doubts remain about the actual robustness properties of the scheme. Vatsa et al. employ robust embedding techniques: in [44], they embed voice features in colour facial images, the same group [33, 43, 45] propose to embed facial template data (and additional text data in the first work) into fingerprint sample data using a robust (multiple) watermarking approach. Park et al. [34] suggest to use robust embedding of iris templates into face image data to enable various functionalities, among them proof of database membership, steganographic issues, and of course multibiometric fusion. Kim et al. [21] propose a blind and robust spread spectrum watermarking technique for embedding face template data into fingerprint sample. Maiorana et al. [30] embed dynamic signature properties

into the image of a static signature using a robust DCT-Radon transform based embedding technique.

The approach of Zebbiche et al. [49], employing a robust wavelet based watermarking method for hiding fingerprint minutiae data in fingerprint images actually fits better into this scenario than into the steganographic one it has originally been suggested for.

Most schemes propose to use robust WMs for embedding and therefore do not provide the capacity for sample embedding (for many modalities, not even for template data). It seems that for this application case, it would therefore be better to abandon the idea of providing robustness but to use fragile or steganographic embedding techniques (eventually protected by error correction coding to provide some limited resistance against channel errors or lightweight signal processing). Key-dependent embedding is discussed explicitly only in some algorithms (e.g. [5, 31, 33]) but has turned out to be of importance in this setting. It is somewhat questionable if WM is in fact a suited technology in this context due to its potential impact on recognition performance (since the aim of the entire technique is the improvement of recognition !) and the existence of a sound cryptographic alternative.

A variant of the described ideas is to embed (template or sample) data into samples of the same modality. In this case, two different biometric templates of a single subject can be used in biometric matching which can also lead to improved recognition performance. In this case, only a single sensor has to be applied.

3.3 Two-Factor Authentication

The idea is to enable a two-factor authentication by requiring an additional token providing authentication data. Additional authentication data can be stored on a smart-card which has to be submitted by the holder at the access control site. The smart-card embeds the authentication data into the host sample data using WM technology which is sent to the biometric feature extraction / matching module. Another possibility for the second authentication factor could be a password which is submitted by the user. As a special case, the second authentication factor can be a biometric template stored on a smart-card. In this case, the advantages with respect to improved recognition performance due to the multibiometric nature of the scheme apply here as well.

Security of the overall system is increased simply by introducing an additional but different authentication scheme. The aim of watermarking in this scenario is similar as in the previous one since it is used to embed additional information into sample data (which in the present case comes from a smart-card or from a users' password, whereas in the former scenario data is generated by a different sensor). Again, the appropriateness of WM to be applied in this context is not obvious, for the same reasons as discussed before. This scenario is a case of sample watermarking, only as a special case template embedding can be applied.

Attack. The attacker can utilise a stolen smart-card (or sniffed password) and additional sniffed sample data of the attackers' target subject to fool the

system. He uses the biometric system pretending to be a legitimate user, but after WM embedding (e.g. of the data stored on the card), the attackers' sample data is *tampered* to match that of the sniffed sample data while not destroying the WM.

WM properties and content. As a consequence, the WM has to be capable of carrying the additional authentication data (capacity requirement: passphrase, ID, or template data) and extraction has to be blind, since otherwise, the advantage of transferring only a single data item would be lost by the required re-transmission of the original sample for extraction. In order to resist against a manipulation of the attackers' sample acquired by the sensor as described in [11], the WM scheme employed must not be robust. Therefore, only semi-fragile or fragile WMs fit all requirements.

Crypto alternative. The situation is perfectly identical to the multibiometric scenario and shares all corresponding problems discussed before.

A scheme embedding additional classical authentication data with robust watermarking is described in [41]. Here, the embedded signature is used as an additional security token like an additional password. In principle, all techniques developed for the multibiometric scenario could be used in this context, however, the majority of all schemes proposed also employ robust embedding, which is subject to tampering as described before.

Jain and Uludag [18] propose to embed face template data stored on a smartcard in fingerprint images (called senario 2 in the paper while scenario 1 is a steganographic one). Instead of embedding an additional security token also biometric template data from a second sensor can be embedded – in [36] an encrypted palmprint template is embedded into a fingerprint image, where the key is derived from palmprint classes. Since these additional data are not used in multibiometric fusion but serve as independent second token coming from a second sensor, this approach can be interpreted as being both, a multibiometric recognition scheme or a two factor authentication scheme. Since the employed WM system is a non-blind one, the applicability in a real system remains questionable.

It has to be pointed out that for this as well as for the former scenario, WM is not the only means (and probably not the best one) to communicate the "embedded" data in addition to sample data. Therefore, WM cannot be seen as the key-enabling technology for both of these scenarios. In case WM is selected as the means of transportation for the scenario investigated in this section, fragile or semi-fragile schemes have to be used which usually also have less or no impact on recognition performance.

3.4 Sample Replay Prevention

The aim of watermarking in this scenario is to prevent the use of sniffed sample data to fool the sensor. During data acquisition, the sensor (i.e. camera) embeds a watermark into the acquired sample image before transmitting it to the feature extraction module. In case an intruder interferes the communication channel

(AS 2), sniffs the image data and presents the fake biometric trait (i.e. the sniffed sample image) to the sensor (AS 1), it can detect the watermark, will deduce non-liveness and will refuse to process the data further. Therefore, the sensor needs to be capable of extracting the WM in addition to embedding it. As a consequence, all image data eventually additionally stored in the template database for some reason also carry the WM which stems from the enrolment process in this case. Consequently, image data from a compromised database cannot be used as fake biometric traits either. In this scenario, sample watermarking is applied.

Attack. An attacker aims at *removing* the WM in order to be able to use sniffed data for replay attacks or as fake traits.

WM properties and content. As a consequence, the WM has to be robust. It has to be detectable in the image as long as the image can be used in the recognition process (note that this corresponds well to the DRM scenario where a robust WM has to be detectable as long as the image is of sufficient quality). The extracted mark needs to carry at least the information "yes, I have been acquired by a sensor" (so eventually zero-bit WM could be used), but could also carry actual sensor IDs. The WM must be detected in blind manner, other wise the sample would have to be sent unprotected a second time.

Crypto alternative. Encrypting the data after acquisition for transmission provides similar functionality, however, the data needs to be decrypted for feature extraction and matching, which is a severe disadvantage. In any case, the WM may serve as additional "second line of defence" as it is suggested in the DRM context as well. Generic liveness detection techniques also target the attempt of using sniffed image data to fool the sensor and are a possible alternative method, however, usually at a much higher cost.

The proposed solution by Bartlow et al. [3] exactly targets the database environment. All robust WM algorithms, e.g. those proposed by Uludag and Gunsel [42, 9] could be employed in this scenario. Of course, problems with respect to impact of robust WM on recognition performance are valid in this scenario as well.

3.5 Sensor and Sample Authentication

The aim of watermarking in this scenario is to ensure the integrity of the sample data acquisition and transmission process. During data acquisition, the sensor (i.e. camera) embeds a watermark into the acquired sample image before transmitting it to the feature extraction module. The feature extraction module only proceeds with its tasks if the WM can be extracted correctly (which means that (a) the data has not been tampered with and (b) the origin of the data is the correct sensor). If an attacker tries to inject a sniffed image into the communication channel between sensor and feature extraction module for a replay attack or modifies a correctly acquired image in some malicious manner (AS 2), this is prevented by this approach. The same strategy can be applied to raw image data in the template database (if present). In this scenario, sample watermarking is applied.

Attack. An attacker aims at *inserting* the WM in order to mimic correctly acquired sensor data.

WM properties and content. In contrast to the previous scenario, the WM needs to be unique in the sense that it has to uniquely identify the sensor and carry a unique transaction number or timestamp. Resistance against a WM insertion attack can be achieved by sensor-key dependent embedding. Since the watermarking scheme has to be able to detect image manipulations, semi-fragile techniques are the method of choice. The WM could also eventually be fragile resulting in every minor modification attempt to be detected. However, each channel error or compression after embedding during transmission will destroy the WM in this case leading to a high false negative rate during authentication which is a highly undesired effect. Therefore, fragile WM can only by used in definitely lossless environments. Especially in semi-fragile watermarking it was found to be highly advantageous to embed image-dependent watermark data in order to prevent an embedding of image-independent watermarks after modifications have been conducted. WM extraction should be blind, otherwise the sample would need to be sent a second time (which would be possible in principle but is undesired due to transmission overhead).

Crypto alternative. Classical authentication protocols can be used to secure the communication between sensor and feature extraction module – a digital signature signed with the private key of the acquisition device can ensure the authenticity of the sensor and the integrity of the image data. However, digital signatures represent separate data which has to be taken care of separately. Cryptographic digital signatures are not capable of providing any robustness against channel errors and unintentional signal processing "attacks" like compression, which is the same as with fragile WM. Additionally, WM eventually provide information about the location where image data tampering has occurred, which could be used to determine if recognition-critical data has been affected (e.g. the iris texture in rectangular eye images) or to gain information if an intentional attack is the cause of the damaged WM (e.g., if the image data is corrupted by transmission noise "only", we result in evenly distributed tamper locations).

Yeung et al. [48] propose a fragile watermarking technique to add the ability for integrity verification of the captured fingerprint images against altering during transmission or in a database. Also, the method is shown to have little impact on recognition performance. Ratha et al. [37, 38] propose to embed a response to an authentication challenge sent out by a server into a WSQ compressed fingerprint image in order to authenticate the sensor capturing the fingerprint image. If the (fragile) watermark cannot be extracted, either the image has been tampered with or the image does not come from the correct sensing device. Since the approach is bound to a specific format, it is not even robust against lossless format conversions. Wang et al. [47] also introduce a fragile watermarking

scheme, however, they propose to embed image dependent data (i.e. SVD data) into the image contrasting to the former to approaches by Yeung et al. and Ratha et al.

Also, semi-fragile watermarking has been suggested to verify authenticity of biometric sample data. Two different embedding techniques for embedding both, a sample image dependent signature as well as a template are proposed by Komninos et al. [23]. PCA features are used as embedded data in [28], the embedded data can as well be used for an approximate recovery of the sample data in [6], and [2] proposes the embedding of robust signatures into fingerprint images.

Finally, [27] use two embedding techniques, the first for checking integrity on a block level using CRC checks, the second providing reversible watermarking (i.e. the sample is recntructed to the original before data embedding took place) in case the first rates the sample as being authentic.

It has to be noted, that a combination of the last two scenarios (sample replay prevention and sample and sensor authentication) seems to be highly sensible and desirable. This can also easily done at low cost since in both cases the sensor embeds WM information. Kim et al. [22, 31] is the first approach somewhat addressing this issue by proposing a dual WM scheme to protect fingerprint images by using a robust and fragile method.

Overall, it has to be stated that this scenario is not at all specific to biometric systems. The general discussion if (semi-)fragile WM is a sensible alternative to classical authentication protocols in case of image data to be protected applies to the discussed scenario, as well as all corresponding arguments do.

4 Discussion and Conclusion

In this paper we have discussed the available literature on the use of watermarking technology to enhance biometric systems. While the majority of proposals in the field employs watermarking to enable multibiometric scenarios (with the primary aim of increasing recognition accuracy) or to facilitate steganographic-like scenarios (in order to conceal the transfer of biometric data), we have also identified three scenarios where watermarking has been suggested to help in improving the security of classical uni-modal biometric systems.

We have found that the WM schemes as suggested to be used in the context of biometric systems often exhibit somewhat adhoc properties and specific requirements are not analysed in detail. In many cases, the actual WM method proposed does not lead to the desired effect or at least not in an optimal manner. A more thorough analysis of concrete attack scenarios is desirable for many environments in order to tailor required WM properties better to specific demands.

For most scenarios considered, WMs are not the only means to achieve the desired goals (and for some scenarios, WM are definitely not the best means to do so). For the covert (template) transmission scenario, we have found that the steganographic aim itself does only make sense under specific circumstances.

In case of making sense, there is no other technique that achieves the same effect. If confidentiality is actually what the aim of WM is, only (additional) encryption can provide this in a sound manner.

Two scenarios use WM as a means to transport additional information embedded into sample data (i.e. multibiometric recognition and two factor authentication). Of course, this can be done in some classical way involving cryptography or not, however, these alternatives usually exhibit the classical disadvantages as compared to watermarking since they cannot provide any robustness even against format conversions. Since we have seen that robustness on the other hand potentially impacts on recognition performance and should therefore be avoided if possible, semi-fragile embedding techniques remain as the only sensible ones here. If schemes of this type can meet corresponding capacity requirements depends on the actual amount of data to be embedded.

With respect to sample replay prevention, WM is an attractive choice from the application viewpoint, since it prevents the encryption–decryption effort of classically securing template or sample confidentiality. However, sniffing of template data is not prevented and in case a different biometric system is not able to detect the embedded mark or does not even support this feature, the approach is flawed (contrasting to encrypting the data, where an attacker simply is not able to access the plaintext at all). Additionally, as already mentioned, in this approach robustness is absolutely a must, therefore caution needs to be paid not to degrade recognition.

For sample and sensor authentication, (semi-)fragile WM can be used as a means to provide the desired aims. The alternative cryptographic techniques provide more security, but on the other hand absolutely no robustness against the slightest modification of the data is achievable. In addition, the "classic" techniques are not able to localise possible data manipulations and authentication data has to be communicated separately.

With respect to WM properties, a disadvantage inherent in many (robust) watermarking schemes when applied to biometric sample data is that of possible negative impact on recognition performance [10] – besides the design of specific watermarking approaches taking this problem into account (e.g. [42, 9]) an entirely different solution is to rely on *reversible* WM schemes [27, 8] which enable to reconstruct the original signal after WM extraction. This property (which is important e.g. in applying WM to medical imagery) fits perfectly into the biometric scenario since it enables recognition with entirely unaffected sample data. However, such schemes typically do not provide enough capacity to embed template data. Another option is to use (semi-)fragile or steganographic embedding techniques, which do offer sufficient capacity and exhibit almost no impact on recognition accuracy. Therefore, in case robustness is not a vital requirement but only a desired property (i.e. multibiometric embedding), it should be either employed in a way not affecting recognition or avoided. Furthermore, blind WM extraction is a must for most scenarios and is highly desirable for the remaining ones. For the WM application cases where template data or even sample data has to be embedded, capacity turns out to be a critical and even limiting factor.

Summarising, more thorough investigations are required in this field to (a) identify sensible application scenarios for watermarking in biometrics and to (b) select and/or design appropriate WM schemes to support the desired functionalities and we hope that this survey can represent a first step in this direction.

References

[1] Ahmad, S., Lu, Z.-M.: A joint biometrics and watermarking based framework for fingerprinting, copyright protection, proof of ownership, and security applications. In: Proceedings of the 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), pp. 676–679 (2007)

[2] Ahmed, F., Moskowitz, I.S.: Composite signature based watermarking for fingerprint authentication. In: Proceedings of the ACM Workshop on Multimedia and Security (MMSEC 2005), pp. 799–802 (2005)

[3] Bartlow, N., Kalka, N., Cukic, B., Ross, A.: Protecting iris images through asymmetric digital watermarking. In: IEEE Workshop on Automatic Identification Advanced Technologies, vol. 4432, pp. 192–197. West Virginia University, Morgantown (2007)

[4] Blythe, P., Fridrich, J.: Secure digital camera. In: Digital Forensic Research Workshop, Baltimore, MD, USA (August 2004)

[5] Chung, Y., Moon, D., Moon, K., Pan, S.: Hiding biometric data for secure transmission. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3683, pp. 1049–1057. Springer, Heidelberg (2005)

[6] Ding, S., Li, C., Liu, Z.: Protecting hidden transmission of biometrics using authentication watermarking. In: Proceedings of the 2010 WASE International Conference on Information Engineering (ICIE), pp. 105–108 (2010)

[7] Dong, J., Tan, T.: Effects of watermarking on iris recognition performance. In: Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008), pp. 1156–1161 (2008)

[8] Feng, J.-B., Lin, I.-C., Tsai, C.-S., Chu, Y.-P.: Reversible watermarking: current status and key issues. International Journal on Network Security 2(3), 161–171 (2006)

[9] Gunsel, B., Uludag, U., Tekalp, A.M.: Robust watermarking of fingerprint images. Pattern Recognition Journal 35(12), 2739–2747 (2002)

[10] Hämmerle-Uhl, J., Raab, K., Uhl, A.: Experimental study on the impact of robust watermarking on iris recognition accuracy (best paper award, applications track). In: Proceedings of the 25th ACM Symposium on Applied Computing, pp. 1479–1484 (2010)

[11] Hämmerle-Uhl, J., Raab, K., Uhl, A.: Attack against Robust Watermarking-Based Multimodal Biometric Recognition Systems. In: Vielhauer, C., Dittmann, J., Drygajlo, A., Juul, N.C., Fairhurst, M.C. (eds.) BioID 2011. LNCS, vol. 6583, pp. 25–36. Springer, Heidelberg (2011)

[12] Hassanien, A.E., Abraham, A., Grosan, C.: Spiking neural network and wavelets for hiding iris data in digital images. Soft Computing - A Fusion of Foundations, Methodologies and Applications 13(4), 401–416 (2008)

[13] Hoang, T., Tran, D., Sharma, D.: Remote multimodal biometric authentication using bit priority-based fragile watermarking. In: Proceedings of the 19th International Conference on Pattern Recognition, Tampa, Florida, USA, pp. 1–4 (December 2008)

[14] Hong, S., Kim, H., Lee, S., Chung, Y.: Analyzing the secure and energy efficient transmissions of compressed fingerprint images using encryption and watermarking. In: Proceedings of the 2008 International Conference on Information Security and Assurance, pp. 316–320 (2008)
[15] Hui, L., Yu-ping, H.: Wavelet tree quantization-based biometric watermarking for offline handwritten signature. In: Proceedings of the 2009 International Asia Symposium on Intelligent Interaction and Affective Computing, pp. 71–74 (2009)
[16] Jain, A.K., Uludag, U.: Hiding fingerprint minutiae in images. In: Proceedings of AutoID 2002, 3rd Workshop on Automatic Identification Advanced Technologies, Tarrytown, New York, USA, pp. 97–102 (March 2002)
[17] Jain, A.K., Ross, A., Uludag, U.: Biometric template security: Challenges and solutions. In: Proceedings of the 13th European Signal Processing Conference, EUSIPCO 2005, Antalya, Turkey (September 2005)
[18] Jain, A.K., Uludag, U.: Hiding biometric data. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1494–1498 (2003)
[19] Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security. EURASIP Journal on Advances in Signal Processing, 1–17 (2008)
[20] Khan, M.K., Xie, L., Zhang, J.S.: Robust hiding of fingerprint-biometric data into audio signals. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 702–712. Springer, Heidelberg (2007)
[21] Kim, W.-G., Lee, H.K.: Multimodal biometric image watermarking using two-stage integrity verification. Signal Processing 89(12), 2385–2399 (2009)
[22] Kim, W.-G., Lee, S.H., Seo, Y.-S.: Image Fingerprinting Scheme for Print-and-Capture Model. In: Zhuang, Y.-t., Yang, S.-Q., Rui, Y., He, Q. (eds.) PCM 2006. LNCS, vol. 4261, pp. 106–113. Springer, Heidelberg (2006)
[23] Komninos, N., Dimitriou, T.: Protecting biometric templates with image watermarking techniques. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 114–123. Springer, Heidelberg (2007)
[24] Kundur, D., Hatzinakos, D.: Digital watermarking using multiresolution wavelet decomposition. In: Proceedings of the 1998 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1998), Seattle, WA, USA, vol. 5, pp. 2969–2972 (May 1998)
[25] Kutter, M., Jordan, F., Bossen, F.: Digital signature of color images using amplitude modulation. In: Sethi, I.K., Jain, R.C. (eds.) Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases, San Jose, USA, vol. 2952, pp. 518–526 (1997)
[26] Lang, A., Dittmann, J.: Digital watermarking of biometric speech references: impact to the eer system performance. In: Delp, E.J., Wong, P.W. (eds.) Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, p. 650513 (2007)
[27] Lee, H., Lim, J., Yu, S., Kim, S., Lee, S.: Biometric image authentication using watermarking. In: Proceedings of the International Joint Conference SICE-ICASE 2006, pp. 3950–3953 (2006)
[28] Li, C., Ma, B., Wang, Y., Zhang, Z.: Protecting biometric templates using authentication watermarking. In: Qiu, G., Lam, K.M., Kiya, H., Xue, X.-Y., Kuo, C.-C.J., Lew, M.S. (eds.) PCM 2010. LNCS, vol. 6297, pp. 709–718. Springer, Heidelberg (2010)
[29] Low, C.-Y., Teoh, A.B.-J., Tee, C.: Fusion of LSB and DWT Biometric Watermarking Using Offline Handwritten Signature for Copyright Protection. In: Tistarelli, M., Nixon, M.S. (eds.) ICB 2009. LNCS, vol. 5558, pp. 786–795. Springer, Heidelberg (2009)

[30] Maiorana, E., Campisi, P., Neri, A.: Biometric signature authentication using radon transform-based watermarking techniques. In: Proceedings of the 2007 Biometrics Symposium (2007)

[31] Moon, D., Kim, T., Jung, S.-H., Chung, Y., Moon, K., Ahn, D., Kim, S.K.: Performance evaluation of watermarking techniques for secure multimodal biometric systems. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005 part II. LNCS (LNAI), vol. 3802, pp. 635–642. Springer, Heidelberg (2005)

[32] Motwani, R.C., Harris, F.C., Bekris, K.E.: A proposed digital rights management system for 3d graphics using biometric watermarks. In: Proceedings of the 7th IEEE Conference on Consumer Communications and Networking Conference CCNC 2010, pp. 1075–1080 (2010)

[33] Noore, A., Singh, R., Vatsa, M., Houck, M.M.: Enhancing security of fingerprints through contextual biometric watermarking. Forensic Science International 169, 188–194 (2007)

[34] Park, K.R., Jeong, D.S., Kang, B.J., Lee, E.C.: A Study on Iris Feature Watermarking on Face Data. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4432, pp. 415–423. Springer, Heidelberg (2007)

[35] Qi, M., Lu, Y., Du, N., Zhang, Y., Wang, C., Kong, J.: A novel image hiding approach based on correlation analysis for secure multimodal biometrics. Journal of Network and Computer Applications 33(3), 247–257 (2010)

[36] Rajibul, M.I., Shohel, M.S., Andrews, S.: Biometric template protection using watermarking with hidden password encryption. In: Proceedings of the International Symposium on Information Technology 2008 (ITSIM 2008), pp. 296–303 (2008)

[37] Ratha, N.K., Connell, J.H., Bolle, R.M.: Secure data hiding in wavelet compressed fingerprint images. In: ACM Multimedia 2000, Los Angeles, CA, USA (November 2000)

[38] Ratha, N.K., Figueroa-Villanueva, M.A., Connell, J.H., Bolle, R.M.: A secure protocol for data hiding in compressed fingerprint images. In: Maltoni, D., Jain, A.K. (eds.) BioAW 2004. LNCS, vol. 3087, pp. 205–216. Springer, Heidelberg (2004)

[39] Ratha, N.K., Connell, J.H., Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal 40(3), 614–634 (2001)

[40] Roberts, C.: Biometric attack vectors and defenses. Computers & Security 26, 14–25 (2007)

[41] Satonaka, T.: Biometric watermark authentication with multiple verification rule. In: Proceedings of the 12th IEEE Workshop on Neural Networks in Signal Processing, pp. 597–606 (2002)

[42] Uludag, U., Gunsel, B., Ballan, M.: A spatial method for watermarking of fingerprint images. In: Proceedings of the 1st International Workshop on Pattern Recognition in Information Systems, PRIS 2001, Setubal, Portugal, pp. 26–33 (July 2001)

[43] Vatsa, M., Singh, R., Noore, A.: Improving biometric recognition accuracy and robustness using DWT and SVM watermarking. IEICE Electronics Express 2(12), 362–367 (2005)

[44] Vatsa, M., Singh, R., Noore, A.: Feature based RDWT watermarking for multimodal biometric system. Image and Vision Computing 27(3), 293–304 (2009)

[45] Vatsa, M., Singh, R., Noore, A., Houck, M.M., Morris, K.: Robust biometric image watermarking for fingerprint and face template protection. IEICE Electronics Express 3(2), 23–28 (2006)
[46] Vielhauer, C., Steinmetz, R.: Approaches to biometric watermarks for owner authentification. In: Proceedings of SPIE, Security and Watermarking of Multimedia Contents III, San Jose, CA, USA, vol. 4314 (January 2001)
[47] Wang, D.-S., Li, J.-P., Hu, D.-K., Yan, Y.-H.: A novel biometric image integrity authentication using fragile watermarking and Arnold transform. In: Li, J.P., Bloshanskii, I., Ni, L.M., Pandey, S.S., Yang, S.X. (eds.) Proceedings of the International Conference on Information Computing and Automatation, pp. 799–802 (2007)
[48] Yeung, M.M., Pankanti, S.: Verification watermarks on fingerprint recognition and retrieval. Journal of Electronal Imaging, Special Issue on Image Security and Digital Watermarking 9(4), 468–476 (2000)
[49] Zebbiche, K., Ghouti, L., Khelifi, F., Bouridane, A.: Protecting fingerprint data using watermarking. In: Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2006, Istanbul, Turkey, pp. 451–456 (June 2006)
[50] Zebbiche, K., Khelifi, F.: Region-based watermarking of biometric images: Case study in fingerprint images. International Journal of Digital Multimedia Broadcasting (March 2008)
[51] Zebbiche, K., Khelifi, F., Bouridane, A.: An efficient watermarking technique for the protection of fingerprint images. EURASIP Journal on Information Security (February 2008)
[52] Zeitz, C., Scheidat, T., Dittmann, J., Vielhauer, C.: Security issues of internet-based biometric authentication systems: risks of man-in-the-middle and BioPhishing on the example of BioWebAuth. In: Delp, E.J., Wong, P.W., Dittmann, J., Nemon, N.D. (eds.) Proceedings of SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, vol. 6819, pp. 0R-1 –0R12, (2008)

Capacity-Approaching Codes for Reversible Data Hiding

Weiming Zhang[1,2], Biao Chen[1], and Nenghai Yu[1,⋆]

[1] Department of Electrical Engineering & Information Science, University of Science and Technology of China, Hefei 230026, China
[2] Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China
zwmshu@gmai.com, chb893@mail.ustc.edu.cn, ynh@ustc.edu.cn

Abstract. By reversible data hiding, the original cover can be losslessly restored after the embedded information is extracted. Kalker and Willems established a rate-distortion model for reversible data hiding, in which they proved the capacity bound and proposed a recursive code construction. In this paper we improve the recursive construction by designing a data embedding method for all-zero covers and a more efficient compression algorithm. We prove that the proposed codes can approach the capacity bound under various distortion constraints. We also apply this coding method to RS method for spatial images, and the experimental results show that the novel codes can significantly reduce the embedding distortion.

Keywords: data hiding, watermark, reversible data hiding, recursive construction, arithmetic coder.

1 Introduction

Data hiding is a technique for embedding information into a cover media such as images, audio and video files, which can be used for the purpose of media notation, copyright protection, integrity authentication and covert communication, etc. Most data hiding methods embed messages into the cover media to generate the marked media by only modifying the least significant part of the cover, and thus keep perceptual transparency. The embedding process will usually introduce permanent distortion to the cover, that is, the original cover can never be reconstructed from the marked cover. However, in some applications, such as medical imagery, military imagery and law forensics, no degradation of the original cover is allowed. In these cases, we need a special kind of data hiding methods, referred to as reversible data hiding or lossless data hiding, by which the original cover can be losslessly restored after the embedded message is extracted.

⋆ Corresponding author.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 255–269, 2011.

Many reversible data hiding methods have been proposed since it was introduced. Fridrich et al. [1] presented a universal framework for reversible data hiding, in which the embedding process is divided into three stages. In the first stage, extract losslessly compressible features (or portions) from the original cover. The second stage compresses the features with a lossless compression method, and thus saves space for the payload (message). In the third stage, embed messages into the feature sequence and generate the marked cover. One direct reversible embedding method is to compress the feature sequence and append messages after it to form a modified feature sequence, then replace the original features by the modified features, and thus generate the marked cover. Therefore, after extracting the message, the receiver can restore the original cover by decompressing the features. Fridrich et al. [1] suggested the features that can exploit characteristics of certain image formats, e.g., texture complexity for spatial images and middle frequency coefficients for JPEG images. Celik et al. [2] extended Fridrich's scheme by predicting multiple LSB planes. The same idea proposed in [1] can also be used for reversible data embedding into binary images [3,4] or videos [5,6].

To increase embedding capacity, the researchers desire to construct a longer feature sequence that can be perfectly compressed. One of such constructions is difference expansion (DE), first proposed by Tian [7], in which the features are the differences between two neighboring pixels of pixel pairs. The features are compressed by expansion, i.e., the differences are multiplied by 2, and thus the least significant bits (LSBs) of the differences can be used for embedding messages. The methods proposed in [8] and [9] can achieve better performance by applying DE to the prediction-errors.

Another well-known strategy for reversible data hiding is histogram shift (HS), in which the histogram of the image is used as the compressible feature because the distribution of the pixel values of an image is usually uneven. To compress the histogram, Ni et al. [10] proposed selecting a peak bin and a zero bin, and shifting the bins between them toward the zero bin by one step. Therefore, the bin neighboring to the peak bin is emptied out, which with the peak bin can be used to represent 1 and 0 respectively. It is easy to see that steeper histogram implies better compression rate, and usually the histogram of residuals is quite steep. Thus, most state-of-the-art methods apply histogram shift to residuals of the image [11,12].

According to the stage where distortion happens, we divide reversible data hiding into two types as follows. Type-I: all distortion is introduced in the stage of message embedding. Type-II: both compression stage and embedding stage will introduce some distortion to the cover, and compression stage is responsible for major distortion. The methods in [1,2,3,4,5,6] belong to Type-I; and yet both DE-based methods [7,8,9] and HS-based methods [10,11,12] belong to Type-II. Both types process the feature compression and data embedding in a separate way. In the present paper, we pay our attention to designing joint coding of compression and embedding for a binary feature sequence, which can be directly

used to improve Type-I methods. In fact, schemes in Type-II can be converted to Type-I, and we will discuss how to do such conversion in one of our coming papers.

For the Type-I reversible data hiding, the core problem is how to reversibly embed data into a compressible feature sequences with good performance. The performance is measured by embedding rate versus distortion, which is a special rate-distortion coding problem. A formal model for this problem has been established by Kalker and Willems [13]. In [13], the authors obtained the rate-distortion function, i.e., the upper bound of the embedding rate under a given distortion constraint, and they also proposed a joint compression and embedding code construction, called recursive code construction [13,14], which consists of a non-reversible data embedding code and a conditional compression code.

In this paper, we improve the recursive construction to approach the rate-distortion bound proved in [13]. In the novel construction, we use a data embedding code for all-zero covers and a backward extraction strategy, which enable us to design a very efficient conditional compression algorithm. The proposed codes can be directly applied to Type-I reversible data hiding schemes and significantly reduce the distortion for various given embedding rates.

The rest of this paper is organized as follows. The coding model, theoretical upper bound of embedding rate and recursive construction are briefly introduced in Section 2. The proposed coding method with the analysis of embedding rate versus distortion is elaborated in Section 3. The experiment results on improving RS schemes are given in Section 4. The paper is concluded with a discussion in Section 5.

2 Coding Model and Recursive Construction

2.1 Coding Model

Throughout this paper, we denote matrices and vectors by boldface fonts, and use the same notation for the random variable and its realization for simplicity. We denote the entropy by $H(X)$ and conditional entropy by $H(X|Y)$.

To do reversible data hiding, a compressible feature sequence should first be extracted from the original cover. For Type-I schemes, the features can usually be represented by a binary sequence. Therefore we directly take the binary feature sequence as the cover to discuss the coding method. A formal setup and theory limit for reversible data hiding into a compressible sequence have been established in [13] by Kalker and Weillems, and we follow their notation.

Assume that there is a memoryless source producing binary compressible cover sequence $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ such that $x_i \in \{0, 1\}$ with the probability $P(x_i = 0) = p_0$ and $P(x_i = 1) = p_1, 1 \leq i \leq N$. The assumption of $\mathbf{x}$ being compressible implies that the ratios of 0's and 1's are biased. Without loss of generality, we assume that $p_0 > 1/2$. We use Hamming distance to measure the embedding distortion on the cover $\mathbf{x}$. Because the message $\mathbf{m}$ is usually compressed and encrypted before being embedded, we assume that the

message is a binary random sequence. If we can reversibly embed L-length message $\mathbf{m} = (m_1, m_2, \cdots, m_L)$ into $\mathbf{x}$ to get the marked cover $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ with d modifications on average, we define the embedding rate as $\rho = L/N$ and the distortion as $\Delta = d/N$. Furthermore, we define the embedding efficiency as $e = \rho/\Delta$, which means the average number of bits embedded per unit distortion. We hope to get high embedding efficiency for various given embedding rates.

A direct construction for reversible data hiding is proposed by Fridrich et al. [1] as follows. First compress the cover $\mathbf{x}$ into a string $Comp(\mathbf{x})$ with a lossless compression algorithm $Comp(\ )$. The length of $Comp(\mathbf{x})$ is approximately equal to $NH(p_0)$. Therefore we can append $N(1 - H(p_0))$ bits of message $\mathbf{m}$ after $Comp(\mathbf{x})$ to obtain $\mathbf{y} = Comp(\mathbf{x})||\mathbf{m}$. The recipient can extract the message $\mathbf{m}$ from $\mathbf{y}$ and reconstruct $\mathbf{x}$ by decompressing $Comp(\mathbf{x})$. As the bits of $Comp(\mathbf{x})$ are uncorrelated with those of $\mathbf{x}$ and the message $\mathbf{m}$ is random, the expectation of distortion between $\mathbf{x}$ and $\mathbf{y}$ is 0.5. The embedding rate is equal to $(1 - H(p_0))$, which in fact is the maximum achievable embedding rate. If we only need to embed a shorter message with length equal to $\alpha N(1 - H(p_0))$ for some $\alpha < 1$, we can execute the above-mentioned method on a fraction α of the symbols in $\mathbf{x}$. In this case, the embedding rate $\rho = \alpha(1 - H(p_0))$ and the distortion $\Delta = \alpha/2$. Therefore, for the distortion constraint Δ, this simple method can achieve a rate-distortion line

$$\rho_{sim}(p_0, \Delta) = 2\Delta(1 - H(p_0)) \ . \tag{1}$$

Virtually, the simple method above is not optimal. In fact, this method achieves only a lower and fixed embedding efficiency $2(1 - H(p_0))$.

The maximum embedding rate achievable within the distortion constraint Δ is called the capacity under the distortion Δ. The following theorem proved by Kalker et al. [13] gives expression of the capacity.

Theorem 1. [13] *The reversible embedding capacity ρ_{rev} for a memoryless binary source with $p_0 \geq 1/2$ is, for $0 \leq \Delta \leq 1/2$, given by*

$$\rho_{rev}(p_0, \Delta) = H(\max(p_0 - \Delta, 1/2)) - H(p_0) \tag{2}$$

Note that the above bound can be increased for non-memoryless sequences, but we assume the binary cover is memoryless throughout this paper and this assumption in fact is suitable for most schemes.

2.2 Recursive Construction

To improve the embedding efficiency, Kalker et al. [13] proposed a recursive embedding method, which consists of a non-reversible embedding algorithm and a conditional compression algorithm. First select a non-reversible embedding code $\mathcal{E}$ with distortion D and embedding rate R. Assume the binary cover sequence $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ is sufficiently long. The sequence is segmented into disjoint blocks of length K, such that $\mathbf{x} = \mathbf{x}_1||\mathbf{x}_2||\cdots||\mathbf{x}_{N/K}$. Without loss of generality, we assume that N/K is a sufficiently large integer. By using the embedding code $\mathcal{E}$, KR bits of message $\mathbf{m}_1$ can be embedded into the first host block $\mathbf{x}_1$,

resulting to the first marked block $\mathbf{y}_1$. The recipient can reconstruct $\mathbf{x}_1$ under the condition of known-$\mathbf{y}_1$ because she can observe $\mathbf{y}_1$. Therefore the amount of information needed to reconstruct $\mathbf{x}_1$ is equal to $H(\mathbf{x}_1|\mathbf{y}_1)$, which means we can compress $\mathbf{x}_1$ into a sequence of length $H(\mathbf{x}_1|\mathbf{y}_1)$. This compressed sequence is embedded into the second block $\mathbf{x}_2$, leaving room for $KR-H(\mathbf{x}_1|\mathbf{y}_1)$ bits of auxiliary message. Similarly, the information for reconstructing $\mathbf{x}_2$ is embedded into $\mathbf{x}_3$. This process is continued recursively until $\mathbf{x}_{K-1}$. For the last block $\mathbf{x}_K$, the simple method described in Subsection 2.1 is used to complete a full reversible data hiding method. When N and N/K are large enough, the distortion of this method is equal to distortion of the code $\mathcal{E}$, and the embedding rate is equal to $R - H(\mathbf{x}_1|\mathbf{y}_1)/K$.

This recursive construction can achieve higher embedding efficiency than the simple method because of two key points: 1) the data is embedded by an efficient non-reversible embedding code; 2) the cover block is compressed under the condition of corresponding marked block. However the recursive construction proposed above cannot still approach the upper bound of embedding efficiency.

3 Improved Recursive Construction

3.1 Motivations and Overall Framework

In this section, we will improve the recursive construction to approach the upper bound of embedding efficiency for various embedding rates. To do that, we first observe the rate-distortion function ([2]), which shows that the maximum capacity is equal to $1 - H(p_0)$, and it can be achieved when distortion $\Delta = p_0 - 1/2$. In Fig. 1, we draw the rate-distortion lines for $p_0 = 0.7$ and 0.9, which show that the capacity increases with distortion Δ for $0 \leq \Delta \leq p_0 - 1/2$, but keeps equal to $1 - H(p_0)$ for $p_0 - 1/2 < \Delta \leq 1/2$. Therefore, we only need to consider how to construct codes for $0 \leq \Delta \leq p_0 - 1/2$.

In Corollary 1 of [13], Kalker et al. proved that the optimal embedding manner for $0 \leq \Delta \leq p_0 - 1/2$ is that only the most probable symbol is allowed to be modified. In other words, only zeros are allowed to be changed if $p_0 \geq 1/2$.

Inspired by the observation above, we improve the recursive construction as follows. We only embed messages into zeros of the cover block $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,k})$ to obtain the marked block $\mathbf{y}_i = (y_{i,1}, \cdots, y_{i,k})$, and thus only zeros in $\mathbf{x}_i$ will be modified for the ith block such that $1 \leq i \leq N/K - 1$. Therefore, for the position j such that $y_{i,j} = 0$, the corresponding $x_{i,j}$ must also be equal to 0. This property can be used to compress $\mathbf{x}_i$ under the condition of known-$\mathbf{y}_i$. In fact, we can first delete the symbol $x_{i,j}$ in $\mathbf{x}_i$ at position j such that $y_{i,j} = 0$ and obtain a subsequence of $\mathbf{x}_i$, denoted by $\mathbf{x}'_i$, and then compress $\mathbf{x}'_i$ by a lossless compression algorithm $Comp(\ )$. This method will extremely improve the compression rate because most symbols in $\mathbf{x}_i$ have been compressed by deletion. The compressed $\mathbf{x}'_i$, denoted by $Comp(\mathbf{x}'_i)$, cascaded with an auxiliary message are embedded into the next block $\mathbf{x}_{i+1}$ to get the next marked block $\mathbf{y}_{i+1}$. To extract the message and reconstruct the cover, the extraction process must be performed via a backward manner. To extract message from $\mathbf{y}_i$, we first extract

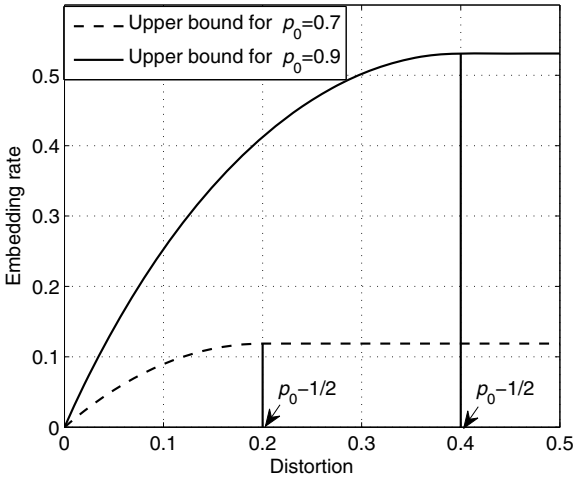


Fig. 1. Maximum capacity lines for $p_0 = 0.7$ and 0.9

message from $\mathbf{y}_{i+1}$ and obtain $\mathbf{x}'_i$ by decompression. Combining $\mathbf{x}'_i$ and $\mathbf{y}_i$, we can reconstruct $\mathbf{x}_i$ and know the positions of zeros in $\mathbf{x}_i$. According to these positions of zeros of $\mathbf{x}_i$, we can extract message from $\mathbf{y}_i$.

The detailed process and an example for embedding and extraction will be described in Subsection 3.3. We now deal with the first problem, that is, how to construct efficient codes for embedding data into an all-zero cover.

3.2 Data Embedding into All-Zero Covers

Data embedding into all-zero covers is just a special case of the coding model in Section 2 with taking $p_0 = 1$, which can be realized by a decompression algorithm, e.g., the reverse zero-run length (RZL) coding proposed by Wong et al. [6]. In this section, we proposed a more efficient embedding code by improving the RZL method.

Assume the cover sequence $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ is an N-length all-zero cover, which means every symbol $x_i = 0$ for $1 \le i \le N$. We want to embed a part of the message sequence $\mathbf{m} = (m_1, m_2, \cdots, m_L, \cdots)$ into $\mathbf{x}$. Two pointers, $P1$ and $P2$ are needed in the embedding process. $P1$ is used to label the last cover symbol that has been modified, and $P2$ is used to count the number of message bits that have been embedded. The following embedding construction is an rate-variable coding method, in which the embedding rate is determined by a parameter k, $k \ge 1$. First set $P1 = 0$ and $P2 = 0$. The encoder reads the message bit m_{P2+1}, and there are two embedding cases according to the value of m_{P2+1}.

Case 1. If $m_{P2+1} = 0$, set $P1 = P1 + 2^k$, $P2 = P2 + 1$ and one bit m_{P2+1} is embedded. In this case, no cover symbol is modified.

Case 2. If $m_{P2+1} = 1$, read the next k bits $(m_{P2+2}, \cdots, m_{P2+k+1})$, which can be represented by a decimal integer belonging to $[0, 2^k - 1]$, denoted by $(m_{P2+2}, \cdots, m_{P2+k+1})_{\text{int}}$. Set $P1 = P1 + (m_{P2+2}, \cdots, m_{P2+k+1})_{\text{int}} + 1$, $P2 = P2 + k + 1$, and flip x_{P1} from "0" to "1". Thus, $k + 1$ bits $(m_{P2+1}, \cdots, m_{P2+k+1})$ are embedded, and only one cover symbol, x_{P1}, is modified.

For both cases, we have embedded the first $P2$ bits of message into the first $P1$ cover symbols. In the same manner, we continue to embed the rest message bits $(m_{P2+1}, m_{P2+2}, \cdots)$ into the rest cover symbols $(x_{P1+1}, \cdots, x_N)$, until $N - P1 < 2^k$. The obtained marked cover is denoted by $\mathbf{y} = (y_1, y_2, \cdots, y_N)$.

To extract the message from $\mathbf{y}$, first set pointers $P1 = 0$ and $P2 = 0$. With $P1+1$ as the start point, read a 2^k length block from $\mathbf{y}$, i.e., $(y_{P1+1}, \cdots, y_{P1+2^k})$. There are also two cases according to whether the block $(y_{P1+1}, \cdots, y_{P1+2^k})$ includes "1".

Case 1. If all symbols in $(y_{P1+1}, \cdots, y_{P1+2^k})$ are equal to "0", let the $(P2+1)$th message bit $m_{P2+1} = 0$, $P1 = P1 + 2^k$, and $P2 = P2 + 1$.

Case 2. If there exists symbol "1" in $(y_{P1+1}, \cdots, y_{P1+2^k})$, search for the first index i such that $y_{P1+i} = 1$ and $y_{P1+1} = y_{P1+2} = \cdots = y_{P1+i-1} = 0$. The integer $i - 1$ can be represented by a binary sequence consisting of k bits, denoted by $(i-1)_{\text{bin}}$. Let the $(P2+1)$th message bit $m_{P2+1} = 1$, the next k bits $(m_{P2+2}, \cdots m_{P2+k+1}) = (i-1)_{\text{bin}}$, and $P1 = P1 + i$, $P2 = P2 + k + 1$.

In the same manner, extract messages from the rest symbols $(y_{P1+1}, \cdots, y_N)$ until $N - P1 < 2^k$ and there is no symbol "1" in the rest $N - P1$ symbols of the marked cover. Now we use a simple example to show the embedding and extraction process of the method above.

Example 1. Take the parameter $k = 2$. Assume the cover is a 9-length all-zero cover, i.e. N=9, and the message consists of 7 bits, as shown in Fig. 2. To embed the message, first set pointers $P1 = 0$ and $P2 = 0$, and then do the following three steps.

Step 1. Read $m_1 = 0$, thus set $P1 = P1 + 2^2 = 4$, and $P2 = P2 + 1 = 1$.

Step 2. Read $m_{P2+1} = m_2$. Because $m_2 = 1$, read the next two message bits $(m_3, m_4) = (0, 1)$, which is interpreted as a decimal integer $(0, 1)_{\text{int}} = 1$. Set $P1 = P1 + (0, 1)_{\text{int}} + 1 = 6$, $P2 = P2 + k + 1 = 4$, and flip $x_{P1} = x_6$ to "1".

Step 3. Read $m_{P2+1} = m_5$. Because $m_5 = 1$, read the next two message bits $(m_6, m_7) = (1, 0)$, interpreted as a decimal integer $(1, 0)_{\text{int}} = 2$. Set $P1 = P1 + (1, 0)_{\text{int}} + 1 = 9$, $P2 = P2 + k + 1 = 7$, and flip $x_{P1} = x_9$ to "1". As $N - P1 = 9 - 9 = 0 < 4$, the embedding process stops. The marked cover is denoted by $\mathbf{y}$ that is obtained by modifying the sixth and ninth bits of the cover $\mathbf{x}$.

To extract the message from $\mathbf{y}$, we first set $P1 = 0$ and $P2 = 0$, and then do the following three steps.

Index	1	2	3	4	5	6	7	8	9
Message **m**	0	1	0	1	1	1	0		
Cover **x**	0	0	0	0	0	0	0	0	0
Marked cover **y**	0	0	0	0	0	1	0	0	1
Embedding steps	Step 1				Step 2		Step 3		

Fig. 2. Example of data embedding into all-zero covers

Step 1. Read the first four bits $(y_1, y_2, y_3, y_4) = (0, 0, 0, 0)$ from $\mathbf{y}$. Because this is an all-zero block, set $m_1 = 0$, $P1 = P1 + 4 = 4$, and $P2 = P2 + 1 = 1$.

Step 2. With $P1 + 1 = 5$ as the start point, read successive four bits $(y_5, y_6, y_7, y_8) = (0, 1, 0, 0)$. Because the first symbol "1" appears at the second position in this block, i.e., the index $i = 2$ and $(i-1)_{\text{bin}} = (0, 1)$. Let $m_{P2+1} = m_2 = 1$ and $(m_3, m_4) = (i-1)_{\text{bin}} = (0, 1)$. Set $P1 = P1 + i = 6$ and $P2 = P2 + 1 + k = 4$.

Step 3. Although $N - P1 = 9 - 6 = 3 < 4$, the extraction process will continue because the last three symbols of $\mathbf{y}$ includes a "1". The "1" appears at the third position in $(y_7, y_8, y_9) = (0, 0, 1)$, so let the index $i = 3$ and thus $(i-1)_{\text{bin}} = (1, 0)$. Thus, we extract the last three bits of messages such that $m_{P2+1} = m_5 = 1$ and $(m_6, m_7) = (i-1)_{\text{bin}} = (1, 0)$.

In this example, we embed 7 bits of message into a 9-length cover with only 2 modifications. We denote this embedding method by $\mathcal{E}_0$. To analyze the embedding rate and distortion of $\mathcal{E}_0$, we investigate the two cases in the embedding process. In Case 1, we embed one bit into a 2^k-length cover without making any modification; in Case 2, we embed $k+1$ bits of messages by expending $n = (m_{P2+2}, \cdots, m_{P2+k+1})_{\text{int}} + 1$ cover symbols and one modification. Because the message block $(m_{P2+2}, \cdots, m_{P2+k+1})$ is random, the probability $P(n = j) = 1/2^k$ for any $j \in \{1, 2, \cdots, 2^k\}$, and thus the expectation of n is equal to

$$\frac{1}{2^k}(1 + 2 + \cdots + 2^k) = \frac{2^k + 1}{2} \ . \tag{3}$$

Therefore in Case 2, we on average embed $k+1$ bits into $(2^k+1)/2$ cover symbols with one modification. Because $P(m_{P2+1} = 0) = P(m_{P2+1} = 1) = 1/2$, the two cases occur with equal probability . In summary, for one embedding step, the average number of embedded bits is equal to

$$N_{\text{mess}} = \frac{1}{2} \times 1 + \frac{1}{2} \times (k+1) = \frac{k+2}{2} \ . \tag{4}$$

The average number of expended cover symbols is equal to

$$N_{\text{cover}} = \frac{1}{2} \times 2^k + \frac{1}{2} \times \frac{2^k + 1}{2} = \frac{2^{k+1} + 2^k + 1}{4} \ , \tag{5}$$

and the average number of modifications is equal to

$$N_{\text{modi}} = \frac{1}{2} \times 0 + \frac{1}{2} \times 1 = \frac{1}{2} \ . \tag{6}$$

Therefore the embedding rate R and distortion D of code $\mathcal{E}_0$ can be calculated as follows

$$R_0 = \frac{N_{\text{mess}}}{N_{\text{cover}}} = \frac{2k+4}{2^{k+1}+2^k+1}, \quad D_0 = \frac{N_{\text{modi}}}{N_{\text{cover}}} = \frac{2}{2^{k+1}+2^k+1} \ . \tag{7}$$

3.3 Improved Recursive Construction

We use $\mathcal{E}_0$ as the non-reversible embedding code and design a corresponding compression algorithm to improve the recursive construction in [13,14]. Assume that the binary cover sequence $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ is generated from a memoryless source satisfying $P(x_i = 0) = p_0$ and $P(x_i = 1) = p_1$. Firstly we divide $\mathbf{x}$ into N/K disjoint blocks of length K, such that $\mathbf{x} = \mathbf{x}_1||\mathbf{x}_2||\cdots||\mathbf{x}_{N/K}$. In every block we only embed messages into zero symbols via the embedding code $\mathcal{E}_0$. Therefore, we can on average embed Kp_0R_0 bits into the Kp_0 "0's" of $\mathbf{x}_1 = (x_{1,1}, \cdots, x_{1,K})$ by flipping Kp_0D_0 "0's" to "1's" because the code $\mathcal{E}_0$ has embedding rate R_0 and distortion D_0, yielding the first marked block $\mathbf{y}_1 = (y_{1,1}, \cdots, y_{1,K})$. Therefore, at the position j, such that $1 \le j \le K$ and $y_{1,j} = 0$, the corresponding $x_{1,j}$ must also be equal to "0", because no "1" in $\mathbf{x}_1$ has been flipped to "0" in $\mathbf{y}_1$. We use this property to compress $\mathbf{x}_1$ under the condition of known-$\mathbf{y}_1$ by deleting all symbols in $\mathbf{x}_1$ at position j's such that $y_{1,j} = 0$, resulting to a subsequence of $\mathbf{x}_1$. Denote this subsequence by $\mathbf{x}'_1$, and thus $\mathbf{x}'_1 = \{x_{1j} | 1 \le j \le K,\ y_{1j} = 1\}$. The ratio of zeros in $\mathbf{y}_1$ is equal to the ratio of non-modified zeros of $\mathbf{x}_1$, that is $p_0(1 - D_0)$. Thus, the ratio of ones in $\mathbf{y}_1$ is equal to $1 - p_0(1 - D_0) = p_1 + p_0D_0$, and the average length of $\mathbf{x}'_1$ is equal to $K(p_1 + p_0D_0)$. In other words, under the condition of known-$\mathbf{y}_1$, the block $\mathbf{x}_1$ is compressed to $\mathbf{x}'_1$ with compression rate $p_1 + p_0D_0$, and we can reconstruct $\mathbf{x}_1$ by replacing ones of $\mathbf{y}_1$ by the symbols of $\mathbf{x}'_1$.

Furthermore we compress $\mathbf{x}'_1$ with a lossless compression algorithm $Comp(\ )$, e.g., an arithmetic coder. Denote the ratio of zeros and ones in $\mathbf{x}'_1$ by q_0 and q_1 respectively, which can be easily computed as follows.

$$q_0 = \frac{p_0D_0}{p_1 + p_0D_0}, \quad q_1 = \frac{p_1}{p_1 + p_0D_0} \ . \tag{8}$$

Therefore $\mathbf{x}'_1$ can be compressed with the rate about equal to $H(q_0)$. In summary, when $\mathbf{y}_1$ is known, the amount of information needed to reconstruct $\mathbf{x}_1$ is equal to

$$K(p_1 + p_0D_0)H(q_0) \ . \tag{9}$$

By using the code $\mathcal{E}_0$, the compressed information $Comp(\mathbf{x}'_1)$ is embedded into the zeros of the next block $\mathbf{x}_2$, leaving room for $K(p_0R_0 - (p_1 + p_0D_0)H(q_0))$ bits of auxiliary message and resulting to the second marked block $\mathbf{y}_2$. The information for reconstructing $\mathbf{x}_2$, denoted by $Comp(\mathbf{x}'_2)$ is embedded into the third block $\mathbf{x}_3$. This process is continued recursively until the one but the last block

$\mathbf{x}_{N/K}$. For the last block $\mathbf{x}_{N/K}$, we directly compress the block and make room for $K(1-H(p_0))$ bits, into which we embed $Comp(\mathbf{x}'_{N/K-1})$ for reconstructing the second last block $\mathbf{x}_{N/K-1}$. In addition, we also embed the overhead information, such as the value of p_0, the block length K, and the parameter k used by code $\mathcal{E}_0$, into the last block.

To extract the message and reconstruct the cover, the extraction process must be performed in a backward manner. To extract messages from the ith block $\mathbf{y}_i$, for $1 \le i \le N/K-1$, we must first extract messages from $\mathbf{y}_{i+1}$ and obtain $\mathbf{x}'_i$ by decompression. Combining $\mathbf{x}'_i$ and $\mathbf{y}_i$, we can reconstruct $\mathbf{x}_i$ and know the positions of zeros in $\mathbf{x}_i$, according to which we can extract messages from $\mathbf{y}_i$ by using the code $\mathcal{E}_0$.

When N and N/K are large enough, the embedding rate and distortion of the method above can be estimated by ρ_{rec} and D_{rec} such that

$$\rho_{rec} = p_0 R_0 - (p_1 + p_0 D_0) H(q_0), \quad D_{rec} = p_0 D_0 \ . \tag{10}$$

When varying the parameter k in the code $\mathcal{E}_0$, we can get variable rate (R_0, D_0) by Eq. (7), and thus yield rate-variable recursive construction with embedding rate $\rho_{rec}(k)$ and distortion $D_{rec}(k)$ as follows.

$$\rho_{rec}(k) = p_0 \frac{2k+4}{2^{k+1}+2^k+1} - \left(p_1 + p_0 \frac{2}{2^{k+1}+2^k+1}\right) H(q_0), \tag{11}$$

$$D_{rec}(k) = p_0 \frac{2}{2^{k+1}+2^k+1}, \quad k \ge 1 \ . \tag{12}$$

Now we use an example with only two blocks to illustrate the embedding and extraction process of the method described above.

Example 2. This example is based on Example 1. As shown in Fig.3, the first cover block $\mathbf{x}_1$ consists of ten symbols with only one "1". The first seven message bits are the same as in Example 1, which are embedded into the zeros of $\mathbf{x}_1$ and generate a nine-length marked block as we have obtained in Example 1. We denote this marked block in interim step by $\mathbf{y}'_1$. Replace zeros of $\mathbf{x}_1$ by $\mathbf{y}'_1$ and generate the ultimate marked block $\mathbf{y}_1$. Denote the index set of 1's in $\mathbf{y}_1$ by Ind_1, and thus $Ind_1 = \{3, 7, 10\}$, according to which we extract bits from $\mathbf{x}_1$ and get $\mathbf{x}'_1 = (x_3, x_7, x_{10}) = (1, 0, 0)$. The sequence $\mathbf{x}'_1$ is compressed to $Comp(\mathbf{x}'_1)$ and then is embedded into the second block.

To reconstruct the cover block and extract messages from the marked block $\mathbf{y}_1$, we first count the number of ones in $\mathbf{y}_1$ that is equal to 3. Second, we extract messages from the second marked block and decompress the extracted messages successively until we get a 3-length decompressed sequence which is just $\mathbf{x}'_1$. Thus, we can reconstruct $\mathbf{x}_1$ by replacing the ones of $\mathbf{y}_1$ by $\mathbf{x}'_1$. After that we know the index set of zeros in $\mathbf{x}_1$ such that $Ind_0 = \{1, 2, 4, 5, 6, 7, 8, 9, 10\}$, according to which we extract bits from $\mathbf{y}_1$ and get the sequence $\mathbf{y}'_1$. Finally, we can extract the seven message bits from $\mathbf{y}'_1$ by using the code $\mathcal{E}_0$.

Index	1	2	3	4	5	6	7	8	9	10	Second Block
message	0	1	0	1	1	1	0	…			…
						⇕					
$\mathbf{y}_1'$	0	0		0	0	0	1	0	0	1	
						⇕					
$\mathbf{x}_1$	0	0	1	0	0	0	0	0	0	0	…
						⇕					
$\mathbf{y}_1$	0	0	1	0	0	0	1	0	0	1	Comp($\mathbf{x}_1'$)…
						⇕					
$\mathbf{x}_1'$			1				0			0	
Comp($\mathbf{x}_1'$)						Comp($\mathbf{x}_1'$)					

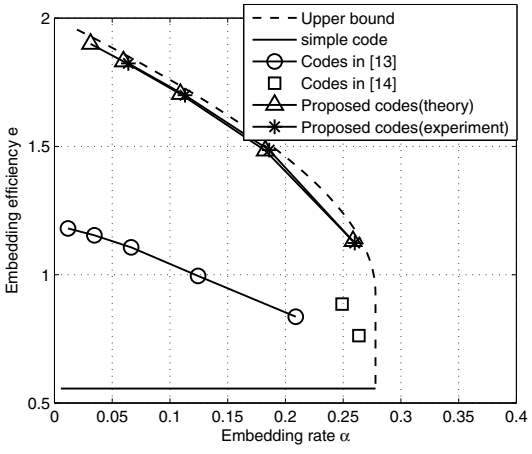
Fig. 3. Example of improved recursive construction

3.4 Performance Comparison

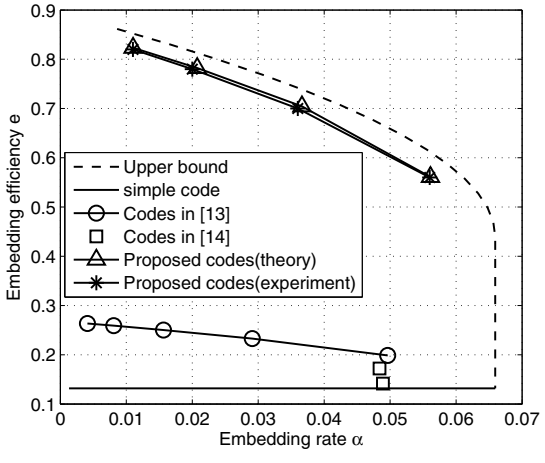
We compare the coding method above with the codes proposed in [13] and [14]. In the original recursive construction [13], Kalker and Willems used Hamming matrix embedding [16] as the non-reversible data embedding code, by which we can embed k bits of message into $2^k - 1$ cover bits by at most one modification. The Hamming codes modify zeros and ones with equal probability. Maas et al. [14] improved the original recursive construction by adjusting Hamming code to change more zeros than ones for the case $k = 2$.

Both theoretical and simulation results of the proposed method are compared with the codes in [13,14] for $p_0 = 0.8$ and 0.65. The simulation results are obtained by embedding random messages into a 2^{16}-length cover. In the experiments, we set the length of cover blocks K=200, and an adaptive arithmetic coder [15] is used as the compression algorithm $Comp(\ )$. We compare the codes by using the measurement of embedding efficiency versus embedding rate. As shown in Fig.4, the proposed codes significantly outperform the codes presented in [13,14]. For small p_0 ($p_0 = 0.6$), the embedding efficiency of codes in [13,14] is close to that of simple codes, while the embedding efficiency of proposed codes is still close to the upper bound. However, we note that the proposed codes with small parameter k will perform poor when the value of p_0 decreases. Therefore we generate codes for $p_0 = 0.8$ by using $k = 1, 2, 3, 4, 5$, while for $p_0 = 0.65$ we only use $k = 2, 3, 4, 5$. The suitable coding parameters for $p_0 \in [0.54, 1]$ are proposed in Table 1. When p_0 is smaller than 0.54, the minimal parameter k should further increase, but the capacity for such cases usually is too small for practical applications.

Note that this construction can not reach the upper bound of embedding efficiency because $\mathcal{E}_0$ is not optimal. If we use the decompressionor of an optimal compression algorithm as coding method for all-zero covers, we have $R_0 = H(D_0)$ and then it is easy to prove that Eq. (10) is equivalent to Eq. (2). Therefore, the proposed construction in fact is optimal for reversible data hiding.



(a) $p_0 = 0.8$



(b) $p_0 = 0.65$

Fig. 4. Comparison of embedding efficiency vs. embedding rate between the proposed codes and codes in [13, 14]

Table 1. Coding parameter k according to p_0

p_0	[0.66, 1]	[0.58, 0.66)	[0.54, 0.58)	...
k	≥ 1	≥ 2	≥ 3	...

4 Applications in Type-I Schemes

The coding method above can be directly applied to data hiding schemes that belong to Type-I, such as the schemes in [1,2,3,4,5,6]. Taking the regular-singular (RS) method in [1] as an example, we illustrate the ability of the proposed codes for reducing embedding distortion.

The RS method [1] is proposed for spatial images by constructing compressible features based on texture complexity. Assume the cover is a 8-bit grayscale image. The image first is divided into small groups, e.g., n pixels per group. A permutation F is used to flip the gray values, the amplitude of which is controlled by a positive inter A. For instance, when $A = 1$, the flipping function is as follows:

$$F: \ 0 \leftrightarrow 1, \ 2 \leftrightarrow 3, \ 4 \leftrightarrow 5, \cdots, \ 254 \leftrightarrow 255 \ . \tag{13}$$

For a pixel group $G = (x_1, \cdots x_n)$, $F(G) = (F(x_1), \cdots F(x_n))$. A distinguishing function f is used to detect the changing direction of the variation of the group.

$$f(G) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \ . \tag{14}$$

By using the functions F and f, the pixel group can be defined as regular (R) , singular (S), or unusable (U) such that

$$\begin{array}{l} G \in R \Leftrightarrow f(F(G)) > f(G) \\ G \in S \Leftrightarrow f(F(G)) < f(G) \\ G \in U \Leftrightarrow f(F(G)) = f(G) \end{array} \ . \tag{15}$$

For typical picture, adding some noise will lead to an increase of the variation, so we expect a bias between the number of regular groups and the number of singular groups. By assigning a "0" to a regular group and a "1" to a singular group, we can generate a binary cover sequence satisfying $p_0 > 1/2$. Flipping between "0" to "1" can be realized by applying F to the corresponding pixel group.

Usually larger amplitude A implies larger capacity but also larger embedding distortion. In our experiments, we set A from 1 to 4, and the group size $n = 4$. For each value of A, we embed messages with the original RS method and the proposed codes into 10 test images [17] with size of 512×512 (see Fig.5), and calculate the average embedding rate and average PSNR. We observed that the ratio of zeros p_0 in the RS sequence varies from 0.54 to 0.87. In our coding method, we use the minimal parameter k according to p_0 as proposed in Table

Fig. 5. Text images with size of 512×512

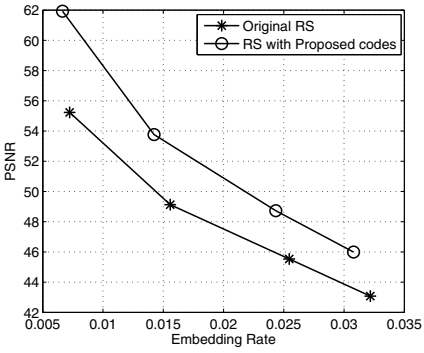


Fig. 6. Experimental results on improving RS method

1. As shown in Fig.6, the proposed codes can significantly increase the PSNRs for various embedding rates. Herein, the embedding rate is defined as bits carried by per pixel (bpp).

5 Conclusion

Most state-of-the-art reversible data hiding schemes use a strategy with separate processes of feature compression and message embedding. Kalker and Willems noted that higher embedding rate under given distortion constraint may be achieved by using joint encoding of feature compression and message embedding and thus proposed the recursive code construction. In this paper we improve the recursive construction by using not only the joint encoding above but also a joint decoding of feature decompression and message extraction. The improved codes can approach the capacity bound and significantly outperform previous codes [13,14] by embedding efficiency.

The current codes are designed for embedding data into a biased 0-1sequence. These kinds of codes cannot be directly used for the Type-II schemes such as DE-based schemes and HS-based schemes, because the Type-II schemes generate a binary feature sequence in a special compression manner that accounts for the majority of the distortion. In one of our subsequent papers, we will discuss how to convert the Type-II schemes to fit the coding model established by Kalker and Willems [13] and apply the codes propose in the present paper to improve these schemes.

Acknowledgments. This work was supported by the Natural Science Foundation of China (60803155) and the National Science and Technology Major Project (No.2010ZX03004-003). The authors would also like to sincerely thank Dr. Tomáš Filler and anonymous reviewers for their valuable comments.

References

1. Fridrich, J., Goljan, M.: Lossless Data Embedding for All Image Formats. In: Proc. of EI SPIE, Security and Watermarking of Multimedia Contents IV, San Jose, vol. 4675, pp. 572–583 (2002)
2. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless Generalized-LSB Data Embedding. IEEE Trans. on Image Processing 14(2), 253–266 (2005)
3. Xuan, G., Shi, V., Chai, P., et al.: Reversible Binary Image Data Hiding by Run-Length Histogram Modification. In: 19th International Conference on Pattern Recognition, ICPR 2008 (2008)
4. Li, S., Kot, A.C.: Privacy Protection of Fingerprint Database Using Lossless Data Hiding. In: Proceedings of the 2010 IEEE International Conference on Multimedia and Expo., pp. 1293–1298 (2010)
5. Du, R., Fridrich, J.: Lossless Authentication of MPEG-2 Video. In: Proc. of IEEE International Conference on Image Processing, vol. 2, pp. 893–896 (2002)
6. Wong, K., Tanaka, K., Takagi, K., Nakajima, Y.: Complete Video Quality-Preserving Data Hiding. IEEE Trans. on Circuits and Systems for Video Technology 19(10), 1499–1512 (2009)
7. Tian, J.: Reversible Data Embedding Using a Difference Expansion. IEEE Trans. Circuits Syst. Video Technol. 13(8), 890–896 (2003)
8. Thodi, D.M., Rodriguez, J.J.: Expansion Embedding Techniques for Reversible Watermarking. IEEE Trans. Image Process. 16(3), 721–730 (2007)
9. Hu, Y., Lee, H.-K., Li, J.: DE-based Reversible Data Hiding with Improved Overflow Location Map. IEEE Trans. Circuits Syst. Video Technol. 19(2), 250–260 (2009)
10. Ni, Z., Shi, Y.Q., Ansari, N., Wei, S.: Reversible Data Hiding. IEEE Trans. Circuits Syst. Video Technol. 16(3), 354–362 (2006)
11. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting. Signal Process. 89, 1129–1143 (2009)
12. Luo, L.X., Chen, Z.Y., Chen, M., et al.: Reversible Image Watermarking Using Interpolation Technique. IEEE Trans. Inf. Forensics and Security 5(1), 187–193 (2010)
13. Kalker, T., Willems, F.M.: Capacity Bounds and Constructions for Reversible Data-Hiding. In: Proc. of 14th International Conference on Digital Signal Processing, DSP 2002, pp. 71–76 (2002)
14. Maas, D., Kalker, T., Willems, F.M.: A Code Construction for Recursive Reversible Data-Hiding. In: Proc. Multimedia and Security Workshop at ACM Multimedia, Juan-les-Pins, France (December 6, 2002)
15. Sayood, K.: Introduction to Data Compression, pp. 87–94. Morgan Kaufmann Publishers, San Francisco (1996)
16. Crandall, R.: Some Notes on Steganography. Posted on steganography mailing list (1998), http://os.inf.tu-dresden.de/~westfeld/crandall.pdf
17. Miscelaneous gray level images, http://decsai.ugr.es/cvg/dbimagenes/g512.php

Code Obfuscation against Static and Dynamic Reverse Engineering

Sebastian Schrittwieser[1] and Stefan Katzenbeisser[2]

[1] Vienna University of Technology, Austria
sebastian.schrittwieser@tuwien.ac.at
[2] Darmstadt University of Technology, Germany
katzenbeisser@seceng.informatik.tu-darmstadt.de

Abstract. The process of reverse engineering allows attackers to understand the behavior of software and extract proprietary algorithms and data structures (e.g. cryptographic keys) from it. Code obfuscation is frequently employed to mitigate this risk. However, while most of today's obfuscation methods are targeted against static reverse engineering, where the attacker analyzes the code without actually executing it, they are still insecure against dynamic analysis techniques, where the behavior of the software is inspected at runtime. In this paper, we introduce a novel code obfuscation scheme that applies the concept of software diversification to the control flow graph of the software to enhance its complexity. Our approach aims at making dynamic reverse engineering considerably harder as the information an attacker can retrieve from the analysis of a single run of the program with a certain input, is useless for understanding the program behavior on other inputs. Based on a prototype implementation we show that our approach improves resistance against both static disassembling tools and dynamic reverse engineering at a reasonable performance penalty.

Keywords: Code obfuscation, reverse engineering, software protection, diversification.

1 Introduction

Today, software is usually distributed in binary form which is, from an attacker's perspective, substantially harder to understand than source code. However, various techniques can be applied for analyzing binary code. The process of reverse engineering aims at restoring a higher-level representation (e.g. assembly code) of software in order to analyze its structure and behavior. In some applications there is a need for software developers to protect their software against reverse engineering. The protection of intellectual property (e.g. proprietary algorithms) contained in software, confidentiality reasons, and copy protection mechanisms are the most important examples. Another important aspect are cryptographic algorithms such as AES. They are designed for scenarios with trusted end-points where encryption and decryption are performed in secure environments and withstand attacks in a black-box context, where an attacker does not have knowledge

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 270–284, 2011.

of the internal state of the algorithm (such as round keys derived from the symmetric key). In contrast to traditional end-to-end encryption in communications security, where the attacker resides between the trusted end-points, many types of software (e.g. DRM clients), have to withstand attacks in a white-box context where an attacker is able to analyze the software while its execution. This is particularly difficult for software that runs on an untrusted host.

Software obfuscation is a technique to obscure the control flow of software as well as data structures that contain sensitive information and is used to mitigate the threat of reverse engineering. Collberg et al. [8] define an obfuscating transformation τ as a transformation of a program P into a program P' so that P and P' have the same observable behavior. The original program P and the obfuscated program P' must not differ in their functionality to the user (aside from performance losses because of the obfuscating transformation), however, non-visible side effects, like the creation of temporary files are allowed in this loose definition. Another formal concept of software obfuscation was defined by Barak et al. [3]. Although this work shows that a universal obfuscator for any type of software does not exist and perfectly secure software obfuscation is not possible, software obfuscation is still used in commercial systems to "raise the bar" for attackers. In the context of Digital Rights Management systems it is the prime candidate for the protection against attackers who have full access to the client software. While the research community developed a vast number of obfuscation schemes (see e.g. [5] and [16]) targeted against static reverse engineering, where the structure of the software is analyzed without actually executing it, they are still insecure against dynamic analysis techniques, which execute the program in a debugger or virtual machine and inspect its behavior.

In this work we introduce a novel code obfuscation technique that effectively prevents static reverse engineering and limits the impact of dynamic analysis. Technically, we apply the concept of code diversification to enhance the complexity of the software to be analyzed. Diversification was used in the past to prevent "class breaks", so that a crack developed for one instance of a program will most likely not run on another instance and thus each copy of the software needs to be attacked independently. In this work we use diversification for the first time for a different purpose, namely increasing the resistance against dynamic analysis.

The main contribution of the paper is a novel code obfuscation scheme that provides strong protection against automated static reverse engineering and which uses the concept of software diversification in order to enhance the complexity of dynamic analysis. Note that we do not intend to construct a perfectly secure obfuscation scheme, as dynamic analysis can not be prevented. However, our aim is to make attacks significantly more difficult so that knowledge derived from one run of the software in a virtual machine does not necessarily help in understanding the behavior of the software in runs on other inputs.

The remainder of the paper proceeds as follows. After a short overview of related work (Section 2) we introduce our approach in Section 3. In Section 4 we explain how performance is influenced by our method and evaluate security aspects. Finally, a conclusion is given in Section 5.

2 Related Work

There are a number of publications on software obfuscation and their implementation. A comprehensive taxonomy of obfuscating transformations was introduced in 1997 by Collberg et al. [8]. To measure the effect of an obfuscating transformation, Collberg defined three metrics: *potency*, *resilience* and *cost*. *Potency* describes how much more difficult the obfuscated program P' is to understand for humans. Software complexity metrics (e.g. [6,12,22,11,13,21,19]), which were developed to reduce the complexity of software, can be used to evaluate this rather subjective metric. In contrast to potency that evaluates the strength of the obfuscating transformation against humans, *resilience* defines how well it withstands an attack of an automatic deobfuscator. This metric evaluates both the programmer effort (how much effort is required to develop a deobfuscator) and the deobfuscator effort (the effort of space and time required for the deobfuscator to run). A perfect obfuscating transformation has high potency and resilience values, but low *costs* in terms of additional memory usage and increased execution time. In practice, a trade-off between resilience/potency and costs (computational overhead) has to be made. However, the main problem of measuring an obfuscation technique's strength is that a well-defined level of security does not exist, even though it can make the process of reverse engineering significantly harder and more time consuming. Several other theoretical works on software obfuscation can be found in [17] and [23].

As preventing disassembling is nearly impossible in scenarios where attackers have full control over the host on which the software is running, the common solution is to make the result of disassembling worthless for further static analysis by preventing the reconstruction of the control flow graph. To this end, [16] and [5] use so-called branching functions to obfuscate the targets of CALL instructions: The described methods replace CALL instructions with jumps (JMP) to a generic function (*branching function*), which decides at runtime which function to call. Under the assumption that for a static analyzer the branching function is a black box, the call target is not revealed until the actual execution of the code. This effectively prevents reconstruction of the control flow graph using static analysis. However, the concept of a branching function does not protect against dynamic analysis. An attacker can still run the software on various inputs and observe its behavior. Medou et al. [18] argue that recently proposed software protection models would not withstand attacks that combine static and dynamic analysis techniques. Still, code obfuscation can make dynamic analysis considerably harder.

An attack is called a class break, if it was developed for a single entity, but can easily be extended to break any similar entity. In software, for example, we would speak of a class break if an attacker can not only remove a copy protection mechanism on the software purchased, but also can write a generic patch that removes it from every copy of the software. For software publishers, class breaks are dreaded, because they allow mass distribution of software cracks (e.g. on the Internet) to people who would otherwise not be able to develop cracks themselves. The concept of diversification for preventing class breaks of software

was put forth by Anckaert [1]. An algorithm for automated software diversification was introduced by De Sutter et al. [9]. Their approach uses optimization techniques to generate different, but semantically equivalent, assembly instructions from code sequences. While software diversification is an effective solution (see e.g. [2]), it raises major difficulties in software distribution, because each copy has to be different. There is no efficient way for the distribution of diversified copies via physical media (e.g. DVD), and software updates for diversified software are difficult to distribute as well. Franz [10] proposes a model for the distribution of diversified software on a large scale. The author argues that the increasing popularity of online software delivery makes it feasible to send each user a different version of the software. However, a specific algorithm for the diversification process is not given.

Another approach to protect cryptographic keys embedded in software is the use of White-Box Cryptography (WBC), which attempts to construct a decryption routine that is resistant against a "white-box" attacker, who is able to observe every step of the decryption process. In WBC, the cipher is implemented as a randomized network of key dependent lookup tables. A white-box DES implementation was introduced by Chow et al. [7]. Based on this approach, other white-box implementations of DES and AES have been proposed, but all of them have been broken so far (see e.g. Jabob et al. [14], Wyseur et al. [24] and Billet et al. [4]). Michiels and Gorissen [20] introduce a technique that uses white-box cryptography to make software tamper-resistant. In their approach, the executable code of the software is used in a white-box lookup table for the cryptographic key. Changing the code would result in an invalid key. However, due to the lack of secure WBC implementations, the security of this construction is unclear.

Hardware-based approaches would allow to completely shield the actual execution of code from the attacker. However, this only moves attacks to the tamper resistance of the hardware, while raising new challenges like difficult support for legacy systems and high costs. Therefore, hardware-based software protection is out of scope of this work.

3 Approach

Our approach combines obfuscation techniques against static and dynamic reverse engineering. Within this paper, the term static analysis refers to the process of automated reverse engineering of software without actually executing it. Using a disassembler, an attacker can translate machine code into assembly language, a process that makes machine instructions visible, including ones that modify the control flow such as jumps and calls. This way, the control flow graph of the software can be reconstructed without executing even a single line of code. By inserting indirect jumps that do not reveal their jump target until runtime and utilizing the concept of a branching function we make static control flow reconstruction more difficult.

Employing code obfuscation to prevent static analysis is a first step towards running code securely, even in the presence of attackers who have full access

to the host. However, an attacker is still able to perform dynamic analysis of the software by executing it. The process of disassembling and stepping through the code reveals much of its internal structure, even if obfuscating transformations were applied to the code. Preventing dynamic analysis in a software-only approach is not fully possible as an attacker can always record executed instructions, the program's memory, and register values of a single run of the software. However, in our approach we aim at making dynamic analysis considerably harder for the attacker by applying concepts from diversification. In particular, the information an attacker can retrieve from the analysis of a single run of the program with certain inputs is useless for understanding the trace of another input. It thus increases costs for an attacker dramatically, as the attacker needs to run the program many times and collect all information to obtain a complete view of the program. This concept can be considered as diversification of the control flow graph.

3.1 Protection against Static Reverse Engineering

In our approach we borrow the idea of a branching function to statically obfuscate the control flow of the software. While previous implementations replace existing CALL instructions with jumps to the branching function, we split the code into small portions that implement only a few instructions and then jump back to the branching function. While this increases the overhead, it makes the blocks far more complex to understand. Because of the small size of code blocks, they leak only little information: A single code block usually is too small for an attacker to extract useful data without knowing the context the code block is used inside the software. The jump from the branching function to the following code block is indirect, i.e. it does not statically specify the memory address of the jump target, but rather specifies where the jump target's address is located at runtime. Static disassembling results in a huge collection of small code blocks without the information on how to combine them in the correct order to form a valid piece of software.

Figure 1 explains this approach. The assembly code of the software is split into small pieces, which we call *gadgets*. At the end of each gadget we add a jump back to the branching function. At runtime, this function calculates, based on the previously executed gadget, the virtual memory address of the following gadget and jumps there. The calculation of the next jump target should not solely depend on the current gadget, but also on the history of executed gadgets so that without knowing every predecessor of a gadget, an attacker is not able to calculate the address of the following one. We achieve this requirement by assigning a signature to each gadget (see Section 3.3). During runtime, the signatures of executed gadgets are summed up and this sum is used inside the branching function as input parameter for a lookup table that contains the address of the subsequent gadget. Without knowing the signature sum of all predecessors of a gadget, it is hard to calculate the subsequently executed gadget.

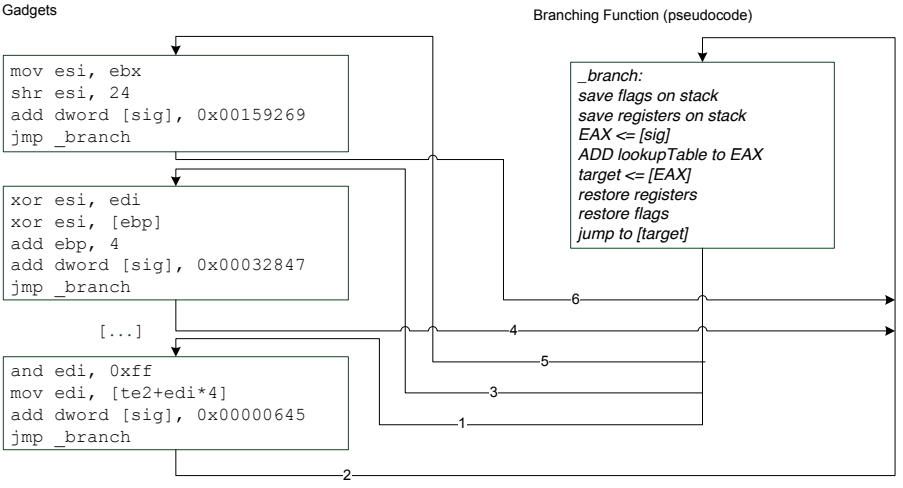


Fig. 1. Overall architecture of the obfuscated program: small code blocks (gadgets) are connected by a branching function

3.2 Protection against Dynamic Reverse Engineering

The approach effectively prevents static analysis, as a debugger is not able to connect gadgets to each other without calculating signature sums and executing the branching function. Dynamic analysis, however, reveals all gadgets used in a single invocation of the software as well as their order. An attacker can easily remove the jumps to the branching function by just concatenating called gadgets in their correct order. By performing this task for several inputs, he gets significant information on the software behavior.

To mitigate that risk, we diversify the control flow graph of the software so that it contains many more control flow paths than the original implementation. We diversify gadgets (i.e. add semantically identical but syntactical different gadgets to the code) and add input dependent branches so that different gadgets get executed upon running the software with different inputs. We can symbolize this by a gadget graph, where the actual gadget code is stored in the edges that connect two nodes, which symbolize the state of a program. Figure 2 shows the multi-target branching concept before gadget diversification. For every node, we create outgoing edges and fill them with gadgets (i.e. instructions from the original code). All outgoing edges of one node start with the same instruction and only differ in gadget length. In a further step, these gadgets are diversified. Every path through the graph is a valid trace of the program. The branches are input dependent: based on the program's input the branching function decides which path through the graph has to be taken. For a logical connection between gadgets, we implement a path signature algorithm that uniquely identifies the currently executed node and all its predecessors (see Section 3.3).

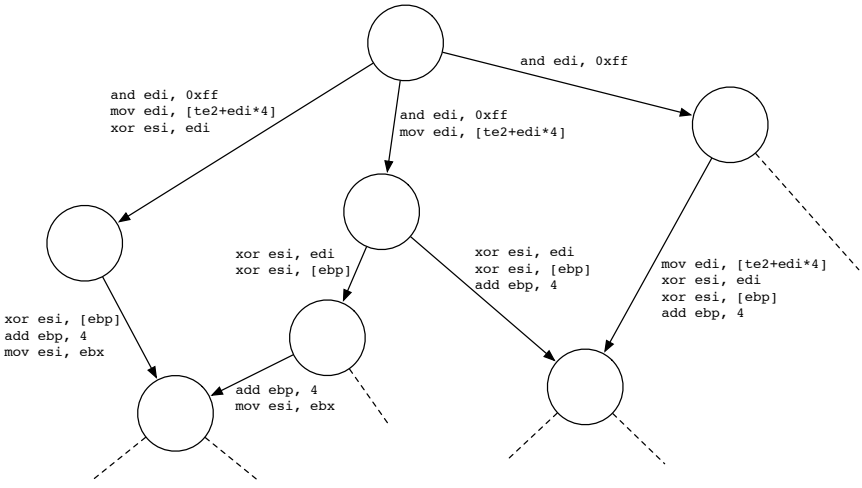


Fig. 2. Gadget graph

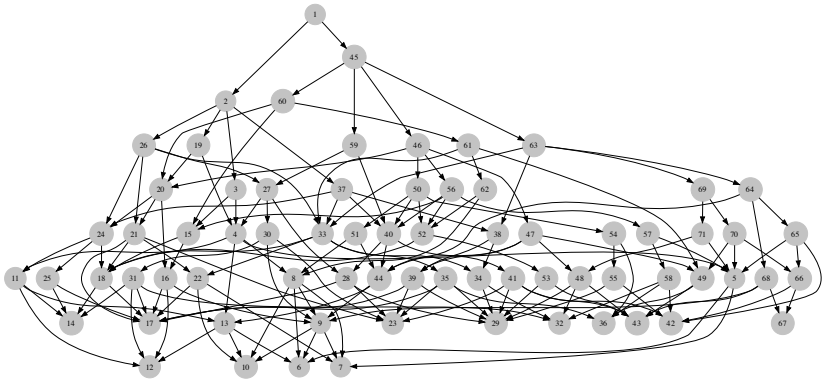


Fig. 3. Diversified control flow graph

In order to increase the security of the obfuscation, we prevent that a path that is valid for one input is also valid for other inputs. We do this by modifying some instruction's operands and automatically compensate these modifications during runtime by corrective input data. Consider, for example, the assembly instruction `add eax, 8`. If we replace this instruction with `add eax, ebx; sub eax, 1`, where the content of the register `eax` is derived from the program's input, only a value of 9 in `ebx` would yield to the correct value in register `eax`. Figure 3 shows a more complex control flow graph.

All paths through this graph are valid and semantically equal traces of the program. However, because of the inserted modifications to operands, one specific path yields correct computation only for a specific input (or a group of inputs) and fails otherwise. If an attacker would use the trace of one input for running the program in the context of another input (e.g. by diverting the control flow in the

branching function), our modifications to operands would not be compensated by the new input and the program would show unexpected behavior and might crash at some point (e.g. because of access to miscalculated memory addresses). The process of creating the diversified gadget graph is much easier and faster than breaking the obfuscation as an attacker has to obtain each trace individually.

At the beginning of our obfuscation algorithm, a random gadget graph is created from the software to be obfuscated, based on the input parameters for branching level and gadget size. We then generate unique path signatures (for details see Section 3.3) inside a depth-first search that traverses through all possible paths of the graph. Furthermore, we diversify the gadget code (see Section 3.4), assign the path signature to the gadget and add the gadget to the output file. For every possible path that can be taken to reach a gadget, we add the gadget's memory address and path signature sum to the lookup table. Finally, we attach the branching function and the lookup table to the obfuscated code. Algorithm 1 shows the obfuscation algorithm in pseudocode.

Algorithm 1. Obfuscation algorithm in pseudocode

```
create random gadget graph
DepthFirstSearch (graph)
  while path signature of current gadget is not unique do
    create random path signature
  end while
  diversify gadget code
  add path signature to gadget
  output gadget code
  add gadget's memory address and path signature sum to lookup table
end DepthFirstSearch
output branching function
output lookup table
```

3.3 Graph Construction

The main challenge of our approach against dynamic reverse engineering is the performance of the obfuscation algorithm. One the one hand, our approach aims to significantly delay dynamic analysis of an attacker by making it hard to traverse the entire graph within a reasonable time frame (i.e. a brute force attack). However, on the other hand, the initial construction of the graph has to be dramatically less time consuming than an attack. We solve this problem with full knowledge of the structure of the graph at obfuscation time compared to runtime. The obfuscation algorithm creates the graph and stores its structure in memory, allowing very efficient graph traversal at obfuscation time. In contrast, an attacker only has access to the binary code of the software that does not contain an explicit description of the graph's structure. An attacker has to execute all (or at least most) paths of the graph through the branching function, including the gadget's entire code, in order to rebuild the graph and obtain a complete view of the software.

Our graph construction algorithm takes the original program code as well as a minimum and maximum gadget size and a minimum and maximum branching size as input parameters and is based on a depth-first search. Starting at the root node, the algorithm adds a random number of child nodes (within the bounds of the branching size) and assigns a gadget to each connecting edge. All edges to child nodes contain the same code by means of being filled with a random number of instructions (within the given bounds on the gadget size) from the original code. Only the gadget size and therefore the number of instructions differ at this stage. Gadgets are not diversified at graph construction time. We define the absolute number of instructions executed until reaching a node of the graph as *node level*. Before adding a new node to the graph, the algorithm calculates the node level of the new node and checks if it already exists anywhere in the graph. It that case, instead of creating the node, the algorithm links to the existing node. This method prevents a continually growing width of the graph.

During gadget graph construction, we calculate and store a path signature in each node. We make it unique (see below) so that it clearly identifies the node and all its predecessors. The signature is based on simple ADD and SUB assembly instructions on a fixed memory location. Each gadget adds (or subtracts) a random value to (or from) the value stored in memory. When traversing through the graph, the value stored at the memory location identifies the currently executed gadget and the path that was taken through the graph to reach this gadget. A node can have more than one signature, as more than one path of the graph could reach this node. In that case, each node signature uniquely identifies one of the possible paths from the root to the node. During signature assignment we prevent collisions (two nodes sharing the same signature), by comparing the current signature to all previously calculated signatures and choosing a different value for the ADD or SUB instruction if needed. We decided to implement a trail-and-error approach instead of an algorithm that generates provable distinct signatures to avoid performances bottlenecks at runtime. Figure 4 shows the path signature for a small graph.

We further add a second input parameter to the branching function described in the static part of our approach. Now, both the program's input and the path signature are input parameters for a lookup table that determines the next gadget to be called. To eliminate any information leakage from the branching function's input value, only a hash value of the program's input and the path signature is stored in the lookup table.

3.4 Automatic Gadget Diversification

An efficient generation of semantically equivalent mutations of gadgets is the key challenge for software diversification. This process has to be fully automatic to be able to process large amounts of source code and the transformation function is preferably one-way to prevent differential analysis of gadgets. Pattern-based diversification algorithms (e.g. [9]) are a reasonable first code replacement step. However, the fact that an attacker only has local view on a gadget, can help to

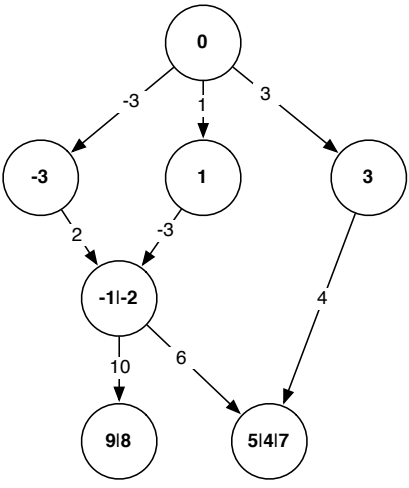


Fig. 4. Path signatures

improve the strength of the diversification by inserting code dependency problems that are locally undecidable for an attacker.

We propose a combination of dummy code insertions and a process we call *instruction splitting*. The idea is to split basic instructions into two ore more instructions that are in combination semantically equivalent to the original instruction and then insert dummy code instructions in between them. We create bogus dependencies between the actual gadget code and dummy instructions by accessing data of split instructions inside the dummy code. To identify and remove dummy instructions, an attacker has to be sure that the code does not perform any vital operations on the code that is executed afterwards. However, this problem is hard to decide due to dependencies between gadgets. Because of the small gadgets sizes, an attacker only has local view on a gadget without knowledge of the subsequently executed gadget.

A simple example is the instruction `add eax, 5` that can be split into the two instructions `add eax, 2` and `add eax, 3`. Of course, this simple transformation provides only very limited security against automatic gadget matching algorithms. We can, however, tremendously improve the strength of the transformation by inserting dummy code. For example, the instruction `mov dword [0x0040EA00], eax` can be considered as dummy code, if the value that is stored in `0x0040EA00` is not used anywhere later in the software. The instruction sequence `add eax, 2; mov dword [0x0040EA00], eax; add eax, 3` is only semantically equivalent to `add eax, 5`, if `mov dword [0x0040EA00], eax` is dummy code. For an attacker with only local knowledge, this is an ambiguous problem.

Simple pattern based transformations do not withstand automated attacks aiming at reversing the diversification. The instructions `test eax, eax` and `cmp eax, 0` are semantically equivalent, but the transformation is weak,

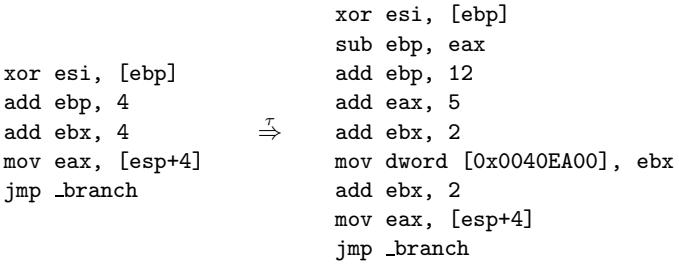


Fig. 5. Code block diversification and obfuscation

because a very simple matching algorithm can easily identify them as equivalent. However, analogous to the instruction splitting method, multi-instruction patterns can be combined with dummy code insertions to enable strong diversification. To provide an example, consider the instructions `push ebp; mov ebp, esp`. A semantically equivalent expression would be `push ebp; push esp; pop ebp`. A simple substitution transformation of one version for the other would most likely not withstand an automated attack. However, if the transformation is combined with dummy code insertion (e.g. `push ebp; push esp; add esp, [0x0040EA00]; pop ebp`, where `0x0040EA00` is `0`), an attacker with local knowledge of the gadget can not reveal the dummy code instructions and hence can not decide gadget equivalence locally.

Figure 5 shows the transformation of a small code block. The transformation function τ adds dummy code (lines 4 and 6) and modifies the instruction `add ebp, 4` so that it only provides the correct functionality if the corresponding input `8` is loaded into register `eax`. This modification prevents an attacker from extracting this specific (and fully functional) trace and using it with other inputs. To be able to generalize a trace, all input dependent operand modifications would have to be removed, thus the entire code would have to be analyzed instruction by instruction.

4 Discussion

The following section discusses the impact of our obfuscation scheme on performance and size of the resulting program and evaluates security aspects.

Performance and Size. To demonstrate the effectiveness of our approach, we implemented a prototype that reads assembly source code and generates an obfuscated version of it. We measured the performance losses of a simple benchmarking tool as well as a standard AES implementation using 8 different gadgets sizes. While the dynamic part of our approach accounts for an increase in required memory space because of diversified copies of gadgets, execution time heavily depends on the size and implementation of the branching function, as it inserts additional instructions. The performance decreases with the number

of gadgets, due to calls to the branching function, which are required to switch between gadgets. In contrast, the strength of the obfuscation is directly proportional to the number of gadgets, so a trade-off between obfuscation strength and performance has to be made. We compared different gadget sizes from 1 to 50 with the execution times of the non-obfuscated programs (see Figure 6). While very small gadgets result in significant performance decreases, the execution time for a program with a gadget sizes of 10 and bigger approximates the execution time for the original program.

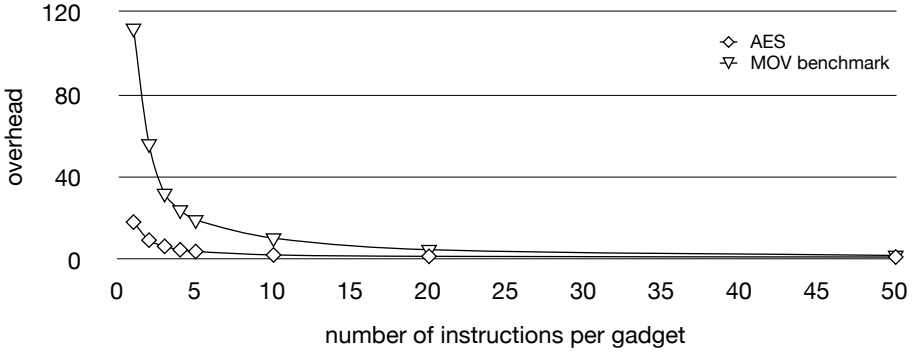


Fig. 6. Execution time for different gadget sizes

Security. We classified our method with Collberg's metric. Potency (strength against humans) can be evaluated with software complexity metrics. *Program Length* [11], *Nesting Complexity* [12], and *Data Flow Complexity* [22] are increased by our obfuscating transformation and we rate its potency level similar to Collberg's transformation "Parallelize Code" (potency level: *high*). Both methods hide the control flow graph and allow the attacker only local view on small code blocks.

Resilience (strength against automated deobfuscators) is based on the runtime of a deobfuscator and the scope of the obfuscation transformation. The runtime grows *exponentially* with the size of the software and the branching level of the resulting graph, as a deobfuscator has to traverse through the entire graph to reconstruct the control flow. For example, splitting a small program (100 assembly instructions) into gadgets of 12 to 15 instructions and building a gadget graph where every node has 2 to 3 child nodes, yields to more than 1800 different paths through this graph. In Collberg's classification, the scope of our transformation is "*global*". The combination of both measures results in the resilience level "*strong*".

We furthermore used two state-of-the-art reverse engineering tools to evaluate the strength of the static part of our approach. At first, we tried to reconstruct the program's control flow with the disassembler IDA Pro 5.6. Table 1 compares the automated disassembling rates for the original versions of the code and the

Table 1. Amount of successfully reconstructed code areas (IDA Pro)

AES algorithm		MOV benchmark	
original	obfuscated	original	obfuscated
37.96%	10.27%	100%	0.13%

obfuscated ones. The values in the table are the percent of successfully reconstructed areas. While IDA Pro was able to reconstruct nearly 38% of the original AES code, the percentage for the obfuscated version declined to about 10%. For the MOV benchmark, the difference was even larger. The results show that for both the AES algorithm and the MOV benchmark, the obfuscated version was much more difficult to reconstruct for IDA Pro. The huge differences between the two examples was caused by different amount of obfuscated code. While for the MOV benchmark the entire code was obfuscated, in the AES example only the algorithm itself was obfuscated. IDA Pro was able to reconstruct non-obfuscated parts of the code correctly, but failed at reconstructing obfuscated code. The disassembler is not able to determine the jump targets of the branching function without actually executing it.

The second tool we used for evaluation is Jakstab [15] which aims at recovering control flow graphs. Jakstab was not able to resolve the indirect jump at the end of the branching function of our sample program. Although it successfully extracted some of the jump targets from the lookup table, the correct order of the jumps still remained unknown to Jakstab.

Although both tools implement methods for disassembling software and reconstructing control flow graphs, it is not surprising to see them fail at breaking our proposed obfuscation technique as they are not tailored to our particular implementation. Hence, for a more realistic evaluation we also discuss on what a possible deobfuscator for our approach would look like.

One of the main strengths of our approach is that obfuscated software does not contain an explicit representation of the graph structure. It is hidden inside the lookup table, which only reveals the direct successor of a gadget within a single trace during runtime. If an attacker wants to manipulate the software (e.g. remove a copy protection mechanism) he could pursue the following two strategies:

- **Reconstructing the entire graph.** Without obfuscation, an attacker would search for the copy protection code inside the software and then remove it. In our diversified version of the software, however, multiple different versions of the copy protection are distributed over the entire code. Moreover, they are split into small blocks to fit into the gadgets. An attacker could execute every possible trace of the software and so reconstruct the entire control flow graph. The result would, without doubt, reveal the structure of the code as the individual traces can be analyzed separately. However, the enormous number of possible paths through the graph makes this approach time consuming.

– **Removing diversity of a single trace.** Alternatively, the attacker could remove the copy protection code from one trace and then make this trace valid for all inputs (i.e. remove diversity). The main challenge of this approach is, that the attacker has to analyze and understand the entire trace to be able to identify and remove modifications to operands that were inserted during obfuscation time to bind the code to a specific input.

Neither strategy can likely be performed without human interaction. In the first one, a large number of variants of the same copy protection mechanism would have to be identified and removed manually from the individual traces. In the second strategy, a human deobfuscator would have to analyze an entire trace to be able to identify the inserted modifications that make the trace specific to a single input. We believe, that this high amount of manual effort significantly raises the bar for reverse engineering attacks.

5 Conclusion

This paper proposed a novel software obfuscation method, based on control flow diversification, which makes it difficult for an attacker to relate structural information obtained by running a program several times and logging its trace. By splitting code into small portions (gadgets) before diversification, we achieve a complex control flow graph and static analysis can only reveal very limited local information of the program. We practically evaluated the strength of our approach against automated deobfuscators and showed that it can dramatically increase the effort for an attacker. A performance evaluation showed observable slowdowns for very small gadgets sizes, due to the vast amount of inserted jumps. Versions with bigger gadgets, however, yield to very reasonable performance results.

Future work includes the development of more sophisticated diversification techniques. In contrast to the current implementation where diversification is done only inside gadgets, we consider inter-gadget diversification as an even more effective method against automated gadget matching algorithms.

References

1. Anckaert, B., De Bosschere, K.: Diversity for Software Protection
2. Anckaert, B., De Sutter, B., De Bosschere, K.: Software piracy prevention through diversity. In: Proceedings of the 4th ACM Workshop on Digital Rights Management, DRM 2004, pp. 63–71. ACM, New York (2004)
3. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (Im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
4. Billet, O., Gilbert, H., Ech-Chatbi, C.: Cryptanalysis of a white box AES implementation. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 227–240. Springer, Heidelberg (2005)

5. Cappaert, J., Preneel, B.: A general model for hiding control flow. In: Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management. ACM, New York (2010)
6. Chidamber, S., Kemerer, C.: A metrics suite for object oriented design. IEEE Transactions on Software Engineering 20(6) (2002)
7. Chow, S., Eisen, P., Johnson, H., van Oorschot, P.: A white-box DES implementation for DRM applications. In: Digital Rights Management, pp. 1–15 (2003)
8. Collberg, C., Thomborson, C., Low, D.: A taxonomy of obfuscating transformations (1997)
9. De Sutter, B., Anckaert, B., Geiregat, J., Chanet, D., De Bosschere, K.: Instruction set limitation in support of software diversity. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 152–165. Springer, Heidelberg (2009)
10. Franz, M.: E unibus pluram: massive-scale software diversity as a defense mechanism. In: Proceedings of the 2010 Workshop on New Security Paradigms. ACM, New York (2010)
11. Halstead, M.: Elements of software science. Elsevier, New York (1977)
12. Harrison, W., Magel, K.: A complexity measure based on nesting level. ACM Sigplan Notices 16(3) (1981)
13. Henry, S., Kafura, D.: Software Structure Metrics Based on Information Flow. IEEE Transactions on Software Engineering 7(5), 510–518 (1981)
14. Jacob, M., Boneh, D., Felten, E.: Attacking an obfuscated cipher by injecting faults. In: Digital Rights Management, pp. 16–31 (2003)
15. Kinder, J., Veith, H.: Jakstab: A static analysis platform for binaries. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 423–427. Springer, Heidelberg (2008)
16. Linn, C., Debray, S.: Obfuscation of executable code to improve resistance to static disassembly. In: Proceedings of the 10th ACM Conference on Computer and Communications Security. ACM, New York (2003)
17. Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004)
18. Madou, M., Anckaert, B., De Sutter, B., De Bosschere, K.: Hybrid static-dynamic attacks against software protection mechanisms. In: Proceedings of the 5th ACM Workshop on Digital Rights Management, pp. 75–82. ACM, New York (2005)
19. McCabe, T.: A complexity measure. IEEE Transactions on Software Engineering (1976)
20. Michiels, W., Gorissen, P.: Mechanism for software tamper resistance: an application of white-box cryptography. In: Proceedings of the 2007 ACM Workshop on Digital Rights Management, pp. 82–89. ACM, New York (2007)
21. Munson Taghi, M., John, C.: Measurement of data structure complexity. Journal of Systems and Software 20(3), 217–225 (1993)
22. Oviedo, E.: Control flow, data flow and program complexity. McGraw-Hill, Inc., New York (1993)
23. Wee, H.: On obfuscating point functions. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. ACM, New York (2005)
24. Wyseur, B., Michiels, W., Gorissen, P., Preneel, B.: Cryptanalysis of white-box DES implementations with arbitrary external encodings. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007. LNCS, vol. 4876, pp. 264–277. Springer, Heidelberg (2007)

Countering Counter-Forensics: The Case of JPEG Compression

ShiYue Lai and Rainer Böhme

European Research Center for Information System (ERCIS)
University of Münster, Leonardo-Campus 3, 48149 Münster, Germany
{shiyue.lai,rainer.boehme}@ercis.uni-muenster.de

Abstract. This paper summarizes several iterations in the cat-and-mouse game between digital image forensics and counter-forensics related to an image's JPEG compression history. Building on the counter-forensics algorithm by Stamm et al. [1], we point out a vulnerability in this scheme when a maximum likelihood estimator has no solution. We construct a targeted detector against it, and present an improved scheme which uses imputation to deal with cases that lack an estimate. While this scheme is secure against our targeted detector, it is detectable by a further improved detector, which borrows from steganalysis and uses a calibrated feature. All claims are backed with experimental results from 2×800 never-compressed never-resampled grayscale images.

Keywords: Image Forensics, Counter-Forensics, JPEG Compression.

1 Introduction

Advances in information technology, the availability of high-quality digital cameras, and powerful photo editing software have made it easy to fabricate digital images which appear authentic to the human eye. Such forgeries are clearly unacceptable in areas like law enforcement, medicine, private investigations, and the mass media. As a result, research on computer-based forensic algorithms to detect forgeries of digital images has picked up over the past couple of years.

The development of digital image forensics soon became accompanied by so-called "tamper hiding" [2] or digital image counter-forensics. The aim of digital image counter-forensics, as the name suggests, is to fool current digital image forensics by skillfully taking advantage of their limitations against intelligent counterfeiters. In other words, counter-forensics challenge the reliability of forensic algorithms in situations where the counterfeiter has a sufficiently strong motive, possesses a background in digital signal processing, and disposes of detailed knowledge of the relevant digital forensic algorithms. Hence, studying counter-forensics is essential to assess the reliability of forensics methods comprehensively, and eventually to improve their reliability.

However, almost all counter-forensic algorithms are imperfect. This means they may successfully mislead specific forensic algorithms against which they are designed. But their applications either skips over some traces of counterfeit

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 285–298, 2011.

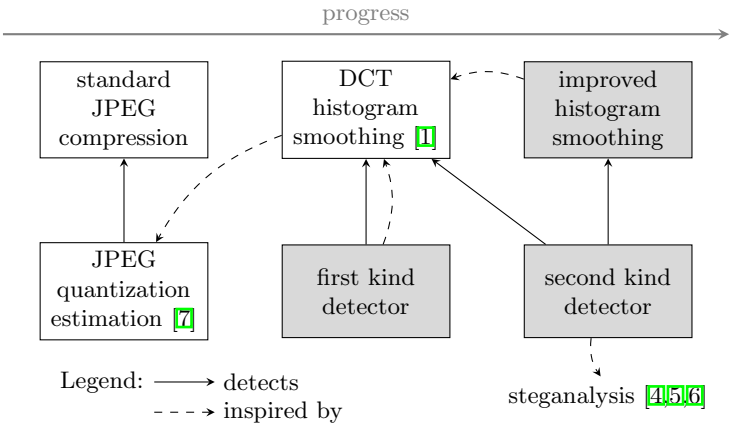


Fig. 1. Succession of advances in JPEG forensics and counter-forensics: contributions of this paper are shaded in gray

uncovered or leaves additional traces in the resulting images. These traces, in turn, can be exploited to detect that counter-forensics has been applied. This limitation is mentioned in the early literature on counter-forensics [2] and we are aware of one specific publication in the area of digital camera identification [3]. This paper contributes new targeted detectors of counter-forensics against the forensic analysis of an image's JPEG compression history.

Due to its popularity for storing digital images of natural scenes, JPEG compression is an image processing operation that is of great interest to researchers in both forensics and counter-forensics. In this paper, first, by exploring features of the high-frequency AC coefficient distributions, a novel targeted detector is presented to detect counter-forensics of JPEG compression proposed by Stamm et al. [1]. We shall refer to this detector as "first kind" detector throughout this paper. Then we improve the original counter-forensic algorithm by estimating the relation between the distributions of AC coefficients. This novel algorithm avoids obvious identifying traces of the original counter-forensics method and, as a result, it is undetectable by our first kind detector. Finally, a more powerful "second kind" detector is presented, which can be used to detect both the original method and the improved method. This novel forensic tool borrows from techniques in steganalysis, namely the so-called "calibration" of JPEG DCT coefficient distributions [4,5,6]. To maintain better overview, Figure 1 visualizes the above-described succession of advances in forensics and counter-forensics by depicting the relations between our contributions and existing work. The effectiveness of all presented methods is validated experimentally on a large set of images for different JPEG quality factors.

This paper is organized as follows. Section 2 briefly reviews related work. A targeted detector against Stamm et al.'s [1] counter-forensics of JPEG compression is presented in Section 3. An improved variant of the counter-forensics

method and a more powerful detector are proposed in Sections 4 and 5, respectively. Section 6 concludes with a discussion.

2 Related Work

We refrain from repeating the details of standard JPEG compression here. Forensics and counter-forensics of JPEG compression have attracted the interest of researchers recently. Regarding *forensic* techniques, Fan and Queiroz [7] propose a way to detect JPEG compression history of an image. The authors provide a maximum-likelihood (ML) method to estimate the quantization factors from a spatial domain bitmap representation of the image. Farid [8] verifies the authenticity of JPEG images by comparing the quantization table used in JPEG compression to a database of tables employed by selected digital camera models and image editing software. He et al. [9] demonstrate how local evidence of double JPEG compression can be used to identify image forgeries. Pevny and Fridrich [10] present a method to detect double JPEG compression using a maximum likelihood estimator of the primary quality factor.

Note that all these fruits of researches on forensics of JPEG compression are obtained under the assumption that no counter-forensics algorithms has been applied to suppress evidence of image tampering or to change other forensically significant image properties.

Regarding *counter-forensics* of JPEG compression, Stamm et al. [1] propose a counter-forensics operation capable of disguising key evidence of JPEG compression by restoring the DCT coefficients according to their model distribution. In a separate publication [11], the same team further proposes an anti-forensic operation capable of removing blocking artifacts from a previously JPEG-compressed image. They also demonstrate that by combining this operation with the method in [1], one can mislead a range of forensic methods designed to detect a) traces of JPEG compression in decoded images, b) double JPEG compression, and c) cut-and-paste image forgeries.

At the time of submission we were not aware of any prior work on countering counter-forensics specific to an image's JPEG compression history. Only after the presentation of our work at the Information Hiding Conference we learned about independent research of another counter-forensic technique against Stamm et al.'s [1] method: Valenzise et al. [12] present a detector based on noise measures after recompression with different quality factors.

3 Detecting Stamm et al.'s Counter-Forensics of JPEG Compression

3.1 Stamm et al.'s DCT Histogram Smoothing Method

The intuition of Stamm et al.'s counter-forensic method is to smooth out gaps in the individual DCT coefficient histograms by adding noise according to a distribution function which is conditional to the DCT subband, the quantization

factor, and the actual value of each DCT coefficient [1]. The conditional noise distribution for the AC coefficients is derived from a Laplacian distribution model of AC DCT coefficients in never-quantized digital images.[1] The maximum likelihood estimates of the Laplacian parameter $\lambda_{(i,j)}$ for the frequency subbands $I_{(i,j)}$, $(i,j) \in \{0,\ldots,7\}^2$, are obtained from the quantized AC coefficients using the formula:

$$\hat{\lambda}_{\mathrm{ML}(i,j)} = -\frac{2}{Q_{(i,j)}} \ln\left(\gamma_{(i,j)}\right), \tag{1}$$

where $\gamma_{(i,j)}$ is defined as

$$\gamma_{(i,j)} = \frac{\sqrt{Z_{(i,j)}^2 Q_{(i,j)}^2 - (2V_{(i,j)}Q_{(i,j)} - 4S_{(i,j)})(2NQ_{(i,j)} + 4S_{(i,j)})}}{2NQ_{(i,j)} + 4S_{(i,j)}} - \frac{Z_{(i,j)}Q_{(i,j)}}{2NQ_{(i,j)} + 4S_{(i,j)}}. \tag{2}$$

Here, $S_{(i,j)} = \sum_{k=1}^{N} |y_k|$, y_k is the k-th quantized coefficient in subband (i,j), N is the total number of the quantized (i,j) AC coefficients. $Z_{(i,j)}$ is the number of coefficient that take value zero and $V_{(i,j)}$ is the number of nonzero coefficients in this subband. $Q_{(i,j)}$ is the quantization factor for subband (i,j) which is either known (if the image is given in JPEG format) or has to be estimated, e. g., with the method in [7]. Once $\hat{\lambda}_{\mathrm{ML}(i,j)}$ is known, the AC coefficients can be adjusted within their quantization step size to fit the histogram as close as possible to a Laplacian distribution with parameter $\hat{\lambda}_{\mathrm{ML}(i,j)}$.

In the high frequency subbands, because $Q_{(i,j)}$ is high, it is more likely that all coefficients are quantized to zero. This happens more often the smaller the overall JPEG quality factor (QF) is. If all the coefficients are zero, i. e., $V = 0, Z = N, S = 0$, then according to Eq. (2), $\gamma_{(i,j)} = 0$. Hence the parameter estimate $\hat{\lambda}_{\mathrm{ML}(i,j)}$ is undefined in Eq. (1). Our experiments suggest that this happens pretty often. Table 1 shows the average share of all-zero frequency subbands with different QF for 1600 tested images.

In this situation, Stamm et al.'s method just allows these coefficients to remain unmodified for that the perturbations to each DCT coefficient caused by mapping all decompressed pixel values to the integer set $\{0,\ldots,255\}$ in the spatial domain will result in a plausible DCT coefficient distribution after retransformation into the DCT domain by the forensic investigator. However, as we shall see, this conjecture is too optimistic. It leaves us a clue to build our first kind targeted detector.

[1] A non-parametric model is employed for the DC coefficients, which in general do not follow any parsimonious distribution model. This part is not touched in this paper.

Table 1. Average share of all-zero frequency subbands for different JPEG qualities

JPEG quality factor (QF)	60	70	80	90
All-zero frequency subbands (in %)	23.4	19.2	11.5	2.5

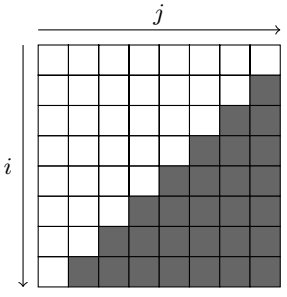


Fig. 2. The high frequency subbands according to our definition (shaded cells)

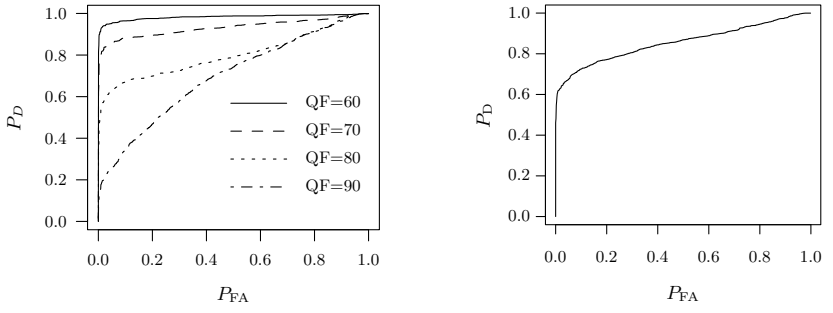
3.2 Targeted Detector Based on Zeros in High Frequency AC Coefficients

Throughout this paper, we define a *high frequency subband* as subband that lies below the anti-diagonal of the 8×8 matrix of DCT subbands (cf. Fig. 2). Under this definition, there are 28 high frequency subbands.

All high frequency subbands where $\hat{\lambda}_{\mathrm{ML}(i,j)}$ is undefined exhibit a high rate of zero coefficients—even though the perturbations to each DCT coefficient caused by mapping all decompressed pixel values to the set $\{0, \ldots, 255\}$ will result in a plausible DCT coefficient distribution. Based on this observation, a targeted detector with different threshold T can be constructed as follows. If the rate R of zero coefficients in high frequency subbands is less or equal to T ($R \leq T$), then the detector classifies a given input image as unsuspicious. Otherwise, it is flagged as image after compression and counter-forensics.

All experiments reported in this paper draw on a set of 1600 test images captured with a single Minolta DiMAGE A1 camera. The images were stored in raw format and extracted as 12-bit grayscale bitmaps to avoid inference of color filter interpolation or noise reduction. One test image sized 640×480 pixels has been cropped from each raw image at a random location to avoid resizing artifacts. The test images were made available to us by Ker [13]. All JPEG compression and decompression operations were carried out by us using `libjpeg` version 6b with its default DCT method on an Intel Mac OS 10.6 platform.

The 1600 never-compressed images were randomly and equally divided into two groups. The 800 images in the first group were first compressed with different quality factor (QF= 60, 70, 80, 90), then subject to counter-forensics of JPEG compression using Stamm et al.'s method. Finally the original uncompressed images in the first group and the images after counter-forensics were tested together. Let P_{FA} and P_{MD} ($P_{\mathrm{D}} = 1 - P_{\mathrm{MD}}$) denote the probabilities of false



(a) ROC for different QF (b) The average ROC

Fig. 3. The ROC of the detector based on the rate of zeros in high frequency subbands

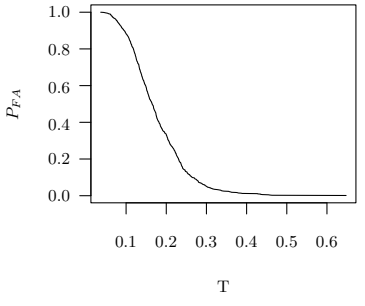


Fig. 4. Empirical P_{FA} as a function of threshold T

alarm and missed detection, respectively. Fig. 3 shows the receiver operation characteristics (ROC) curve for this first kind detector. Fig. 4 shows the empirical P_{FA} of detectors with different threshold T, and Fig. 5 shows P_{MD} as a function of threshold T for different QF.

The figures suggest that to minimize the overall error (P_{E}, which is calculated by dividing the number of the tested images by the number of misclassifid images), T should be chosen around $T \approx 0.2$. The 800 images in the second group were used to test the targeted detectors with $T = \{0.15, 0.20, 0.25\}$. Table 2 reports the results.

3.3 Targeted Detector Based on the Magnitude of High Frequency AC Coefficients

Another feature of the frequency subbands whose $\hat{\lambda}_{\mathrm{ML}(i,j)}$ are undefined is that the magnitude of the their AC coefficients is very small. Experiments on the

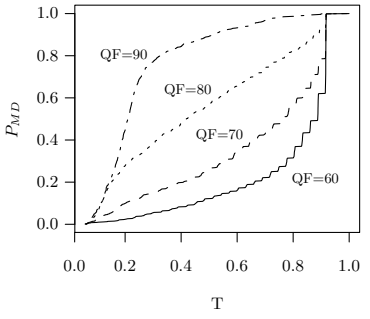


Fig. 5. Empirical P_{MD} as a function of threshold T

Table 2. P_{FA}, P_{MD} and P_{E} for the detectors with $T = \{0.15, 0.20, 0.25\}$: (%)

T	P_{E}	P_{FA}	P_{MD}				
			QF=60	QF=70	QF=80	QF=90	Average
0.15	20.6	54.6	1.4	5.4	20.0	21.6	12.1
0.20	21.8	24.7	2.3	9.9	27.5	44.5	21.0
0.25	25.0	7.6	3.8	12.1	32.8	68.8	29.3

Table 3. P_{FA}, P_{MD} and P_{E} for the second targeted detector: (%)

P_{E}	P_{FA}	P_{MD}				
		QF=60	QF=70	QF=80	QF=90	Averagel
22.9	0.3	0.8	2.5	20.3	90.4	28.5

first 800 images indicated that the magnitude is never larger than $|y_k| \leq 2$. Based on this observation, another targeted detector can be established like this: for a given image, the absolute values of the (unquantized) AC coefficients are examined. If there exists a subband where the maximum absolute value of its AC coefficients is not larger than 2, then the image is judged to be a forgery. Otherwise it can be classified as authentic. To test this detector, the second group of 800 images were used. Table 3 shows the value of the two kinds of errors for different QF. It is can be seen that the P_{MD} is very high if QF is larger than 80. This is so because with increasing QF, there are fewer subbands where $\hat{\lambda}_{\mathrm{ML}(i,j)}$ is undefined.

3.4 Combination of the Two Targeted Detectors

Judging from the above discussion, it can be seen that with QF increasing, both detectors become less sensitive to the counter-forensics operation. This is intuitive because the higher the QF the more similar are the original image and the image after counter-forensics. It is also noticeable that P_{MD} for the first detector with proper T are lower than the second one, whereas the second detector

Table 4. The performance of the detectors combining the first targeted detector with different T and the second targeted detector: (%)

T	P_E	P_{FA}	P_{MD}				
			QF=60	QF=70	QF=80	QF=90	Average
0.15	19.6	54.1	0.8	2.5	18.9	21.6	10.9
0.20	18.4	24.3	0.8	2.5	20.2	44.4	16.9
0.25	19.6	7.3	0.8	2.5	20.2	67.5	22.8

generates a very low P_{FA}. Based on this observation, a targeted detector combining the two approaches can be built like this: we first pass a given image though the second detector. If the result is negative (that is, the image is considered a forgery), then we trust the decision because P_{FA} of the second detector is very low. Conversely, if the result is positive, then the image is passed on to the first detector with threshold T. Table 4 reports the performance of a detector combining the first detector with different T and the second detector. The experiment is done on the second group of images. It can be seen that the performance of the combined detector is better than of any single detector. Our detector is good at detecting the counter-forensics operation of JPEG compression when QF is not too high, while keeping the P_{FA} reasonably low.

4 Improved Counter-Forensics of JPEG Compression

To overcome the obvious vulnerability in the high frequency subbands, which persists in Stamm et al.'s method due to undefined $\hat{\lambda}_{\text{ML}(i,j)}$, we propose a way to impute estimates for the Laplacian parameter λ if Eq. (1) has no solution.

This part is builts on prior research by Lam and Goodman [14]. They argue that the distribution of DCT coefficients $I_{(i,j)}$ is approximately Gaussian for stationary signals in the spatial domain. Hence, if the variance of all 8×8 blocks σ^2_{block} in a image was constant, then the variance $\sigma^2_{(i,j)}$ of frequency subband $I_{(i,j)}$ would be proportional to σ^2_{block}. Suppose $\sigma^2_{\text{block}} = K_{(i,j)} \cdot \sigma^2_{(i,j)}$ and let $K_{(i,j)} = k^2_{(i,j)}$ to simplify notation. Then $\sigma_{\text{block}} = k_{(i,j)} \cdot \sigma_{(i,j)}$, where $k_{(i,j)}$ is a scaling parameter. Now adding one important characteristic of natural images, namely that the variance of the blocks varies between blocks, leads us to Lam and Goodman's double-stochastic model [14]. Let $p(.)$ denote the probability density function, then

$$p(I_{(i,j)}) = \int_0^\infty p(I_{(i,j)} \mid \sigma^2_{\text{block}}) p(\sigma^2_{\text{block}}) d(\sigma^2_{\text{block}}). \tag{3}$$

As discussed above, $p(I_{(i,j)} \mid \sigma^2_{\text{block}})$ is approximately zero-mean Gaussian, i. e.,

$$p(I_{(i,j)} \mid \sigma^2_{\text{block}}) = \frac{1}{\sqrt{2\pi}\sigma_{(i,j)}} \exp\left(-\frac{I^2_{(i,j)}}{2\sigma^2_{(i,j)}}\right). \tag{4}$$

Table 5. The performance of targeted detectors with different T for improved method: (%)

T	P_E	P_{FA}	P_{MD}				
			QF=60	QF=70	QF=80	QF=90	Average
0.15	28.7	54.1	24.0	22.1	22.0	21.0	22.3
0.20	43.1	24.3	50.1	48.2	50.0	42.3	47.8
0.25	56.5	7.3	68.8	70.1	68.7	67.5	68.8

Following the argument in [14] further, we can approximate the distribution function of the variance $p(\sigma^2_{\text{block}})$ by exponential distributions and half-Gaussian distributions. Adapted to our notation we obtain for the exponential distribution:

$$p(x) = \lambda \exp(-\lambda x); \tag{5}$$

and for the half-Gaussian distribution:

$$p(x) = \begin{cases} \frac{2\lambda}{\sqrt{2\pi}} \exp\left\{-\frac{x^2\lambda^2}{2}\right\} & \text{for } x \geq 0 \\ 0 & \text{for } x < 0. \end{cases} \tag{6}$$

Plugging Eqs. (5) or (6) into Eq. (4), respectively, yields the same Laplacian shape for the AC coefficient distribution in natural images (proofs in [14]),

$$p(I_{(i,j)}) = \frac{k_{(i,j)}\sqrt{2\lambda}}{2} \exp\left(-k_{(i,j)}\sqrt{2\lambda}|I_{(i,j)}|\right). \tag{7}$$

So the relationship of the Laplacian parameters $\lambda_{(i,j)}$ for frequency subbands $I_{(i,j)}$ are only determined by $k_{(i,j)}$. For any two AC DCT subbands $I_{(i,j)}$ and $I_{(m,n)}$ it holds that

$$\frac{\lambda_{(i,j)}}{\lambda_{(m,n)}} = \frac{k_{(i,j)}\sqrt{2\lambda}}{k_{(m,n)}\sqrt{2\lambda}} = \frac{k_{(i,j)}}{k_{(m,n)}}. \tag{8}$$

To learn the scaling parameters $k_{(i,j)}$ from our images, we first select blocks with different variance randomly from our image database while keeping the number of blocks for each variance bracket approximately constant. We then transform the selected blocks to the DCT domain and calculate the variance $\sigma_{(i,j)}$. With σ_{block} and $\sigma_{(i,j)}$ known for a balanced sample, $\hat{k}_{(i,j)}$ can be estimated with the least squares method. Once $\hat{k}_{(i,j)}$ is determined, $\hat{\lambda}_{(i,j)}$ can be imputed using Eq. (8) in cases where $\hat{\lambda}_{\text{ML}(i,j)}$ cannot be calculated from Eq. (1). This procedure avoids the obvious vulnerability in the high frequency subbands that made the original DCT histogram smoothing method of [1] detectable with our first kind detector. All other parts of the algorithm remain the same in our improved variant of Stamm et al.'s method.

As the obvious singularity in the high frequency subbands no longer exist with the improved method, our targeted detector of Sect. 3.4 fails to detect it or produces unacceptably many false positives (see Table 5).

5 Second Kind Detector Using Calibration

The final contribution of this paper is a more powerful detector. This time we seek inspiration from steganalysis of JPEG images, since both forensics and steganalysis strive for closely related goals [15,16].

The notion of calibration was coined by Fridrich et al. in 2002 in conjunction with their attack against the steganographic algorithm F5 [4]. The idea is to desynchronize the block structure of JPEG images, e. g., by slightly cropping it in the spatial domain. This way one obtains a reference image which, after transformation to the DCT domain, shares macroscopic characteristics with the original unprocessed JPEG image. Kodovsky and Fridrich [6] give a detailed analysis of the nature of calibration in steganalysis. While pointing out some fallacies of calibration, they also studied in depth how, why, and when calibration works. Based on their research and the fact that calibration has been successfully used to construct many steganalysis schemes [4,5,17], we decided to investigate the usefulness of calibrated features to set up a detector for JPEG compression counter-forensics.

In this paper, a single calibrated feature, the ratio of the variance of high frequency subbands, is utilized to establish a detector. For a given image I, first, in spatial domain, we crop it by 4 pixels in both horizontal and vertical direction (similar to [5]) to obtain a new image J. Then we calculate the variance of 28 high frequency subbands for both images I and J. The calibrated feature F is calculated as follows:

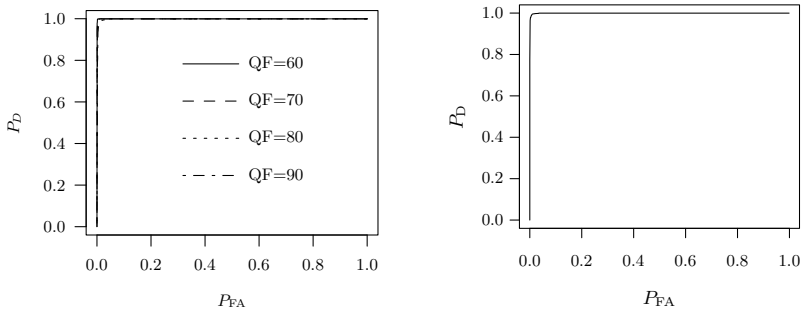
$$F = \frac{1}{28} \sum_{i=1}^{28} \left(\frac{v_{I,i} - v_{J,i}}{v_{I,i}} \right), \tag{9}$$

where $v_{I,i}$ is the variance of i-th high frequency subband in I and $v_{J,i}$ is the variance of i-th high frequency subband in J.

According to Lam and Goodman [14], the variance of frequency subbands is determined by the variance of pixels in the spatial domain (see also Sect. 4). In digital images of natural scenes, the variances of pixels in spatial proximity are often very similar. Hence, cropping the image can be treated as a transposition of the spatial pixels. So, for a authentic image, the changes of the variance of frequency subbands should be negligible. However, for images that have undergone counter-forensics, even though DCT coefficients have been smoothed to match the marginal distribution of authentic images, this variance feature is violated after cropping. The change of the variance of frequency subbands is notable, especially in high frequency subbands.

Our second kind detector with threshold G works as follows. If $F \leq G$, then the detector decides for a normal image. Otherwise the detector flags an image as after compression and counter-forensics.

To test our new detector, the first 800 images are first compressed with different quality factor (QF= 60, 70, 80, 90), then processed with counter-forensics of JPEG compression using our improved method of Sect. 4. Finally the original uncompressed images, together with images after counter-forensics are used for



(a) ROC for different QF (b) The average ROC

Fig. 6. The ROC of detectors with different G

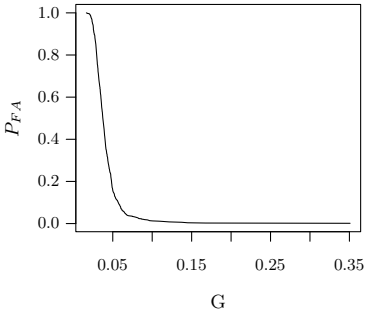


Fig. 7. The P_{FA} of detectors with different G

testing. Figure 6 shows the ROC curve for the new detector. Figure 7 shows the P_{FA} of detectors with different G and Fig. 8 reports the information of P_{MD} for different QF of detectors with different G.

It can be seen from the ROC curve (Fig. 6) that our new detector is pretty good at detecting our improved method of JPEG compression counter-forensics. The two kinds of errors P_{FA} and P_{MD} can be kept very low simultaneously. Table 6 shows the result using the second group of 800 images with $G = \{0.10, 0.15, 0.20\}$. Observe that both P_{FA} and P_{MD} are very low for our choice of G. Even for QF=95, with $G = 0.1$, P_{MD} is below 0.2 %, which is a satisfactory value.

We also tested our new detector against the original histogram smoothing method by Stamm et al. [1]. Table 7 reports the results, which are also very satisfactory. We have no explanation, though, why the calibrated feature produces slightly higher P_{MD} for the original methods compared to the improved method which uses imputed values if $\hat{\lambda}_{\mathrm{ML}(i,j)}$ is undefined.

Note that all results refer to detection rates of counter-forensics. Our new methods do not aim to detect JPEG compression as such, but attempts to

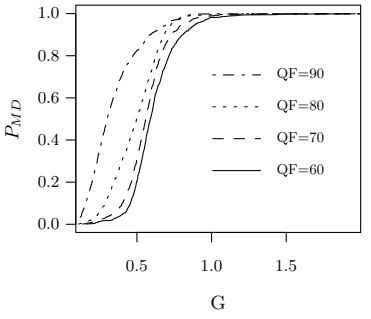


Fig. 8. The P_{MD} of detectors with different G

Table 6. P_{FA}, P_{MD} and P_{E} for the detector with $G = \{0.10, 0.15., 0.20\}$: (%)

G	P_{E}	P_{FA}	P_{MD}				
			QF=60	QF=70	QF=80	QF=90	Average
0.10	0.3	1.3	0.0	0.0	0.0	0.3	0.1
0.15	0.8	0.4	0.0	0.0	0.1	3.8	1.0
0.20	2.9	0.3	0.0	0.1	1.5	12.4	3.5

Table 7. Performance of the detectors with $G = \{0.10, 0.15, 0.20\}$ for detecting the method by Stamm et al. [1]: (%)

G	P_{E}	P_{FA}	P_{MD}				
			QF=60	QF=70	QF=80	QF=90	Average
0.10	0.5	1.3	0.0	0.0	0.0	1.3	0.3
0.15	1.7	0.4	0.0	0.0	0.8	7.5	2.0
0.20	4.2	0.3	0.0	0.3	1.5	18.9	5.2

conceal it. Therefore, in practice, our detectors should be used in combination with existing methods to detect plain JPEG compression, e. g., [7].

6 Discussion and Conclusion

The contribution of this paper is to document several incremental advances in the area of forensics and counter-forensics of JPEG compression. We decided to cut the dialectical iterations at a stage where a sufficiently powerful detector of counter-forensics could be found. This second kind detector uses a single calibrated feature, which was inspired from the literature on JPEG steganalysis.

According to the performance measures of this second kind detector and the preceding analysis, it is reasonable to state that our second kind detector is effective in detecting counter-forensics of JPEG compression, which modifies DCT coefficients independently to smooth the marginal distributions of AC subbands.

In reaction to this detector, future research on counter-forensic of JPEG compression needs to find way to keep both our calibrated feature F and the distribution of frequency subbands similar to authentic (i. e., never-compressed) images. This is a similar problem as faced by steganographic embedding functions for JPEG images. As for JPEG compression forensics, it is likely that more reliable and distinguishable calibrated features can be found so that detectors keep on a level playing field with advances in counter-forensics. Like in almost any security field, the cat-and-mouse game between attackers and defenders will continue.

Another open research question is a systematic comparison of the detection performance of our method and the more recent approach by Valenzise et al. [12].

As for every digital forensics paper, we deem it appropriate to close with the usual limitation that digital forensics, despite its mathematical formalisms, remains an inexact science. Hence the results of its methods should rather be understood as indications (subject to underlying technical assumptions and social conventions), never as outright proofs of facts.

Acknowledgements. The first author gratefully receives a CSC grant for PhD studies in Germany and an IH student travel grant supporting her trip to Prague.

References

1. Stamm, M., Tjoa, S., Lin, W., Ray Liu, K.J.R.: Anti-forensics of JPEG compression. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2010), pp. 1694–1697. IEEE Press, Los Alamitos (2010)
2. Kirchner, M., Böhme, R.: Tamper hiding: Defeating image forensics. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 326–341. Springer, Heidelberg (2008)
3. Goljan, M., Fridrich, J., Chen, M.: Sensor noise camera identification: Countering counter-forensics. In: Memon, N.D., Dittmann, J., Alattar, A.M., Delp, E.J. (eds.) Proceedings of SPIE Media Forensics and Security II, vol. 7541, p. 75410S. SPIE, Bellingham (2010)
4. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG images: Breaking the F5 algorithm. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
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. Kodovsky, J., Fridrich, J.: Calibration revisited. In: Proceedings of the 11th ACM Workshop on Multimedia and Security, pp. 63–73. ACM Press, New York (2009)
7. Fan, Z., Queiroz, R.: Identication of bitmap compression history: JPEG detection and quantizer estimation. IEEE Transactions on Image Processing 12(2), 230–235 (2003)
8. Farid, H.: Digital image ballistics from JPEG quantization. Tech.Rep. TR2006-583 (2006)
9. He, J., Lin, Z., Wang, L., Tang, X.: Detecting doctored JPEG images via DCT coefficient analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 423–435. Springer, Heidelberg (2006)

10. Pevny, T., Fridrich, J.: Detection of double-compression in JPEG images for applications in steganography. IEEE Transactions on Information Forensics and Security 3, 247–258 (2008)
11. Stamm, M., Tjoa, S., Lin, W., Ray Liu, K.J.R.: Undetectable image tampering through JPEG compression anti-forensics. In: IEEE Int. Conf. on Image Processing, ICIP (2010)
12. Valenzise, G., Nobile, V., Tagliasacchi, M., Tubaro, S.: Countering JPEG anti-forensics. In: IEEE Int. Conf. on Image Processing, ICIP (to appear, 2011)
13. Ker, A.D., Böhme, R.: Revisiting weighted stego-image steganalysis. In: Delp, E.J., Wong, P.W., Dittmann, J., Memon, N.D. (eds.) Proc. of SPIE Security, Forensics, Steganography and Watermarking of Multimedia Contents X, San Jose, CA, vol. 6819 (2008)
14. Lam, E.Y., Goodman, J.: A mathematical analysis of the DCT coefficient distributions for images. IEEE Transactions on Image Processing 9, 1661–1665 (2000)
15. Kirchner, M., Böhme, R.: Hiding traces of resampling in digital images. IEEE Transactions on Information Forensics and Security 3, 582–592 (2008)
16. Barni, M., Cancelli, G., Esposito, A.: Forensics aided steganalysis of heterogeneous images. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2010), pp. 1690–1693. IEEE Press, Los Alamitos (2010)
17. Pevny, T., Fridrich, J.: Merging Markov and DCT features for multi-class JPEG steganalysis. In: Proceeding SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents (IX), pp. 313–314 (January 2007)

Stegobot: A Covert Social Network Botnet

Shishir Nagaraja[1], Amir Houmansadr[2], Pratch Piyawongwisal[2], Vijit Singh[1], Pragya Agarwal[1], and Nikita Borisov[2]

[1] Indraprastha Institute of Information Technology, New Delhi, India
{nagaraja,vijit,pragya}@iiitd.ac.in
[2] University of Illinois at Urbana-Champaign, Urbana, IL, USA
{ahouman2,piyawon1,nikita}@illinois.edu

Abstract. We propose Stegobot, a new generation botnet that communicates over probabilistically unobservable communication channels. It is designed to spread via social malware attacks and steal information from its victims. Unlike conventional botnets, Stegobot traffic does not introduce new communication endpoints between bots. Instead, it is based on a model of covert communication over a social-network overlay – bot to botmaster communication takes place along the edges of a social network. Further, bots use image steganography to hide the presence of communication within image sharing behavior of user interaction. We show that it is possible to design such a botnet even with a less than optimal routing mechanism such as restricted flooding. We analyzed a real-world dataset of image sharing between members of an online social network. Analysis of Stegobot's network throughput indicates that stealthy as it is, it is also functionally powerful – capable of channeling fair quantities of sensitive data from its victims to the botmaster at tens of megabytes every month.

1 Introduction

Malware is an extremely serious threat to modern networks. In recent years, a new form of general-purpose malware known as *bots* has arisen. Bots are unique in that they collectively maintain communication structures across nodes to resiliently distribute commands and data through a *command and control* (C&C) channel. The ability to coordinate and upload new commands to bots gives the botnet owner vast power when performing criminal activities, including the ability to orchestrate surveillance attacks, perform DDoS extortion, sending spam for pay, and phishing.

The evolution of botnets has primarily been driven by botnet responses based on the principle of 'whatever-works'. Early botnets followed a centralized architecture however the growing size of botnets led to scalability problems. Additionally, the development of mechanisms that detect centralized command-and-control servers further accelerated their demise [6,11,9]. This led to the development of a second generation of decentralized botnets. Examples are Storm and Conficker [25,19,20] that are significantly more scalable and robust to churn.

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 299–313, 2011.

We believe that one of the main design challenges for future botnets will be covertness — the ability to evade discovery will be crucial to a botnet's survival as organizations step up defense efforts. While there are several covertness considerations involved, one of the most important ones is hiding communication traces. This is the topic of the present paper. We hope to initiate a study in the direction of defenses against covert botnets by designing one in the first place.

We discuss the design of a decentralized botnet based on a model of covert communication where the nodes of the network only communicate along the edges of a social network. This is made possible by recent advances in malware strategies. Social malware refers to the class of malware that propagate through the social network of its victims by hijacking social trust. Instances include targeted surveillance attacks on the Tibetan Movement [15] and the non-targeted attack by the Koobface [4] worm on a number of online social networks including Facebook [1].

By adopting such a communication model, a malicious network such as a botnet can make its traffic significantly more difficult to be differentiated from legitimate traffic solely on the basis of communication end-points. Additionally, to frustrate defense efforts based on traffic flow classification, we explore the use of covert channels based on information hiding techniques. What if criminals used steganographic information hiding techniques that exploit human social habits in designing botnets? Would it be possible to design such a botnet? Would it be weaker or stronger than existing botnets? These are some of the questions we hope to answer in this paper.

The rest of this paper is organized as follows: in Section 2 we describe our threat model along with an overview on JPEG steganography primitives, which is essential in the design of the social botnet introduced in this paper, *Stegobot*. In Section 3 we describe the design and construction of various components. We evaluate the use of of image steganography in designing the command and control channel of Stegobot using a real world dataset in Section 4.1; and the routing mechanism in Section 4.2. This is followed by related work in Section 5 and conclusions in Section 6.

2 Preliminaries

2.1 Threat Model

We assume the threat model of a global passive adversary. Since a botnet is a distributed network of compromised machines acting cooperatively, it is fair to assume that the defenders will also cooperate (ISPs and enterprises) and hence have a global view of communication traffic (strong adversary).

We also assume that botnet infections are not detected. As with any botnet Stegobot cannot withstand hundred-percent clean up of all infected machines. However we expect it to easily withstand random losses of a considerable numbers of bots. This assumption is due to the fact that online social networks are often scale-free graphs. In a seminal paper [5], Albert and Baraba'si showed that

scale-free graphs are highly robust to the removal of randomly selected nodes. Indeed the real world social graph considered in this paper (see dataset description in section 4.2) has a power-law degree-distribution with a slope of $\gamma = 2.3$.

2.2 JPEG Steganography

A primary goal of this paper is to show that a botnet based on covert channels can be constructed with a simple design and successfully operated. We use JPEG steganography to construct communication channels between the bots. We now review the main results in JPEG steganography that are of relevance to this paper. A full discussion on the relative merits and demerits of various design choices is defered until section 5.

We considered the JSteg scheme [3,21] but the resulting steganographic capacity of the communication is rather low; steganographic images are detectable [13] even at low embedding rates of 0.05 bits per non-zero non-one coefficients. A better scheme is proposed by Fridrich et al. [8] who showed that the average steganographic capacity of grayscale JPEG images with quality factor of 70 can be approximated to be 0.05 bits per non-zero AC DCT coefficient. The most recent scheme based on the same principle (of minimal distortion embedding) as the Fridrich scheme is the YASS [23] scheme, which has been shown undetectable at payloads of 0.05 bits per non-zero DCT coefficient.

3 Stegobot Construction

A botnet is a distributed network of a number of infected computers. It is owned by a human controller called the **botherder** and consists of three essential components: the botmaster(s), the bots, and the Command and Control (C&C) channel. **Bots** are compromised machines running a piece of software that implement commands received from one or more **botmasters**; they also send **botcargo** – information acquired by the bot such as the result of executing botherder commands – to the botmaster. Botmasters refer to compromised machines that the botherder interacts with in order to send commands via a **C&C** channel. The botmaster sends instructions to the bots to carry out tasks and receives botcargo sent back to it by the bots.

3.1 Design Goals

A distinguishing feature of Stegobot is the design of the communication channel between the bots and the botmaster. Stegobot is designed for *stealth*, therefore we do not wish to include *any* C&C communication links between computers that do not already communicate.

A further goal is to design *probabilistically unobservable* communication channels connecting the botmaster and the bots. If the C&C communication is unobservable then botnet detection can be significantly more difficult than where communication is not hidden. This is because in the latter case, traffic-flow signatures and the changes in the structure of traffic connectivity induced by the presence of the botnet can lead to easier detection and removal of the botnet [10,16].

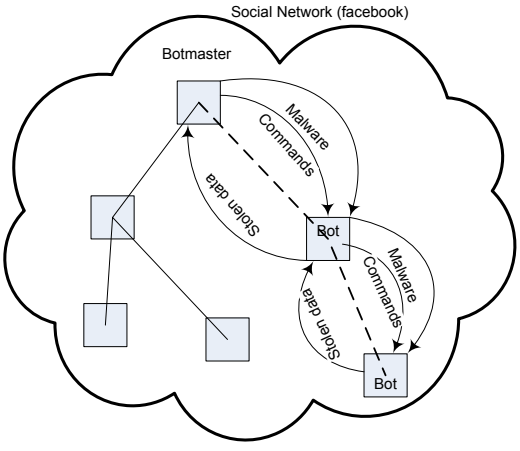


Fig. 1. The topology of the Stegobot botnet

3.2 Malware Propagation and Bots

The first step in botnet creation is malware deployment. The malware is an executable which infects the machine and performs the activities necessary of a bot. Stegobot is designed to infect users connected to each other via *social links* such as an email communication network or an online social network that allows friends to exchange emails. The propagation of malware binaries takes place via social-malware attacks [15].

Social-malware attacks refer to the use of carefully written email lures to deliver botnet malware combined with the use of email communication networks to propagate malware. This works when the attackers take the trouble to write emails that appear to come from the co-workers or friends of the victim (social phish). A more effective attack is to replay a stolen email containing an attachment that was genuinely composed by a friend back to the victim after embedding a malicious payload within the attachment.

Once the attacker secures an initial foothold (deploy the malware on at least one victim's machine), the attacker can expand the list of compromised machines with little additional effort. Whenever one of the initial set of victims sends an email containing an attachment to one of their colleagues, the bot quickly embeds a malicious payload to the attachment. Upon opening the attachment, the receiver's computer also gets infected and the process continues. Therefore once a single user is compromised (and the compromised machine continues to be operated for sending emails), the attacker can recruit additional bots in an automated fashion. Indeed this was the modus operandi behind the Ghostnet surveillance attacks on both Google and the Tibetan administration in 2009 [15].

Of course the attacker's attempts at composing email lures can fail with non-zero probability. However this exercise needs to succeed only once (as explained in the previous paragraph) to generate a botnet containing thousands of nodes, and the risk of failure is offset by targeting multiple people within a social group.

3.3 Bots

In Stegobot, bots possess a pre-programmed list of activities such as harvesting email addresses and passwords, or credit card numbers or simply to log all keystrokes. Alternatively, in a more flexible design the bots execute commands received from the botmaster. For instance, bots receive search keywords from the botmaster and respond with matches from the filesystem, as in the case of the Tibetan attacks [15].

As explained in the previous paragraph, Stegobot spreads along the social network of its victims. While we have used emails to explain social-malware attacks, the attacks are by no means restricted to email communication networks alone; online social networks are equally good targets. For instance, Koobface [4] is a worm that propagates on Facebook over social links, demonstrating that migrating from conventional email to social network messaging does not insulate users from social malware attacks. Further, it is noteworthy that Facebook is adding email extensions to its existing service; and Google added a social networking service — Google Buzz — to its popular email service in 2010. This allows bots to communicate with each other and the botmaster over the social network as explained in the next section.

3.4 Message Types

Stegobot uses two types of message constructions. First, **Bot-commands** are broadcast messages from the botmaster. Examples include search strings for file contents or within keylogged data.

Second, *botcargo* messages return information requested by the botmaster such as files matching search strings. Botcargo messages can be divided further into two types: locally generated (*botcargo-local*) or forwarded messages (*botcargo-fwd*) on a multi-hop route to the botmaster.

3.5 Communication Channel

In Stegobot, we use the images shared by the social network users as a media for building up the C&C channel. Specifically, we use image steganography techniques to set up a communication channel within the social network, and use it as the botnet's C&C channel.

A bot running on a computer can communicate with a bot running on a different computer if both the computers are being used by people connected by an edge in the social network. The social network acts as a peer-to-peer overlay over which the information is transferred from each bot to the botmaster. In Stegobot, information between bots must only be transferred using steganographic channels. In our case, this channel is constructed by hiding the botcargo within images using standard techniques reviewed in earlier sections. By keeping the size of the botcargo within certain limits, it is possible to make the presence of bot communication difficult to discover by examining the communication channel alone (section 4.1).

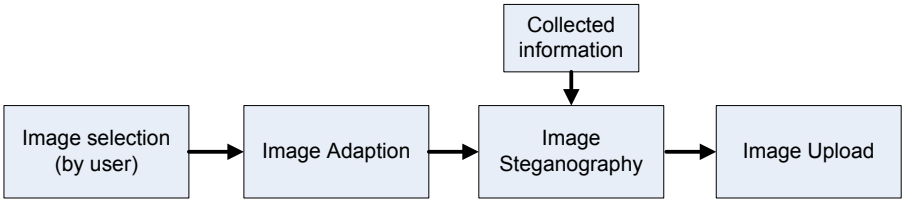


Fig. 2. Process of sending a one-hop message

One-hop communication takes place according to a *push-pull* model. Consider the example of Facebook (see figure 2). When a user *pushes* (uploads) an image to Facebook from an infected host, the bot intercepts the image and inserts the botcargo into the image using an image steganography technique as previously discussed. In our prototype this was done by uploading botcargo into all pictures on the victim's computer; a more practical approach might be to concentrate on a subset of directories where the user stores pictures. Upon completion of image upload, all the neighbors of the user are notified (by Facebook). When a neighbor of the publisher logs into Facebook from an infected machine and views the picture, the bot *pulls* (intercepts) the image and extracts the steganographically embedded botcargo from the image. All botcargo is finally destined for the botmaster who downloads the cargo by viewing newly posted pictures from her neighbors. When the botmaster intends to issue a command, she does so by preparing a botcargo message and uploading it to her Facebook account. It is worth noting that Facebook presently downloads all the images on to your computer automatically when a Facebook page is visited; the embedded images don't need to be clicked on by the victim for botcargo transfer.

While the communication channel used in our design and experiments is based on Facebook, any social communication mechanism involving rich content can be utilized in its place. In theory, blocking access to Online Social Networks (OSNs) will stop Stegobot. In practice, efforts to limit access is not easy since the use of OSNs for furthering business goals is on the increase. Additionally, such measures are easily circumvented by determined users leveraging open anonymizing proxies.

Multi-hop communication: In Stegobot, routing is based on a very simple algorithm namely **restricted flooding**.

Congestion control: Each bot maintains a bandwidth *throttle* which is used to control the ratio of *botcargo-local* to *botcargo-fwd* messages.

Metrics: We measure the effectiveness of the routing strategy using a set of metrics.

- *Channel efficiency* the percentage of *botcargo-fwd* messages that reach the botmaster averaged over all bots.
- *Channel bandwidth* is similar to efficiency, but it is the absolute number of *botcargo-fwd* messages that reach the botmaster averaged over all bots.

– *Duplication count* is the number of duplicate *botcargo-fwd* messages received by the botmaster.
– *Botnet bandwidth* is the total number of *botcargo-fwd* reaching the botmaster every month excluding duplicates.

4 Experiments

In order to convince ourselves that a Stegobot deployment could indeed be profitably operated in a real world setting, we performed a number of experiments which are detailed below.

4.1 Steganography Experiments

We use YASS [23] as the image steganography scheme of the C&C channel over the Facebook social network. Facebook's image processing can interfere with the bots' steganographic communication channel. In order to minimize this, the bot performs an image adaption process as follows before embedding a payload: 1) each image is converted to the JPEG format, 2) images are resized to meet the maximum resolution limits performed by Facebook, i.e., 720×720[1]. This is performed keeping the aspect ratio of the images.

We use a database of 116 different images to perform our experiments. In each experiment an image is adapted to Facebook constraints, as mentioned before, and then the hidden information is embedded into that image using YASS scheme. The stego image is then uploaded into Facebook through a Facebook user account, and then downloaded from the Facebook using another Facebook account. Finally, the downloaded image is evaluated by the YASS detector described in [23] in order to extract the hidden message. To evaluate the robustness of our steganographic process we calculate the bit error rate (BER) metric which is defined as the ratio of error message bits to the total number of message bits for each image.

Table 1 summarizes the average of the BER parameter (over all of the images) for different metrics of YASS scheme. Q is the quality factor of YASS scheme and represents the amount of compression performed by YASS during the steganography process. Q has a range of $[0, 100]$ and directly impacts the quality of the stego image, i.e., higher Q results in images with higher quality/size. Based on the results of our experiments, Facebook's uploading process is equivalent to the application JPEG compression over the image with a quality factor of Q_f. For $Q > Q_f$ Facebook applies extra compression on the image which results in loosing some of hidden information bits. On the other hand decreasing Q results in lower number of bits being inserted by the YASS scheme. So, there should be an optimum value for Q within the range of $[0, 100]$ which minimizes the BER rate, i.e., maximizes the robustness to Facebook perturbations. As table 1 shows

[1] More recently, Facebook is allowing uploading of higher-resolution images that increase the steganographic capacity at least 10 times based on our preliminary experiments.

Table 1. Average BER (over 116 images) without removing 'bad images'

q	2	4	6	8	10	12	14	16	18	20
Q=65	0.3073	0.1320	0.0520	0.0227	0.0097	0.0047	0.0022	0.0010	0.0006	0.0003
Q=70	0.2966	0.1318	0.0529	0.0219	0.0096	0.0049	0.0025	0.0010	0.0005	0.0002
Q=75	0.3015	0.1557	0.0680	0.0283	0.0101	0.0067	0.0027	0.0010	0.0004	0.0000
Q=80	0.3086	0.1839	0.0846	0.0347	0.0143	0.0089	0.0034	0.0015	0.0008	0.0000
Q=85	0.3512	0.2618	0.1777	0.0854	0.0372	0.0183	0.0127	0.0053	0.0024	0.0013
Q=90	0.4287	0.3917	0.3639	0.3390	0.3146	0.2906	0.2567	0.2122	0.1591	0.1262

Table 2. Number of bits inserted in each image for different values of q

q	2	4	6	8	10	12	14	16	18	20
Data bits	40280	20140	13426	10070	8056	6173	5754	5035	4475	4028

Table 3. Average BER after removing 'bad images'

q	2	4	6	8	10	12	14	16	18	20
Q=65	0.2945	0.1088	0.0311	0.0092	0.0022	0.0002	0.0000	0.0000	0.0000	0.0000
Q=70	0.2836	0.1105	0.0340	0.0095	0.0016	0.0002	0.0000	0.0000	0.0000	0.0000
Q=75	0.2892	0.1372	0.0492	0.0136	0.0011	0.0001	0.0000	0.0000	0.0000	0.0000
Q=80	0.2977	0.1686	0.0662	0.0175	0.0020	0.0003	0.0000	0.0000	0.0000	0.0000
Q=85	0.3436	0.2512	0.1631	0.0646	0.0165	0.0029	0.0012	0.0000	0.0000	0.0000
Q=90	0.4255	0.3877	0.3589	0.3331	0.3074	0.2823	0.2464	0.1978	0.1396	0.1035

the BER values are minimized for a $Q = 75$, hence we approximate the quality factor of the Facebook compression to be $Q_f \approx 75$.

We also investigate the effect of the redundancy parameter of YASS, q, on the BER. The parameter q represents the number of times an information bit is repeated inside an image by the YASS scheme. Intuitively, we expect that larger q results in reducing the BER, since more redundant bits can help better in reconstructing a noisy message; this is confirmed through our experiments as table 1 shows. In fact, the q parameter makes a tradeoff between robustness and steganographic capacity: increasing q improves robustness by reducing BER while it also reduces the number of data bits inserted by the YASS scheme. Table 2 shows the number of bits inserted by YASS for different values of q.

Our experiments show that a small number of image, namely *bad images*, are responsible for a majority of errors in the average BER. Excluding these images in the steganography process can significantly reduce average BER. We define and use a metric, *SelfCorr*, in order to decide whether an image is 'bad' or 'good'. The *SelfCorr* metric evaluates the cross correlation of an image by a noise-filtered version of itself. We declare images with *SelfCorr*> 0.9964 as 'bad' images. Table 3 illustrates the BER results after excluding the small number of 'bad' images determined by the *SelfCorr* metric. As can be seen, the average BER is significantly improved, e.g, the average BER is 0 for $q \geq 12$.

4.2 Routing Results

Combining social-malware with steganographic channels yields a covert botnet where new bots are recruited as infections spread along the edges of the social network, while existing bots communicate using the well understood image based steganographic channels. In this section, we study the routing capabilities of such a botnet using a real-world example.

Dataset: We chose to study the Flickr[2] social network [2], an online friendship network that facilitates image sharing. We crawled the Flickr website and downloaded on a fraction of the Flickr social network. Specifically, our dataset contains 7200 nodes (people), the social network edges (online friendship relations) between them, and the number of images posted per person per month. The dataset corresponds to user activity on Flickr over a period of 40 months. The Flickr dataset will be made available on our website for the research community.

In our simulation, each bot node generates K *botcargo-local* (see section 3.4) messages per month. $K = 10$ corresponds to say ten files that the bot plans to route to the botmaster across the social overlay network. *ttl* is fixed at $log(N = 7000) \approxeq 3$ hops. Each bot reserves a minimum of 5% of node bandwidth to forward *botcargo-fwd* messages received from neighbors. Further, we assume *botcommand* messages broadcast from the botmaster at a rate of one message per month. This means that the botmaster can instruct her bots to change operation no more than once a month.

Stegobot's infection strategy is based on social malware attacks. In our experiments, we have assumed an infection rate of 50%. While this number might appear high to some readers, it is actually a conservative estimate; social-malware has been known to have infection rates approaching 90-95% in real-world attacks [15].

Botcargo preparation: Each bot gathers botcargo (both from the host as well as from its neighbors). It then encodes as much of the botcargo in a single image as allowable according to a detection threshold ℓ bits. The practically possible values for the number of bits is given in table 2 and a discussion in section 4.1.

Routing: In Stegobot, routing is carried out by restricted flooding. Each bot publishes (floods) botcargo to all neighbors (joined the botnet) within *ttl* hops in the social network. Finally, the botmaster receives botcargo through the one of its infected neighbors. We assume that the botmaster is a randomly chosen node in the network. For each of the graphs below, we averaged the results over fifty different botmaster nodes.

Figure 3 shows the efficiency of botcargo transmission for increasing amounts of *ttl* and various numbers of *botcargo-local* messages. For $K = 5$ *botcargo-local* messages, the efficiency peaks at 30% and decreases and then stabilizes for higher *ttl* values as the resulting increase in the number of *botcargo-fwd* messages begins

[2] Unfortunately, we did not have access to the Facebook topology or the upload patterns of users.

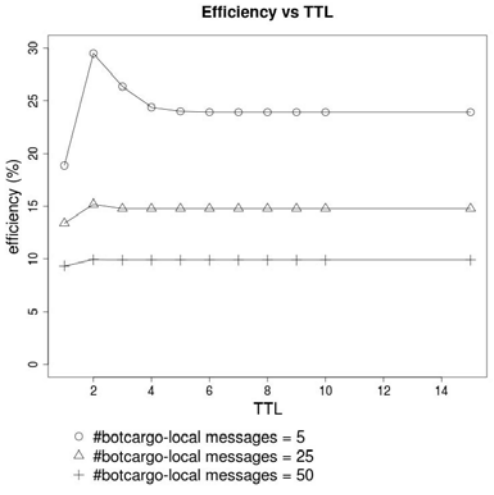
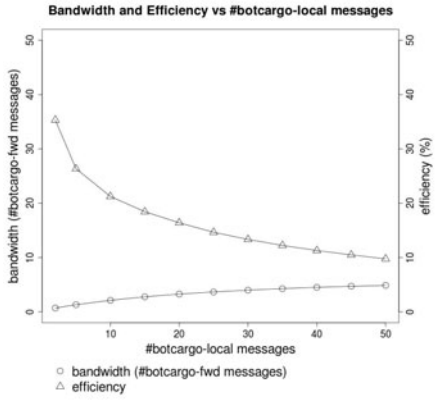


Fig. 3. Average channel efficiency against ttl

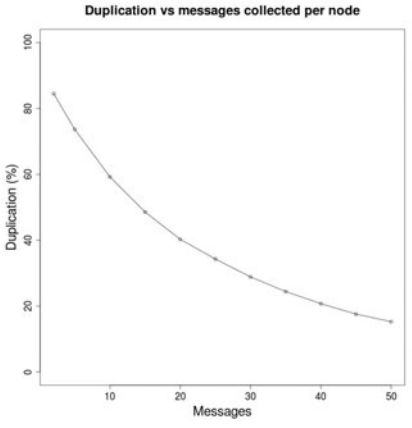
to cause congestion. Congestion effects are also felt when the number of *botcargo-local* messages increase even at a smaller *ttl*. This justifies our intuition for using $ttl = log(N)$ where N is the number of infected nodes in the botnet.

In restricted flooding, high-degree nodes in the topology play the role of hubs and are able to pull and collect large amounts of botcargo. As such they become a natural point where stolen information is collected and can then be siphoned off to the botmaster.

Channel Bandwidth and Efficiency: Figure 4 shows the bandwidth and efficiency of the communication channel in the average case. Figure 4.a shows the monthly average number of *botcargo-fwd* messages received by the botmaster (normalized by the size of the botnet) for various amounts of *botcargo-local* messages collected per bot (constant across bots). Figure 4.a also shows the average efficiency of the communication channel from a bot to the botmaster as the size of the botcargo changes. The network seems to operate at an average efficiency of 30% of collected botcargo reaching the botmaster when $K = 2$ (#botcargo per bot per month). This decreases with increase in K although the absolute number of messages delivered at the botmaster increases marginally from .75 per bot for $K = 2$ to 2.5 per bot for $K = 10$. Further increases result in even more marginal increases as the effects of congestion result in decreasing routing efficiency. A positive effect of increasing per node botcargo collection sizes (K) is the reduction in duplicate messages reaching the botmaster. This is shown in figure 4.b, the proportion of duplicate messages rapidly decreases until $K = 10$ and further reduces to 40% at $K = 20$. We observe that the positive effects of duplication reduction correspond with an increase in normalized bandwidth as the number of *botcargo-local* messages collected per node increase.



(a) Normalized Bandwidth and Efficiency

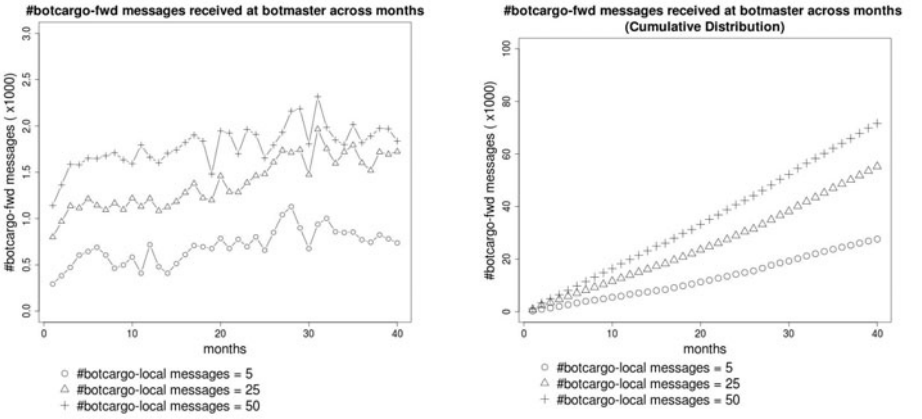


(b) Duplication

Fig. 4. Communication channel bandwidth and efficiency

The main result of our experiments is shown in figure 5. Figure 5.a shows the average number of botcargo messages delivered to the botmaster. This shows an increasing trend. This can be traced to the increasing number of users and the number of average number of photo updates per user increase over the months in our dataset. The sharp drops and increases are related to routing performance under **churn**, when a few large uploaders suddenly stop using uploading for certain periods of time, or dormant users being uploading in larger numbers (say from one-two images to twelve-fifteen images per month). Figure 5.b indicates the cumulative amount of traffic received by the botmaster over the years and gives a sense of the total amount of sensitive material she can steal and the long-term trends. Combining the total number of messages reaching the botmaster (18000 *botcargo-fwd*) with the number of bits embedded in each message, we obtain a monthly bandwidth of between 21.60MB/month in the average case ($q = 8$) to 86.13MB ($q = 2$) for lower interference from the image adaption process.

Overall, it is easy to see that even with a simple and naive routing algorithm such as restrictive flooding, the botmaster is easily able to collect around 10% of the total amount of stolen information. With a slightly more sophisticated algorithm that exploits the presence of medium and high degree hub nodes as super-peers, one could design a better routing algorithm. For instance, in the current implementation all nodes behave the same way, hence hub nodes also locally flood all the botcargo they receive. This is replayed back and forth between hubs and the rest of the network causing severe congestion. By ensuring that super-peers carefully route incoming botcargo only to other super-peers, we believe it should be possible to significantly improve network throughput.



(a) Botcargo delivered (b) Cumulative botcargo delivered

Fig. 5. Experimental results for the number of delivered batcargo

5 Related Work

Most current botnets use a peer-to-peer architecture [19,20] which improves robustness and scalability. Botnet detection techniques exploit inter-bot interaction patterns [16] or exploit the statistical characteristics of traffic flows [10,28] to localize bots. Both these approaches require access communication traffic between the bots. By using (probabilistically) unobservable communication channels, Stegobot evades all these detection approaches.

The work closest to ours is the work of Nappa et al. [17] who describe the design of a resilient botnet using the Skype protocol for inter-bot communication. The use of Skype for VoIP communication is popular and is hence difficult to block without annoying legitimate users. By hijacking active (logged in) Skype sessions, the botnet is able to bypass firewalls that might otherwise prevent bots from directly communicating with each other. Our design goes a lot further due to the unobervability properties of our communication channel. Unlike the design of Nappa et al., we do not add new connection end-points – no communication between user-accounts (bots) that do not already communicate, and no additional communication is introduced beyond what that users already exchange, resulting in a stealthy design.

5.1 JPEG Steganography

Practical steganography schemes are based either on heuristic methods or provide some provable security based on some specific model. One of the first practical steganography schemes for the JPEG images is the JSteg scheme [3,21]. It is based on using the Least Significant Bit (LSB) components of the quantized DCT coefficients. More specifically, the message bits are simply replaced with

the LSBs of the DCT coefficients of an image, considering some exclusions for preserving the image quality. The embedding path for the LSBs was originally sequential while the use of pseudo-random path was suggested in later implementations. Even with pseudo-random path the LSB steganography techniques are shown to be detectable through different kind of attacks [27,29,14,13] that exploit artifacts made in the first order statistics of the DCT coefficients.

These attacks led the next generation of the JPEG steganography schemes, namely statistical restoration-based schemes, to consider preserving statistical behavior of the cover images [24]. The main idea is to divide the cover image into two disjoint parts, which one part is used to embed the message and the other part is used to make corrections in order to preserve the selected statistical behavior of the image. A related approach for preserving the statistical behavior is used in the Model Based Steganography [22], where some specific *model* is preserved for the DCT coefficients.

As an example of the heuristic steganography schemes we can mention the F5 scheme [26], which was developed to address the detectability of the LSB-based embedding schemes. By decreasing the absolute value of the coefficients by 1 and using some other tricks the F5 scheme avoids the obvious artifacts on different features of the image. To increase the embedding efficiency F5 uses a coding scheme, namely Matrix Embedding.

Another approach for steganography, recently attracting more attention, is the minimal distortion embedding [7,12]. These schemes focus on increasing the embedding efficiency by decreasing the embedding distortion. Newman et al. in [18] propose JPEG-compatibility-steganalysis resistant method, which embeds the message into the spatial domain of the image before performing the JPEG compression. YASS [23] is a more recent scheme based on the approach of minimal distortion embedding.

6 Conclusions

The essence of communication security lies not merely in protecting content but also unobservability. In this paper, we have presented and analyzed the design of a covert botnet using unobservable communication channels that aims to steal sensitive information. The proposed botnet deploys innovative social malware infection strategies to create an overlay network over the social communication network of victims. A critical aspect of our design is the use of image based steganographic techniques to hide bot communication and make it indistinguishable from image noise. While techniques for image steganography are well known, we go one step further to show that it is possible to design an effective covert network by exploiting the social network connecting users and the social habits of individual users.

Acknowledgements. The authors would like to thank Anindya Sarkar for providing the source code for the YASS image steganography scheme.

References

1. Facebook, http://www.facebook.com
2. Flickr, http://www.flickr.com
3. JSteg, http://zooid.org/~paul/crypto/jsteg/
4. Koobface, http://en.wikipedia.org/wiki/Koobface
5. Albert, R., Jeong, H., Barabasi, A.-L.: Error and attack tolerance of complex networks. Nature 406(6794), 378–382 (2000)
6. Binkley, J.R., Singh, S.: An algorithm for anomaly-based botnet detection. In: SRUTI 2006: Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet, p. 7. USENIX Association, Berkeley (2006)
7. Fridrich, J.J., Goljan, M., Soukal, D.: Perturbed quantization steganography. Multimedia Syst. 11(2), 98–107 (2005)
8. Fridrich, J.J., Pevný, T., Kodovský, J.: Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In: Kundur, D., Prabhakaran, B., Dittmann, J., Fridrich, J.J. (eds.) Proceedings of the 9th workshop on Multimedia & Security, MM&Sec 2007, Dallas, Texas, USA, September 20-21, pp. 3–14. ACM, New York (2007)
9. Goebel, J., Holz, T.: Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In: HotBots (2007)
10. Gu, G., Perdisci, R., Zhang, J., Lee, W.: BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proceedings of the 17th USENIX Security Symposium, Security 2008 (2008)
11. Karasaridis, A., Rexroad, B., Hoeflin, D.: Wide-scale botnet detection and characterization. In: HotBots (2007)
12. Kim, Y., Duric, Z., Richards, D.: Modified Matrix Encoding Technique for Minimal Distortion Steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 314–327. Springer, Heidelberg (2007)
13. Lee, K., Westfeld, A.: Generalised category attack—improving histogram-based attack on JPEG LSB embedding. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 11–13. Springer, Heidelberg (2008)
14. Lee, K., Westfeld, A., Lee, S.: Category attack for lsb embedding of jpeg images. In: Shi, Y.Q., Jeon, B. (eds.) IWDW 2006. LNCS, vol. 4283, pp. 35–48. Springer, Heidelberg (2006)
15. Nagaraja, S., Anderson, R.: The snooping dragon: social-malware surveillance of the tibetan movement. Technical Report UCAM-CL-TR-746, University of Cambridge (March 2009)
16. Nagaraja, S., Mittal, P., Hong, C.-Y., Caesar, M., Borisov, N.: Botgrep: finding p2p bots with structured graph analysis. In: Proceedings of the 19th USENIX Conference on Security, USENIX Security 2010, p. 7. USENIX Association, Berkeley (2010)
17. Nappa, A., Fattori, A., Balduzzi, M., Dell'Amico, M., Cavallaro, L.: Take a Deep Breath: A Stealthy, Resilient and Cost-Effective Botnet Using Skype. In: Kreibich, C., Jahnke, M. (eds.) DIMVA 2010. LNCS, vol. 6201, pp. 81–100. Springer, Heidelberg (2010)
18. Newman, Moskowitz, Chang, Brahmadesam: A steganographic embedding undetectable by JPEG compatibility steganalysis. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 258–277. Springer, Heidelberg (2003)
19. Porras, P., Saidi, H., Yegneswaran, V.: A multi-perspective analysis of the Storm (Peacomm) worm. In: SRI Technical Report 10-01 (2007)

20. Porras, P., Saidi, H., Yegneswaran, V.: A foray into Conficker's logic and rendezvous points. In: 2nd Usenix Workshop on Large-Scale Exploits and Emergent Threats, LEET 2009 (2009)
21. Provos, N., Honeyman, P.: Hide and seek: An introduction to steganography. IEEE Security and Privacy 1, 32–44 (2003)
22. Sallee, P.: Model-based steganography. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) IWDW 2003. LNCS, vol. 2939, pp. 154–167. Springer, Heidelberg (2004)
23. Solanki, K., Sarkar, A., Manjunath, B.S.: YASS: Yet Another Steganographic Scheme That Resists Blind Steganalysis. In: Furon, T., Cayre, F., Doërr, G.J., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 16–31. Springer, Heidelberg (2008)
24. Solanki, K., Sullivan, K., Madhow, U., Manjunath, B., Chandrasekaran, S.: Provably secure steganography: Achieving zero k-l divergence using statistical restoration. In: ICIP (2006)
25. Stover, S., Dittrich, D., Hernandez, J., Dietrich, S.: Analysis of the Storm and Nugache trojans: P2P is here. Login 32(6) (December 2007)
26. Westfeld, A.: F5–A steganographic algorithm: High capacity despite better steganalysis. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 289–302. Springer, Heidelberg (2001)
27. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 61–75. Springer, Heidelberg (2000)
28. Yen, T.-F., Reiter, M.K.: Traffic aggregation for malware detection. In: Zamboni, D. (ed.) DIMVA 2008. LNCS, vol. 5137, pp. 207–227. Springer, Heidelberg (2008)
29. Yu, X., Wang, Y., Tan, T.: On estimation of secret message length in jsteg-like steganography. In: International Conference on Pattern Recognition, vol. 4, pp. 673–676 (2004)

CoCo: Coding-Based Covert Timing Channels for Network Flows

Amir Houmansadr and Nikita Borisov

Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign
{ahouman2,nikita}@illinois.edu

Abstract. In this paper, we propose CoCo, a novel framework for establishing covert timing channels. The CoCo covert channel modulates the covert message in the inter-packet delays of the network flows, while a coding algorithm is used to ensure the robustness of the covert message to different perturbations. The CoCo covert channel is adjustable: by adjusting certain parameters one can trade off different features of the covert channel, i.e., robustness, rate, and undetectability. By simulating the CoCo covert channel using different coding algorithms we show that CoCo improves the covert robustness as compared to the previous research, while being practically undetectable.

1 Introduction

A covert channel intends to conceal the very existence of a hidden message, *covert message*, by communicating it through a legitimate communication channel, i.e., the overt channel. Considering the computer networks, covert channels lie within two major categories: *covert storage channels*, and *covert timing channels* [5]. Covert storage channels work by modifying some unused/random bits in the packet header of a network flow [19,12,9]. Alternatively, a covert timing channel modulates the covert message into the timing pattern of network flow packets [3,21]. A covert channel needs to be *undetectable*, meaning that the covert traffic should not be distinguishable from a legitimate traffic. Murdoch et al. show that many of the storage covert channels can be detected easily since the covert message modifies the benign pattern of the utilized header fields [17]. Also, different statistical tests have been utilized to detect covert timing channels [2,7]. In addition to undetectability, a covert message needs to be *robust* to the perturbations caused by the communicating network and/or an adversary. Under given undetectability and robustness requirements a covert channel aims in maximizing its *rate*, i.e., the number of covert message bits sent per covert flow packets.

In this paper, we design CoCo as a reliable and adjustable framework for establishing covert timing channels. CoCo modulates the covert message in the inter-packet delays (IPD) of a network flow, while a coding algorithm is used to ensure the covert message robustness. A main contribution of CoCo over the previous work is that CoCo is adjustable: a user of the covert channel can tradeoff

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 314–328, 2011.

different features of the covert channel, e.g., undetectability, rate, and robustness, considering the application and network conditions. Recent research tends towards developing covert channels with *provable* undetectability [15], however, for many applications of covert channels a *practical* undetectability is sufficient. Instead of providing a provable undetectability for all applications CoCo adjusts its level of undetectability to what is needed for each specific application, resulting in significant improvements in robustness and/or the rate of the covert message. The adjustments are performed based on the channel models for noise and adversarial perturbations, and also the performance priorities of the covert channel users.

The CoCo covert channel is fast and easy to implement. To communicate a covert message, m, a *sender* S generates a network flow f for a given traffic model using a keyed IPD generator. The sender, then, *slightly* manipulates f according to c which is an encoded version of the covert message m, resulting in the covert flow, f^c. Finally, f^c is sent to a *receiver* R through a noisy channel, e.g., the Internet. The receiver extracts the covert message from a noisy version of f^c using a secret key which is only shared between the sender and the receiver. The channel noise is composed of two components: the natural noise of the overt network, e.g., network delays, and the perturbations performed by an adversary in order to degrade/destroy covert channels. CoCo is robust to both natural and adversarial perturbations of the communication channel by taking advantage of efficient coding algorithms. By adjusting the encoding algorithm and the *gain* of the covert message, as defined later, CoCo is able to balance between rate, undetectability, and robustness of the covert message. Our experiments show that the choice of the coding algorithm and also the encoding rate impacts the robustness performance of the covert message. We show through experiments that under reasonable requirements for practical undetectability CoCo improves the robustness performance significantly compared to similar previous research.

The rest of this paper is organized as follows; we mention some related work in Section 2. In Section 3, we discuss the system model and features of the covert timing channel. In Section 4, we describe a detailed description of the CoCo covert timing channel. In Section 5, we evaluate the detection performance of our covert channel using different coding algorithms. The undetectability of the CoCo covert channel is evaluated in Section 6, and the paper is concluded in Section 7.

2 Related Work

The very first covert timing channels are based on having a sender either send or remain silent in specific time intervals in order to communicate covert message bits 1 and 0, respectively [18,11]. These schemes not only are limited by the need for accurate synchronization between the sender and the receiver, but also are easily detectable using statistical tests [3]. Later research on covert timing channels leans toward inserting the covert message directly in the inter-packet delays (IPD). In [4] a covert timing channel is proposed that tries to mimic the

empirical distribution of the legitimate traffic by splitting the empirical IPDs distribution into two equally-sized *small-delay* and *large-delay* groups and sending a 0-bit (1-bit) covert message by randomly replaying an IPD from the small-delay (large-delay) set. Berk et al. also suggest to encode the messages directly into the inter-packet delays in order to maximize the covert capacity [2]. Shah et al. propose the keyboard JitterBug, a low capacity channel for leaking the typed information over an existing interactive network connection [21].

To defend against covert timing channels two different approaches have been taken in the literature. The first approach is to actively disrupt the network communication in order to destroy possibly existing covert timing channels. An example for this is the use of network *jammers* that apply random delays over the network packets [10]. Even though such disruptions can seriously devote the capacity of covert timing channels they also disrupt the expected performance of the legitimate traffic. This is more of a problem in the case of interactive traffic and real-time traffic. In addition, such disruptions can result in changing the traffic shape, hence, revealing the existence of the traffic disruptors to the covert communicating parties. Another approach in defending covert channels is to passively monitor network traffic in order to detect such channels. This is done using different statistical tests including shape tests, regularity tests and entropy tests [3,2,7].

To thwart these detection mechanisms Gianvecchio et al. devise a *model-based* covert timing channel that models the network traffic in order to generate the covert traffic [8]. The scheme, however, requires frequent communication of traffic model parameters from the sender to the receiver. Recent research has considered simple encoding of messages into the inter-packet delays in order to increase the data rate of covert communication. Sellke et al. use a simple *Geometric code* in conjunction with pseudo random generators to build a low-rate undetectable covert timing channel for i.i.d. traffic [20]. The scheme is limited by the strong assumption of the additive jitter being bounded. Liu et al. in [14] also modulate covert messages in the IPDs of the traffic by using spreading codes in order to increase the robustness of the covert channel to the network perturbations. A similar covert timing channel is proposed in [15] for i.i.d. traffic that provides *provable* undetectability for the covert message.

3 Preliminaries

3.1 System Model

A covert channel consists of a *sender (S)* sending a *covert message* to a *receiver (R)* through a steganographic channel. In this paper, we consider the design of a covert timing channel for computer networks, i.e., the steganographic communication is established through the packet timing information of a network flow, the *cover flow*. The designed covert channel is *active*, in a sense that it generates the timings of the network traffic used for embedding the covert message. CoCo is based on modulating the covert message into the inter-packet delays (IPD) of network flows. Some secret information, the *secret key*, is shared between the

sender and the receiver so that a third-party is not able to either extract the covert message, or detect the presence of the covert communication channel.

3.2 Adversary Model

We assume that the CoCo covert traffic is monitored by an adversary that tries to detect the presence of covert timing channels. This requires the covert channel to be *undetectable*, as defined later. Also, we consider *active* adversaries that perturbs network traffic in order to destroy the possibly existing covert timing channels. An effective covert timing channel should be *robust* to the *channel noise* which is combined of the adversarial perturbations and the natural noise of the network.

The adversarial perturbations are constrained in two ways; first of all, they should not interfere with the benign traffic, as such perturbations are usually performed by entities providing services to legitimate users, e.g., intrusion detection systems. Secondly, the adversarial perturbations should be concealed from covert parties: having knowledge of the active adversaries the covert parties can evade the detection by taking alternative covert mechanisms which are not affected by the known perturbations. Based on this, we assume that the adversarial perturbations can amplify the effect of the network noise in the channel noise, but they do not change the behavior of the channel noise drastically. In our simulations throughout this paper we model the channel noise as an amplified version of the network noise.

3.3 Features of the Covert Timing Channel

Undetectability. We define a covert channel to be *undetectable* by a test algorithm A with a *Cross-Over Error Rate (COER)* of C if the algorithm A is not able to distinguish between a legitimate flow and a covert flow with a COER smaller than C. A test A has a COER of C if it returns the same rates of false alarms and misses in detecting a covert traffic. This is further elaborated in Section 6.

Robustness. As mentioned above, a cover flow containing the covert message is perturbed by the channel noise which is composed of the network noise and the adversary noise. We define a covert timing channel to be *robust* to the channel noise with a bit error rate (BER) of ϵ if the receiver is able to extract the covert message with a BER of at most ϵ. BER is defined as the number of errored covert bits at the receiver divided by the total number of covert bits.

Rate. Another feature of a covert channel is the *rate* of the covert communication. We define the rate of the CoCo covert channel to be

$$r = \lim_{N \to \infty} \frac{K}{N} \tag{1}$$

where K is the number of covert message bits sent using $N + 1$ packets of a cover flow. Under given undetectability and robustness requirements, a covert channel designer aims in maximizing the rate of the covert channel. As will be discussed later, the choice of the coding algorithm of CoCo trades off the rate for the robustness of the covert channel.

4 The CoCo Scheme

In this section we describe the scheme of the CoCo covert timing channel.

4.1 The CoCo Sender Scheme

The sender of the covert channel, S, inserts the covert message into the inter-packet delays of the covert traffic. Figure 1 illustrates the scheme of the CoCo sender that consists of the following components:

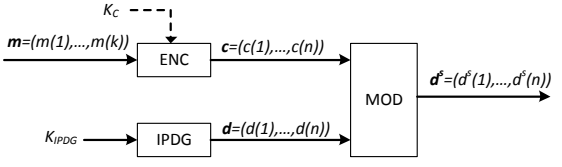


Fig. 1. The block diagram of the CoCo sender

- IPD Generator (IPDG): pseudo-randomly generates the inter-packet delays (IPD) according to a given traffic model and using a secret key.
- Message Encoder (ENC): encodes the covert message bits using a coding algorithm in order to make the covert channel robust to the channel noise.
- IPD Modulator (MOD): modulates the output of the ENC encoder into the IPDs generated by the IPDG block.

A covert flow is generated in three steps:

A. Encoding the message. An encoder ENC is called an (n, k) coding algorithm if for any block of k message bits, $\mathbf{m} = (m(1), \ldots, m(k))$, it generates a block of n encoded bits, $\mathbf{c} = (c(0), \ldots, c(n))$. Such an encoder has a *coding rate* of k/n which also determines the rate of CoCo, i.e., $r = k/n$. A CoCo sender divides a covert message into blocks of k bits and encodes each block using the ENC (n, k) coding algorithm. The ENC encoder accepts inputs in the binary format and generates output in the bipolar format, i.e., $m(i) \in \{0, 1\}$, and $c(j) \in \{-1, 1\}$ $(1 \leq i \leq k, 1 \leq j \leq n)$. Note that the ENC encoder may be protected by an encoding key K_c in order to protect the message confidentiality in case of a compromise. This is not necessary as the covert message $\mathbf{m}$ can be encrypted before encoding. In Section 5, we investigate the use of different coding algorithms as the ENC encoder of CoCo and compare the functionality of CoCo for different schemes.

B. Generating the inter-packet delays. The IPDG block generates sequences of inter-packet delays pseudo-randomly according to a given traffic model. The IPDG takes a traffic model and a secret key, K_{IPDG}, as input and generates sequences of IPDs according to the given traffic model. The secret key K_{IPDG}

is only shared between the sender S and the receiver R, that enable S and R to generate the same pseudo-randomly generated sequences of IPDs. In Section 4.3 we design an IPDG for generating i.i.d. traffic. In fact, the i.i.d. assumption is for the sake of simplicity and the IPDG can be designed in a similar manner for any type of network traffic.

C. Modulating the IPDs with the encoded message. Once the coded message $\mathbf{c}$ is generated (step A) it is embedded within the inter-packet delays generated by the IPDG (step B). In order to send a length n encoded message $\mathbf{c} = (c_1, ..., c_n)$, the sender requires a length n IPD sequence $\mathbf{d} = (d(1), ..., d(n))$. The modulated IPDs are given by:

$$d^S(i) = d(i) + a \times c(i) \qquad 1 \leq i \leq n \tag{2}$$

where $d^S(i)$ is the i^{th} modulated IPD and a is the *amplitude* of the covert channel. We define the *gain* of the covert channel, γ, as:

$$\gamma = \log_2 \frac{a}{\sigma} \tag{3}$$

where σ is the standard deviation of the channel *jitter* (channel jitter is the effect of the channel noise on the inter-packet delays). In fact, γ represents the signal-to-noise ratio of the covert communication and is used in our evaluations in the following sections. Finally, the timings of the covert flow, $\mathbf{t^S} = (t^S(1), ..., t^S(n+1))$, are evaluated as:

$$t^S(i) = \sum_{j=1}^{i-1} d^S(j) + t^S(1) \qquad (2 \leq i \leq n+1) \tag{4}$$

where $t^S(1)$ is the timing of the first packet which is selected at random from the interval $[0, 2]$ seconds.

4.2 The CoCo Receiver Scheme

After being perturbed by the channel noise, the covert traffic receiver, R, receives the packets of the covert flow at times $\mathbf{t^R} = (t^R(1), ..., t^R(n+1))$. The IPDs of the received flow, $\mathbf{d^R} = (d^R(1), ..., d^R(n))$, are evaluated as:

$$d^R(i) = t^R(i+1) - t^R(i) \qquad (1 \leq i \leq n) \tag{5}$$

The receiver intends to extract the covert message form the received IPDs $\mathbf{d^R}$ which is a noisy version of the sent IPDs $\mathbf{d^S}$. The CoCo receiver scheme is shown in Figure 2 which consists of three components:

- IPD Generator (IPDG): the same IPD generator used by the CoCo sender.
- Message Decoder (DEC): a decoder for the ENC encoded messages. A key might be shared between ENC and DEC algorithms.
- IPD Demodulator (DMOD): extracts the modulated bits from the received IPDs.

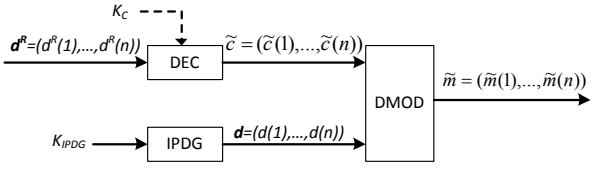


Fig. 2. The CoCo receiver block diagram

The CoCo receiver scheme works in three steps in order to extract the covert message bits from a received covert flow.

A. Re-generating the IPDs. The receiver uses the same IPD generator used by the sender in order to re-generate the IPDs used by the sender to modulate the covert message, i.e., $\mathbf{d}$. The IPDG block uses a key K_{IPDG} which is only shared between the sender and receiver and is essential for the covert channel undetectability. In Section 4.3, we describe the design of the IPDG for an i.i.d. traffic model.

B. Demodulating the encoded message. The received IPDs sequence $\mathbf{d^R}$ can be represented as:

$$d^R(i) = d^S(i) + \delta(i) \tag{6}$$
$$= d(i) + a \times c(i) + \delta(i) \tag{7}$$

As mentioned in part A, the IPD sequence $\mathbf{d}$ can be regenerated by the receiver using some shared information. So, the receiver is able to evaluate $\tilde{\mathbf{c}} = (\tilde{c}(1), \ldots, \tilde{c}(n))$ which is a noisy version of the encoded message $\mathbf{c}$:

$$\tilde{c}(i) = \frac{d^R(i) - d(i)}{a} \tag{8}$$
$$= c(i) + \delta(i)/a \tag{9}$$

C. Decoding the covert message. The final step in extracting the covert message is to decode the noisy encoded message, $\tilde{\mathbf{c}}$. The DEC algorithm is used to decode the length-n $\tilde{\mathbf{c}}$ derived in the previous step into a length-k *recovered message* $\tilde{\mathbf{m}} = (\tilde{m}(1), \ldots, \tilde{m}(k))$. The goal of the DEC algorithm is to minimize the bit-error rate (BER) which is defined as:

$$BER = \frac{\sum_{i=1}^{k} e(m(i), \tilde{m}(i))}{k} \tag{10}$$

where $e(x, y) = 1$ for $x \neq y$, and $e(x, y) = 0$ for $x = y$.

4.3 IPD Generator for an i.i.d. Traffic Model

The IPDG is responsible for generating IPDs according to a given traffic model in a pseudo-random manner. Using a secret key, K_{IPDG}, the sender and receiver are able to use IPDG to generate the same IPDs sequences, while it is cryptographically infeasible for a third party to generate the same sequences of IPDs. The IPDG can be set up to generate any kind of traffic in a random manner that allows the CoCo covert channel to be implemented for different types of traffic. For the sake of simplicity we design an IPDG for generating i.i.d. network traffic. In fact, i.i.d. traffic models are used in many analysis of network traffic and they constitute fundamental elements of many advanced network traffic models.

The IPDG uses a keyed cryptographically secure pseudo-random number generator (CSPRNG), being known to anyone including the attacker. However, the key of this SCPRNG, K_{IPDG}, is only shared between the sender S and the receiver R. Also, S and R agree on the cumulative density function (CDF) of the legitimate traffic, $g(\cdot)$. To generate the i^{th} IPD value, $d(i)$, the IPDG uses the CSPRNG along with its key K_{IPDG} to draw a number $u(i)$ *uniformly* at random from the range of $[0, 1]$. The IPD value $d(i)$ is then generated as:

$$d(i) = g^{-1}(u(i)) \tag{11}$$

where $g^{-1}(\cdot)$ is the inverse of the CDF function $g(\cdot)$. Note that since $g(\cdot)$ is a one-to-one function with output range of $[0, 1]$ its inverse function g^{-1} is also a one-to-one function with input domain of $[0, 1]$. It is easy to show that the IPD sequence $\mathbf{d}$ generated in this manner has an empirical distribution according to the CDF function $g(\cdot)$. A more elaborated approach can be taken to generate IPDs according to non-i.i.d. traffic models.

5 CoCo Performance for Different Coding Schemes

A covert timing channel can be considered as a noisy communication channel for the cover message. The use of the ENC/DEC encoding algorithms is to improve the detection performance of the covert message at the receiver. In this paper, we consider the use of different types of linear encoding schemes. The linear codes are classified into two main groups: *block codes* and *convolutional codes*. A third class of linear codes is derived by combining block codes and convolutional codes, the *Turbo codes*, which are known to approach the channel capacity. We also consider *Low-density parity check* (LDPC) codes, another class of capacity-approaching codes.

Simulation methodology. In each of the simulations a random covert message is generated with a length appropriate for the selected ENC algorithm. The random message is, then, used to generate the covert traffic using the CoCo sender scheme described in Section 4.1. In order to simulate the effect of the channel noise we randomly select samples of network jitter from a large database and apply them

to the IPDs of the covert traffic (note that as discussed in Section 3.2 we consider the channel noise to be an amplified version of the natural network noise). The jitter database is collected over the Planetlab [1] and contains 100000 packets (jitters have an average standard deviation of approximately $12msec$). Finally, a receiver R uses the receiver scheme mentioned in Section 4.2 to extract the covert message bits from the perturbed covert traffic. Note that for the sake of simulations we do not need to generate and add the IPDs to the encoded message, as the IPDs are regenerated by the receiver and are canceled out from the received noisy message before performing the decoding algorithm.

5.1 Block Codes

Reed-Solomon (RS) Codes. Reed-Solomon (RS) codes [13] are a class of linear block correcting-codes that are maximum distance separable (MSD), e.g., they meet the equality criteria of the *singleton bound* [13]. RS codes have been used in satellite communications for many years because of their strength regarding bursty errors. For an (n, k) RS code, each code symbol is m bits, where $n = 2^m - 1$ is the size of the coded message (e.g., an n-bit RS code consists of $m \times n$ binary bits).

Table 1 shows the detection performance of the CoCo scheme using RS encoders with different parameters (each simulation is run 1000 times and the gain of the covert scheme is $\gamma = 0$). Instead of BER, for evaluating RS codes we use a similar metric, *Block Error Rate* (BLER). This is because each error in an RS code affects a whole block of data. As can be seen from the table, decreasing the rate of the RS encoder improves the detection performance of the covert channel, as more redundant bits are inserted to compensate for the channel noise. Note that larger symbol size m requires more processing resources for the encoder and decoder; hence for similar BLER and rate a code with smaller m is preferred. To illustrate the effect of covert gain on the detection performance we also run the simulations for different values of γ. Table 2 shows the detection results for a $(7, 3)$ RS code. As can be seen, increasing γ significantly improves the detection performance. In Section 6, we investigate the effect of the gain on the undetectability of the CoCo covert channel.

Golay codes. Golay codes [13] are one of the few existing *perfect codes*, i.e., they meet the equality criteria of the *hamming bound* [13]. Unfortunately, there are only two instances of the Golay codes: a binary $(23, 12)$ code, and a ternary $(11, 6)$ code. We use the binary Golay code of $(23, 12)$ which is able to correct 3 errors in a block of 23 encoded bits. Considering the high running speed of Golay codes we concatenate them with simple *Hamming* codes [13] in order to improve the CoCo robustness. Tables 3 and 4 illustrate the detection performance achieved by concatenating the binary Golay code with 2-bit and 3-bit parity check codes, respectively. A 2-bit parity check reduces the rate from 12/23 to 10/23, however significantly improves the BER. As an example, for $\gamma = 0$ the the BER is reduced from 0.13 to 0.02. The results are even better using 3 bits of parity.

Table 1. RS codes for different parameters (1000 runs, $\gamma = 0$)

rate (r)	m	k	n	BLER
0.57	3	4	7	0.363
0.43	3	3	7	0.136
0.29	3	2	7	0.135
0.14	3	1	7	0.029
0.26	4	4	15	0.155
0.20	4	3	15	0.074
0.13	4	2	15	0.060
0.07	4	1	15	0.018
0.16	5	5	31	0.087

Table 2. BLER of the (7,3) RS code for different gains (1000 runs) ($r = 0.43$)

γ	BLER
-1	0.129
0	0.036
1	0.001
1.6	0.008
2	0.0002

Table 3. Binary $(23, 12)$ Golay code and 2-bits parity check ($r \approx 0.43$)

γ	Correct blocks	Detected errors	BER
-1	0.4712	0.8078	0.1016
0	0.8637	0.8437	0.0213
1	0.9827	0.8959	0.0018
1.6	0.9950	0.9400	0.0003
2	0.9989	1.0000	0.0000

Table 4. Binary $(23, 12)$ Golay code and 3-bits parity check ($r \approx 0.39$)

γ	Correct blocks	Detected errors	BER
-1	0.4712	0.9030	0.0504
0	0.8639	0.9213	0.0107
1	0.9853	0.9863	0.0002
1.6	0.9964	0.9722	0.0001
2	0.9989	1.0000	0.0000

5.2 Convolutional Codes

Convolutional codes are another class of linear error-correcting codes that have use in many different applications [13]. An (n, k) convolutional code is a device with k inputs and n outputs. The input stream of a message $\mathbf{m}$ is split into k streams entering the inputs of the encoder, and each of the n output streams is evaluated by convolving some of the input streams with a generator sequence $\mathbf{G}$. The length of the generator function is called the *constraint length* v, and $u = v - 1$ is the memory of the encoder. An easy to implement decoder for convolutional codes is an ML decoder based on the Viterbi algorithm [13].

The convolutional codes simulated in this paper use a constraint length of $v = 7$. This is a popular value which is also used in the Voyager program and also the IEEE 802.16e standard. Larger v results in more powerful codes but is only used in space missions because of the decoder's high complexity. Convolutional codes also use *puncturing* which is a method to make a k_2/n_2-rate code out of a k_1/n_1 code by deleting some of the encoded bits based on a puncturing matrix. An M/N puncture means that out of M code bits only N bits are used. Table 5 shows the BER performance of CoCo using different convolutional codes, and Table 6 shows the results for a specific code, but for different γ.

Table 5. Average BER of Convolutional code (1000 runs, each 10000 bits) for different rates ($\gamma = 0$)

Rate (r)	k/n	puncture	BER
0.67	1/3	6/3	0.2029
0.6	1/2	6/5	0.1653
0.57	1/2	8/7	0.1514
0.5	1/2	1/1	0.1098
0.5	1/3	6/4	0.1414
0.4	1/3	6/5	0.0815
0.33	1/3	1/1	0.0413
0.25	1/4	1/1	0.0351
0.2	1/5	1/1	0.0200

Table 6. Average BER of CoCo using a Convolutional code with $k/n = 1/2$ and puncturing of 1/1 (1000 runs, each 10000 bits) ($r = 0.5$)

γ	BER
-1	0.3825
0	0.1095
1	0.0076
1.6	0.0006
2	0.0001

5.3 Turbo Codes

Turbo codes are a class of high-performance error correction codes and are the first practical capacity-approaching codes [16]. A turbo code is generated by concatenating two or more *constituent* codes, where each constituent code can be a convolutional or a block code. Usually some *interleaver* reorders the data at the input of the inner encoders. Turbo codes are decoded through iterative schemes. There are two types of Turbo codes: *Block Turbo Codes* (BTC), and *Convolutional Turbo codes* (CTC). In Figure 3 we draw the BER of the CoCo covert channel using BTC and also CTC codes. The simulated BTC and CTC codes are used in the IEEE 802.16*e* standard. Due to the space constraints we leave the use of other convolutional codes for the future research.

5.4 Low Density Parity-Check Codes (LDPC)

First designed by Gallager in 1962, LDPC codes are linear error correcting codes which are considered as another class of capacity-approaching codes [16]. Compared with the turbo codes the LDPC codes outperform for high code rates while the turbo codes are better suited for lower code rates. The LDPC encoders are represented by randomly generated sparse parity-check matrices and their decoding is performed iteratively using message-passing decoders [16].

We simulate the CoCo covert channel using the LDPC codes used in the IEEE 802.16*e* standard. In particular, we use a rate 0.5 LPDC code with $k = 384$. The BER performance of this covert channel is illustrated in Figure 3; we leave the simulation of other LDPC codes for the future research due to the space constraints.

5.5 Comparisons and Tradeoffs

Figure 3 shows the average BER of the CoCo covert scheme using different coding algorithms and for different values of γ. The coding schemes are selected such that they result in a covert channel with a rate close to $r = 0.5$ (note that not all the codes can be designed for an exact rate of $r = 0.5$). As can be seen,

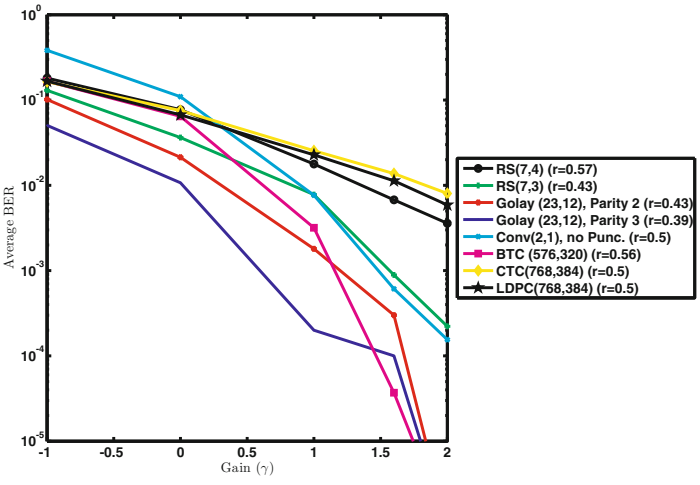


Fig. 3. The average BER of the CoCo using different coding schemes

the BER performance depends on the type of the coding scheme used by the CoCo. As an example, the CoCo covert channel using "Goaly(23,12)-Parity 2" outperforms the one using "RS(7,3)" code, even though they both have a rate of $r = 0.43$. We observe that in all of the cases, for a given rate, increasing γ reduces the BER of the CoCo. We also observe that for a given rate the choice of the coding algorithm depends on the gain γ. As an instance, the "Conv(2,1)-no Punc" code outperforms all of the 0.5-rate codes for gains larger than 1, but is outperformed by all of them for $\gamma \leq 0$.

Unlike previous covert channels that the BER performance depends on the noise power [20,14,15], the BER performance of the CoCo only depends on the signal-to noise ratio of the covert channel, i.e., γ (see equation (9)). This, unlike the other schemes, enables CoCo to provide similar performance for different noise powers by adjusting the covert channel amplitude a. In contrast, other schemes lose performance as the channel noise power increases [20,14,15]. For a covert rate of $r = 0.5$ the covert channel of [15] results in BER rates varying from 0.04 to 0.16 for a channel with Gaussian noise $N(m_N, \sigma_N)$ where σ_N varies between $50ms$ and $500ms$. The BER performance is even worse for the covert channel of [20], as compared in [15]. [14] also results in BERs of approximately 0.07 to 0.32 for a gaussian noise with $1ms \leq \sigma_N < 20ms$. On the other side, as mentioned above the BER performance of CoCo does not depend on the noise power, but on the gain of the covert channel. For a similar rate of $r = 0.5$ CoCo can achieve BER rates less than 10^{-4} for a gain parameter of 2. In fact, the gain parameter makes a tradeoff between the robustness and undetectability of the CoCo covert scheme. Larger γ reduces the BER, hence improves the robustness of the CoCo, while degrades the undetectability of CoCo as discussed in Section 6. In fact the CoCo covert channel sacrifices the provable undetectability achieved by recent research [15] for a better robustness/rate performance and a practical undetectability.

Another feature of the CoCo covert channel is being adjustable: based on the application of the covert channel and the adversarial model one can tradeoff undetectability, rate, and robustness of the CoCo covert channel. As discussed in Section 6 the choice of γ trades off the undetectability and robustness of the CoCo covert channel. Also, for a specific coding scheme reducing the rate improves the covert channel robustness.

6 Undetectability Analysis

We use the two-sample Kolmogorov-Smirnov (K-S) test [6] to evaluate the undetectability of the CoCo covert channel. We simulate the CoCo covert channel for sending SSH covert traffic and use the K-S test to distinguish between covert SSH traffic and legitimate SSH traffic. To model the legitimate traffic we use SSH traces collected by the CAIDA project from its equinix-chicago monitor — an OC192 link of a Tier 1 ISP — in January 2009 [22]. Our evaluations show that 84.6% of the SSH flows have a flow rate almost uniformly distributed between $0.2pps$ and $4.2pps$. We select 100 SSH flows with rates uniformly distributed within this range to represent our sample for the legitimate traffic, as required by the two-sample K-S tests. Each of the selected flows have at least 100 packets.

We then use CoCo to generate the covert traffic. The IPDG of CoCo simply models each SSH flow as a Poisson process with a rate randomly selected from the range of the samples, i.e., $[0.2pps, 4.2pps]$. Note that a more complicated traffic model can be generated in order to better match the behavior of a certain traffic, e.g., by matching the statistical behavior of the legitimate traffic [14]. Each flow is then generated as described in Section 4.3 and is used to modulate the covert message. Also, we use the same IPDG to generate legitimate traffic for the target traffic.

For different values of γ we run the two-sample K-S test between the traffic sample and the CoCo covert flows. Also, we run the same K-S tests between the legitimate flows and the traffic sample. We use the K-S test to distinguish between the legitimate traffic and the CoCo covert traffic by setting up a threshold for the K-S test, η_{KS}. If the K-S test of a flow passes η_{KS} the flow is declared covert, otherwise legitimate. The test produces a *false alarm* if the K-S test result is higher than the η_{KS} for a legitimate traffic, and produces a *miss* if a test value is smaller than η_{KS} for a covert flow. For some value of η_{KS}, the K-S test results in the same rates of false alarms and misses; we call this error rate as the *Cross-Over Error Rate (COER)* of the K-S test. A good test should result in very small COER rates, e.g., orders of 10^{-2}, while a bad test has COER values close to 0.5 (a random guess has a COER of 0.5). Table 7 shows the COER of the K-S test for our simulations. As can be seen, even for $\gamma = 2$ the K-S test is very poor in distinguishing between legitimate traffic and the CoCo covert traffic. In fact, this gives a *practical* undetectability as compared to the *provable* undetectability provided by [15]. In many applications of the covert channels a practical undetectability is sufficient. By adjusting the γ parameter the CoCo covert channel can be designed such that it achieves the undetectability requirements for a specific application .

Table 7. COER of the K-S in detecting the CoCo covert flows

γ	-1	0	1	1.6	2
K-S test	0.4690	0.4660	0.3390	0.3480	0.3700

7 Conclusions

In this paper, we design CoCo, an adjustable framework for covert timing channels. Using efficient coding algorithms we show that CoCo can reliably transfer covert messages with bit error rates as low as 10^{-4}, while remaining practically undetectable. Also, the robustness of the CoCo covert channel depends on the signal-to-noise ratio, not the noise power, making it suitable for establishing very noisy covert channels.

References

1. Bavier, A., Bowman, M., Chun, B., Culler, D., Karlin, S., Muir, S., Peterson, L., Roscoe, T., Spalink, T., Wawrzoniak, M.: Operating systems support for planetary-scale network services. In: Morris, R., Savage, S. (eds.) Symposium on Networked Systems Design and Implementation, pp. 253–266. USENIX (March 2004)
2. Berk, V., Giani, A., Cybenko, G.: Detection of covert channel encoding in network packet delays. Tech. Rep. TR2005-536, Dartmouth College, Computer Science, Hanover, NH (August 2005), http://www.cs.dartmouth.edu/reports/TR2005-536-rev1.pdf
3. Cabuk, Brodley, Shields: IP covert timing channels: Design and detection. In: SIGSAC: 11th ACM Conference on Computer and Communications Security. ACM SIGSAC (2004)
4. Cabuk, S.: Network covert channels: Design, analysis, detection, and elimination (January 2006), http://docs.lib.purdue.edu/dissertations/AAI3260014
5. Department of Defense: DoD 5200.28-STD: Department of Defense (DoD) Trusted Computer System Evaluation Criteria (TCSEC) (1985)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, Chichester (2001), http://www.rii.ricoh.com/~stork/DHS.html
7. Gianvecchio, S., Wang, H.: Detecting covert timing channels: an entropy-based approach. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) ACM Conference on Computer and Communications Security, pp. 307–316. ACM, New York (2007), http://doi.acm.org/10.1145/1315245.1315284
8. Gianvecchio, S., Wang, H., Wijesekera, D., Jajodia, S.: Model-based covert timing channels: Automated modeling and evasion. In: Lippmann, R., Kirda, E., Trachtenberg, A. (eds.) RAID 2008. LNCS, vol. 5230, pp. 211–230. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-87403-4_12
9. Giffin, J., Greenstadt, R., Litwack, P., Tibbetts, R.: Covert messaging through TCP timestamps. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 194–208. Springer, Heidelberg (2003)
10. Giles, J., Hajek, B.: An information-theoretic and game-theoretic study of timing channels. IEEE Transactions on Information Theory 48(9), 2455–2477 (2002)
11. Girling, C.G.: Covert channels in LAN's. IEEE Transactions in Software Engineering SE-13(2), 292–296 (1987)

12. Handel, Sandford.: Hiding data in the OSI network model. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, Springer, Heidelberg (1996)
13. van Lint, J.H.: Introduction to Coding Theory, 3rd edn. Springer, Berlin (1998)
14. Liu, Y., Ghosal, D., Armknecht, F., Sadeghi, A.R., Schulz, S., Katzenbeisser, S.: Hide and seek in time — robust covert timing channels. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 120–135. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04444-1
15. Liu, Y., Ghosal, D., Armknecht, F., Sadeghi, A.R., Schulz, S., Katzenbeisser, S.: Robust and undetectable steganographic timing channels for i.i.d. Traffic. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 193–207. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-16435-4
16. MacKay, D.: Information Theory, Inference, and Learning Algorithms (September 2003)
17. Murdoch, S.J., Lewis, S.: Embedding covert channels into TCP/IP. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 247–261. Springer, Heidelberg (2005), http://dx.doi.org/10.1007/11558859_19
18. Padlipsky, M., Snow, D., Karger, P.: Limitations of end-to-end encryption in secure computer networks. Tech. Rep. ESD TR-78-158, Mitre Corporation (1978)
19. Rowland, C.H.: Covert channels in the TCP/IP protocol suite. First Monday 2(5) (1997), http://firstmonday.org/issues/issue2_5/rowland/index.html
20. Sellke, S.H., Wang, C.C., Bagchi, S., Shroff, N.B.: Tcp/ip timing channels: Theory to implementation. In: INFOCOM, pp. 2204–2212. IEEE, Los Alamitos (2009)
21. Shah, G., Molina, A., Blaze, M.: Keyboards and covert channels. In: Proceedings of the 15th Conference on USENIX Security Symposium, vol. 15, USENIX Association, Berkeley (2006), http://portal.acm.org/citation.cfm?id=1267336.1267341
22. Walsworth, C., Aben, E., Claffy, K.C., Andersen, D.: The CAIDA anonymized 2009 Internet traces—January (March 2009), http://www.caida.org/data/passive/passive_2009_dataset.xml

LinL:Lost in n-best List

Peng Meng[1,2,3], Yun-Qing Shi[2], Liusheng Huang[1,3], Zhili Chen[1,3], Wei Yang[1,3], and Abdelrahman Desoky[4]

[1] NHPCC, Depart. of CS. & Tech., USTC, Hefei 230027,China
[2] New Jersey Institute of Technology, Newark, New Jersey, 07102, USA
[3] Suzhou Institute for Advanced Study, USTC, Suzhou, 215123,China
[4] CSEE, University of Maryland, Baltimore County, MD 21250, USA
mengpeng@mail.ustc.edu.cn

Abstract. Translation-based steganography (TBS) is a new kind of text steganographic scheme. However, contemporary TBS methods are vulnerable to statistical attacks. Differently, this paper presents a novel TBS, namely Lost in n-best List, abbreviated as LinL, that is resilient against the current statistical attacks. LinL employs only one Statistical Machine Translator (SMT) in the encoding process which selects one of the n-best list of each cover text sentence in order to camouflage messages in stegotext. The presented theoretical analysis demonstrates that there is a classification accuracy upper bound between normal translated text and the stegotext. When the text size is 1000 sentences, the theoretical maximum classification accuracy is about 60%. The experiment results also show current steganalysis methods cannot detect LinL.

Keywords: LinL, natural language steganography, translation-based steganography (TBS), text steganography, linguistic steganography.

1 Introduction

The demand for translating fueled the necessity of machine translation (MT) systems in business, science, World Wide Web, education, news, etc. As a result, the popular use of MT by a wide variety of people creates a high volume of traffic for accessing and generating translation. Such huge traffic allows communicating parties to establish a covert channel to transmit steganographic covers and the adversary is impossible to investigate all of them. This renders translation an attractive steganographic carrier.

The core idea of Translation-Based Steganography (TBS) [1,2,3] is: "When translating a non-trivial text between a pair of natural languages, there are typically many possible translations. Selecting one of these translations can be used to encode information" [1]. So the methods to generate the various translations for a given sentence are very important for the security of TBS and its embedding rate.

Contemporary TBS methods have used many different machine translators and a post-processing pass to obtain various translations. The translations obtained by these methods are much different from each other, so the stegotexts

T. Filler et al. (Eds.): IH 2011, LNCS 6958, pp. 329–341, 2011.

generated by TBS are also much different from normal translated text. Consequently, Meng *et al.* [4] and Chen *et al.* [5] successfully got their methods (STBS and NFZ-WDA) to detect TBS.

Like the relation between cryptography and cryptanalysis, steganography and steganalysis is a cat-and-mouse game. Although the statistical methods (STBS and NFZ-WDA) seem to be promising on steganalysis of TBS, translated text is still an attractive steganographic carrier due to demand for translation. Because translated texts have been widely used on the Internet, using translated text as a covert channel will draw less attention. For example, the translators of Google [6], Systran [7], Linguatec [8], just name a few, are widely used on the Internet, and in Google's vision, people will be able to translate documents instantly into the world's main languages in the future. So it is attractive to research much securer TBS.

To enhance the security of TBS, the most important work is to obtain various and similar translations for each cover text sentence. We find the n-best list [9] is a promising method to generate the similar translations.

Generally, the machine translator just generates the best translation for a given input. However, the second best translation, third best translation, and so on, can also be generated according to the applications. The first "n" best translations are known as n-best list, which has been widely used for improving the quality of machine translation and automatic speech recognition [9].

The following is an example of n-best list which is generated by Moses [10], and the n-best list is compared with the translations by other on-line machine translators.

Listed below is a German sentence: hierbei handelt es sich nicht nur um einen statistischen fehler oder um glückliche umstände. Translating this sentence to English by Moses, the 5-best list and the translations from Google, Systran, Linguatec are:

1-best: this is no mere statistical error or lucky coincidence.
2-best: this is not mere statistical error or lucky coincidence.
3-best: this is not just statistical error or lucky coincidence.
4-best: this is not only of a statistical error or lucky coincidence.
5-best: this is not only a statistical error or lucky coincidence.

Google: This is not just a statistical error-or lucky circumstances.

Systran: here it does not only concern around a statistic error or happy would stand around itself.

Linguatec: this is not only a statistical fault or happy circumstances.

The example shows the sentences of the n-best list are more similar to each other than sentences from different translators. So using n-best list to improve the security of TBS seems to be feasible.

Therefore, this paper presents a novel TBS, namely lost in n-best list (i.e. LinL), which employs the n-best list to resist the current statistical detection. LinL just uses one Statistical Machine Translator (SMT) in the encoding process and selects one of the n-best list of each cover text sentence to encode the

secret message. The difference between normal translated text and stegotext is defined by a mathematical model, and finally we give a theoretically maximum classification accuracy between normal translated text and stegotext. A series of experiments also performed to show current steganalysis methods cannot detect LinL.

The organization of this paper is as follows: Section 2 presents an overview of the related work. Section 3 briefly covers the basic operations of the TBS algorithm and some of the steganalysis methods. Section 4 focuses on the Statistical Machine Translation (SMT), and shows why n-best list is suitable for TBS. In Section 5, we use a mathematical model to define the difference between normal translated text and stegotext, and get a formula to compute the classification accuracy upper bound between normal translated text and stegotext. In Section 6 we present the results of using STBS and NFZ-WDA to detect LinL. Possible attacks on LinL are discussed in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

Text-based information, like web pages, academic papers, emails, e-books and so on, exchanged or distributed on Internet plays an important role in people's daily life. Because there are a huge number of texts available in which one can hide information, a covert communication known as linguistic steganography [11] has attracted more and more people's attention.

2.1 Linguistic Steganography

Linguistic steganography is a text steganography method that specifically considers the linguistic properties when generated and modified text, and in many cases, uses linguistic structure as the space in which messages are hidden [11]. TEXTO [12] is an early linguistic steganography program. It works just like a simple substitution cipher, with each of the 64 ASCII symbols or uuencode from secret data replaced by an English word. Wayner [13] introduced a method which uses precomputed context-free grammars to generate steganographic text without sacrificing syntactic and semantic correctness. Chapman and Davida [14] gave another steganographic method called NICETEXT. The texts generated by NICETEXT not only had syntactic and lexical variation, but whose consistent register and "style" could potentially pass a casual reading by a human observer. Chang and Clark [15] introduced a method to integrate text paraphrasing into a linguistic steganography system.

Non-linguistic approaches to text steganography have also been researched. Liu and Tsai [16] proposed a steganographic method for data hiding in Microsoft Word documents by a change tracking technique. Desoky [17, 18, 19, 20] has introduced a series of text steganography methods , which are named as noiseless steganography (Nostega).

2.2 Statistical Steganalysis

For detecting the above linguistic steganography, some steganalytic algorithms have been proposed. Taskiran et al. [21] used a universal steganalytic method based on language models and support vector machines to differentiate sentences modified by a lexical steganography algorithm from unmodified sentences. Chen et al. [22] used the statistical characteristics of correlations between the general service words gathered in a dictionary to classify given text segments into stegotexts and normal texts. This method can accurately detect NICETEXT and TEXTO systems. The paper [23] also brought forward a detection method for NICETEXT, which took advantage of distribution of words. Another effective linguistic steganography detection method [24] uses an information entropy-like statistical variable of words together with its variance as two features to classify text segments.

3 Translation-Base Steganography and Steganalysis

This section briefly presents an overview of the translation-based steganography (TBS). To introduce TBS, we focuse on the "Lost in Just the Translation (LiJtT)" [2] which extends the original "Lost in Translation (LiT)" [1] into one which allows the sender to only transmit the stegotext. The encoding processes of both LiT and LiJtT are selecting the translation results by various translators to encoding bits.

Conceptually, TBS works as follows: First, the sender obtain a cover text in the source language. The cover text could be a secret of the sender or could have been obtained from public sources — for example, a news website. Then, the sender translates the sentences in the source language into the target language using multiple different translators. Because a sentence translated by different translators may generate different translation results, the sender essentially creates multiple translations for each sentence and ultimately selects one of these to encode some bits of the hidden message.

The encoding process of LiJtT specifically works as follows. After generating multiple translations for a given cover text sentence, the sender uses the secret key (which is shared between the sender and receiver) to hash the individual translated sentences into bit strings. The lowest h bits of the hash strings, referred to as header bits, are interpreted as an integer $b \geq 0$. Then the sentence whose lowest $[h+1, h+b]$ bits corresponds to the bit-sequence that is to be encoded is selected.

When the receiver receives a translation which contains a hidden message, he first breaks the received text into sentences. Then applies a keyed hash to each received sentence. The lowest $[h+1, h+b]$ bits in this hash contain the next b bits of the hidden message. Figure 1 illustrates the protocol.

These methods to generate different translations for data hiding can be detected by statistical methods. Papers [25, 26] present the first steganalysis method on TBS, which needs to know the MT set and the source language of the cover text. Due to the source language and the translator set may be part

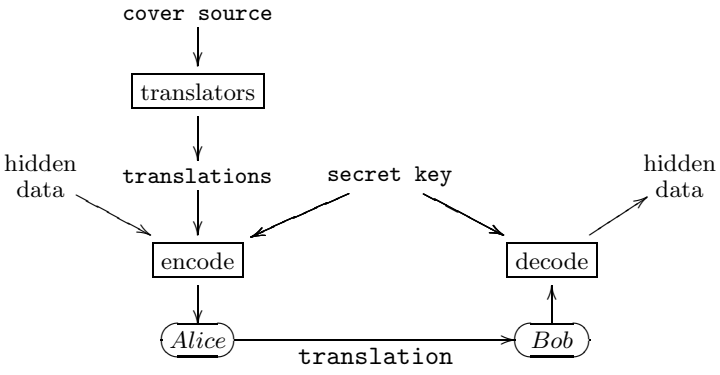


Fig. 1. Illustration of the basic protocol (from [2]). The adversary can observe the message between Alice and Bob containing the selected translation.

of the private secret of the sender [2], the method cannot be used in general. To blind detection of TBS, Meng *et al.* [4] introduced a statistical steganalysis method which was named STBS. STBS is based on the word and 2-gram frequency difference between normal text and stegotext, the average classifying accuracy is about 80% when the text size is 20K bytes . To accurately detect TBS when the text size is much smaller, Chen *et al.* [5] gave another statistical steganalysis method, which is named natural frequency zoned word distribution analysis (NFZ-WDA). When the text size is 5K bytes, the detection accuracy is above 90%.

The steganalysis methods have demonstrated that the security of TBS is based on the methods to generate various translations. The more similarity between the translations, it is the more difficult to classify normal translated text and stegotext. The contemporary TBS uses different translators and a post-processing pass to generate the various translations for a cover text sentence. Because the translations resulted from different translators are much different to each other, Meng *et al.* [4] and Chen *et al.* [5] successfully introduced their methods to detect TBS. So it becomes clear that generating similar translations for the cover text sentence is pivotal for the security of TBS.

To generate the various and similar translations of a cover text sentence, n-best list of statistical machine translation (SMT) [9] seems to be a good strategy. To thoroughly study the security of using n-best list in TBS encoding process, we introduce the process of statistical machine translation.

4 Statistical Machine Translation

Statistical Machine Translation (SMT) as a research area started in the late 1980s. Lately, most competitive statistical machine translation systems use phrase-based translation [27].

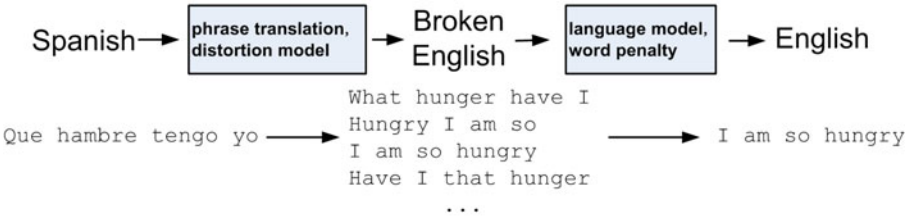


Fig. 2. An illustration of phrase-based translation

SMT working process can be simply summarized as follows(by translating a different language to English as an example): For all the candidate English sentences of a foreign language sentence, SMT counts a probability cost for each of them and outputs the sentence with the highest probability cost as the translations.

Figure 2 illustrates the process of phrase-based translation.

The probability cost that is assigned to a translation is a product of the probability costs of four models: phrase translation table, language model, reordering model, and word penalty.

Each of the four models contributes information over one aspect of the characteristics of a good translation:

"The phrase translation table ensures that the English phrases and the foreign language phrases are good translations of each other.

The language model ensures that the output is fluent English.

The distortion model allows for reordering of the input sentence.

The word penalty provides means to ensure that the translations do not get too long or too short" [27].

Each of the models can be given a weight that sets its importance. Mathematically, the cost of translation is:

$$p(e|f) = \Phi(f|e)^{weight_\Phi} \times LM^{weight_{LM}} \times D(e,f)^{weight_d} \times W(e)^{weight_w}$$

The probability cost of the English translation e given the foreign input f, $p(e|f)$, is broken up into four models, phrase translation $\Phi(f|e)$, language model $LM(e)$, distortion model $D(e,f)$, and word penalty $W(e) = exp(length(e))$. Each of the four model is weighted by a weight [27].

To translate a sentence, the main process of SMT is to search the best translation from hundreds and thousands of candidate translations. An upper bound for the number of candidate English sentences can be estimated by $N \sim 2^{n_f}|V_e|n_f$ [27] where n_f is the number of foreign words of the translated sentence , and $|V_e|$ the size of the English vocabulary. Because the search space is very large, one can imagine that the best translation, the second best translation, the third best translation, and so on, will be very similar to each other. Thus, the stegotext generated by TBS that is based on n-best-list would be difficult to be differentiated from normal translated text.

To validate the security of using n-best list in TBS, we provide both theory analysis and experiment study. In the next section, we give a theory analysis of using n-best list in TBS.

5 Theoretically Analyze the Security of LinL

In this section, we estimate the difference between normal translated text and stegotext by establishing a mathematical model, and we finally give a formula to compute the classification accuracy upper bound of LinL.

The translation process of SMT shows each candidate English sentence is associated with a probability cost, i.e., from SMT point of view each candidate English sentence is just treated as a probability, SMT just outputs the sentence with the highest probability as the translations. From the perspective of SMT, the probability cost is considered as the only feature of the translations. So the difference between the n-best list can be defined by the difference between each sentence's probability cost, and the difference between normal translated text and stegotext can be defined by the difference between their probability cost distributions.

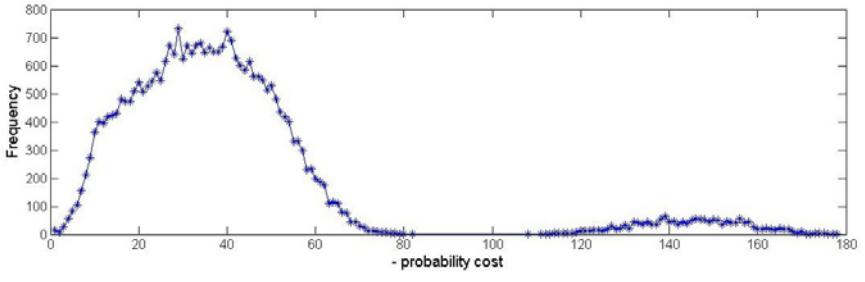


Fig. 3. The distribution of the probability cost of normal translated sentences

Figure 3 shows the distribution of the probability cost of the normal translated sentences. Except some very high values, the distribution of the probability cost can be approximatively considered as normal distribution. Because the difference of the probability cost of n-best list is very small, he distribution of the probability cost of stegotext sentences can also be approximatively considered as normal distribution.

For a text segment which contains m sentences, there are totaly m probability cost features. Because each probability cost feature can be considered as a normal distribution variable, the vector of the m probability cost features can be considered as m-variate multivariate normal. The m-vector is the only measurement of the text. So the problem of classifying between normal translated text and stegotext is turned to the classification of two multivariate normal distributions.

Table 1. The means and variances of the probability cost of normal translated texts and stegotexts

Type	Ave	Var
normal	-44.49	42.89
Li2L	-45.16	42.77
Li4L	-46.12	44.99
Li8L	-46.79	47.85

Suppose the distributions of the probability cost of the normal translated texts and stegotexts are denoted by two normal distributions: $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, where μ_1 and μ_2 are the means, and σ_1 and σ_2 are the variances of the first and second populations, respectively. The means and variances of normal translated texts and stegotexts can be obtained by a statistical method. Table 1 shows the means the variances of different type of texts that we have obtained from more than 10 thousands of sentences of each type. Li2L, Li4L and Li8L represent TBS with 2-best, 4-best and 8-best list to generate the stegotext, respectively.

Assume that the text contains m sentences and the probability cost of all sentences are independent, so the normal translated texts and stegotexts can be denoted by two m-variate multivariate normal distributions: $N(\mu_{m1}, \Sigma_1)$ and $N(\mu_{m2}, \Sigma_2)$, where,

$$\mu_{m1} = \begin{bmatrix} \mu_1 \\ \mu_1 \\ \vdots \\ \mu_1 \end{bmatrix} \qquad and \qquad \mu_{m2} = \begin{bmatrix} \mu_2 \\ \mu_2 \\ \vdots \\ \mu_2 \end{bmatrix}$$

are the mean vectors (each contains m values),

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_1 & & \\ & & \ddots & \\ & & & \sigma_1 \end{bmatrix} \qquad and \qquad \Sigma_2 = \begin{bmatrix} \sigma_2 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_2 \end{bmatrix}$$

are the covariance matrices of the first and second populations, respectively.

The problem of classification of two multivariate normal distribution has been thoroughly researched in multivariate statistical analysis. For the two m-variate multivariate normal distributions, as defined above, the maximum classification accuracy can be computed by the following formula [28]:

$$Accuracy = \int_{-\infty}^{\sqrt{m}\frac{|\mu_1 - \mu_2|}{\sigma_1 + \sigma_2}} (2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}t^2} dt$$

Table 2. Maximum classification accuracy of LinL

Type \ Length (Sen.)	100	200	300	400	500	600	700	800	900	1000
Li2L	0.53	0.54	0.55	0.56	0.57	0.58	0.58	0.59	0.59	0.60
Li4L	0.57	0.60	0.63	0.65	0.66	0.68	0.69	0.70	0.71	0.72
Li8L	0.60	0.64	0.67	0.70	0.72	0.73	0.75	0.77	0.78	0.79

Using this formula to compute the maximum classification accuracy of normal translated texts and stegotexts, which only needs to know the means and variances of the probability cost.

With the data of Table 1, the maximum classification accuracy between normal translated text and stegotext can be couputed. Table 2 shows the maximum classification accuracy with the data of Table 1. From the data of Table 2, the following can be concluded:

- The classification accuracy increases with the text size increases.
- The less n-best list used in the TBS encoding process, the more secure for LinL.

6 Experiment

A series of experiments were performed to show the security of LinL. The experiments use the steganalysis methods which have successfully detected contemporary TBS to detect LinL.

Moses [10] was used to translate from German to English to generate the n-best list. The WMT08 News Commentary data set [29], about 55k sentences were used to train Moses and as the source text of the experiment. Li2L, Li4L and Li8L were tested. The normal translated texts and stegotexts were split to 10K bytes segment. STBS [4] and NFZ-WDA [5] methods were tested respectively. Table 3 shows the detection results.

The experiment results in Table 3 shows both STBS and NFZ-WDA cannot detect LinL. When using STBS to detect Li2L and Li4L, the detection accuracy is no better than random guess. Even using STBS to detect Li8L, the detection accuracy is still very low. When using NFZ-WDA to detect Li2L, Li4L and Li8L, respectively, it would classify most of the test texts to normal translated text.

7 Discussion

This section discusses the various possible attacks on LinL. As one of the serial TBS methods, some the discussions about LiT [1] and LiJtT [2], like future machine translation and repeated sentence problems, are also suitable for LinL. We just discuss the problems that may come out with LinL in this section.

Table 3. Experiment results of using STBS and NFZ-WDA to detect LinL

	Type	Train	Test	Non-stego	Stego	Accuracy(%)
STBS	Normal	50	229	155	74	51.02
	Li2L	50	212	142	70	
	Normal	50	229	99	130	48.49
	Li4L	50	169	75	94	
	Normal	50	229	133	96	61.36
	Li8L	50	110	35	75	
NFZ-WDA	Normal	50	229	224	5	51.02
	Li2L	50	212	211	1	
	Normal	50	229	194	35	54.02
	Li4L	50	169	148	21	
	Normal	50	229	178	51	59.29
	Li8L	50	110	87	23	

7.1 Translation Quality

Whether the translation quality of stegotext is worse than normal translated text? From SMT point of view, some sentences of stegotext are not the best translation, but the second best translaion, third best translation, and so on, the answer is yes. However, translation quality is difficult to be used as a feature to classify a text to normal translated text and stegotext. First, the translation quality is difficult to count, and the translation quality of different machine translator or the same machine translator with different training database is much different. Second, the best translation given by a MT may not be the best translation from human's perspective. So using translation quality to attack LinL seems impossible.

7.2 Statistical Attacks

Statistical attacks have been extremely successful at all area of steganography, such as image [30], video [31] and text [22]. We also cannot preclude the existence of yet-undiscovered statistical methods for defeating LinL. However, a classification accuracy upper bound between normal translated text and stegotet is given, it can be used as a reference when use LinL. For steganography and steganalysis, it is an arm race. Once a statistical steganalysis is known, it is actually easy to modify the steganography method to resist its attacks.

8 Conclusion

This paper introduces a novel translation based steganography, namely LinL, which uses the n-best list of a statistical machine translator (SMT) to encode the secret message. We just use one machine translator in the encoding process, the generated texts (stegotexts) of LinL are very similar to normal translated text, so it is difficult to classify normal translated texts and stegotexts. To show the security of LinL, we have derived a detection accuracy upper bound of

LinL, and some steganalysis methods are tested on LinL, the experiment results show current steganalysis methods cannot classify normal translated text and stegotext.

Comparing with contemporary TBS, LinL can resist statistical detection and the embedding rate can be changed easily. Further more, LinL does not need post-processing algorithms either. To enhance the embedding rate, we can select a bigger "n" of the n-best list. To enhance the security of LinL, we just select a smaller "n" of the n-best list. However, if we just select the 1-best translation result, LinL will just be a normal translator.

The security of LinL maybe can continue to improve, for example, according to the sentence length or the probability cost of each translations, to select a different number of "n" for each sentence will be better for the security and embedding rate of LinL. This problem will be investigated in the future work. Although there is still some research work to be done for LinL, the theory analysis and experiment results shown have demonstrated that using n-best list to enhance the security of TBS is promising.

Acknowledgment. This work was partly supported by the Major Research Plan of the National Natural Science Foundation of China (No. 90818005) and the National Natural Science Foundation of China (No. 60903217).

References

1. Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R., Atallah, M.: Translation-based steganography. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 219–233. Springer, Heidelberg (2005)
2. Stutsman, R., Atallah, M., Grothoff, K.: Lost in just the translation. In: Proceedings of the 2006 ACM Symposium on Applied Computing, pp. 338–345. ACM, New York (2006)
3. Grothoff, C., Grothoff, K., Stutsman, R., Alkhutova, L., Atallah, M.: Translation-based steganography. Journal of Computer Security 17(3), 269–303 (2009)
4. Meng, P., Hang, L., Chen, Z., Hu, Y., Yang, W.: STBS: A statistical algorithm for steganalysis of translation-based steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 208–220. Springer, Heidelberg (2010)
5. Chen, Z., Huang, L., Meng, P., Yang, W., Miao, H.: Blind linguistic steganalysis against translation based steganography. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 251–265. Springer, Heidelberg (2011)
6. Google: Google translator (2009), http://translate.google.cn
7. Systran: Systran translator (2009), https://www.systransoft.com
8. Linguatec: Linguatec translation, http://www.linguatec.de
9. Chen, B., Zhang, M., Aw, A., Li, H.: Exploiting n-best hypotheses for smt self-enhancement. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 157–160. Association for Computational Linguistics (2008)

10. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics (2007)
11. Bennett, K.: Linguistic steganography: Survey, analysis, and robustness concerns for hiding information in text. Purdue University, CERIAS Tech. Report (2004)
12. Maker, K.: TEXTO, ftp://ftp.funet.fi/pub/crypt/steganography/texto.tar.gz
13. Wayner, P.: Disappearing cryptography: information hiding: steganography and watermarking. Morgan Kaufmann Pub., San Francisco (2008)
14. Chapman, M., Davida, D.: Hiding the hidden: A software system for concealing ciphertext as innocuous text. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 335–345. Springer, Heidelberg (1997)
15. Chang, C., Clark, S.: Linguistic steganography using automatically generated paraphrases. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010)
16. Liu, T., Tsai, W.: A new steganographic method for data hiding in microsoft word documents by a change tracking technique. IEEE Transactions on Information Forensics and Security 2(1), 24–30 (2007)
17. Desoky, A.: Nostega: a novel noiseless steganography paradigm. Journal of Digital Forensic Practice 2(3), 132–139 (2008)
18. Desoky, A.: Listega: list-based steganography methodology. International Journal of Information Security 8(4), 247–261 (2009)
19. Desoky, A.: NORMALS: normal linguistic steganography methodology. Journal of Information Hiding and Multimedia Signal Processing 1(3), 145–171 (2010)
20. Desoky, A.: Matlist: mature linguistic steganography methodology. Security and Communication Networks
21. Taskiran, C., Topkara, U., Topkara, M., Delp, E.: Attacks on lexical natural language steganography systems. In: Proceedings of SPIE, vol. 6072, pp. 97–105 (2006)
22. Zhili, C., Liusheng, H., Zhenshan, Y., Wei, Y., Lingjun, L., Xueling, Z., Xinxin, Z.: Linguistic steganography detection using statistical characteristics of correlations between words. In: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 224–235. Springer, Heidelberg (2008)
23. Zhili, C., Liusheng, H., Zhenshan, Y., Lingjun, L., Wei, Y.: A statistical algorithm for linguistic steganography detection based on distribution of words. In: Third International Conference on Availability, Reliability and Security, ARES 2008, pp. 558–563 (2008)
24. Zhili, C., Liusheng, H., Zhenshan, Y., Xinxin, Z.: Effective linguistic steganography detection. In: IEEE 8th International Conference on Computer and Information Technology Workshops, CIT Workshops 2008, pp. 224–229 (2008)
25. Meng, P., Hang, L., Yang, W., Chen, Z.: Attacks on translation based steganography. In: IEEE Youth Conference on Information, Computing and Telecommunication, YC-ICT 2009, pp. 227–230. IEEE, Los Alamitos (2010)
26. Meng, P., Hang, L., Chen, Z., Yang, W., Yang, M.: Analysis and detection of translation-based steganography. Chinese Journal of Electronics 38(8), 1748–1752 (2010)

27. Koehn, P.: MOSES, Statistical Machine Translation System, User Manual and Code Guide (2010)
28. Anderson, T., Bahadur, R.: Classification into two multivariate normal distributions with different covariance matrices. The Annals of Mathematical Statistics 33(2), 420–431 (1962)
29. WMT08: Wmt08 news commentary (2008), http://www.statmt.org/wmt08/training-parallel.tar
30. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG images: Breaking the F5 algorithm. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 310–323. Springer, Heidelberg (2003)
31. Budhia, U., Kundur, D., Zourntos, T.: Digital video steganalysis exploiting statistical visibility in the temporal domain. IEEE Transactions on Information Forensics and Security 1(4), 502–516 (2006)

Author Index