LNCS 8948
Journal Subline

Transactions on
Data Hiding and Multimedia Security X

Yun Q. Shi
Editor-in-Chief

Springer

Lecture Notes in Computer Science 8948

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

More information about this series at http://www.springer.com/series/7870

Yun Q. Shi (Ed.)

Transactions on Data Hiding and Multimedia Security X

Springer

Editor
Yun Q. Shi
New Jersey Institute of Technology
Newark, NJ
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-662-46738-1 ISBN 978-3-662-46739-8 (eBook)
DOI 10.1007/978-3-662-46739-8

Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media
(www.springer.com)

Transactions on Data Hiding and Multimedia Security
Tenth Issue

In this volume we present the tenth issue of the *LNCS Transactions on Data Hiding and Multimedia Security,* which includes six papers.

The first paper presents a new method to reduce mutual information via embedding watermark in the key controlled wavelet domain. The second paper presents a perceptual image hashing algorithm based on wave atom transform, which can distinguish maliciously attacked images from content-preserving ones. In the third paper, specular reflection for short-wavelength-pass-filter detection is proposed to prevent rerecording screen images. The remaining three papers deal with steganography. While most steganographic research has been done in the field of non-real-time mediums, an algorithm that enables data hiding in G.711, the most commonly used voice codec for VoIP devices, is presented in the fourth paper. The fifth paper addresses adaptive steganography and steganalysis with fixed-size embedding, where a two-player zero-sum game between a steganographer and a steganalyst is analyzed. The sixth paper addresses permutation steganography in the File Allocation Table (FAT) file system.

We hope that this issue will be of great interest to the research community and will trigger new research in the field of data hiding and multimedia security.

Finally, we want to thank all the authors, reviewers, and editors who have devoted their valuable time to the success of this sixth issue. Special thanks go to Springer Verlag and Dr. Alfred Hofmann for their continuous support.

December 2014

Yun Q. Shi
Hyoung-Joong Kim
Stefan Katzenbeisser

LNCS Transactions on Data Hiding and Multimedia Security Editorial Board

Advisory Board Members

Contents

Strengthening Spread Spectrum Watermarking Security via Key Controlled Wavelet Filter

Bingbing Xia[✉], Xianfeng Zhao, Dengguo Feng, and Mingsheng Wang

State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, People's Republic of China
{xiabingbing,xfzhao,feng,mswang}@is.iscas.ac.cn

Abstract. Spread spectrum watermarking security can be evaluated via mutual information. In this paper, we present a new method to reduce mutual information by embedding watermark in the key controlled wavelet domain. Theoretical analysis shows that the watermark signals are diffused and its energy is weakened when they are evaluated from the attacker's observation domain, and it can lead to higher document-to-watermark energy ratio and better watermark security without losing robustness. Practical algorithms of security tests using optimal estimators are also applied and the performance of the estimators in the observation domain is studied. Besides, we also present a novel method of calculating the key controlled wavelet filter, and give both numerical and analytical implementations. Experiment results show that this method provides more valid parameters than existing methods.

Keywords: Watermarking security · Spread spectrum · Key controlled wavelet · Parameterizations · Mutual information

1 Introduction

Watermarking security has received much more attention in recent years [1,11]. Various mathematical frameworks such as Fisher's information [2], Shannon's equivocation [9] have been used to perform theoretical analysis on spread spectrum watermarking schemes. In spread spectrum watermarking scheme, the watermarker owns a secret key that he or she repeatedly uses to watermark contents. The attacker can obtain several observations watermarked by the same key to get information about the secret key, and then they can implement optimal attacks on the watermarking scheme. Thus, watermarking security can be evaluated by the difficulty of estimating the secret key in the attacker's view [2].

The information about the secret key revealed by the observations can be quantified by Shannon's mutual information [9]. The calculation of the mutual information for the various existing spread spectrum watermarking scheme is

This work was supported by the NSF of China under 61170281, NSF of Beijing under 4112063, Strategic and Pilot Project of CAS under XDA06030601, and the Project of IIE, CAS, under Y1Z0041101 and Y1Z0051101.

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 1–20, 2015.
DOI: 10.1007/978-3-662-46739-8_1

given in [9]. When they focus only on the number of the observations needed to achieve certain estimation accuracy (regarded as “security level”), we present a new way to reduce the mutual information by increasing the document-to-watermark energy ratio, using the method of embedding watermark on key-controlled wavelet domain. Our method leads to better security of spread-spectrum watermarking scheme. To embed watermark in the key-controlled wavelet domain, we firstly calculate parameterized wavelet filters by some parameters and then embed watermark signals in the coefficients of the wavelet decomposition sub-bands called **embedding domain**. Parameters used to calculate wavelet filters are kept secret as part of the secret key. Attackers can only manipulate on the wavelet decomposition sub-bands created by arbitrarily decided parameters, which is called **observation domain**. Based on some results in [9], we prove that the watermarking signals are diffused inside and between the wavelet sub-bands when evaluated from the attacker’s observation domain, and this will result in a reduction of the watermark energy. On the theoretical side, the watermark diffusion effect can lead to higher document-to-watermark energy ratio and thus strengthen the security of spread-spectrum watermarking scheme without losing robustness. On the practical side, the watermark diffusion effect in the observation domain can practically change the optimal condition in a pattern unknown to the attacker, which makes the existing practical estimators of the secret key become less effective. This watermark diffusion effect is independent from the specific watermark embedding algorithms. Thus this method can be integrated in any existing wavelet domain watermark algorithm to strengthen the security of the scheme without losing robustness. This watermarking scheme involving key controlled wavelets can also be combined with methods of watermark synchronization, such as in [18], to survive geometric attacks.

Some methods about wavelet parameterizations have been presented. Zou and Tewfik [19] proposed a principle to achieve wavelet parameterizations by constructing wavelet filter frequency response. Based on it, Schneid and Pittner [17] designed an iterative form implementation to calculate the parameterized wavelet filter. Dietl *et al.* [6] applied this implementation into spread spectrum watermarking and thus designed a parameterized wavelet domain watermarking-scheme, without a detailed analysis to the security introduced by the parameterized wavelet filters. The main drawback of this implementation is its inconvenient iterative calculation process. To obtain parameterized wavelet filters with length N, one has to calculate a series of parameterized wavelet filters with length from 2 to $N - 1$ consequently. Regensburger [16] presented another method to calculate parameterized wavelet filters by introducing discrete moment as parameters. Gröbner basis is used to gain analytical resolutions to the parameterized equations. Though the discrete moment is unnecessary to the scheme of spread-spectrum watermarking, calculating parameterized wavelet filters by solving nonlinear equations suggests a better way than the iterative process in [6]. In this paper, we follow the method in [16], but replace the unnecessary discrete moment with arbitrary parameters. In this way, the constraints contained in the nonlinear equations are reduced as much as possible. Therefore, the solution space covers more usable wavelets than the method in [6], which provides a bigger key space of the watermarking scheme. We provide both numerical and analytical methods to solve the nonlinear equations. Experimental results show that the key space approximates to 5×10^5 roughly for wavelet filter with length of 6, and it will increase with the wavelet filter length.

The structure of this paper is organized as follows: in Sect. 2, we overview how to evaluate the watermarking security using mutual information and express the basic idea of reducing the mutual information. In Sect. 3, we theoretically analyze the watermark diffusion and the energy reduction effects brought by involving key controlled wavelets. In Sect. 4, estimations to the secret keys of the spread-spectrum watermarking schemes are applied. As the optimal conditions change in the observation domain, the optimal estimators become less effective. In Sect. 5, we present the principle of obtaining key-controlled wavelet filters by solving nonlinear equations with arbitrary parameters, and give both numerical and analytical implementations. Section 6 covers rough estimations to the key space of the key-controlled wavelet based watermarking scheme. Conclusions are given in Sect. 7.

2 Evaluating Spread-Spectrum Watermarking Security Using Mutual Information

Spread spectrum watermark embedding process can be summarized as $y = x + w$ where x and y are sample values of the embedding domain before and after watermark embedding, respectively; w is the watermark sequence with length n generated from a secret key. The watermarker uses w repeatedly to watermark a set of contents denoted by $\{x_1, x_2, \cdots, x_N\}$, and then produces a set of watermarked contents denoted by $\{y_1, y_2, \cdots, y_N\}$, which can be obtained by the attacker. The attacker's goal is to estimate the watermark information and the corresponding secret key, by using some optimal estimators to deal with the observations containing watermarks derived from the same secret key. So the spread-spectrum watermarking security can be evaluated by the difficulty of estimating the secret key for the attacker's view. The evaluation can be achieved by means of the Shannon's mutual information $I(y_1, y_2, \cdots, y_N; w)$.

Freire and González [9] studied the various existing spread-spectrum watermarking schemes [4,12,13] in two different scenarios called known message attack (KMA) and watermarked only attack (WOA), and gave the calculation of mutual information in both case. In the KMA scenario, the mutual information between the watermarked contents $\{y_1, y_2, \cdots, y_N\}$ and the watermark signal w can be calculated as

$$\frac{1}{n} I\left(y_1, y_2, \cdots, y_N; w\right) = \frac{1}{2} \log\left(1 + N \cdot \xi^{-1}\right) \tag{1}$$

where $\xi = \sigma_x^2 / D_w$ is the so-called document-to-watermark ratio(DWR) for quantifying the relative powers between the host and the watermark signals.

σ_x^2 denotes the variance of the host signal x, and $D_w = (1/n)E[\|w\|^2]$ is the embedding distortion per dimension defined in [9], which is called 'watermark energy' in this paper for simplification.

As seen from Eq. (1), there are two factors that affect the mutual information values: the number of the observations N owned by the attacker, and the document-to-watermark energy ratio ξ. Since Freire and González [9] studied the number of the observations N needed to achieve certain estimation accuracy (regarded as "security level"), we mainly focus on the other factor. Obviously, the mutual information in the KMA scenario is a decreasing function of ξ, which suggests that increasing ξ will lead to a reduction on mutual information, and thus achieve better security for spread-spectrum watermarking.

Similar result holds for the WOA scenario. Although the exact expression for the mutual information cannot be obtained, Freire and González [9] derived the upper and lower bounds. Both the upper and lower bounds contains the same part as in Eq. (1), as well as two other terms consist of some statistics of the original contents $\{x_1, x_2, \cdots, x_N\}$. Based on these results, we can assert roughly that increasing ξ will also lead to a reduction on mutual information in the WOA scenario, which is further supported by the experimental result in [9].

Given a set of the original contents to be watermarked, it is straightforward to increase ξ by decreasing the watermark energy D_w. However, doing this will also reduce the robustness of the watermarking scheme. In Sect. 3, we describe in details a new approach to increase this ratio ξ without reducing the embedding capacity and robustness by using key controlled wavelet filter.

3 Watermark Diffusion and Energy Reduction in the Observation Domain

The framework of the watermark embedding and extracting scheme using key controlled wavelet filter is basically the same to the watermarking scheme on standard wavelet domains, except we use the parameterized wavelet filters instead of the standard ones for decomposition and reconstruction. Given an original content to be watermarked, we firstly calculate the parameterized wavelet filters using a set of parameters, and then obtain the wavelet decomposition sub-bands determined by these parameterized wavelet filters. We use the Improved Spread Spectrum (ISS) watermarking scheme proposed in [12] to embed watermark signals in the wavelet decomposition coefficients. Other existing spread spectrum watermarking methods can also be utilized in the similar way as well. The watermarked content is finally obtain by wavelet reconstruction. On the watermark extracting side, the same parameterized wavelet filters are generated using the same parameters, thus the watermark can be extracted correctly.

Since the parameters used both in watermark embedding and extracting are kept secret as part of the watermarking key, attackers can only manipulate on wavelet decomposition sub-bands generated by the arbitrarily decided parameters. We use **embedding domain** to denote the wavelet decomposition sub-bands where the watermark is truly embedded, and **observation domain** for the wavelet decomposition sub-bands where attackers can manipulate. In observation domains, the watermark signals are diffused inside and between the sub-bands, and the watermark energy is weakened. In this way, we can raise the document-to-watermark energy ratio in the attacker's perspective while maintaining the embedding strength and capacity in the recipient perspective.

We will discuss this in details in two steps: the single dimension watermark scenario and the multiple dimension watermark scenario. Images are chose as the original content for carrying the watermark without loss of generality.

3.1 Single Dimension Watermark Scenario

In this scenario, we limit the length of the watermark to one. Although this is not a practical scenario, we can further discuss the more practical scenario where the length of the watermark is not constrained based on the conclusions of this stage. Without loss of generality, we embed a single watermark element W into the LH sub-band of wavelet multi-resolution decomposition of the cover image I, with D_w denoting the original watermark energy. When evaluated from an arbitrary observation domain, the watermark signals in LH sub-band changes to $\tilde{W}$ and the corresponding watermark energy is $\tilde{D}_w$. Given that the watermark W can be dependent or independent to I depending on the specific embedding algorithm, our theoretical analysis stated below will always hold.

Let $\{h_0(k), h_1(k)\}$ and $\{{h_0}'(k), {h_1}'(k)\}$ denote the wavelet decomposition and reconstruct filter coefficients corresponding to the embedding domain, respectively. The four sub-bands of wavelet multi-resolution decomposition of the cover image I are as follows:

$$C_1(x,y) = \sum_m \sum_n h_0(m-2x)h_0(n-2y)I(m,n) \tag{2a}$$

$$H_1(x,y) = \sum_m \sum_n h_0(m-2x)h_1(n-2y)I(m,n) \tag{2b}$$

$$V_1(x,y) = \sum_m \sum_n h_1(m-2x)h_0(n-2y)I(m,n) \tag{2c}$$

$$D_1(x,y) = \sum_m \sum_n h_1(m-2x)h_1(n-2y)I(m,n) \tag{2d}$$

The embedding process of a single dimension watermark in LH sub-band can be written as $\hat{H}_1(x_d, y_d) = H_1(x_d, y_d) + W$, where (x_d, y_d) stands for the embedding position. After the wavelet reconstruction, we obtain the watermarked image $\hat{I}$ as

$$\begin{aligned}\hat{I}(x,y) = &\sum_m \sum_n C_1(m,n){h'}_0(x-2m){h'}_0(y-2n)\\ &+\sum_m \sum_n \hat{H}_1(m,n){h'}_0(x-2m){h'}_1(y-2n)\\ &+\sum_m \sum_n V_1(m,n){h'}_1(x-2m){h'}_0(y-2n)\end{aligned} \tag{3}$$

$$
\begin{aligned}
&+\sum_m \sum_n D_1(m,n) {h'}_1(x-2m) {h'}_1(y-2n) \\
&= I(x,y) + W \cdot {h'}_0(x-2x_d) {h'}_1(y-2y_d)
\end{aligned}
$$

Let $\{\tilde{h}_0(k), \tilde{h}_1(k)\}$ and $\{\tilde{h}'_0(k), \tilde{h}'_1(k)\}$ denote the wavelet decomposition and reconstruction filter coefficients corresponding to the attacker's observation domain. The wavelet decomposition sub-band in the observation domain is

$$
\begin{aligned}
\tilde{C}_1(x,y) =& C_1(x,y) \\
&+ W \cdot \sum_m \tilde{h}_0(m-2x) {h'}_0(m-2x_d) \\
&\cdot \sum_n \tilde{h}_0(n-2y) {h'}_0(n-2y_d)
\end{aligned} \tag{4a}
$$

$$
\begin{aligned}
\tilde{H}_1(x,y) =& H_1(x,y) \\
&+ W \cdot \sum_m \tilde{h}_0(m-2x) {h'}_0(m-2x_d) \\
&\cdot \sum_n \tilde{h}_1(n-2y) {h'}_1(n-2y_d)
\end{aligned} \tag{4b}
$$

$$
\begin{aligned}
\tilde{V}_1(x,y) =& V_1(x,y) \\
&+ W \cdot \sum_m \tilde{h}_1(m-2x) {h'}_0(m-2x_d) \\
&\cdot \sum_n \tilde{h}_0(n-2y) {h'}_1(n-2y_d)
\end{aligned} \tag{4c}
$$

$$
\begin{aligned}
\tilde{D}_1(x,y) =& D_1(x,y) \\
&+ W \cdot \sum_m \tilde{h}_1(m-2x) {h'}_0(m-2x_d) \\
&\cdot \sum_n \tilde{h}_1(n-2y) {h'}_1(n-2y_d)
\end{aligned} \tag{4d}
$$

As can be seen from the above, the watermark in the observation domain diffuses to all four wavelet decomposition sub-bands, i.e. the watermark signals diffuses between sub-bands. Hereafter we are going to discuss the details of the diffusion inside the sub-band. We choose LH sub-band for further discussion, while similar analysis can be applied to other sub-bands.

From Eq. (4b), the watermark signals in LH sub-band are

$$
\begin{aligned}
\tilde{W} =& W \cdot \sum_m \tilde{h}_0(m-2x) {h'}_0(m-2x_d) \\
&\cdot \sum_n \tilde{h}_1(n-2y) {h'}_1(n-2y_d) \\
\triangleq& W \cdot \delta_{x,y}
\end{aligned} \tag{5}
$$

We call $\delta_{x,y}$ wavelet diffusivity. As is seen, the watermark signals in LH sub-band in the observation domain diffuse to a square area centered on the original embedding position, i.e. the watermark signal diffuses inside wavelet decomposition sub-bands.

Now we can calculate the corresponding watermark energy in the observation domain as

$$\begin{aligned}\tilde{D}_w &= \frac{1}{XY}E\left[\left\|\tilde{W}\right\|^2\right] \\ &= \frac{1}{XY}E\left[\|W\|^2 \cdot \sum \delta_{x,y}^2\right] \\ &= \frac{1}{XY} \cdot \sum \delta_{x,y}^2 \cdot E\left[\|W\|^2\right] \\ &= \frac{1}{XY} \cdot \sum \delta_{x,y}^2 \cdot D_w\end{aligned} \tag{6}$$

To compare $\tilde{D}_w$ against the original watermark energy D_w in the embedding domain, we proceed from Eq. (5) to further derivation.

$$\begin{aligned}|\delta_{x,y}| &= \left|\sum_m \tilde{h}_0(m-2x){h'}_0(m-2x_d)\right. \\ &\quad \left.\cdot \sum_n \tilde{h}_1(n-2y){h'}_1(n-2y_d)\right| \\ &\le \sum_m \left|\tilde{h}_0(m-2x){h'}_0(m-2x_d)\right| \\ &\quad \cdot \sum_n \left|\tilde{h}_1(n-2y){h'}_1(n-2y_d)\right| \\ &\le \sum_m \frac{\tilde{h}_0^2(m-2x)+{h'}_0^2(m-2x_d)}{2} \\ &\quad \cdot \sum_n \frac{\tilde{h}_1^2(n-2y)+{h'}_1^2(n-2y_d)}{2} \\ &= \frac{1}{4}\left[\sum_m \tilde{h}_0^2(m-2x)+\sum_m {h'}_0^2(m-2x_d)\right] \\ &\quad \cdot \left[\sum_n \tilde{h}_1^2(n-2y)+\sum_n {h'}_1^2(n-2y_d)\right]\end{aligned} \tag{7}$$

Considering the normalization property of the double shift orthogonal wavelet filter coefficients, i.e. $\sum\limits_k h_0(k)^2 = 1$ and $\sum\limits_k h_1(k)^2 = 1$ [15], Eq. (7) can be further derived as

$$|\delta_{x,y}| \le \frac{1}{4}\left[\sum_m \tilde{h}_0^2(m)+\sum_m {h'}_0^2(m)\right]$$

$$
\begin{aligned}
&\cdot \left[\sum_n \tilde{h}_1^2(n) + \sum_n {h'_1}^2(n)\right] \qquad (8)\\
&= \frac{1}{4}\,[1+1]\cdot[1+1]\\
&= 1
\end{aligned}
$$

Based on Eqs. (8) and (6), we can derive that

$$
\begin{aligned}
\tilde{D}_w &\leq \frac{1}{XY}\cdot\sum 1\cdot D_w\\
&= \frac{1}{XY}\cdot XY\cdot D_w\\
&= D_w
\end{aligned}
\tag{9}
$$

Note that the equivalence in Eq. (7) holds if and only if the wavelet filters of the observation domain and the embedding domain are the same. Thus in single dimension watermark scenario, the watermark on observation domain will diffuse both inside and between the wavelet decomposition sub-bands. The watermark energy in the observation domain will decrease, as long as the embedding domain is unknown to the attacker.

3.2 Multiple Dimension Watermark Scenario

In practical scenarios where the length of the watermark is unconstrained, each sample value of the watermark signal will diffuse in the observation domain following the manners described in the previous part. Thus the diffusion of adjacent positions will superimposes with each other. We begin the study of this scenario by further discussing the wavelet diffusivity $\delta_{x,y}$ in Eq. (5).

$$
\begin{aligned}
\delta_{x,y} = &\sum_m \tilde{h}_0(m-2x)h'_0(m-2x_d)\\
&\cdot\sum_n \tilde{h}_1(n-2y)h'_1(n-2y_d)
\end{aligned}
$$

The valid sum range of the two summations is limited by the length of the wavelet filter. That is

$$
\begin{cases} m-2x_d \in [0, H-1] \\ m-2x \in [0, H-1] \end{cases} \tag{10a}
$$

$$
\begin{cases} n-2y_d \in [0, H-1] \\ n-2y \in [0, H-1] \end{cases} \tag{10b}
$$

where H denotes the length of the wavelet filter. We can then derive the diffuse range in x-axis for every single watermarked position in the observation domain. From Eq. (10a) we have

$$
\begin{cases} m \in [2x_d, 2x_d + H - 1] \\ m \in [2x, 2x + H - 1] \end{cases} \tag{11}
$$

The wavelet diffusivity will be zero unless the inequalities below are satisfied.

$$\begin{cases} 2x + H - 1 \geq 2x_d \\ 2x \leq 2x_d + H - 1 \end{cases} \Rightarrow |x - x_d| \leq H/2 - 1 \tag{12}$$

The same result holds for the y-axis.

As shown in Eq. (12), the diffusion range for every single dimension of the watermark in the observation domain is a square area centered on the original embedding position with $H - 1$ as the side length. When the watermark of the position (x_o, y_o) is calculated, all the diffused watermark pieces generated by the embedding position fall into the square area $\mathbf{D} = \{(x, y) \,|\, |x - x_o| \leq H/2 - 1, |y - y_o| \leq H/2 - 1\}$ should be added together as

$$\tilde{W}(x_o, y_o) = \sum_{(x,y) \in \mathbf{D}} W(x, y) \delta_{x,y}(x_o - x, y_o - y) \tag{13}$$

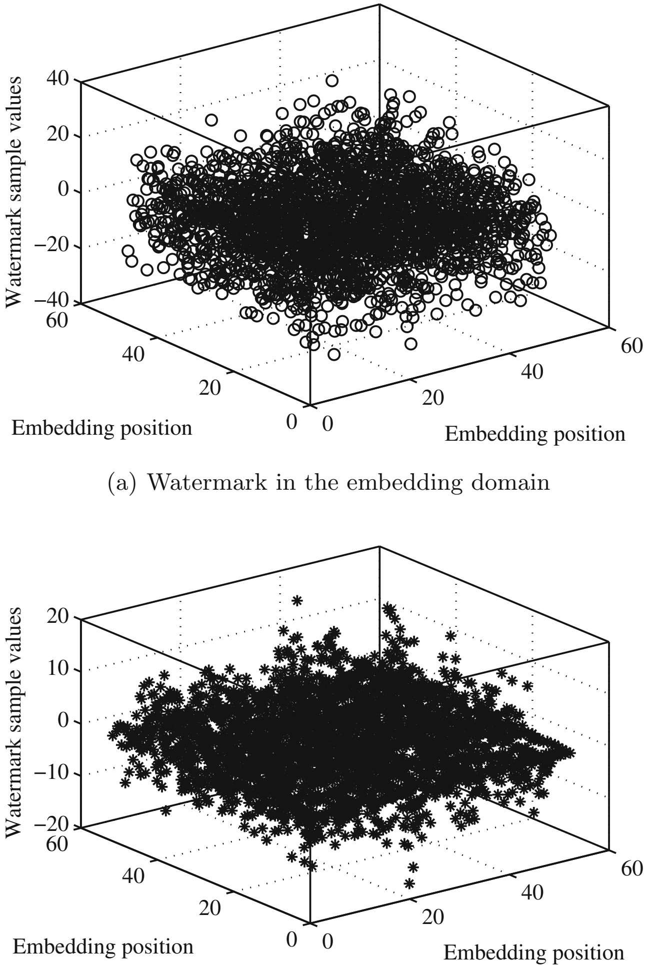
Though it is too complicated to analyze the details of the watermark energy diffusion in Eq. (13), we can roughly assert that the introduced effect to watermark in the observation domain is similar to the low-pass filtering. Figure 1(a) shows the original 50 × 50 watermark signals in normal distribution in the embedding domain, and Fig. 1(b) gives the diffused watermark in an arbitrary observation domain constructed by the key-controlled wavelets.

Since watermarks are always embedded in the mid-frequency region of the cover image to achieve balance between robustness and imperceptiveness, the low-pass filtering effects on the cover signal introduced by key-controlled wavelets are less significant than those on the watermark signal. In other words, the document (cover) energy is basically unchanged while the watermark energy is reduced. Due to this difference, the document-to-watermark energy ratio ξ is increased in the observation domain in most cases, as shown in Fig. 2.

4 Optimal Estimation Performance in the Observation Domain

As the watermark signal in the observation domain changes due to the watermark diffusion effect discussed in the previous section, optimal estimations to the secret key of the spread-spectrum watermarking scheme become less effective. The optimal conditions in the embedding domain relied by those estimators are no longer achievable to the attacker who can only manipulate in the observation domain, and thus the efficiency of the optimal estimations is reduced. We introduce some practical algorithms that are useful to hack the spread-spectrum based watermarking schemes, such as principal component analysis (PCA) [7], blind independent component analysis (blind ICA) [10] and informed ICA [9], and then watch their performance in the observation domain.

ICA and PCA are well-known statistical tools for performing blind source separation (BSS) [14], and they give a method to estimate the watermark signals in the spread-spectrum watermarking schemes. PCA was first applied to the



(a) Watermark in the embedding domain

(b) Watermark in the observation domain

Fig. 1. Watermark diffusion under multiple dimension scenarios

watermarking security problem in [7], and was later refined in [2] by means of a two-step procedure, which involved both PCA and blind ICA. Freire and González [9] developed new estimators that worked in scenarios where PCA and blind ICA failed, thus leading to a wider battery of methods called informed ICA to perform practical security tests.

In the following discussion, we will focus on the ISS watermarking algorithm presented in [12] without loss of generality, since the attacks devised for it are applicable to other existing algorithms. Hence, the embedding function we consider is

$$y = x + \upsilon w - \lambda(x^T w)w \tag{14}$$

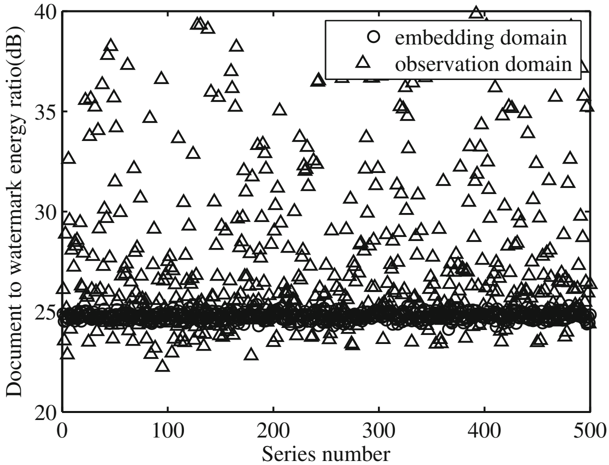


Fig. 2. Document-to-watermark energy ratio on embedding and observation domain

where $0 \leq \lambda \leq 1$ is the host-rejection parameter, and υ is a parameter for fixing the embedding distortion. For fair comparison with other spread spectrum watermarking algorithm, it is suggested that $\upsilon = (n\sigma_w^2 - \lambda^2\sigma_x^2)^{1/2}$ in [12].

To test the performance of the optimal estimators on our watermarking scheme using key-controlled wavelet filters, we generate a set of watermarked gray-scale images by embedding the same watermark signal in the parameterized wavelet decomposition sub-bands of each original image, using the ISS watermarking algorithm with optimal parameter choice described in [12]. The optimal estimators are then applied to these watermarked images to estimate the watermark signal from the embedding domain and the observation domain, respectively.

4.1 PCA Estimator

Let Q denote the covariance matrix of the N observations acquired by the attacker. The eigenvalue decomposition of Q is

$$\begin{array}{l} Q = VDV^T, with \\ V = [w, V_w] \in \mathbf{R}^{n\times n} \\ D = \begin{bmatrix} \upsilon^2 + (1-\lambda)^2\sigma_x^2 & 0 \\ 0 & \sigma_x^2 \cdot I_{n-1} \end{bmatrix} \end{array} \tag{15}$$

where $V_w \in \mathbf{R}^{n\times n-1}$ is a unitary matrix whose columns span the orthogonal complement of the subspace spanned by watermark w. Assuming that there is only one watermark in each cover (Further application of ICA is used to handle the scenario that each cover takes several watermarks), the PCA estimator as follows gives a simple estimation to the watermark w [2].

$$\hat{w} = V[\arg\max_i D_{i,i}] \tag{16}$$

where $V[k]$ denotes the kth column of the matrix V, and $D_{i,i}$ is the ith element in the diagonal of the matrix D.

The PCA estimator in (16) will give a correct estimation when $\hat{w} = w$ holds. From the definition of the matrix V, we can derive that

$$\upsilon^2 + (1-\lambda)^2\sigma_x^2 > \sigma_x^2 \tag{17}$$

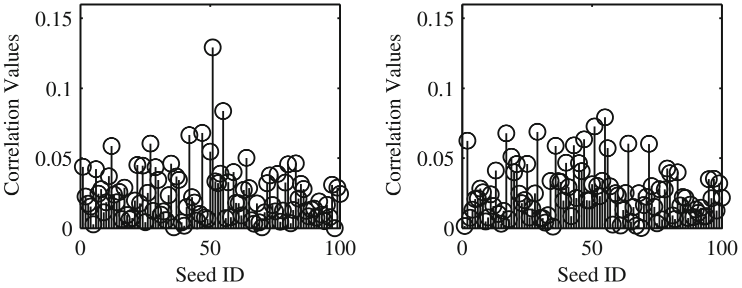
Substituting the optimal value of parameter υ suggested in [12] where $\upsilon = (n\sigma_w^2 - \lambda^2\sigma_x^2)^{1/2}$, we can derive the condition that the PCA estimator being effective as follows.

$$\xi < \frac{n}{2\lambda} \tag{18}$$

As seen in Sect. 3, the document to watermark energy ratio ξ will increase in the observation domain. Hence, the condition in Eq. (18) will become more difficult to meet and thus it makes the PCA estimator less efficient. To analysis the efficiency of the PCA estimator quantitatively, we obtain an estimated watermark signal w_{pca} from the set of gray-scale watermarked images using the PCA estimator, and calculate correlations between w_{pca} and a set of watermarks $\{w_i\}, i = 1, 2, \cdots, 100$ generated by 100 different seeds, including the specific watermark signal used for embedding ($i = 50$). The correlations are defined as

$$corr_i = \frac{\text{cov}(w_{pca}, w_i)}{\sigma_{w_{pca}}\sigma_{w_i}} \tag{19}$$

The experimental results are shown in Fig. 3. When applying the PCA estimator to the embedding domain, as seen from Fig. 3(a), the correlations between the estimated watermark signal w_{pca} and the specific watermark signal used for embedding ($i = 50$) is relatively large compared to other randomly generated watermarks, which means that the embedded watermark signal as well as part of the embedding key is revealed successfully by the PCA estimator. However, the PCA estimator fails to give any valuable information in the case of manipulating on the observation domains, as shown in Fig. 3(b).



(a) Embedding domain (b) Observation domain

Fig. 3. Correlations between the watermark signal estimated by PCA and a set of watermarks generated by 100 different seeds, including the specific watermark signal used for embedding (Seed ID No. 50)

4.2 Blind ICA Estimator

In BSS, the idea behind ICA methods is to optimize a cost function that measures the mutual independence between the separated sources [14]. The ICA estimator used in [2,14] is

$$\hat{w} = \arg\max_{s} J_{ICA}(s) \tag{20}$$

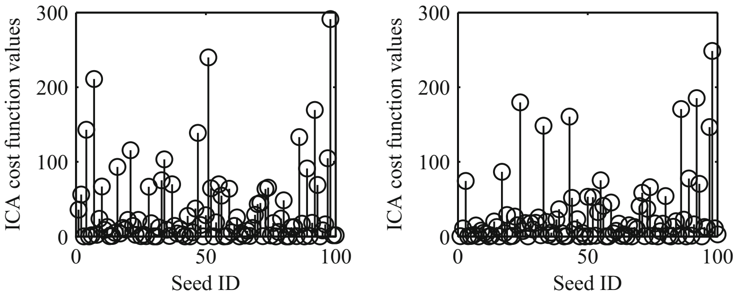
where $J_{ICA}(\cdot)$ is the ICA cost function defined in [13] as

$$J_{ICA}(s) = \left(E[g(\mathbf{Y}^T s)] - E[g(U)]\right)^2 \tag{21}$$

with $U \sim \mathrm{N}(0, \mathrm{var}(\mathbf{Y}^T s))$. The term $\mathbf{Y}^T s$ stands for a binary Gaussian mixture defined in [8] as

$$\begin{aligned} &\mathbf{Y}^T s \sim \tfrac{1}{2}\left(N\left(\upsilon\rho, \sigma_x^2\|\mathbf{t}\|^2\right) + N\left(-\upsilon\rho, \sigma_x^2\|\mathbf{t}\|^2\right)\right), \\ &\text{with}\|\mathbf{t}\|^2 = 1 + \rho^2(\lambda^2 - 2\lambda) \end{aligned} \tag{22}$$

where ρ is the correlation value between the embedded watermark signal w and the estimated ones s obtained by the ICA estimator.



(a) Embedding domain (b) Observation domain

Fig. 4. Cost function values of blind ICA estimators corresponding to a set of watermarks generated by 100 different seeds, including the specific watermark signal used for embedding (Seed ID No.50)

The optimal choice of the so-called "contrast function" is $g(z) = \log(f(z))$ where $f(z)$ is the probability density function of the independent component to be estimated. For an i.i.d. Gaussian host, the optimal ICA cost function results in [8] is

$$\begin{aligned} J_{ICA}(s, a) = \Big(& E[\log\cosh(a \cdot \mathbf{Y}^T s)] \\ & -E[\log\cosh(a \cdot U)]\Big)^2 \end{aligned} \tag{23}$$

where the parameter a can be fixed by the attacker.

To analyze the performance of the blind ICA estimator, we generate a set of watermarks $\{w_i\}, i = 1, 2, \cdots, 100$ using 100 different seeds, including the specific watermark signal used for embedding ($i = 50$). The ICA cost function values corresponding to each watermark are calculated and shown in Fig. 4. In the embedding domain, the 'correct' watermark signal results in a second largest cost function value (Seed ID No.50), which means that the embedded watermark signal as well as the embedding key is partly revealed. In the observation domain, the blind ICA estimator fails as the PCA estimator do, due to the watermark diffusion effect introduced by key-controlled wavelet.

4.3 Informed ICA Estimator

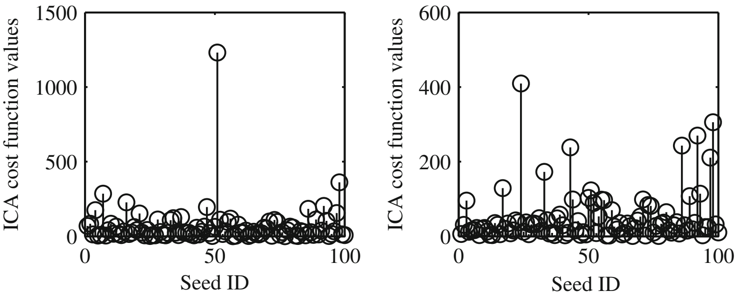
The performance of the blind ICA estimator can be enhanced by introducing the "informed ICA" method [9]. The basic idea of the informed ICA is estimating the cover energy σ_x^2 from the observations held by the attacker, and taking advantage of these estimations in the construction of the cost function in the blind ICA.

The estimation of σ_x^2 is computed from the observations obtained by attackers as

$$\hat{\sigma}_x^2 = \left(\frac{1}{n} tr(\mathbf{Q})\right)^{\frac{1}{2}} = \left(\frac{1}{n} \sum_{i=1}^{n} D(i,i)\right)^{\frac{1}{2}} \tag{24}$$

where $tr(\mathbf{Q})$ is the covariance of the observations, and $D(i,i)$ are the diagonal elements of the matrix of eigenvalues defined in (15). We denote $J_{ICA}^{\inf}(s,a)$ as the cost function of informed ICA estimator. The expression of $J_{ICA}^{\inf}(s,a)$ is the same as (23), with the only difference that $U \sim \mathrm{N}(0, \hat{\sigma}_x^2)$.

We test the performance of the informed ICA estimators by an experiment similar to the previous one, and the result is shown in Fig. 5. As can be seen, although the performance of the informed ICA estimator is better than blind ICA in the embedding domain, it still fails to give any valuable information about the embedded watermark signals and the secret key in the observation domain.



(a) Embedding domain (b) Observation domain

Fig. 5. Cost function values of informed ICA estimators

5 Parameterized Wavelet Filter

Zou and Tewfik [19] proposed a principle to achieve wavelet parameterizations by constructing wavelet filter frequency response. Based on this principle, Schneid and Pittner [17] designed an iterative form implementation to calculate the parameterized wavelet filter coefficients. Dietl *et al.* [6] applied this implementation into spread-spectrum watermarking, and designed a parameterized wavelet domain watermarking scheme, without detailed analysis to the security introduced by the parameterized wavelet filters. The main drawback of this implementation is the inconvenient iterative calculation process. To obtain parameterized wavelet filters of length N, one has to calculate a series of parameterized wavelet filters with length from 2 to $N-1$ consequently. Regensburger [16] presented another method to calculate parameterized wavelet filters by introducing discrete moment as the parameters. Gröbner basis is used to gain analytical resolutions to the parameterized equations. Though the discrete moment is unnecessary to the scheme of spread-spectrum watermarking, it suggests a better way to calculate parameterized wavelet filters by solving nonlinear equations than the iteratively process in [6].

In this paper, we follow the method in [16], but replace the unnecessary discrete moment with arbitrary parameters. In this way, the constraints contained in the nonlinear equations are reduced as much as possible. Therefore, the solution space covers more usable wavelets than the method in [6], which leads to a bigger key space of the watermarking scheme.

Let $\{h_0(k), h_1(k)\}$ and $\{{h_0}'(k), {h_1}'(k)\}$ denote the wavelet decomposition and reconstruction filter coefficients corresponding to the embedding domain, respectively, and N the length of the filters. The relationships between these four filters are shown as follows.

$$h_1(k) = (-1)^{k-1} h_0(N-k-1) \tag{25a}$$

$${h_0}'(k) = h_0(N-k-1) \tag{25b}$$

$${h_1}'(k) = (-1)^k h_0(k) \tag{25c}$$

In other words, solving only one of these four filters will then determine the others and thus we can achieve a valid construction of the parameterized wavelet domain.

We choose $h_0(k)$ for further discussion without loss of generality, describing its properties that the double shift orthogonal wavelet should satisfy [5] in the form of nonlinear equations.

$$\text{Normalization:} \sum_k h_0(k) = \sqrt{2} \tag{26a}$$

$$\begin{aligned}&\text{Double Shift Orthogonality}:\\ &\sum_k h_0(k) h_0(k-2n) = 0,\\ &n \neq 0, n \in Z, 2n \leq N\end{aligned} \tag{26b}$$

$$\text{Low pass:} \sum_k (-1)^k h_0(k) = 0 \tag{26c}$$

As can be seen from Eq. (26), the number of equations is $1 + (N/2) - 1 + 1 = (N/2) + 1$. When the filter length is N, we need $(N/2) - 1$ more equations to solve $h_0(k)$. If we fill in the bank with K-regular conditions below, we obtain the standard Daubechies wavelets.

$$\sum_k k^K (-1)^k h_0(k) = 0, K = 1, 2, \cdots, N/2 - 1 \tag{27}$$

The K-regular conditions in Eq. (27) guarantee that the wavelets are K-level smooth, which is unnecessary for the purpose of the watermarking embedding and extracting. So we can replace some or all of the K-regular conditions by arbitrarily parameterized equations of $h_0(k)$, thus obtaining parameterized wavelets instead of the standard ones. The complete equations of parameterized wavelet filters are shown as follows.

$$\sum_k h_0(k) = \sqrt{2} \tag{28a}$$

$$\begin{gathered}\sum_k h_0(k) h_0(k - 2n) = 0, \\ n \neq 0, n \in Z, 2n \leq N\end{gathered} \tag{28b}$$

$$\sum_k (-1)^k h_0(k) = 0 \tag{28c}$$

$$\begin{gathered}\sum_k (-1)^k \cdot k^K \cdot h_0(k) = 0, \\ K = 1, 2, \cdots, M_0\end{gathered} \tag{28d}$$

$$\begin{gathered}h_0(k_i) = m_i, \text{in which} \\ 0 \leq k_i \leq N - 1, i = 1, 2, \cdots, M_1, \\ M_0 + M_1 = \tfrac{N}{2} - 1\end{gathered} \tag{28e}$$

In this paper, we provide two different methods to solve the nonlinear multivariate equations in Eq. (28) efficiently: the numerical methods using Newton iteration and the analytical methods using Gröbner bases.

5.1 Numerical Method Using Newton Iteration

The numerical solutions of Eq. (28e) can be obtained by Newton iteration. We firstly rewrite equations in (28) as $F(h_0) = 0$, where $F : D \subset R^N \to R^N$, $F = (f_0(h_0), f_1(h_0), \cdots, f_{N-1}(h_0))^T$, $h_0 = (h_0(0), h_0(1), h_0(2), \cdots, h_0(N-1))^T$.

Then the iterative formula can be written as

$$\begin{gathered}h_0^{k+1} = h_0^k - F'(h_0^k)^{-1} F(h_0^k), k = 0, 1, 2, \cdots \\ F'(h_0) = \begin{bmatrix} \frac{\partial f_0}{\partial h_0(0)} & \frac{\partial f_0}{\partial h_0(1)} & \cdots & \frac{\partial f_0}{\partial h_0(N-1)} \\ \frac{\partial f_1}{\partial h_0(0)} & \frac{\partial f_1}{\partial h_0(1)} & \cdots & \frac{\partial f_1}{\partial h_0(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N-1}}{\partial h_0(0)} & \frac{\partial f_{N-1}}{\partial h_0(1)} & \cdots & \frac{\partial f_{N-1}}{\partial h_0(N-1)} \end{bmatrix}\end{gathered} \tag{29}$$

Given a set of parameters $\{m_i\}$, the parameterized wavelet filter coefficients can be calculated from an iteration process described by Eq. (29).

5.2 Analytical Method Using Gröbner Bases

The method of Gröbner bases is an efficient way to solve problems of polynomial ideals in an algorithmic or computational fashion. It is also used in several powerful computer algebra systems to study specific polynomial ideals that arise in applications [3]. In this paper, we use Gröbner bases to solve the nonlinear multivariate equations in Eq. (28). The process is similar to that in [16].

Step 1. Eliminate the variables in the linear equations in Eq. (28) and substitute the solutions into the nonlinear ones. Thus the number of the variables is reduced to $(N/2) - 1$.

Step 2. Calculate the reduced Gröbner basis in *lexicograghical* order [3] to the equations obtained in Step 1. The resulting reduced Gröbner basis contains at least one equation with only a single variable.

Step 3. Solve the equations with a single variable, and then substitute back into other equations to get complete solutions for Eq. (28).

Here is an example of the parameterized wavelet filter coefficients with length 6. We have six equations, consisting of two nonlinear ones and four linear ones, with m and n denoting the two arbitrary parameters, respectively. The multivariate nonlinear equations are as shown in Eq. (30), and the solutions are given in Eq. (31).

$$\begin{aligned}
&h_0(0) + h_0(1) + h_0(2)\\
&+h_0(3) + h_0(4) + h_0(5) = \sqrt{2}\\
&h_0(0) - h_0(1) + h_0(2) - h_0(3) + h_0(4) - h_0(5) = 0\\
&h_0(2) = m\\
&h_0(5) = n\\
&h_0(0)h_0(2) + h_0(1)h_0(3)\\
&+h_0(2)h_0(4) + h_0(3)h_0(5) = 0\\
&h_0(0)h_0(4) + h_0(1)h_0(5) = 0
\end{aligned} \tag{30}$$

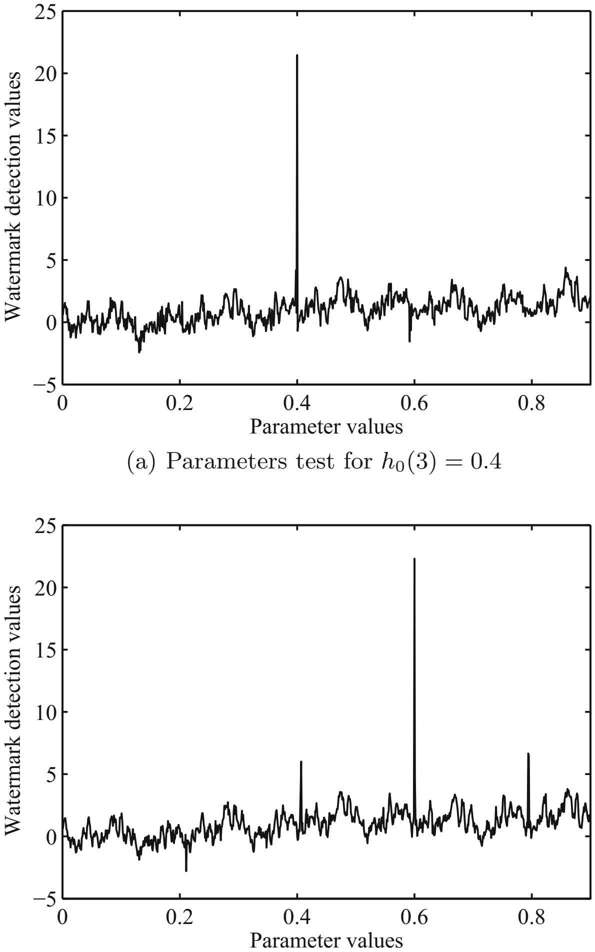
$$\begin{aligned}
&h_0(0) = -\frac{1}{2}m + \frac{\sqrt{2}}{4} + \frac{1}{4}\cdot\Bigg(4m^2 - 4\sqrt{2}m + 2\\
&- 16n^2 + 4\sqrt{2}n + 4n\sqrt{2 - 16m^2 + 8\sqrt{2}m}\Bigg)^{\frac{1}{2}}\\
&h_0(1) = \frac{\sqrt{2}}{4} - n + \frac{1}{4}\sqrt{2 - 16m^2 + 8m\sqrt{2}}\\
&h_0(2) = m\\
&h_0(3) = n^2 - (\frac{\sqrt{2}}{4} + 1)n
\end{aligned} \tag{31}$$

$$- \frac{1}{4}\sqrt{2 - 16m^2 + 8m\sqrt{2}} + \frac{\sqrt{2}}{2}$$

$$h_0(4) = -\frac{1}{2}m + \frac{\sqrt{2}}{4} - \frac{1}{4} \cdot \Big(4m^2 - 4\sqrt{2}m + 2$$

$$- 16n^2 + 4\sqrt{2}n + 4n\sqrt{2 - 16m^2 + 8\sqrt{2}m}\Big)^{\frac{1}{2}}$$

$$h_0(5) = n$$



(a) Parameters test for $h_0(3) = 0.4$

(b) Parameters test for $h_0(2) = 0.6$

Fig. 6. Watermark can only be extracted correctly with matching parameters from 1000 parameters on each free degree

6 Key Space Estimation

The key space of the key-controlled wavelet is crucial to the spread-spectrum watermarking schemes. As seen from the previous section, the nonlinear multivariate equations used to solve the wavelet filter coefficients in Eq. (28) contain no additional restrictive conditions, except for the constrains in Eq. (26) which guarantees the wavelet decomposition and reconstruction filters to maintain their basic properties, i.e. normalization, double shift orthogonality and low pass characteristic. Furthermore, the values of the arbitrary parameters in Eq. (28) vary consequently and thus cover all the coefficient values of usable wavelet filters. Therefore, the key space of the proposed method is extended as much as possible.

As the parameters used in Eq. (28) are consecutively distributed in [0, 1] uniformly, it is hard to analyze the key space theoretically, so we design an experiment to estimate the key space roughly instead. Firstly, we embed spread spectrum watermark in the LH sub-band of a key-controlled wavelet decomposition with filter length $N = 6$, using two arbitrary parameters as $h_0(2) = 0.6$ and $h_0(3) = 0.4$. Secondly, we try to extract watermark in the observation domains determined by different parameters. We have tested 1000 uniformly distributed values from [0, 1] for each parameter, holding the other one unchanged. Figure 6 shows that the watermark can only be retrieved correctly with matching parameters on each free degree of parameters. Thus, the result suggests a key space approximate to $1000^2/2 = 5 \times 10^5$ roughly. The maximum free degrees of parameters is $(N/2) - 1$ due to Eq. (28), so the estimation of the key space to the key-controlled wavelet approximates to

$$\frac{1000^{N/2-1}}{N/2-1} \tag{32}$$

7 Conclusions

In this paper, we present a new method to reduce mutual information by embedding watermark in key controlled wavelet domain. Theoretical analysis shows the watermark signal diffusion and energy reduction effect on attacker's observation domain, thus leading to a higher document-to-watermark energy ratio and better watermark security. Since the optimal conditions for estimating watermarks in the embedding domain no longer meet in the observation domain, the performance of the estimators to the secret key of the spread-spectrum watermark scheme is reduced. We also provide two different methods to efficiently solve the nonlinear multivariate equations to construct key controlled wavelet filters: the numerical methods using Newton iteration and the analytical methods using Gröbner bases. Experimental results show that these two methods provide adequate key space for the watermark system.

Acknowledgments. This work was supported by the NSF of China under 61170281, NSF of Beijing under 4112063, Strategic and Pilot Project of CAS under XDA06030601, and the Project of IIE, CAS, under Y1Z0041101 and Y1Z0051101.

References

1. Barni, M., Bartolini, F., Furon, T.: A general framework for robust watermarking security. Sig. Process. **83**(10), 2069–2084 (2003)
2. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: theory and practice. IEEE Trans. Sig. Process. **53**(10, pt. 2), 3976–3987 (2005)
3. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, New York (1997)
4. Cox, I.J., Killian, J., Leighton, T., Shamoon, T.: Secure spread spectrum watermarking for images, audio and video. IEEE Trans. Image Process. **6**(12), 1673–1687 (1997)
5. Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Series in Applied Mathematics, vol. 61. SIAM Press, Philadelphia (1992)
6. Dietl, W., Meerwald, P., Uhl, A.: Protection of wavelet-based watermarking systems using filter parametrization. Sig. Process. **83**, 2095–2116 (2003)
7. Doerr, G., Dugelay, J.L.: Danger of low-dimensional watermarking subspaces. In: Proceedings Iof the EEE International Conference on Acoustics, Speech, Signal Processing, Montreal, QC, Canada, vol. 3, pp. 93–96 (2004)
8. Freire, L.P.: Digital watermarking security. Ph.D. dissertation, Department of Signal Theory and Communications, University of Vigo, Vigo, Spain (2008)
9. Freire, L.P., González, F.P.: Spread-spectrum watermarking security. IEEE Trans. Inf. Forensics Secur. **4**(1), 2–24 (2009)
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. **10**(3), 626–634 (1999)
11. Kalker, T.: Considerations on watermarking security, In: Proceedings of the IEEE International Workshop on Multimedia Signal Processing, Cannes, France, pp. 201–206 (2001)
12. Malvar, H.S., Florêncio, D.A.F.: Improved spread spectrum: a new modulation technique for robust watermarking. IEEE Trans. Sig. Process. **51**(4), 898–905 (2003)
13. Moulin, P., Ivanovic, A.: The zero-rate spread-spectrum watermarking game. IEEE Trans. Sig. Process. **51**(4), 1098–1117 (2003)
14. Oja, E., Hyvärinen, A., Karhunen, J.: Independent Component Analysis. Adaptive and Learning Systems for Signal Processing, Communications and Control. Wiley, New York (2001)
15. Phillips, W.J.: Wavelets and Filter Banks Course Note (2003). http://www.engmath.dal.ca/courses/engm6610/notes/notes.html
16. Regensburger, G.: Parametrizing compactly supported orthonormal wavelets by discrete moments. Applicable Algebra in Engineering, Communication and Computing (2007). http://www.ricam.oeaw.ac.at/people/page/regensburger/papers/regensburger06.pdf
17. Schneid, J., Pittner, S.: On the parametrization of the coeffcients of dilation equations for compactly supported wavelets. Computing **51**, 165–173 (1993)
18. Wang, X.Y., Wu, J.: A feature-based robust digital image watermarking against desynchronization attacks. Int. J. Autom. Comput. **4**(4), 428–432 (2007). Springer
19. Zou, H., Tewfik, A.H.: Parametrization of compactly supported orthonormal wavelets. IEEE Trans. Sig. Process. **41**, 1423–1431 (1993)

Wave Atom-Based Perceptual Image Hashing Against Content-Preserving and Content-Altering Attacks

Fang Liu(✉) and Lee-Ming Cheng

Department of Electronic Engineering, City University of Hong Kong,
83 Tat Chee Avenue, Kowloon Tong, Hong Kong
f.liu@my.cityu.edu.hk, itlcheng@cityu.edu.hk

Abstract. This paper presents a perceptual image hashing algorithm based on wave atom transform, which can distinguish maliciously attacked images from content-preserving ones. Wave atoms are employed due to their significantly sparser expansion and better feature extraction capability than traditional transforms, like discrete cosine transform (DCT) and discrete wavelet transform (DWT). Thus, it is expected to show better performance in image hashing. Moreover, a preprocessing method based on Fourier-Mellin transform is employed to keep the proposed scheme against geometric attacks. In addition, a randomized pixel modulation based on RC4 is performed to ensure the security. According to the experimental results, the proposed scheme is sensitive to content-altering attacks with the resiliency of content-preserving operations, including image compression, noising, filtering, and rotation. Moreover, compared with some other image hashing algorithms, the proposed approach also achieves better performance even in the aspect of robustness, which is more important in some image hashing application, for example image database retrieval or digital watermarking.

Keywords: Image hashing · Authentication · Robustness · Wave atom transform

1 Introduction

Nowadays, the vigorous popularity of image processing techniques has resulted in an explosive growth of image illegal use, such as image forgery and unauthorized utilization. A traditional solution to deal with data illegal issues is to generate a hash using some standard cryptographic hash functions, like MD5 and SHA-1, and form a digital signature by some public key encryption algorithms [1]. This kind of hash functions achieves high sensitivity when applied to data authentication, where even one bit change in the message will result in significant changes in the hash value. Unfortunately, it is the sensitivity that makes these functions not applicable to digital images. Since images will also be considered as the identical one even if they have undergone some content-preserving manipulations, such as image compression, noising, and filtering.

Perceptual image hashing has been therefore presented to provide the content-based authentication, copyright verification and some other protections for digital images.

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 21–37, 2015.
DOI: 10.1007/978-3-662-46739-8_2

The core idea of perceptual image hashing is to construct the hash by extracting characteristics of human perception in images, and use this constructed hash to authenticate or retrieve an image without considering the various variables or formats of this image. This kind of schemes takes the changes of human perception into account and ignores the perceptually unnoticeable changes. They have drawn a lot of attention owing to the outstanding performance against common image processing operations. There are two important performance requirements for strong image hashing schemes, namely robustness and fragility, which influence each other mutually. Robustness is the degree to which an image hashing scheme is invariant to perceptually identical images, while fragility is the degree to which the scheme distinguishes the perceptually different images from the original ones. Consequence, it is expected that images which look like the same or very similar should have the same or very similar hash codes, while images which differ from each other should have distinct hash codes.

At present, many research studies have been carried out on perceptual image hashing based on various transformations, such as DWT [2–7], DCT [8, 9], Radon transform (RT) [10–12], discrete Fourier transform (DFT) [13–15], and others.

In 1998, a scale interaction model is used in wavelet domain to extract visually salient image feature points for image authentication [2]. Venkatesan et al. also extracted the invariant statistics characteristics of wavelet coefficients to construct robust hash in 2000 [3]. In the same year, an invariant relation of the parent and child pair nodes located at multiple scales in DWT decomposition is explored for hash generation as well [4]. Monga and Evans also exploited the features derived from the end-stop wavelet coefficients to detect visually significant feature points [5, 6]. Recently, Ahmed et al. [7] proposed a secure image hashing using both DWT and SHA-1.

Fridrich and Goljan [8] also took the advantage of that low frequency coefficients in DCT can represent the coarse information of a whole image and proposed a robust hash for digital watermarking. Lin and Chang [9] found a desired relation to construct their robust hash. This relation is based on the fact that DCT coefficients in the same position of different blocks are invariant before and after JPEG compression.

Since the Radon transform is also robust against image processing basic attacks and strong attacks, Lefebvre [10] first applied it to image hash. Further research has been taken by Roover [11] based on radial projection of the image pixels and is denoted the Radial hASHing (RASH) algorithm. A new approach is also proposed for image fingerprinting using the Radon transform to make the fingerprint robust against affine transformations by Seo et al. in [12].

There are lots of hashing schemes based on DFT as well. In [13], Swaminathan et al. developed an algorithm to generate a hash based on Fourier transform features and controlled randomization. In [14], a print–scan resistant image hashing algorithm is proposed based on the RT domain combining with DWT and DFT. In [15], moment features are extracted from the RT domain and the significant DFT coefficients of the moments are used to produce hashes.

Besides the transformations employed above, the matrix factorization is also prevalent in the field of perceptual image hashing [16–18]. Kozat et al. [16] proposed the hashing scheme based on matrix invariants as embodied by Singular Value Decomposition (SVD) and viewed images as well as attacks as a sequence of linear operators. Monga and Mihcak [17] first employed the low-rank decomposition of nonnegative matrix factorization (NMF) with NMF of pseudo-randomly selected subimages to derive hashes. Tang et al. [18] explored the invariance relation existing in the NMF for constructing robust image hashes too. Recently, Tang et al. also proposed an efficient image hashing with a ring partition and a NMF, which is claimed with both the rotation robustness and good discriminative capability.

Moreover, there are many other significant methods for perceptual hashing as well. For instance, Lv and Wang [19] proposed a robust SIFT-Harris detector for selecting the most stable SIFT key points. The image hashes are then generated by embedding the detected local features into shape-contexts-based descriptors. And SIFT features are also used in the work of forensic hashing [20] to estimate geometric transform, while the block-based features are employed to detect and localize the image tampering. Khelifi and Jiang [21] proposed a robust and secure hash algorithm based on virtual watermarking detection which can detect the malicious changes in relatively large areas. Zhao et al. [22] employed Zernike moments representing the luminance and chrominance of an image as the global features, and position and texture information of salient regions as the local features to form their hashes.

However, compromise has always been made between robustness and fragility among those hashing schemes. Fortunately, it is expected that wave atom transform can achieve better performance than these conventional transforms in image hashing. Demanet and Ying introduced wave atom transform in 2007 [23], which are a recent addition to the repertoire of mathematical transforms of computational harmonic analysis. They have been proved to have a dramatically sparser expansion of wave equations than traditional transformations, which come either as an orthonormal basis or a tight frame of directional wave packets, and are particularly suitable for representing oscillatory patterns in images. Motivated by these attractive characteristics, this paper demonstrated the feasibility of wave atom transform applied in perceptual hashing based on our previous work [24]. In addition, a preprocessing image authentication method is proposed to further ensure the proposed scheme against geometric attacks using Fourier-Mellin transform.

The rest of this paper is structured as follows. Section 2 shows a brief overview and implementation of wave atom transform. The proposed algorithm is described in Sect. 3. The experimental analysis is presented in Sect. 4, whereas the conclusions are giving in Sect. 5.

2 Wave Atom Transform

Demanet and Ying introduced wave atoms as a variant of 2-D wavelet packets in 2007 [23], which can adapt to arbitrary local directions of a pattern, and can also sparsely represent anisotropic patterns aligned with the axes. Oscillatory functions and oriented textures in wave atoms have been proved to have a dramatically sparser expansion compared to some other fixed standard representations like Gabor filters, wavelets, and curvelets. Wave atoms interpolate precisely between Gabor atoms [25] and directional wavelets [26]. The period of oscillations of each wave packet is related to the size of essential support via parabolic scaling, i.e. $wavelength \sim (diameter)^2$.

Wave atoms can be constructed from tensor products of adequately chosen 1-D wave packets. Let $\psi^j_{m,n}(x)$ represent a 1-D wave packet, where $j, m \geq 0$, and $n \in Z$, centered in space around $x_{j,n} = 2^{-j}n$ and centered in frequency around $\pm w_{j,m} = \pm\pi 2^j m$ respectively, with $C_1 2^j \leq m \leq C_2 2^j$. The basis function is defined combining dyadic scaled and translated versions of $\hat{\psi}^0_m$ in the frequency domain as the following

$$\psi^j_{m,n}(x) = \psi^j_m\left(x - 2^{-j}n\right) = 2^{j/2}\psi^0_m\left(2^j x - n\right) \tag{1}$$

where

$$\psi^0_m(w) = e^{-iw/2}\left[e^{i\alpha_m} g(\varepsilon_{\mathrm{m}}(w - \pi(m+1/2))) + e^{-i\alpha_m} g(\varepsilon_{\mathrm{m+1}}(w + \pi(m+1/2)))\right] \tag{2}$$

with $\alpha_m = \pi/2(m + 1/2)$, $\varepsilon_{\mathrm{m}} = (-1)^m$ and g a real-value C^∞ bump function is compactly supported on an interval of length 2π such that $\sum_m |\psi^0_m(w)|^2 = 1$.

For each wave $w_{j,m}$ at scale 2^{-j}, the coefficient $c_{j,m,n}$ is treated as a decimated convolution.

$$c_{j,m,n} = \int \psi^j_m(x - 2^{-j}n)u(x)dx = \frac{1}{2\pi}\int e^{i2^{-j}nw}\overline{\hat{\psi}^J_m(w)}\hat{u}(w)dw. \tag{3}$$

Discretize the sample u at $x_k = kh$, $h = 1/N$, $k=1,\cdots, N$, and the discrete coefficients $c^D_{j,m,n}$ are calculated by utilizing a reduced inverse FFT inside an interval of size $2^{j+1}\pi$, centered around the origin

$$c^D_{j,m,n} = \sum\nolimits_{k=2\pi\left(-2^j/2+1:1\ :\ 2^j/2\right)} e^{i2^{-jnk}} \sum\nolimits_{p\in 2\pi Z} \overline{\hat{\psi}^J_m(k+2^J p)}\hat{u}(k+2^j p). \tag{4}$$

There is a simple wrapping technique for implementation of 1-D wave packet as follows:

(1) Perform a FFT of size N of the samples $u(k)$.
(2) For each pair (j,m), wrap the product $\overline{\hat{\psi}^J_m}\hat{u}$ by periodicity inside the interval $[-2^j\pi, 2^j\pi]$ and perform an inverse FFT of size 2^j to obtain $c^D_{j,m,n}$.
(3) Repeat step (2) for all pairs (j,m).

The 2-D orthonormal basis functions with four bumps are formed by individually utilizing products of 1-D wave packets in the frequency plane. Let $\mu = (j, m_1, m_2, n_1, n_2)$, the basis function is modified as

$$\phi^+_\mu(x_1, x_2) = \psi^j_{m1}(x_1 - 2^{-j}n_1)\psi^j_{m2}(x_2 - 2^{-j}n_2). \tag{5}$$

A dual orthonormal basis can be established from the "Hilbert-transformed" wavelet packets as

$$\phi_{\mu}^{-}(x_1, x_2) = H\psi_{m1}^{j}(x_1 - 2^{-j}n_1)H\psi_{m2}^{j}(x_2 - 2^{-j}n_2). \tag{6}$$

By combining Eqs. (5) and (6), basis functions with two bumps are provided in the frequency domain, and directional wave packets oscillate in one single direction

$$\phi_{u}^{(1)} = \left(\phi_{u}^{+} + \phi_{u}^{-}\right)/2,\ \phi_{u}^{(2)} = \left(\phi_{u}^{+} - \phi_{u}^{-}\right)/2 \tag{7}$$

where $\phi_{u}^{(1)}$and $\phi_{u}^{(2)}$ are denoted as ϕ_u together which form the wave atoms frame.

3 Proposed Algorithm

In this section, a compelling image hashing scheme is proposed based on wave atom transform, which satisfies both robustness and fragility for image hashing against content-preserving and content-altering attacks. Some initial work is presented in [24]. Wave atom transform is used since it can represent the image features better than other transforms. It has also been observed that the use of wave atom coefficients in the third scale band gives good robustness to common signal processing attacks except rotation manipulation, since rotation manipulation indeed changes an image perceptually to a certain extent. Consequently, a rotation-invariant preprocessing enhancement is presented in image authentication module to further ensure the robustness of our proposed algorithm against geometric attack. Besides, a randomized pixel modulation (RPM) [7] is used to enhance the security of our proposed scheme. Before applying wave atom transform, all pixels in the spatial domain are randomly modulated using a pseudo-random secret key stream based on RC4. The details of the proposed algorithm are shown below.

3.1 Hash Generation Module

The detailed procedures of hash generation module shown in Fig. 1 are described as follows:

(1) Let I denote the original input image of size $N \times N$.

(2) Then, the RPM [7] is employed to I for the purpose of security. The details are described as follows:
Firstly, divide I into a number of non-overlapping blocks with dimension $J \times J$ for each block. Thus, N^2/J^2 blocks are generated. Denote P_i as the i-th block, where $i = 0, \cdots, N^2/J^2 - 1$. And J is set to 16 in our implementation.

Secondly, the RC4 algorithm governed by a secret key K_1 is employed to generate pseudo-random numbers for each block i. By sorting a $J \times J$ generated number sequence in descending order, the indexes for all these numbers in the original sequence are marked. Let $P_i(x, y)$ represent the gray value of the pixel at spatial domain location (x, y) in block P_i, and further reshaped to one dimension as $S_i(m)$ where $m = x + (y - 1) \times J$. Then $S_i(m)$ is permutated according to the marked indexes and denoted as $S_i'(m)$.

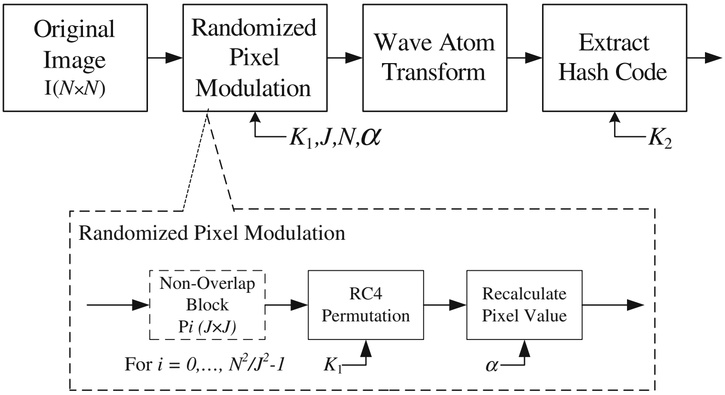


Fig. 1. Hash generation module

Finally, in order to make the image hash code dependent on the secret key, every pixel in each block is modulated as the following

$$P_i'(x,y) = P_i(x,y) + \alpha \times S_i'(m) \tag{8}$$

where $P_i'(x,y)$ is the new pixel value and $1 \le x,y \le J$ and $m = x + (y - 1) \times J$.

(3) By performing wave atom transform to the image of new pixel values, several scale bands could be obtained, which has different frequencies. For each scale band, there are a number of sub-blocks which consists of different numbers of wave atom coefficients. Among these scale bands, the third scale band is selected to compute the hash code, since middle frequency scale coefficients are more robust than high frequency ones, and also more fragile than low frequency ones [26]. Since the energy of wave atom coefficients captures most information of main image features, the intermediate hash could be computed by exploring the mutual relationship of these sub-blocks.

(4) Denote $C(j, m_1, m_2, n_1, n_2)$ as wave atom coefficients, where j is the scale, and m_1, m_2, n_1, n_2 represents the phase. Assign an index i for each sub-block in the third scale band. Let E_i be the energy of the i-th block. For all non-empty blocks in the third scale band

$$E_i = \sum\nolimits_{q=1}^{l_2} \sum\nolimits_{p=1}^{l_1} \mathrm{C}(j, m_1, m_2, p, q)^2 \tag{9}$$

where $l_1$and $l_2$represent the length and width of the sub-block respectively.

To ensure that the extracted features used to generate the hash code cannot be exposed, a random sequence generated by RC4 is XORed with E_i to generate the new sequence E_i'. Let the total number of non-empty blocks in the third scale band be t. The energy difference between each two blocks is used to generate one hash bit. The intermediate hash can be calculated using this equation:

$$h^{(i)} = \begin{cases} 1, & \mathit{if}\, E'_i > E'_{i+1} \\ 0, & \mathit{Otherwise} \end{cases} \tag{10}$$

where $i \in [1, \cdots, t-1]$.

To increase the security of hash code, a pseudo-random sequence generated by the secret key K_2 is employed to XOR the intermediate hash h based on RC4 algorithm and generates the final hash H.

3.2 Image Authentication Module

To keep the robustness of the proposed scheme against common content-preserving attacks and geometric attacks, a rotation-invariant preprocessing is first presented in this section using Fourier-Mellin transform. Then the proposed authentication procedures are presented.

3.2.1 Rotation-Invariant Preprocessing

It is well known that the Fourier-Mellin transform is invariant to rotation, translation and scaling manipulations, which is especially useful for image recognition [27]. In this paper, to ensure the proposed scheme robust to geometric attacks, the rotation-invariant property of Fourier-Mellin transform is employed in the proposed image hashing scheme.

Let $I_1(x, y)$ denote an image, and $I_2(x, y)$ is a translated and rotated replica of $I_1(x, y)$ with translation (x_0, y_0) and rotation angle φ_0, then

$$I_2(x, y) = I_1(xcos\varphi_0 + ysin\varphi_0 - x_0, -xsin\varphi_0 + yco\varphi_0 - y_0). \tag{11}$$

According to Fourier Transform and its properties, transforms of I_1 and I_2 are related by

$$F_2(u, v) = e^{-j2\pi(ux_0+vy_0)} F_1(ucos\varphi_0 + vsin\varphi_0, -usin\varphi_0 + vcos\varphi_0). \tag{12}$$

Denote M_1 and M_2 as the magnitudes of F_1 and F_2, thus we have

$$M_2(u, v) = M_1(ucos\varphi_0 + vsin\varphi_0, -usin\varphi_0 + vcos\varphi_0). \tag{13}$$

Using the polar coordinates, Eq. (13) can be rewritten as

$$M_2(\rho, \varphi) = M_1(\rho, \varphi - \varphi_0). \tag{14}$$

Here, it is evident that there is only a same rotation which results from the image domain. The angle of rotation φ_o can be calculated using phase correlation.

Consequently, this preprocessing is provided using Fourier-Mellin transform which can estimate the rotated angle. And if the estimated angle is not zero, the translated and rotated image is rotated back. Otherwise, the image is not preprocessed.

3.2.2 Authentication Procedure

The image authentication module as illustrated in Fig. 2 is then employed to authenticate the received image. Using the same parameters N, K_1, K_2, α and J, system can calculate the hash code of the received image and make the comparison with the original hash code in terms of normalized Hamming distance. The image authentication procedures are described as follows:

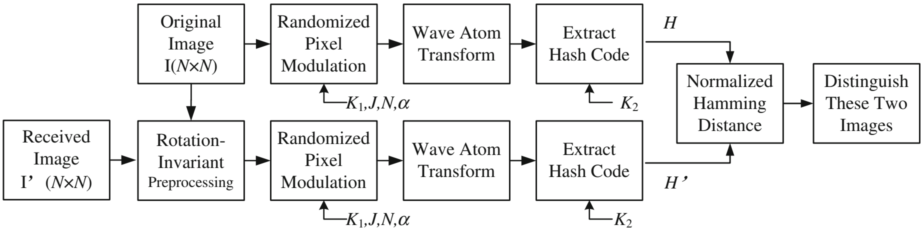

Original Image I(N×N)
Randomized Pixel Modulation
Wave Atom Transform
Extract Hash Code
H
K1,J,N,α
K2
Normalized Hamming Distance
Distinguish These Two Images
Received Image I’ (N×N)
Rotation-Invariant Preprocessing
Randomized Pixel Modulation
Wave Atom Transform
Extract Hash Code
H’
K1,J,N,α
K2

Fig. 2. Image authentication module

(1) The received image goes through the rotation-invariant preprocessing as described in Sect. 3.2.1.

(2) The output image undergoes the same steps as described in Sect. 3.1 in which the hash code $H^{'}$ is calculated.

(3) Denote the i-th hash value of the original image and received image as $H(i)$ and $H^{'}(i)$ respectively, the normalized Hamming distance d is therefore computed by

$$d\left(H, H^{'}\right) = 1/L\sum\nolimits_{i=1}^{L} \delta(H(i), H^{'}(i)) \tag{15}$$

where L is the length of hash and

$$\delta\left(H(i), H^{'}(i)\right) = \begin{cases} 0, H(i) = H^{'}(i) \\ 1, H(i) \neq H^{'}(i) \end{cases} \tag{16}$$

(4) Denote ϑ as a threshold to decide whether the received image could be authenticated. If the calculated normalized Hamming distance is larger than ϑ, the image $I^{'}$ is considered as a tampered or even a different one, which is unauthentic. Otherwise the image $I^{'}$ will be authenticated.

4 Experimental Analysis

In order to test the performance of the proposed algorithm, 21 gray-scale images of size 512 × 512 are used as the original test images, and the total numbers of images for content-preserving operations and content-altering attacks are 1113 and 442, respectively. Moreover, three image hashing algorithms are used for comparison in terms of robustness. The FAR versus FRR curve is also given to demonstrate the global performance of our proposed algorithm where the normalized Hamming distance d is used as the metric.

4.1 Content-Preserving Experimental Analysis and Comparisons

It is important to notice that a good perceptual image hashing scheme can authenticate perceptually identical images from the perceptually different images. In this section, some common content-preserving image processing operations conducted on Stirmark benchmark [28] are applied to illustrate the performance of our proposed scheme, including geometric operations. Table 1 shows the average normalized Hamming distance of the whole 21 original images under those operations based on different α. Different versions of image Lena are also shown in Fig. 3 under different value of α for example. The parameter α in Eq. (8) is used to enhance the security of the hash code such that the new pixel value depends on both the original pixel value and the secret key. Without knowing the secret key, an attacker cannot extract the hash accurately, thus cannot create a forged image.

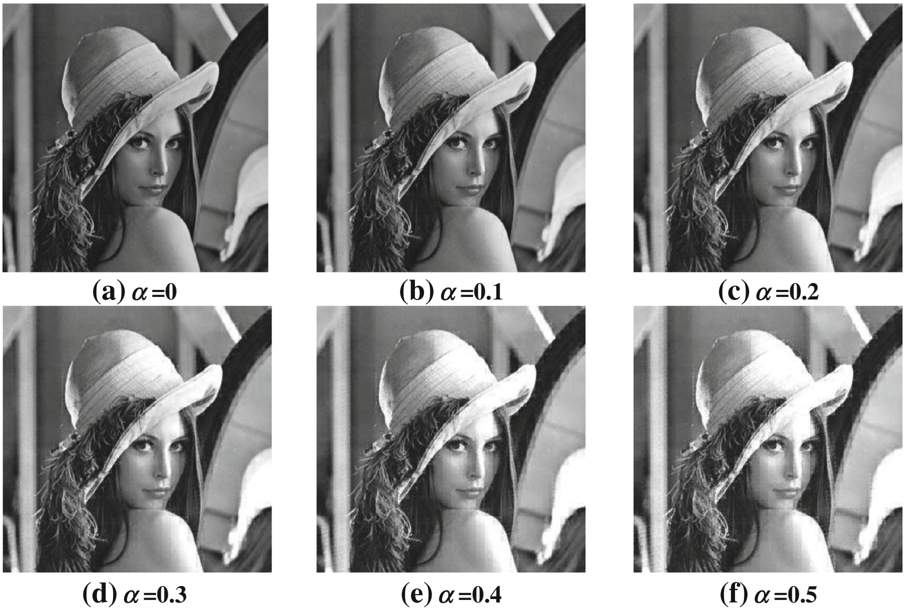

(a) α=0
(b) α=0.1
(c) α=0.2
(d) α=0.3
(e) α=0.4
(f) α=0.5

Fig. 3. Different versions of image Lena under different value of α

Note that the normalized Hamming distance is expected to approach zero for the perceptual identical images and approach 0.5 for different images. It can be also observed that the values of average normalized Hamming distance d between the hashes extracted from the original and processed images are all small under all manipulations, including geometric operations in Table 1, and with the increase of α,

Table 1. Average normalized Hamming distance under different image processing operations conducted on Stirmark benchmark

Image	Parameter	Average Normalized Hamming Distance d					
		α =0	α =0.1	α =0.2	α =0.3	α =0.4	α =0.5
JPEG compression	15	0.0145	0.0207	0.0166	0.0193	0.0207	0.0200
	20	0.0110	0.0186	0.0145	0.0193	0.0200	0.0186
	25	0.0069	0.0097	0.0110	0.0214	0.0166	0.0186
	30	0.0069	0.0124	0.0138	0.0186	0.0186	0.0138
	35	0.0076	0.0124	0.0117	0.0166	0.0173	0.0152
	40	0.0083	0.0124	0.0117	0.0145	0.0159	0.0152
	50	0.0062	0.0090	0.0055	0.0138	0.0152	0.0145
	60	0.0048	0.0076	0.0062	0.0145	0.0138	0.0131
	70	0.0028	0.0083	0.0062	0.0159	0.0124	0.0145
	80	0.0014	0.0055	0.0041	0.0138	0.0131	0.0131
	90	0.0014	0.0048	0.0041	0.0145	0.0110	0.0124
	100	0.0000	0.0062	0.0041	0.0124	0.0124	0.0131
Noise addition	0	0.0000	0.0055	0.0041	0.0124	0.0124	0.0124
	5	0.0462	0.0455	0.0366	0.0359	0.0380	0.0407
	10	0.1028	0.1049	0.0966	0.0897	0.0890	0.0973
	15	0.1656	0.1580	0.1504	0.1456	0.1366	0.1387
	20	0.1656	0.1587	0.1539	0.1511	0.1463	0.1401
Median filtering	3 × 3	0.0193	0.0200	0.0248	0.0290	0.0311	0.0324
	5 × 5	0.0290	0.0311	0.0276	0.0373	0.0331	0.0317
	7 × 7	0.0393	0.0428	0.0373	0.0428	0.0373	0.0359
	9 × 9	0.0476	0.0449	0.0442	0.0483	0.0435	0.0449
Convolution filtering	Gaussian	0.0849	0.1063	0.1104	0.1242	0.1235	0.1318
	Sharpening	0.0386	0.0386	0.0400	0.0380	0.0455	0.0490
Affine transformation	Y-shearing 1	0.0531	0.0559	0.0504	0.0518	0.0428	0.0455
	Y-shearing 2	0.0966	0.0959	0.0876	0.0911	0.0835	0.0814
	X-shearing 1	0.0518	0.0497	0.0545	0.0476	0.0476	0.0455
	X-shearing 2	0.1008	0.1001	0.0945	0.0925	0.0918	0.0939
	XY-shearing	0.0918	0.0835	0.0745	0.0801	0.0683	0.0718
	General 1	0.0856	0.0828	0.0745	0.0773	0.0697	0.0759
	General 2	0.0828	0.0725	0.0669	0.0732	0.0628	0.0642
	General 3	0.0759	0.0683	0.0635	0.0697	0.0676	0.0656
Rescaling	50	0.0014	0.0069	0.0069	0.0179	0.0207	0.0221
	75	0.0104	0.0117	0.0138	0.0166	0.0207	0.0179
	90	0.0145	0.0166	0.0145	0.0173	0.0200	0.0214
	110	0.0014	0.0062	0.0041	0.0159	0.0159	0.0159
	150	0.0097	0.0117	0.0090	0.0166	0.0159	0.0228
	200	0.0069	0.0076	0.0076	0.0159	0.0166	0.0200
Small random distortion	0.95	0.0621	0.0587	0.0504	0.0511	0.0476	0.0504
	1	0.0566	0.0573	0.0490	0.0476	0.0449	0.0455
	1.05	0.0594	0.0573	0.0483	0.0476	0.0455	0.0469
	1.1	0.0552	0.0552	0.0490	0.0497	0.0462	0.0469
Rotation	−2°	0.1001	0.0966	0.0856	0.0883	0.0842	0.0870
	−1°	0.0738	0.0752	0.0676	0.0676	0.0649	0.0725
	1°	0.0828	0.0807	0.0683	0.0718	0.0656	0.0732
	2°	0.0980	0.0897	0.0863	0.0828	0.0801	0.0794
Rotation with cropping	−2°	0.0745	0.0732	0.0649	0.0635	0.0600	0.0649
	−1°	0.0614	0.0566	0.0483	0.0538	0.0524	0.0504
	1°	0.0573	0.0545	0.0524	0.0483	0.0504	0.0545
	2°	0.0766	0.0752	0.0642	0.0656	0.0663	0.0773

(Continued)

Table 1. (*Continued*)

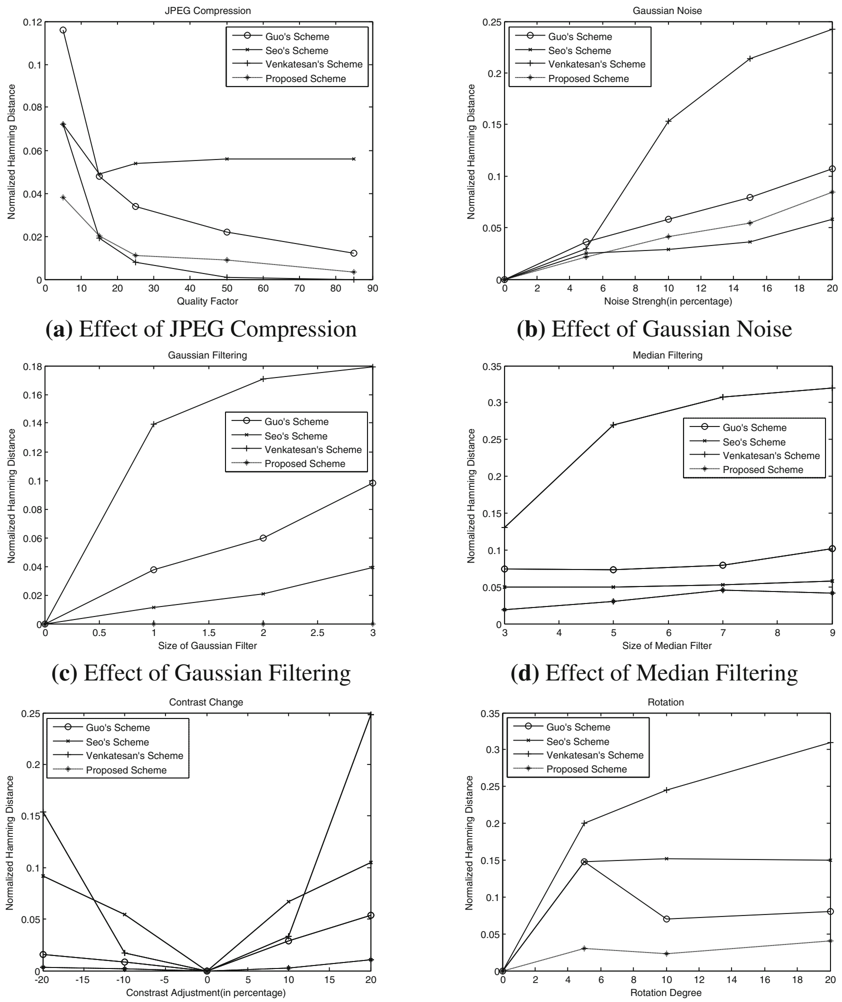
Image	Parameter	Average Normalized Hamming Distance d					
		α =0	α =0.1	α =0.2	α =0.3	α =0.4	α =0.5
Rotation with rescaling	−2°	0.0821	0.0801	0.0690	0.0628	0.0656	0.0690
	−1°	0.0690	0.0718	0.0594	0.0663	0.0656	0.0600
	1°	0.0621	0.0649	0.0538	0.0566	0.0552	0.0621
	2°	0.0794	0.0773	0.0656	0.0718	0.0676	0.0780

the values of d are a little increased. Since the coefficients in the third scale band of wave atom transform cannot be changed greatly without changing the content of image. Aware that the values of d under the parameters of 15 and 20 in Gaussian noise addition operations using Stirmark benchmark are a little larger than others. That's because under these two operations, there is much larger noise all over the images which affects the image quality more severely. By using the rotation-invariant pre-processing, the rotated images have been rotated back, while the non-rotated images have not been processed, thus there are not any perceptually different changes in images except the black background generated from the previous rotation operation, which results to a small d. Therefore, if the threshold ϑ is selected probably, the proposed scheme can authenticate the images which go through all kinds of content-preserving operations conducted on Stirmark benchmark in Table 1.

Moreover, it is well known that image hashing can be used in various applications such as image retrieval or watermarking, in some of which the robustness is the most important standard. Hence, to demonstrate the great robustness of our proposed scheme, three image hashing algorithms proposed by Guo et al. [29], Seo et al. [12] and Venkatean et al. [3] have been compared with our scheme in which α is chosen as 0.1. Guo et al. proposed a content-based image hashing scheme via wavelet and Radon transform, while Seo's scheme and Venkatean's scheme is based on the Radon transform and wavelet transform respectively. And here we employ the same parameters and the same operations as in other compared schemes on Matlab, such as Gaussian noise, Gaussian filtering, contrast change and rotation with cropping. Fig. 4 shows the performance of these image hashing schemes in terms of normalized Hamming distance.

As shown in Fig. 4(a), the robustness performance of proposed scheme is better than Guo's scheme and Seo's scheme but a little worse than Venkatean's scheme under JPEG compression, where the normalized Hamming distance of the proposed scheme is kept below 0.05. With the increase of Gaussian noise strength in Fig. 4(b), the proposed scheme keeps greater robustness than the scheme proposed by Guo and Venkatean, while a little worse than the scheme proposed by Seo. Considering the effect of Gaussian filtering, Median filtering and contrast change, the performance of our proposed method is better than the other three where the normalized Hamming distances are all below 0.05, whereas in other schemes, the normalized Hamming distance is above 0.1 in some cases. Moreover, the proposed scheme also performs the best and far better than others in geometric rotation manipulations.

In conclusion, the simulation results reveal that our scheme is superior to the schemes proposed by Guo et al., Seo et al. and Venkatean et al. The use of third scale band of wave atom transform enables the proposed algorithm to extract invariant



(a) Effect of JPEG Compression

(b) Effect of Gaussian Noise

(c) Effect of Gaussian Filtering

(d) Effect of Median Filtering

(e) Effect of Contrast Change

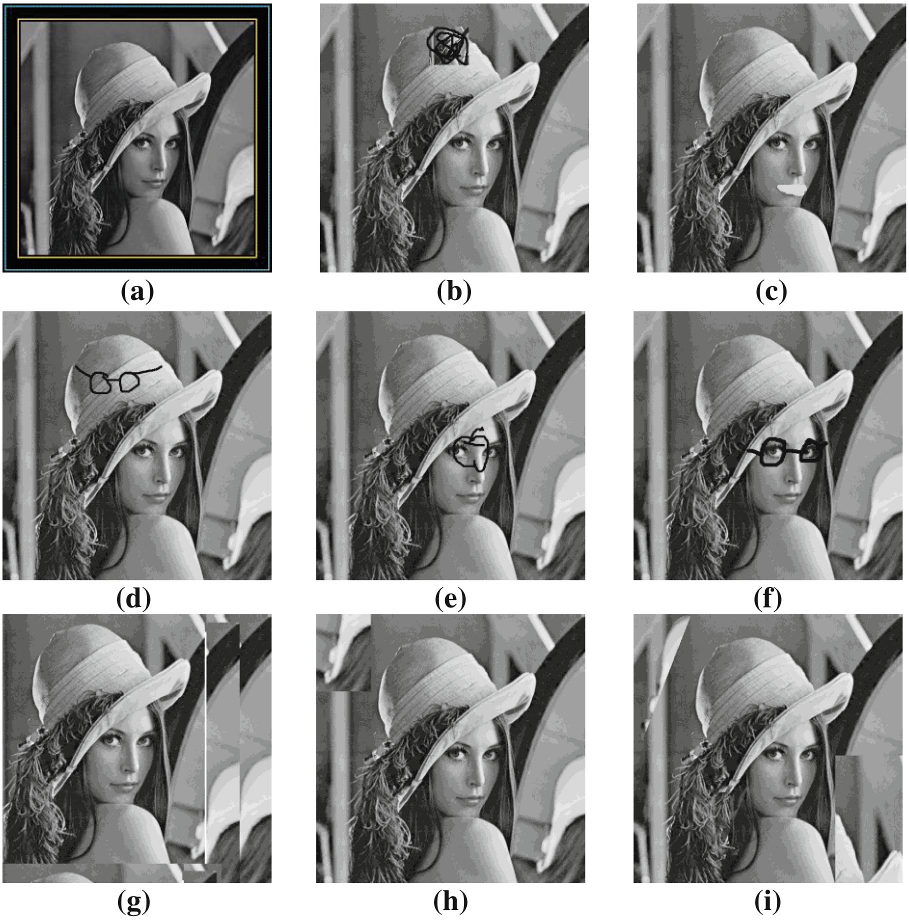
(f) Effect of Rotation

Fig. 4. Comparisons among different image hashing schemes in terms of normalized Hamming distance under different content-preserving manipulations

features from images, which are generally robust against content-preserving image manipulations.

4.2 Content-Altering Experimental Analysis

Although robustness is an important criterion for image hashing, fragility is also an indispensable considered factor. To prove the capability of image content-altering detection, 442 images are used to test in total, and only several tampered versions of image Lena have been shown in Fig. 5. Table 2 shows the normalized Hamming distances between the hashes of Lena and tampered versions of Lena under different values of α.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 5. Tampered versions of image Lena

It is observed that the distances d between the hashes of Lena and the tampered versions of Lena are normally larger than the distances between the original images and content-preserving processed ones. However, the distances decrease with the increase of α. Note that the randomness is increased with the value of α, but the capability of the tampering detection is also decreased. From Eq. (8), it can be seen that the new pixel value $P_I^{'}(x,y)$ is affected by the value of α and increasing the value of α will cause a large offset of $P_I^{'}(x,y)$. In other words, the new pixel value $P_I^{'}(x,y)$ will be less influenced by the original pixel value $P_i(x, y)$ of the original image itself. Therefore, the capability of tampering detection is degraded.

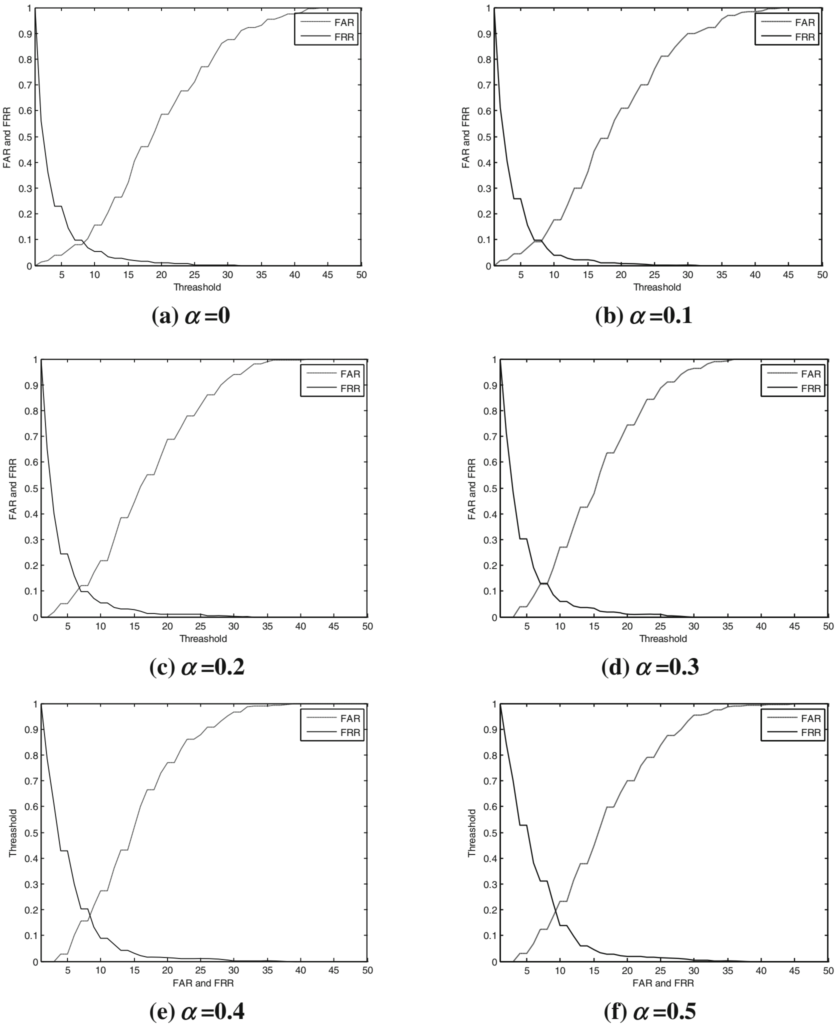
Table 2. Normalized Hamming distance against different tamperings

Image		Normalized Hamming Distance d					
		$\alpha = 0$	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$
Malicious Attack	**(a)**	0.4058	0.3913	0.3333	0.3478	0.3043	0.2899
	(b)	0.1159	0.1014	0.0870	0.0870	0.0580	0.0580
	(c)	0.0870	0.1014	0.1159	0.0870	0.0725	0.0580
	(d)	0.1304	0.1594	0.1014	0.1304	0.1304	0.1449
	(e)	0.1159	0.1449	0.1159	0.1014	0.0870	0.0870
	(f)	0.1449	0.1304	0.1014	0.1014	0.1159	0.1449
	(g)	0.3043	0.3043	0.2464	0.1884	0.1739	0.1449
	(h)	0.1449	0.1304	0.0725	0.0725	0.0580	0.0725
	(i)	0.1739	0.1594	0.1739	0.1449	0.1159	0.1014

It is noteworthy that the threshold ϑ is a tradeoff to evaluate the robustness and capability of the tampering detection. By comparing Tables 1 and 2, it can also be observed that when the value of ϑ decreases, the capability to detect the malicious tampering will be increased, but the sensitivity of content-preserving operations becomes relatively higher. $\vartheta = 0.1$ is found to be an optimal value to discriminate the content-preserving and content-altering attacks. Taking all criteria into consideration, we come to the conclusion that if α and ϑ are chosen as 0.1 and 0.1 respectively, the proposed algorithm is robust to most common image processing manipulations as shown in Table 1 and also able to detect all malicious tampered images as shown in Fig. 5. In addition, the security of the proposed scheme has also been guaranteed.

4.3 FAR versus FRR Curve

In this section, the false acceptance rate and the false rejection rate are calculated to evaluate the global performance of image hashing comprehensively in statistical analysis. The false acceptance rate (FAR) is the probability when the content altered images are identified as the genuine ones, which obtain the authentication. And the false rejection rate (FRR) is the probability when the genuine images are detected as the tampered ones, which cannot obtain the authentication. Determining whether or not the image is authentic is based on normalized Hamming distance. Hence, the distributions of FAR versus FRR can be obtained by varying the value of threshold under different value of α, which are shown in Fig. 6. From Fig. 6, it is expected that the lower the equal error rate, the better the performance, and the scheme performs better when α is smaller. However, it is a tradeoff among robustness, fragility and security. When we consider the security, which is the degree to which an image hashing scheme prevents the attacker from tricking the authentication system with a maliciously modified image, $\alpha = 0.1$ is the optimal compromise in this scheme, while shows almost the same performance when RMP is not used. These experimental results show that the proposed scheme gives a comparable low equal error rate and performs well in both aspects of robustness and fragility in perceptual image hashing authentication.



(a) α=0

(b) α=0.1

(c) α=0.2

(d) α=0.3

(e) α=0.4

(f) α=0.5

Fig. 6. The distributions of FAR versus FRR under different value of α

5 Conclusion

In this paper, we have proposed a perceptual hashing scheme based on wave atom transform and randomized pixel modulation, which is appropriate for image content authentication, image database retrieval and so forth. The proposed algorithm can authenticate the images which have undergone common content preserved image processing operations conducted on Stirmark benchmark, such as compression, filtering, noise addition, affine transformation and also the geometric manipulation. It is simultaneously sensitive to malicious tampering with the guaranty of system security. Instead of using traditional transform like DWT, DCT or other transform, we propose to employ wave atom transform for the sparser expansion and better characteristics to extract texture features when compared with others. The comparison results also show that the proposed scheme achieves better performance than the schemes proposed by Guo et al. [29], Seo et al. [12] and Venkatean et al. [3] even in the aspect of robustness.

References

1. Schneier, B.: Applied Cryptography. John Wiley & Sons Inc, USA (1996)
2. Bhattacharjee, S., Kutter, M.: Compression tolerant image authentication. In: Proceedings of International Conference on Image Processing, vol. 4(7), pp. 435–438, Chicago, USA (1998)
3. Venkatesan, R., Koon, S.M., Jakubowski, M.H., Moulin, P.: Robust image hashing. In: Proceedings of IEEE International Conference Image Processing, vol. 3, pp. 664–666, Vancouver, BC, Canada (2000)
4. Lu, C.S., Liao, H.Y.M.: Structural digital signature for image authentication. IEEE Trans. Multimedia **5**, 161–173 (2003)
5. Monga, V., Evans, B.L.: Robust perceptual image hashing using feature points. IEEE Int. Conf. Image Process. **1**, 677–680 (2004)
6. Monga, V., Evans, B.L.: Perceptual image hashing via feature points: performance evaluation and tradeoffs. IEEE Trans. Image Process. **15**(11), 3452–3465 (2006)
7. Ahmed, F., Siyal, M.Y., Vali, U.A.: A secure and robust hash-based scheme for image authentication. Signal Process. **90**(5), 1456–1470 (2010)
8. Fridrich, J., Goljan, M.: Robust hash functions for digital watermarking. In: Proceedings of IEEE International Conference Information Technology: Coding and Computing, pp. 178–183 (2000)
9. Lin, C.Y., Chang, S.F.: A robust image authentication system distinguishing JPEG compression from malicious manipulation. IEEE Trans. Circuits Syst. Video Technol. **11**(2), 153–168 (2001)
10. Lefebvre, F., Macq, B., Legat, J.D.: RASH: Radon soft hash algorithm. In: Proceedings of European Signal Processing Conference, pp. 299–302 (2002)
11. Roover, C.D., Vleeschouwer, C.D., Lefebvre, F., Macq, B.: Robust video hashing based on radial projections of key frames. IEEE Trans. Signal Process. **53**(10), 4020–4036 (2005)
12. Seo, J.S., Haitsma, J., Kalker, T., Yoo, C.D.: A robust image fingerprinting system using the rado transform. Signal Process. Image Commun. **19**(4), 325–339 (2004)
13. Swaminathan, A., Mao, Y., Wu, M.: Robust and secure image hashing. IEEE Trans. Inf. Forens. Sec. **1**(2), 215–230 (2006)
14. Wu, D., Zhou, X., Niu, X.: A novel image hash algorithm resistant to print–scan. Signal Process. **89**(12), 2415–2424 (2009)
15. Lei, Y., Wang, Y., Huang, J.: Robust image hash in Radon transform domain for authentication. Signal Process. Image Commun. **26**(6), 280–288 (2011)
16. Kozat, S.S., Venkatesan, R., Mihcak, M.K.: Robust perceptual image hashing via matrix invariants. In: Proceedings of IEEE International Conference on Image Processing, pp. 3443–3446 (2004)
17. Monga, V., Mihcak, M.K.: Robust and secure image hashing via non-negative matrix factorizations. IEEE Trans. Inf. Forens. Secur. **2**(3), 376–390 (2007)
18. Tang, Z., Wang, S., Zhang, X., Wei, W., Su, S.: Robust image hashing for tamper detection using non-negative matrix factorization. J. Ubiquitous Convergence Technol. **2**(1), 18–26 (2008)

19. Lv, X., Wang, Z.J.: Perceptual image hashing based on shape contexts and local feature points. IEEE Trans. Inf. Forensics Secur. **7**(3), 1081–1093 (2012)
20. Lu, W., Wu, M.: Multimedia forensic hash based on visual words. In: Proceedings of IEEE Conference Image Processing, pp. 989–992, Hong Kong (2010)
21. Khelifi, F., Jiang, J.: Perceptual image hashing based on virtual watermark detection. IEEE Trans. Image Process. **19**(4), 981–994 (2010)
22. Zhao, Y., Wang, S., Zhang, X., Yao, H.: Robust hashing for image authentication using Zernike moments and local features. IEEE Trans. Inf. Forensics Secur. **8**(1), 55–63 (2013)
23. Demanet, L., Ying, L.: Wave atoms and sparsity of oscillatory patterns. Appl. Comput. Harmonic Anal. **23**(3), 368–387 (2007)
24. Liu, F., Cheng, L.-M.: Perceptual image hashing via wave atom transform. In: Shi, Y.Q., Kim, H.-J., Perez-Gonzalez, F. (eds.) IWDW 2011. LNCS, vol. 7128, pp. 468–478. Springer, Heidelberg (2012)
25. Mallat, S.: A Wavelet Tour of Signal Processing, 2nd edn. Academic Press, Orlando/San Diego (1999)
26. Antoine, J.P., Murenzi, R.: Two-dimensional directional wavelets and the scale-angle representation. Signal Process. **52**, 259–281 (1996)
27. Reddy, B.S., Chatterji, B.N.: An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE Trans. Image Process. **5**, 1266–1271 (1996)
28. Fair evaluation procedures for watermarking systems (2000). http://www.petitcolas.net/fabien/watermarking/stirmark
29. Guo, X.C., Hatzinakos, D.: Content based image hashing via wavelet and radon transform. In: Ip, H.H.-S., Au, O.C., Leung, H., Sun, M.-T., Ma, W.-Y., Hu, S.-M. (eds.) PCM 2007. LNCS, vol. 4810, pp. 755–764. Springer, Heidelberg (2007)

IR Hiding: Use of Specular Reflection for Short-Wavelength-Pass-Filter Detection to Prevent Re-recording of Screen Images

Isao Echizen[1,3](✉), Takayuki Yamada[3], and Seiichi Gohshi[2]

[1] Graduate University for Advanced Studies, Kanagawa, Japan
[2] Kogakuin University, Tokyo, Japan
[3] National Institute of Informatics, Tokyo, Japan
iechizen@nii.ac.jp

Abstract. We previously proposed using infrared (IR) LEDs to corrupt recorded content to prevent the re-recording of images displayed on a screen. This method is based on the difference in sensory perception between humans and devices and prevents re-recording by adding IR noise to the images displayed on the screen without it being detected by the human eye. However, it cannot prevent re-recording using digital camcorders equipped with a short wavelength pass filter to eliminate the noise. We have now improved our method by adding a simple countermeasure against such attacks. It detects IR light reflected off the filter by using the IR specular reflection properties of the filter and thereby detects re-recording using digital camcorders equipped with a short wavelength pass filter. We implemented this countermeasure by adding one of two types of IR LEDs (bullet type and chip type with lens) to our prototype re-recording prevention system, which is installed on a B3-size screen. Testing showed that this enhanced system can detect camcorders with an attached short wavelength pass filter.

Keywords: Short wavelength pass filter · Infrared cut filter · Infrared absorption filter · Specular reflection · Infrared camcorder

1 Introduction

Today's small and highly functional digital camcorders can easily be carried into movie theaters and used to secretly record the displayed images with vivid color and high resolution. This has led in recent years to a growing problem of illegal recording of movies when they are shown in movie theaters. Moreover, the recording media is now digital, often in the form of a memory card. Data recorded on such media can easily be copied, written to other media, such as DVDs, and uploaded to the Internet, without degrading the image or sound. This makes it possible to distribute the data worldwide immediately. The Motion Picture Association of America (MPAA) [1] estimates the damage caused by bootleg film recording to be three billion dollars per year [2]. This has contributed to the reduction in the number of people paying to view movies in a movie theater, buying movies through legal channels, and renting movies for home viewing, resulting in serious financial losses by the movie industry.

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 38–54, 2015.
DOI: 10.1007/978-3-662-46739-8_3

Preventing the re-recording of movies being shown in a movie theater so as to protect the owner's copyright has thus become an even more urgent problem. One approach to solving it is to embed digital watermarks so that copyrighted materials can be detected. Unique information, such as an ID number, is embedded in each image or each voice segment using digital watermarking technology. Information such as the movie theater in which the re-recorded movie was shown and the time it was re-recorded can be obtained by detecting the embedded watermarks [3–7]. Although such methods are effective in terms of obtaining such information, they are ineffective in identifying the person responsible for the re-recording unless the theater was equipped with an audience monitoring system. Moreover, they do nothing to prevent the re-recording act itself.

We previously proposed a different approach: using a re-recording prevention system based on the difference in sensory perception between humans and devices that prevents re-recording by adding infrared (IR) noise to the images displayed on the screen without it being detected by the human eye [8–10]. However, a prototype implementing such a method was unable to prevent re-recording using digital camcorders equipped with a short wavelength pass filter (SWPF) to eliminate the noise. We have now improved our method by adding a simple countermeasure against such attacks: the IR light reflected off the filter is detected using the IR specular reflection properties of the filter. Our system can now detect re-recording using digital camcorders equipped with an SWPF.

Section 2 surveys conventional approaches and Sect. 3 briefly describes our previously proposed method based on the difference in sensory perception between humans and devices and describes how an SWPF mounted on a camcorder eliminates infrared noise. Section 4 describes our countermeasure based on IR specular reflection. Section 5 describes its addition to our prototype system. Section 6 explains how we evaluated the enhanced prototype system, presents the results, and discusses the effectiveness of our countermeasure. Section 7 briefly summarizes the key points and mentions future work.

2 Conventional Approaches

Digital rights management (DRM) is a common approach to preventing unauthorized copying of digital content. It aims to provide persistent access control by encrypting the content and allowing access (e.g., play, view, change, or copy) only by authorized users or devices, i.e., ones with the decryption key [11, 12]. For instance, content such as video data on commercial DVDs is usually encrypted with the Content Scramble System [12] and can only be decrypted and displayed by DVD players with the decryption key. However, when that content is shown on a screen or display, it can be illegally recorded by using a digital camera or camcorder, thereby bypassing the protection offered by DRM. Preventing the illegal recording of content shown on screens or displays is thus essential for preventing copyright violation.

In the digital watermarking approach, a watermark containing identifying information is embedded into the content. By extracting the information from the watermark, an investigator can identify where and when the original content was illegally recorded [3–7]. Various watermarking methods have been studied, including embedding such information as the theater ID into movie frames to trace the flow of illegally recorded content [3–5], using spread-spectrum audio watermarking for multichannel movie sound tracks to estimate the position in the theater of a camcorder being used for recording [6], and using spread-spectrum video watermarking with auto-correlation to estimate the recording position from the distorted coordinates of recorded watermarked video [7]. Although digital watermarking psychologically dampens the motivation to illegally record videos, it does not actually prevent illegal recording. Moreover, content creators are apparently reluctant to add watermarks to their content due to a strong feeling of attachment to their work.

Our proposed method overcomes these problems by directly preventing the illegal recording of videos and movies using camcorders. It is based on the differences in sensory characteristics between humans and devices. It uses IR light to corrupt the recorded content with noise signals that are invisible to the naked eye but are picked up by the CCD or CMOS device of a camera used for illegal recording, thus obviating the need for embedding watermarks into the content or adding a new function to the user-side device.

An approach similar in concept to that of the proposed method was considered by Bourdon et al. [13]. They theoretically investigated the use of spatial and temporal modulation of projected light to prevent illegal recording in movie theaters. An experimental evaluation to validate this approach was not conducted. Moreover, such an approach could be circumvented by manually controlling the camcorder's shutter speed. We are unaware of any other approaches similar to ours.

3 Previously Proposed Method

3.1 Principle

Figure 1 illustrates how the sensory perceptions of humans and sensor devices (e.g., the human eye and a charge-coupled device or the human ear and a microphone) overlap but do not correspond. Although sensor devices are designed so that their pick-up characteristics correspond to those of the human visual and auditory systems, design limitations make it impossible for their characteristics to completely match those of a human.

As mentioned, our previously proposed method for preventing the illegal recording of videos and movies is based on the differences in sensory characteristics between humans and sensor devices [8–10]: a noise signal is added to the shaded area in the figure. The noise signal is created externally; that is, our method does not require new functions to be added to the camera user's equipment.

According to the International Commission on Illumination, the wavelengths of visible light are between 380 and 780 nm [14] while those that can be picked up by image sensor devices, such as the CCDs and CMOSs used in digital cameras and camcorders, are between 200 and 1100 nm. Digital camcorders were designed to react to signals with wavelengths outside the visible range to give them the high level of luminous sensitivity needed for shooting in the dark. Figure 2 illustrates the wavelength ranges of the human visual system and digital video cameras.

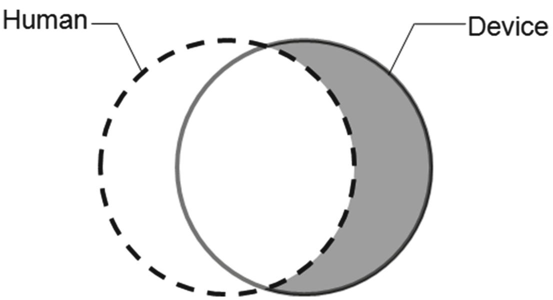


Fig. 1. Illustration of how sensory perceptions of humans and sensor devices overlap but do not correspond.

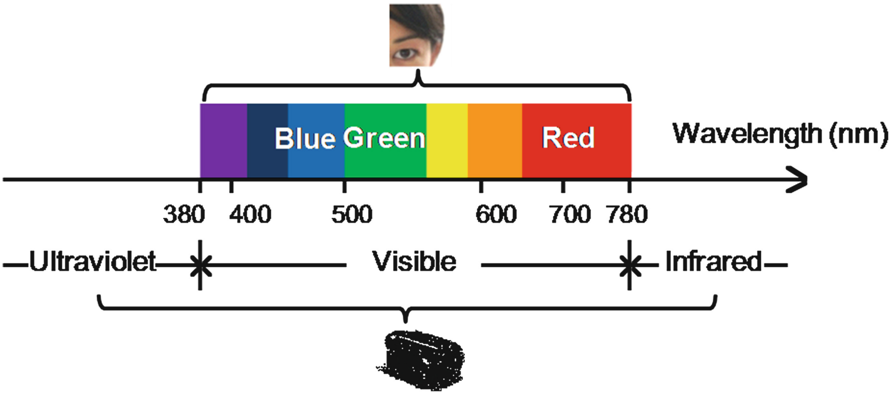


Fig. 2. Wavelength ranges of human visual system and digital video cameras.

The added noise signal created in our previously proposed method corresponds to light wavelengths that humans cannot see but to which sensor devices react. To degrade the quality of images and pictures taken with off-the-shelf digital cameras, a near-infrared light source with a peak wavelength of 870 nm is installed at the center back of the screen to superimpose noise on the photographic images alone, without affecting human vision. Since movie screens feature countless holes approximately 1 mm in diameter ("sound holes") to enable transmission of the sound from the speakers behind the screen, our method can be implemented without having to modify the screen—the infrared light from the source behind the screen can pass through the existing holes. The interference created in the unauthorized copy is heightened by causing the infrared light source to flicker at approximately 10 Hz on the basis of the Bartley effect.

3.2 Elimination of IR Noise Using SWPF

Our previously proposed method is ineffective when an SWPF is used to filter out the IR light. Such a filter allows short wavelength light to pass and blocks long wavelength light, i.e., IR light.

SWPFs can be classified as either “IR cut” or “IR absorption.” An IR cut filter is a planar object with a dielectric multilayer. It reflects the IR light received from several directions back in a single outgoing direction (specular reflection). An IR absorption filter is also a planar object that reflects the incoming IR light back in a single direction. However, since the wavelength penetration depends on the quantity of the absorber mixed into the glass, the IR reflection is lower than that of an IR cut filter, and the reflectance can be almost the same as that of a glass surface.

In contrast, non-specular reflectors, such as scatter plates, have various shapes and surface treatments, so they reflect the incident IR light back in various directions (diffuse reflection). The filter detection algorithm can thus detect the use of an SWPF by analyzing the reflection images picked up by the IR camcorder. Establishing a countermeasure against the use of an SWPF is essential for making this method practical. The simple countermeasure we developed uses the IR specular reflection properties of the SWPF.

4 Proposed Countermeasure

4.1 Principle

Re-recording is basically done by pointing a camcorder towards the screen and pressing the record button. Therefore, as illustrated in Fig. 3, an SWPF attached to the lens of a camcorder would be parallel to the screen.

In our proposed countermeasure, as illustrated in Fig. 3, IR emission units for filter detection and for noise creation are attached at regular intervals on the backside of the screen. An IR camcorder with a visible range cut filter and placed behind the screen captures the IR light reflected by various objects, and an algorithm running on a PC analyzes the reflected light. The SWPF on a camcorder in front of the screen is a planar filter with a dielectric multilayer. As described above, it reflects IR light from various incoming directions in a single outgoing direction (specular reflection). An IR absorption filter also reflects incoming IR light in only one direction. However, since it is a planar object, the wavelengths that it transmits depend on the quality of the absorber used in its glass plates. As mentioned above, its IR reflection is low compared with that of an IR cut filter.

The non-specular reflective objects, which have various shapes and surface treatments, reflect the incident IR light in various directions (diffuse reflection). The filter detection algorithm thus detects the use of an SWPF by analyzing the images picked up by the IR camcorder and identifying the specular reflections. In the following section, we describe the requirements for the countermeasure in more detail.

4.2 Reflections off Object Surfaces

The key to our countermeasure is distinguishing the reflections from an SWPF from those from other objects. The algorithm we use to do this is based on the Phong shading model [15]. In this model, there is a light source, an object, and a camcorder, and the spectral radiance $L_Q(\lambda)$ for one pixel is expressed as follows.

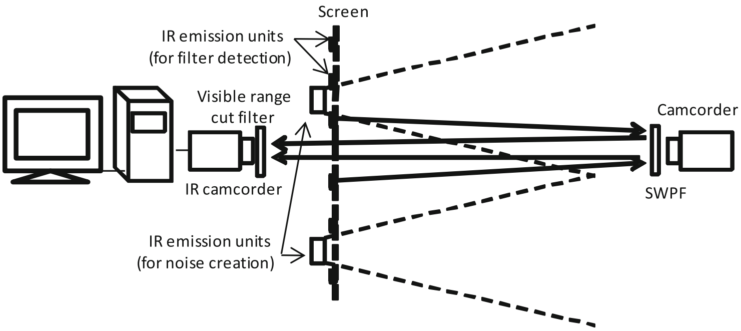


Fig. 3. Principle of SWPF detection.

$$L_Q(\lambda) = I_e(\lambda)K_d(\lambda)cos\theta / r^2 + I_e(\lambda)K_s(\lambda)(cos\varphi)^n + I_a(\lambda)K_a(\lambda). \quad (1)$$

r: distance from light source
θ: angle between light source and normal vector of object surface
φ: angle between camcorder and regular reflection
$I_e(\lambda)$: radian intensity of light source
$I_a(\lambda)$: radian intensity of ambient light source
$K_d(\lambda)$: diffuse reflectance of light source
$K_s(\lambda)$: specular reflectance of light source
$K_a(\lambda)$: reflectance of ambient light

$$(0 \leq K_d(\lambda), K_s(\lambda), K_a(\lambda) \leq 1)$$

The first term in Eq. (1) is called the diffuse reflection element, and it shows that the light reflects randomly and diffuses equally. The second term is called the specular reflection element, and it shows that the light reflects more strongly on object surfaces. The n is the decrease in reflection intensity; when the value is large, the object has properties of specular reflection, and when it is small, the object has those of diffuse reflection. The third term is called the ambient light element, and it shows the brightness of the light on the object surface that is not directly from a light source. Here, the light source is the IR emission units for filter detection, the object is the SWPF, and the camcorder is the IR camcorder. In the enhanced method, a short wavelength cut filter is attached to the IR camcorder to remove the effects of visible range light, thereby excluding the effects of visible light. From the information presented above, K_d, K_a, and φ are

$$K_d(\lambda) \cong 0,\ K_a(\lambda) \cong 0,\ \varphi \cong 0. \quad (2)$$

and Eq. (1) becomes

$$L_Q(\lambda) = K_s(\lambda) \times I_e(\lambda). \quad (3)$$

Here, the reflection of the IR cut filter coefficient, $K_s(\lambda)$, and the reflection of the IR absorption filter coefficient, $K_s{}'(\lambda)$, have the relationship below.

$$0 \leq K_s(\lambda) < K_s{}'(\lambda) \leq 1. \tag{4}$$

If the object surface is curved or is not a specular one, the specular reflection is lower, and the diffuse reflection is higher. Since the increment in the diffuse reflection element is inversely proportional to the square of the object's distance from the light source, the diffuse reflection is smaller than the spectral radiance when the object is an SWPF. We can express these characteristics using the following relationships.

(a) spectral radiance $L_Q(\lambda)$ of IR cut filter:

$$L_Q(\lambda) \cong I_e(\lambda) \tag{5}$$

(b) spectral radiance $L_Q{}'(\lambda)$ of IR absorption filter:

$$L_Q{}'(\lambda) \leq I_e(\lambda) \tag{6}$$

(c) spectral radiance $L_Q{}''(\lambda)$ of curved shape that is not a mirror:

$$L_Q{}''(\lambda) \leq L_Q{}'(\lambda) \tag{7}$$

From these relationships, we can order the spectral radiances:

$$L_Q{}''(\lambda) \leq L_Q{}'(\lambda) \leq L_Q(\lambda). \tag{8}$$

This makes it possible to identify an SWPF and other reflecting objects.

Equality holds in two cases.

(a) specular reflective objects such as a mirror with almost the same reflectance as an IR cut filter
(b) specular reflective objects such as a glass with almost the same reflectance as an IR absorption filter

Both types of reflective objects are unlikely to be in fixed position "facing" the screen during a certain period of time. Even if they did happen to be facing the screen, they would be automatically excluded as candidates by the motion estimation algorithm as soon as they were moved. In the unlikely event that a reflective object remained in a fixed state during a certain period of time, a size-and-shape algorithm would determine whether it was an SWPF.

4.3 Filter Detection Method

As shown in Fig. 3, the relationship between the positions of the IR emission units for filter detection and of the IR camcorder behind the screen is essential because the camcorder must be able to capture the reflected rays of the IR light source. The following section describes their arrangement.

Arrangement of IR Emission Units for Detection. Since the IR camcorder can stably detect the specular reflection from an SWPF with which digital camcorders equipped, the IR emission units for detection should be arranged as shown in Fig. 3. The interval between them is derived as follows.

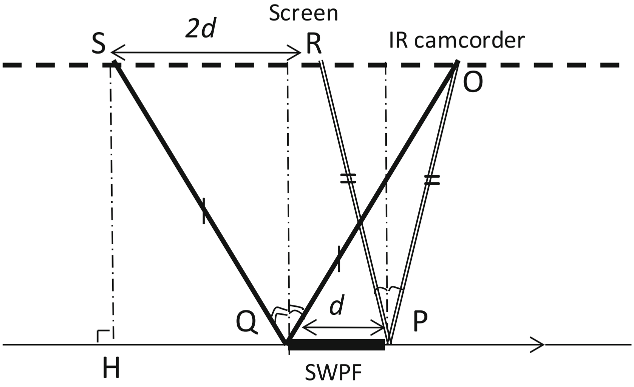
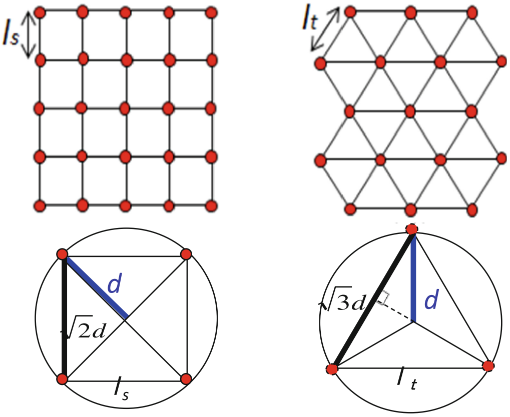


Fig. 4. Relationship between screen and SWPF.



(a) Square lattice (b) Triangular lattice

Fig. 5. Arrangement of IR emission units for filter detection and interval calculation.

Figure 4 illustrates the physical relationship between the screen and an SWPF. The IR reflection from the SWPF is detected along segment QP (length d) using the IR camcorder (point O). Since the IR light incident angle and reflection angle are equal to segment QP (from the property of specular reflection), we must place one or more IR emission units along segment *SR* (length of *2d*). Since a screen is generally flat, the units should be placed on the detection plane.

The detection plane generally has a square lattice or a triangular lattice arrangement, as shown in Fig. 5. Since the position interval of the IR emission units depends on the size of the SWPF and on the filter form (generally square or circular), if the length of one side of a square filter and the diameter of a circle filter are set to *d*, we derive position interval l_s for a square lattice and l_t for a triangular lattice. Given the two types of lattice, it is necessary to determine the position interval of a square lattice and a triangular lattice in a square area with a side length of *2d* and in a circular area with a diameter of *2d* so that at least one IR emission unit is positioned to cover that area. A circle with a diameter of *2d* is inscribed in a square with a side length of *2d*. We thus need to determine the position interval on the basis of a circular area with a diameter of *2d*. Since we have to make the intervals of a square lattice and triangular lattice (l_s and l_t) shorter than the length of one side of the square and triangle that are inscribed in a circle *2d* in diameter, as shown in Fig. 5, we can derive

$$l_s \leq \sqrt{2}d \tag{9}$$

$$l_t \leq \sqrt{3}d. \tag{10}$$

Position of IR Camcorder. Someone attempting to illegally record the images displayed on a screen usually tries to capture the entire displayed image in the camcorder's viewfinder so as to reduce distortion of the recorded images. As a result, the normal vector of the surface of the SWPF attached to the camcorder is generally fixed and facing the center of the screen for a certain period of time. Thus, we can efficiently detect the reflection of IR light by placing the IR camcorder at the center of the screen. There are many tiny holes in the screen for sound transmission, and an IR camcorder behind the screen can capture the video image through them.[1] To determine the locations of the IR light sources, which depend on the screen size (number of IR emission units) and/or the theater size, we performed a preliminary assessment using our prototype SWPF detection system.

4.4 Filter Detection Algorithm

The video images recorded as described above are analyzed using an algorithm that detects specular reflection. It uses two sets of video images.

Video (a): shot in a room without an audience
Video (b): shot in the same room with an audience

By comparing the images between the two, the algorithm can eliminate the reflections from objects already in the room. The detection steps are listed below, and the flow is illustrated in Fig. 6.

[1] The camcorder is positioned as close to the screen as possible, and the focus is set to infinity. As a result, the screen is blurred and the image recorded through the screen holes is sufficiently clear.

Filter Detection Procedure.

- **Step 1.** Input image frames of video (a) and eliminate effect of flashing noise (from IR emission units).
- **Step 2.** Average processed image frames and generate one averaged image frame.
- **Step 3.** Do steps 4 through 8 for each series of image frames of video (b).
- **Step 4.** Input image frames of video (b) and eliminate effect of flashing noise.
- **Step 5.** Subtract pixel values of averaged image frame of video (a) generated in step 2 from those of each image frame of video (b) processed in step 4.
- **Step 6.** Estimate motion areas for video (b) from image frames processed in step 4.
- **Step 7.** Eliminate motion areas for video (b) using results of motion estimation (step 6) and eliminate diffuse reflective objects for video (b).
- **Step 8.** Calculate areas for each reflection area, *S*, for video (b) and compare them with threshold *T*. Do the next step if the area is larger than the threshold.
- **Step 9.** The object for detection is recognized by a labeling algorithm. Do the next step if detected camcorder faces screen for a certain period of time.
- **Step 10.** If reflective object shape is circle or square, detect attack and display position of SWPF-equipped camcorder on PC display for analysis.

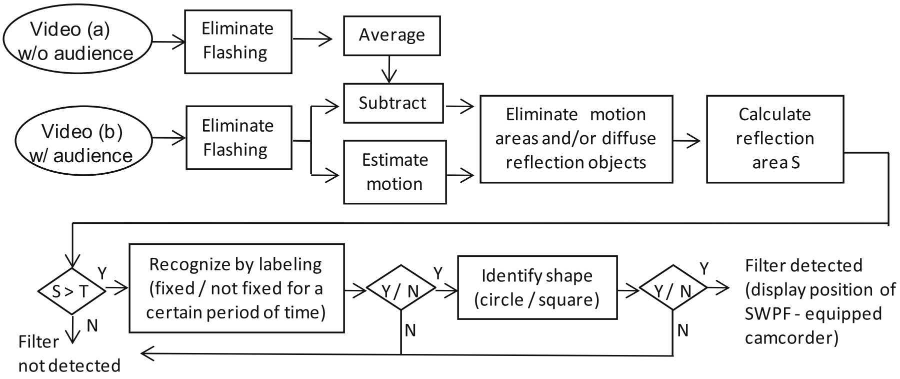

Video (a) w/o audience
Eliminate Flashing
Average
Video (b) w/ audience
Eliminate Flashing
Subtract
Estimate motion
Eliminate motion areas and/or diffuse reflection objects
Calculate reflection area S
S > T
Y
N
Filter not detected
Recognize by labeling (fixed / not fixed for a certain period of time)
Y/ N
Y
N
Identify shape (circle / square)
Y/ N
Y
N
Filter detected (display position of SWPF - equipped camcorder)

Fig. 6. Filter detection flow.

5 Implementation

5.1 Description

We added the proposed countermeasure to our prototype re-recording prevention system, which consists of 3 IR emission units for noise creation, 24 or 48 IR emission units for filter detection, and an IR camcorder for recording images of the reflected IR light. The IR emission units for noise creation comprise 18 reflection-type IR LEDs, a short wavelength cut filter (cut-on wavelength of 870 nm) attached to the front, and a cooling fan attached to the rear. They are arranged horizontally on the back of the screen. We used one of two types of IR-emitting LEDs (bullet type and chip type with lens) as the IR emission units for filter detection, thereby creating two prototypes for evaluation. Their specifications are summarized in Table 1, an overview of the bullet-type system is shown in Fig. 7, and one of the chip-type-with-lens system is shown in Fig. 8.

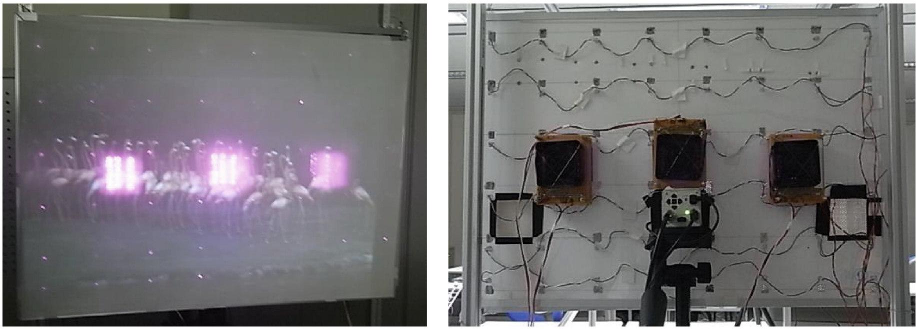
The strength of IR LED radiation can be compared by using a value representing the radiation intensity I_Q, which is the strength of the intensity of a point radiation source. It represents the amount of radiant energy Q per unit time t as the radiant flux φ radiates per unit solid angle Ω and is defined as

$$I_Q = d\varphi/d\Omega. \tag{11}$$

The perceived intensity of the light source depends on its wavelength, the viewer's visual sensitivity, and the camcorder's spectral sensitivity, and it is highly dependent on the system configuration. Moreover, the radiation angle of each IR LED is the half power angle; i.e., the range in which the radiant intensity of an IR LED is reduced by half.

The bullet-type IR LEDs have a radiation angle of ±7°, a wavelength of 940 nm, and an output power 0.13 W, as shown in Table 1. They are arranged behind the screen in a rectangular lattice. The chip-type IR LEDs have a narrower radiation angle (±4°), a wavelength of 940 nm as well, and an output power 0.84 W. They are also arranged behind the screen in a rectangular lattice.

The radiation angle is narrower in the chip-type-with lens system because a lens is placed in front of each LED. As a result, this system has a longer detection range than the bullet-type system. The IR camcorder used for detection has high sensitivity in the IR wavelength range and is placed behind the screen at the center.



(a) Front

(b) Back

Fig. 7. Bullet-type system overview.

5.2 Arrangement of IR-emitting Units

In our evaluation, the detection targets were square SWPFs with a side length of 50 mm and circular SWPFs with a diameter of 50 mm. In accordance with Eq. (9), the distance between the IR LEDs for filter detection was set to 70 mm. The preliminary assessment mentioned above for determining the locations of the IR light sources was done using the following procedure.

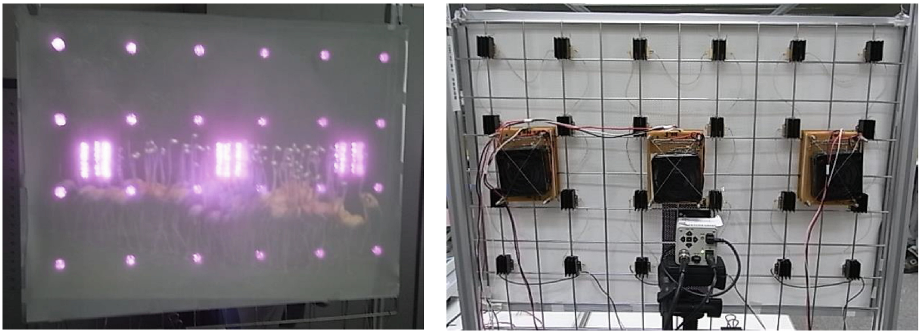

(a) Front
(b) Back

Fig. 8. Chip-type-with-lens system overview.

Table 1. Specifications of IR-emitting LEDs.

	Wavelength	Radiation angle	Output power	Radiant intensity
Bullet type	940 nm	±7°	0.13 W	0.08 W/sr
Chip type with lens	940 nm	±4°	0.84 W	2.08 W/sr

1. An IR absorption filter with lower IR reflectance than an IR cut filter was attached to a camcorder.
2. The camcorder was sequentially placed in five locations in a dark room, and the arrangement of the IR LEDs was adjusted each time so that the IR camcorder could detect the specular reflection from the IR absorption filter. The camcorder was positioned in each case so that it could capture the entire image on the screen.

On the basis of the results of this preliminary assessment, we arranged the bullet-type IR LEDs in a rectangular lattice of eight columns and six rows (Fig. 9(a)) and the chip-type IR LEDs in a rectangular lattice with six columns and four rows (Fig. 9(b)).

6 Evaluation

6.1 Method

For our evaluation, we used a dark room in our laboratory instead of an actual movie theater and the reflective objects shown in Fig. 10.

As shown in Table 2, the objects can be divided into four groups. They were placed anywhere from 2 to 14 m from the screen, except for the three camcorders. They were positioned so that each one could capture the complete image on the screen. We set the comparison threshold at six pixels so that the IR absorption filter, which was placed at a distance of 14 m, could be detected. Using the results of the detection algorithm described in Sect. 4.4, we evaluated the detection ability of each prototype.

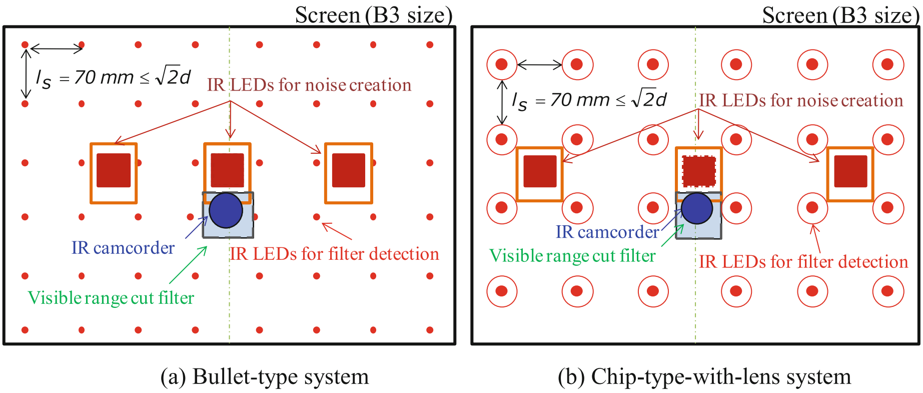

Screen (B3 size)
$l_S = 70\ mm \le \sqrt{2}d$
IR LEDs for noise creation
IR camcorder
Visible range cut filter
IR LEDs for filter detection
(a) Bullet-type system
Screen (B3 size)
$l_S = 70\ mm \le \sqrt{2}d$
IR LEDs for noise creation
IR camcorder
Visible range cut filter
IR LEDs for filter detection
(b) Chip-type-with-lens system

Fig. 9. Arrangement of IR-emitting LEDs for two prototype systems.

6.2 Results

Example evaluation images for a detection distance of 2 m are shown in Fig. 11. The red circles indicate areas with a threshold of six pixels or more. They were detected as an IR cut filter (object 17) and an IR absorption filter (object 18). These areas correctly correspond to the two camcorders with a filter shown in Fig. 10.

The light source of the beam projector (object 1) was eliminated by the background difference step in the detection algorithm, and the moving objects (3–11) were eliminated by the movement detection algorithm; so only the two filters were detected. Filter detection takes about one second, so it is done virtually in real time. The proposed countermeasure is thus effective against attacks using an SWPF.

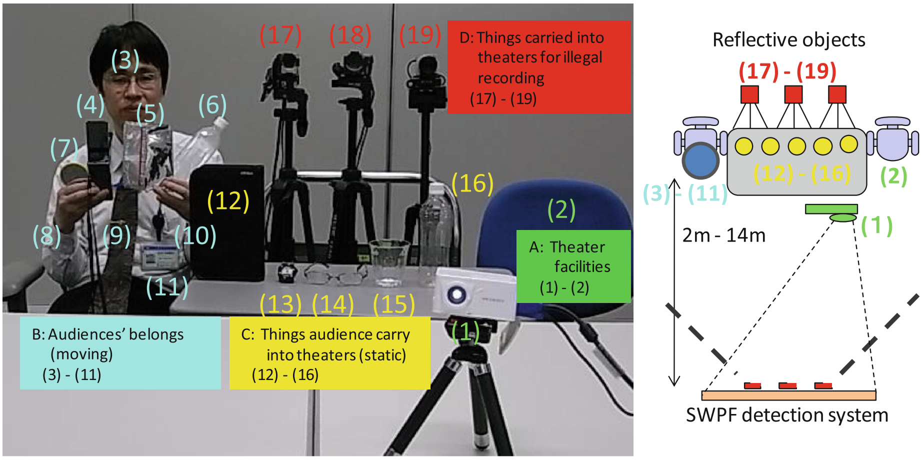
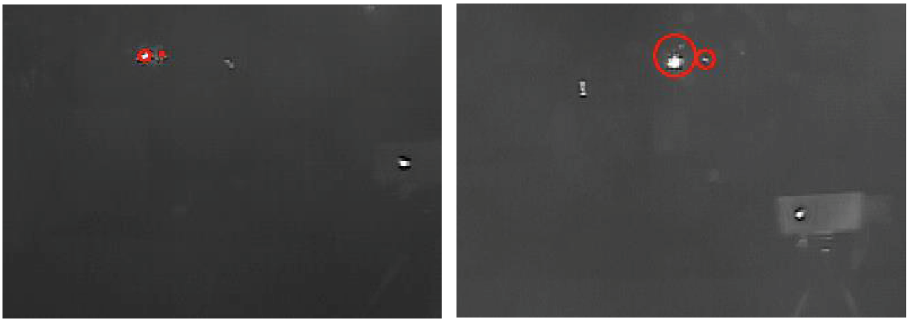

(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13) (14) (15)
(16)
(17) (18) (19)
(1)
(2)
D: Things carried into theaters for illegal recording (17) - (19)
A: Theater facilities (1) - (2)
B: Audiences' belongs (moving) (3) - (11)
C: Things audience carry into theaters (static) (12) - (16)
Reflective objects
(17) - (19)
(12) - (16)
(2)
(3) - (11)
(1)
2m - 14m
SWPF detection system

Fig. 10. Experimental setup.

Table 2. Reflective objects used.

Group	Object type	Objects		
A	Theater facilities	(1) Beam projector	(2) Chair	
B	Audiences' belongings (moving)	(3) Eyeglasses (6) Plastic bottle (9) Tie clip	(4) Mobile phone (7) Hand mirror (10) Pen	(5) Snack package (8) Watch (11) ID card
C	Things audience carry into theaters (static)	(12) Nylon bag (15) Drinking glass	(13) Watch (16) Plastic bottle	(14) Eyeglasses
D	Things carried into theaters for illegal recording	(17) Camcorder with attached IR cut filter (18) Camcorder with attached IR absorption filter (19) Camcorder (without filter)		



(a) Bullet-type system

(b) Chip-type-with-lens system

Fig. 11. Evaluation images for two prototype systems.

6.3 Comparison Between Prototype Systems

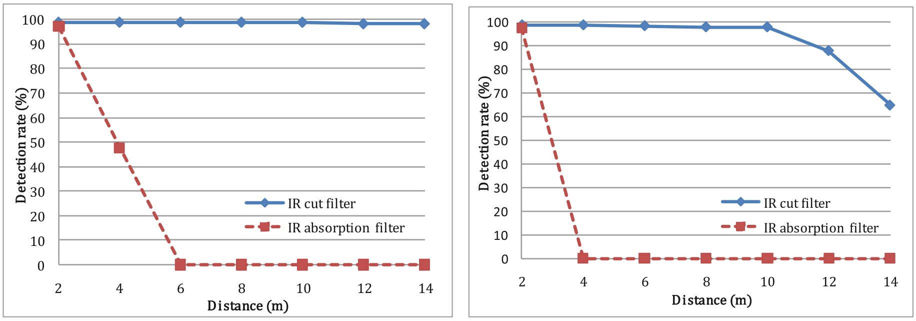
Considering that the target use is in a movie theater, we evaluated the two prototype systems under various conditions. These conditions included lighting, background, size and direction of filter, type and condition of reflective objects, and detection distance. However, the use of a visible cut filter on the IR camcorder makes lighting a moot point. Moreover, the use of the background difference step in the detection algorithm eliminates the effect of the background. And because the SWPF is attached to the camcorder, it is probably about the same size as the camcorder lens. Furthermore, as noted in Sect. 4.1, the SWPF attached to the camcorder is most likely parallel to the screen.

We thus placed the 19 reflective objects listed in Table 2 at various distances from 2 to 14 m from the screen and measured the detection rate. The detection algorithm can independently detect an SWPF attack from three consecutive image frames, which means that an SWPF can be detected ten times per second, assuming a video frame rate of 30 fps. For each measurement of the detection rate, we used a 20-s video clip, so the total number of detections, n, was 200 (= 20 s × 10). The detection rate, r, is thus given by $r = n_c/n \times 100$, where n_c is the number of detections in which the SWPF was correctly detected.

6.4 Results of Comparison

The detection rates are plotted in Figs. 12 and 13 for various distances from the front or the diagonal (5°). The above detection rates from the front for the IR absorption filter dropped at distances greater than 2 m with the bullet-type system (Fig. 12) and at distances greater than 12 m with the chip-type-with-lens system (Fig. 13). This is because IR light sufficiently strong for detection did not reach the filters on the camcorders. In general, LED radiation was centered at zero degrees (peak) and was distributed in a bell shape to the right and left. Therefore, a larger angle between the camcorder and an IR reflective object makes it more difficult to detect the reflected IR light. With both systems, the detection rates were higher when a reflective object was placed directly in front of the system.

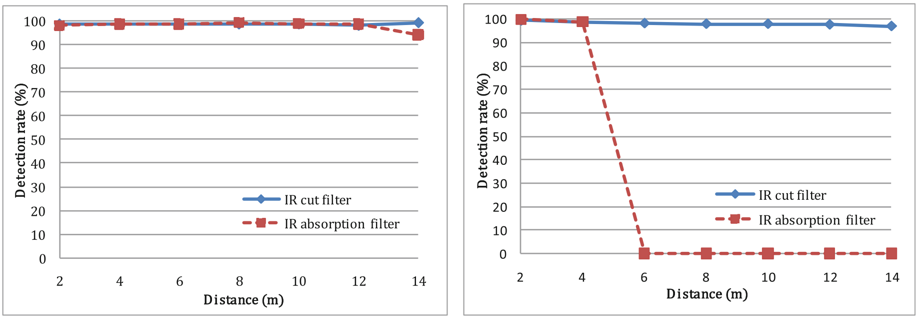
In marketing, it is generally said that the accuracy of a number count by a person is about 90 % [16]. Therefore, we evaluated the detection rate for near distance, middle distance, and far distance for four grades (Excellent, Good, Fair, Poor), as summarized in Table 3. We defined “Good” as more than 90 %, i.e., the accuracy of a number count by a person. The results show that the chip-type-with-lens system was marginally better at far distances than the bullet-type system, meaning that it is better for large places, such as a movie theater. The grades were “Poor” for the chip-type-with-lens system in the diagonal case for the IR absorption filter for middle and far distances. This performance can be improved by widening the radiation angle of the chip-type IR LEDs while maintaining the radiant intensity and by attaching them at different angles.



(a) Front

(b) Diagonal (5°)

Fig. 12. Detection rates for bullet-type system.



(a) Front (b) Diagonal (5°)

Fig. 13. Detection rates for chip-type-with-lens system.

Table 3. Detection grades for two prototype systems.

			Implementation			
			Bullet type		Chip type with lens	
			IR cut filter	IR absorption filter	IR cut filter	IR absorption filter
Grade*	Front	Near distance (2, 4 m)	Excellent	Fair	Excellent	Excellent
		Middle distance (6, 8 m)	Excellent	Poor	Excellent	Excellent
		Far distance (10,12,14 m)	Excellent	Poor	Excellent	Excellent
	Diagonal (5°)	Near distance (2, 4 m)	Excellent	Poor	Excellent	Excellent
		Middle distance (6, 8 m)	Excellent	Poor	Excellent	Poor
		Far distance (10,12,14 m)	Fair	Poor	Excellent	Poor

*The four grades are defined every 2 m in accordance with the average of the detection rate for the measurement point: detection rate over 95% = "Excellent," more than 90 to less than 95% = "Good," more than 50 to less than 90% = "Fair," and less than 50% = "Poor."

7 Conclusion

The re-recording of images shown in a movie theater has become a social problem. Even though existing technical countermeasures using digital watermarking might create a mental deterrence, they are unable to prevent it. Therefore, we developed a method to prevent re-recording that actually prevents re-recording. However, it could be thwarted by attaching a short wavelength pass filter to the camcorder to cut or absorb the IR light used to create noise in the image. We have now developed a countermeasure against such attacks that uses the specular reflection properties of the filter. An evaluation showed that its implementation using chip-type LEDs with a lens system works better than one using bullet-type LEDs in large places, such as a movie theater.

Image digitization continues to advance and sound facilities are growing rapidly, resulting in environments where contents other than movies can be presented at low cost. Realistic and powerful images, such as 3D images of a sporting event, are growing in popularity. The number of people who enjoy viewing sporting events and music concerts in a cinema complex is increasing. Consequently, the re-recording of images displayed on various types of devices in various environments will continue to proliferate. We thus plan to apply our re-recording detection method to various types of display equipment, such as CRTs and LCDs.

References

1. The Motion Picture Association of America (MPAA). http://www.mpaa.org/
2. Ezra, E., Rowden, T. (eds.): Transnational Cinema: The Film Reader. Routledge, London (2006)
3. Haitsma, J., Kaler, T.: A watermarking scheme for digital cinema. In: Proceedings of the 2001 International Conference on Image Processing (ICIP 2001), vol. 2, pp. 487–489 (2001)
4. Gohshi, S., Nakamura, H., Ito, H., Fujii, R., Suzuki, M., Takai, S., Tani, Y.: A new watermark surviving after re-shooting the images displayed on a screen. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 1099–1107. Springer, Heidelberg (2005)
5. Nakamura, H., Gohshi, S., Fujii, R., Ito, H., Suzuki, M., Takai, S., Tani, Y.: A digital watermark that survives after re-shooting the images displayed on a CRT screen. J. Inst. Image Inform. Telev̇. Eng. **60**(11), 1778–1788 (2006)
6. Nakashima, Y., Tachibana, R., Babaguchi, N.: Watermarked movie soundtrack finds the position of the camcorder in theater. IEEE Trans. Multimedia **11**(3), 443–454 (2009)
7. Lee, M., Kim, K., Lee, H.: Digital cinema watermarking for estimating the position of the pirate. IEEE Trans. Multimedia **12**(7), 605–621 (2010)
8. Yamada, T., Gohshi, S., Echizen, I.: Re-shooting prevention based on difference between sensory perceptions of humans and devices. In: Proceedings of the 17th International Conference on Image Processing (ICIP 2010), pp. 993–996 (2010)
9. Yamada, T., Gohshi, S., Echizen, I.: IR hiding: a method to prevent video re-shooting by exploiting differences between human perceptions and recording device characteristics. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 280–292. Springer, Heidelberg (2011)
10. Echizen, I., Yamada, T., Gohshi, S.: IR hiding: method for preventing illegal recording of videos based on differences in sensory perception between humans and devices. In: Shi, Y.Q. (ed.) Transactions on DHMS VII. LNCS, vol. 7110, pp. 34–51. Springer, Heidelberg (2012)
11. Rosenblatt, B., Trippe, B., Mooney, S.: Digital Rights Management - Business and Technology. M&T Books, New York (2003)
12. Content Scramble System (CSS). http://www.dvdcca.org/css.aspx
13. Bourdon, P., Thiebaud, S., Doyen, D.: A theoretical analysis of spatial/temporal modulation-based systems for prevention of illegal recordings in movie theaters. In: Proceedings of the SPIE - Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, vol. 6819, Paper 1389, 9 p. (2008)
14. Schanda, J. (ed.): Colorimetry: Understanding the CIE System. Wiley-Interscience, New York (2007)
15. Zhang, H., Liang, Y.: Computer Graphics Using Java 2D and 3D. Prentice Hall, Upper Saddle River (2006)
16. Holst, G., Lomheim, T.: CMOS/CCD sensors and camera systems. SPIE-International Society for Optical Engineering (2007)

A Reliable Covert Communication Scheme Based on VoIP Steganography

Harrison Neal and Hala ElAarag(✉)

Department of Mathematics and Computer Science, Stetson University,
DeLand, FL, USA
{hneal,helaarag}@stetson.edu

Abstract. Steganography is the science of hiding information in such a way that an adversary wouldn't know it existed. A significant amount of research has been done in this field for non-real time mediums. Research on real time mediums, for example Voice over Internet Protocol (VoIP), isn't as mature. In this paper, we propose an algorithm that enables data hiding in G.711, the most commonly used voice codec for VoIP devices, while gracefully handling packet loss. This would allow two telephone users to covertly transfer multiple pieces of arbitrary information between their respective systems in a reliable manner. We use important performance metrics to evaluate our algorithm, namely, throughput, noise-to-signal ratio and the Perceptual Evaluation of Speech Quality algorithm. We demonstrate that our algorithm performs well compared to other algorithms proposed in the literature in real world environments, where packet loss is inevitable, by maintaining high throughput and good speech quality.

Keywords: VoIP · G.711 · Steganography · Covert communication

1 Introduction

Steganography is the science of hiding messages so to appear as if they don't exist. This is usually achieved by either modifying part of a communication that wasn't intended to hold meaningful data for the user, or by subtly modifying meaningful data to contain a second message [1]. Many early papers on steganography focused on hiding secret information in images, videos and audio. With the rapid expansion of the Internet, later research described hiding data in protocol headers [2]. As interest in steganography has increased, research on steganography within VoIP streams and other real-time media has also increased. One example of recent research in this area is a method by Ito, Abe and Suzuki [3], which uses G.726 in tandem with G.711 to determine a tolerable amount of distortion in the G.711 stream. In their paper, Ito et al. [14] neglected to mention scenarios involving packet loss. As G.726 is a stateful codec, the output is dependent on all previous and current inputs. Should a packet be lost, the state and outputs between the sender and receiver are no longer guaranteed to be equivalent. Without this guarantee, hidden data can no longer be reliably retrieved from the stream. We propose an algorithm that uses a similar technique to that of Ito et al. [14], but allowing for continued reliable retrieval from the stream after a packet has been lost. Some preliminary results have been published in [19, 20]. The rest of the

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 55–68, 2015.
DOI: 10.1007/978-3-662-46739-8_4

paper is organized as follows. In Sect. 2, we give a background about the G.711 codec. Section 3 presents the related work in the literature, while Sect. 4 explains our proposed packet loss tolerant algorithm. In Sect. 5 we demonstrate the evaluation of our algorithm. Finally, we conclude our paper in Sect. 6.

2 Background

G.711 is the oldest voice codec in use not only by today's VoIP systems but also domestic public switching telephone networks (PSTNs) [5]. Some standardization authorities consider G.711 mandatory for VoIP devices as a codec common to all systems for interoperability [6]. In regards to steganography, there appears to be a much greater wealth of available research on G.711 as opposed to other voice codecs. For these reasons, we opt to have our VoIP client use only G.711.

s1abcdefghijkl → s111abcd
s01abcdefghijk → s110abcd
s001abcdefghij → s101abcd
…
s0000001abcdef → s001abcd
s0000000abcdef → s000abcd

Fig. 1. Output of G.711 codec from signed linear audio input, prior to inversion of every other bit

As illustrated in Fig. 1, the G.711 codec takes signed linear audio samples at 8 kHz [7]. For each sample, the codec outputs one byte (eight bits); this results in a 64 kbits/s bit rate. There are three components in the output: the sign (positive or negative, represented by s in Fig. 1), the magnitude on a logarithmic scale and a sub-magnitude on a linear scale (represented by abcd in the figure). The first bit of output is the sign bit and is identical to the sign bit from the input. The second through fourth output bits represent the magnitude of the input on a logarithmic scale. The codec checks the linear sample input for how many "0" bits follow the sign bit before a "1" bit is found, with more "0" bits before the first "1" implying a lesser magnitude. If a "1" bit immediately follows the sign bit, the codec adds "111" to the output. If "01" immediately follows the sign bit, the codec adds "110" to the output, signifying a lesser magnitude than if "1" immediately followed the sign bit. This pattern continues until the case where at least seven consecutive "0" bits follow the sign bit, which would result in an output of "000" for logarithmic magnitude. The remaining 4 bits of output are set to the first four bits in the input that follow the bits used to determine logarithmic magnitude. In transit, systems typically invert every other bit – xor 01010101 (0 × 55) – this is not shown in the figure for simplicity.

3 Related Work

Much work has been done in regards to steganography. They could be classified into two categories. One hides data within TCP/IP packets and the other hides data within the application. The first category allows great flexibility in trying to secretly transmit data.

Ahsan and Kundur [8] showed some simple approaches to manipulating headers in IP packets and reordering packets in such a way that data could be encoded secretly. Reordering packets, while certainly acceptable in many scenarios, would likely create too much jitter for real time communications. Murdoch and Lewis [2] noted that prior work had evaluated the ability for one to use a given TCP or IP field for steganographic purposes, but had neglected to properly evaluate if such manipulation might be more obvious to active wardens. These are people who are familiar with typical TCP traffic on their network, have reason to suspect steganography is taking place and are actively inspecting traffic to look for deviations from the expected typical traffic. Murdoch and Lewis [2] then showed a steganographic approach using initial sequence numbers that takes into consideration methods used by popular operating systems and that would be indistinguishable from packets normally generated by those hosts. This, however, lacks suitability for our purposes, as VoIP typically utilizes UDP. Further, the throughput that could be achieved by only modifying a single packet over the lifetime of a connection is, while incredibly covert, insufficient for our intentions.

In the second category, research has been proposed to hide information on the application level with a focus on audio applications such as those running VoIP. Mazurczyk and Lubacz [9] suggested a method they dubbed Lost Audio paCKets steganography, or LACK. LACK functions by intercepting audio packets at a given interval, replacing the audio payload with data to be transmitted covertly, holding the packet for an interval long enough for a receiver to consider it lost, optionally adding additional jitter to avoid being caught by a simple statistical analysis, and finally sending the modified packet. The interval at which LACK intercepts packets should be based on which audio codec is in use and what rate of packet loss for that codec would result in unacceptable quality; that is to say, LACK is adjustable for any given codec. Optionally, this interval can also be adjusted based on the expected call duration (which can be estimated based on statistics and refined as the call progresses) and amount of data needed to be sent. Mazurczyk [10] thereafter tested the method, showing that G.711 appeared best suited for use with LACK, both due to it having the highest bit rate amongst the codecs tested and it encountering the least degradation of quality as the ratio of lost to total packets increased. At a packet interception rate of 5 %, which would probably lure the attention of someone observing traffic, G.711 still maintained fair mean opinion scores on listening quality objective (MOS-LQO) while achieving 3.2 kbits/s of covert traffic. There is a concern that a mechanism should be in place to ensure packets arriving late on purpose and packets arriving late due to other reasons beyond our control can be differentiated; the jitter introduced in the interval at which packets will be delayed and modified could be produced by a pseudorandom number generator (PRNG) with a seed known to both parties. There is also a concern that a mechanism needs to exist for recovering lost steganographic packets. Additionally, there is an expectation that there is a single static message of known length, which is used during the optional step of regulating the interception interval based on expected call duration and message length. Hamdaqa and Tahvildari [11] suggest ReLACK, which operates in very similar fashion to LACK, but uses a modified version of Shamir's Secret Sharing Scheme [12] on the message being transmitted, increasing the message size to the degree necessary for a specified fault-tolerance, which could be the amount of data that can be comfortably transmitted with LACK without unacceptable quality loss. With this scheme applied, if the receiver obtains at least as much data as was in the original message, the original message can be reconstructed, which addresses the possible issue of lost data. However, if the receiver doesn't obtain at least as much data as was in the original message, the entire message is lost.

Other methods in the literature attempt to subtly manipulate audio data as opposed to manipulating the flow of that audio data across the network and outright replacing the audio. One of the simplest approaches to steganography for many mediums is to tamper with the least significant bit (LSB) of each unit of data [21]. The sender simply replaces the LSB of a unit with the desired hidden bit to send, and the receiver reads the sent LSB. When using the LSB method, you expect that the modification of the least significant portion of the data will be negligible and go unnoticed by any observing parties.

An approach by Aoki [16] works in similar fashion to LSB in a special case. G.711 can transmit both a +0 and −0 signal, and Aoki's [16] algorithm takes advantage of this by using the sign bit as least significant when the magnitude is 0. This technique is virtually lossless in terms of audio quality, but quickly becomes ineffective in areas with moderate background noise. Additionally, Aoki [16] created a semi-lossless method, which works by increasing the absolute magnitude of all non-zero samples by a variable amount, j, allowing zero-magnitude samples to have both their sign bit and true LSBs manipulated for storing hidden data.

Among recent literature, both the method by Miao and Huang [13] and the method by Ito et al. [14] appear promising. While throughput of secret data would be dependent on many factors, including the nature of the audio being used as cover traffic, both of these methods advertise fairly good throughput. Miao and Huang [13] created an approach that isn't related to LSB-tampering. Their approach works by embedding more data in groups of samples that vary wildly (rapid fluctuations in sound samples) and less data in groups of samples that remain consistent and where distortions might be more apparent. For each group of N samples, the approach treats each G.711 sample as a sign bit followed by a seven bit magnitude integer, and obtains the average for the N samples. It then, for all but one sample, determines the difference between that sample and the average, and plans to embed a number of hidden bits equal to the log of the absolute value of the difference in that sample, rounded down. The sample will then be modified to equal the average plus an altered difference that will contain the hidden bits. The altered difference will be the sum of two numbers: some integer with a bit length equal to the number of bits to hide, plus two raised to the number of bits to hide. Finally, the one sample that was originally excluded will be manipulated to restore the average signed magnitude of the group back to the average originally calculated before anything was modified. This allows for a receiver to properly determine the number of bits that would be hidden in each sample, and subtract the correct average to recover the hidden integer in each sample.

The approach by Ito [14] uses a lower bit rate codec, G.726, in conjunction with G.711. The algorithm tests how many least significant bits of output from G.711 can be freely manipulated before transcoding to the lower bit rate codec begins producing different results. This assures the quality of the G.711 samples will at least match that of the lower bit rate codec. As both the sender and receiver for a given audio stream can use the same method to determine how many bits can be tampered with freely before unacceptable degradation would result, both the sender and receiver would come to the same conclusion and retrieve the correct tampered bits.

4 Packet Loss Tolerant Algorithm

In this section, we propose a new data hiding algorithm that is based on the algorithm proposed by Ito et al. [14] but has two main improvements in terms of reliability and throughput. The method suggested by Ito et al. [14] neglected to account for packet loss. As the lower bit rate codec Ito [14] used maintained state (that is, all inputs prior to the current sample being processed can affect the current output), it assumed that state would always match at the sender and receiver. If a scenario included packet loss as a feasible possibility, the sender and receiver would no longer have a guarantee of identical states if the entire audio stream was being considered. That is to say, should any packet loss occur when using the algorithm by Ito et al. [14], any hidden data starting with the first lost packet and extending to the end of the stream would be lost. As such, our proposed algorithm performs the information hiding on a packet-per-packet basis (resetting the state of the codec with each packet) as opposed to the entire stream of audio.

Our packet loss tolerant algorithm consists of three main functions. A function, tamperableBits, that determines the number of bits that could be tampered with without significantly affecting the quality of the audio, a function for embedding secret information into an outgoing audio stream and another for retrieving embedded secret information from an incoming audio stream. The pseudo code for determining a reasonable number of tamperable bits without significant distortion for a given sample is shown in Fig. 2. tamperableBits takes in a G.711 sample and a G.726 state. It outputs an updated G.726 state and the number of bits that can be tampered in the G.711 sample provided. The function named processCodec takes in a G.711 sample, along with the state of the G.726 codec, and outputs both an updated G.726 state and a G.726 output sample. In tamperableBits, the sample is first transcoded to the lower-bit rate codec without alteration. If the absolute value of the transcoding output is as high as possible (given the signed integers that can be represented with the number of bits used per sample by the lower-bit rate codec), this may suggest an overflow has occurred; that is, that the distortion between the expected value and the actual value is too large to be represented. If this is the case, the output from the lower-bit rate codec can’t be used to judge what samples are similar, and the safe option is to assume no bits should be tampered. If this isn’t the case, two copies of the sample are made. The least significant bit is set to 1 on the first copy and cleared (set to 0) on the second copy, we then test if transcoding both altered copies produce the same output. If so, the least significant bit can be tampered, and this check can be repeated for more significant bits in the linear sub-magnitude.

Pseudocode for embedding secret information into an outgoing audio stream is shown in Fig. 3. Prior to sending a VoIP packet with audio, it should be intercepted. First, a new state should be constructed for the lower-bitrate codec. Then, for every sample, the number of bits that can be freely tampered is determined by the tamperableBits algorithm explained above, and the bits are replaced.

```
// This function should take in a g711 sample prepared for transmission
// (every other bit inverted – 0x55) and the state for a lower-bit rate codec.
// It should output a sample transcoded with the lower-bit rate codec, and a
// modified state for the lower-bit rate codec.
function processCodec: IN g711, INOUT state, OUT codecOutput

// This function should take in a g711 sample not prepared for transmission
// and the state for a lower-bit rate codec. It should output the number of bits
// that can be freely modified before it would affect the output from transcoding
// to the lower-bit rate codec, along with the new state of the codec after
// processing a sample representing the lowest modified sample possible.


function tamperableBits: IN sample, INOUT codecState, OUT bits {
        bits ← 0
        lowTamper ← bitwise sample xor 0x55
        highTamper ← bitwise sample xor 0x55
        mask ← 1
        lastLowState ← codecState
        checkMaxDistort ← call processCodec: g711 ← lowTamper, state ↔ lastLowState
        if (absolute value of checkMaxDistort is less than the highest possible) {
                do {
                        lowTamper ← bitwise lowTamper xor 0x55
                        highTamper ← bitwise highTamper xor 0x55
                        lowTamper ← bitwise lowTamper and not mask
                        highTamper ← bitwise highTamper or mask
                        lowTamper ← bitwise lowTamper xor 0x55
                        highTamper ← bitwise highTamper xor 0x55
                        mask ← mask bitwise shifted left 1
                        stateCopy ← codecState
                        highResult ← call processCodec: g711 ← highTamper, state ↔ stateCopy
                        stateCopy ← codecState
                        lowResult ← call processCodec: g711 ← lowTamper, state ↔ stateCopy
                        if (lowResult doesn't equal highResult) {
                                break do-while
                        }
                        lastLowState ← stateCopy
                        bits ← bits + 1
                } while (bits is less than 4)
        }
        codecState ← lastLowState
}
```

Fig. 2. Method to determine number of bits that can be manipulated

The pseudo code for retrieving embedded secret information from an incoming audio stream is similar. For each sample the number of bits that could be freely tampered is determined then extracted as illustrated in Fig. 4.

Using the steganography technique we proposed, we can embed secret data of our choosing into the G.711 stream. Once the audio samples have been modified to contain our information, multiple samples can be bundled into each Real-Time Protocol (RTP)

```
// Should return the next bit of data to be secretly embedded.
function nextBitToInsert: OUT bit

// To be run when packets are available:
while (a new packet is available) {
        samples ← G.711 samples from the packet (every other bit inverted)
        state ← new state for low bitrate codec
        for (each sample in samples) {
                noxmit ← bitwise sample xor 0x55
                bits ← call tamperableBits: sample ← noxmit, codecState ↔ state
                mask ← 1
                for (i from 1 to bits) {
                        noxmit ← bitwise noxmit and not mask
                        insert ← call nextBitToInsert
                        insert ← insert * mask
                        noxmit ← bitwise noxmit or insert
                        mask ← mask bitwise shifted left 1
                }
                sample ← bitwise noxmit xor 0x55
        }
}
```

Fig. 3. Method to embed secret data into an audio packet

packet as normal by the VoIP software and sent to their destination. RTP typically uses UDP, so we risk not receiving a packet either due to a packet not arriving or a packet arriving with an invalid checksum.

5 Evaluation

We collected several audio recordings for testing. These audio recordings were grouped into three categories: no voice with low background noise, high volume voice with low background noise and high volume voice with moderate background noise. The no voice with low background noise category had two recordings. The first recording was a muffled thunderstorm recorded in a sealed building, with volume comparable to light background noise. The second was computer-generated silence. The high volume voice with low background noise category included automated voice mail and operator prompts (VM prompts), and English as a second language study material (ESL). Finally, the high volume voice with moderate background noise category included our peers speaking a short story (Peers) and recordings of telephone calls considered of historical significance and publically available (Historic).

To test the algorithms, we used three performance metrics:

1. **Throughput**: which is the number of bits of the secret data embedded per second.
2. **Noise-to-Signal ratio**: which is calculated according to the following equation,

```
// Should receive the next bit secretly transmitted for processing.
function **nextBitInserted**: IN *bit*

// To be run when packets are available:
while (a new packet is available) {
    *samples* ← G.711 samples from the packet
    *state* ← new state for low bitrate codec
    for (each *sample* in *samples*) {
        *noxmit* ← bitwise *sample* xor 0x55
        *bits* ← call **tamperableBits**: *sample* ← *noxmit*, *codecState* ↔ *state*
        *mask* ← 1
        for (*i* from 1 to *bits*) {
            *recovered* ← bitwise *noxmit* and *mask*
            if (*recovered* equals *mask*) {
                call **nextBitInserted**: bit ← 1
            } else {
                call **nextBitInserted**: bit ← 0
            }
            *mask* ← *mask* bitwise shifted left 1
        }
    }
}
```

Fig. 4. Method to retrieve secret data embedded into an audio packet

$$\frac{\sum_{i=1}^{n}\left\{\left(\frac{A_N}{A_S}\right)^2\right\}}{n} \tag{1}$$

Where:

- S denotes the Signal
- N denotes the Noise
- A_S is the original linear amplitude of a given sound sample (G.711 has 8,000 samples per second)
- A_N is the maximum difference between the amplitude of the original sample and the sample after embedding the data on a linear scale
- n is the number of samples

3. **Perceptual Evaluation of Speech Quality (PESQ)**: PESQ algorithm [15] compares unmodified and degraded audio in a way that aims to report how degraded a human would perceive the audio to be. Higher scores are better, with scores of at least 4 considered good and scores of at least 3 considered acceptable but with noticeable degradation.

A higher throughput, a higher PESQ and a lower Noise-to-Signal ratio means better performance.

We first studied the performance of the algorithms in the worst case scenario, where embedding data would result in the greatest possible noise-to-signal ratio.

For Aoki's [16] approach, we evaluate it in lossless (LL) and semi-lossless (SLL) modes. In the former, only the sign bit is manipulated when the magnitude is 0; in the latter, the sign bit and the least significant bit of the magnitude is manipulated i.e. the variable j in [16] is set to 1.

For the algorithm by Ito et al. [14], G.726 could operate at four different bitrates, namely, 16 kbits/s, 24 kbits/s, 32 kbits/s and 40 kbits/s. Using the 32 kbits/s mode offered more throughput than the 40 kbits/s mode without much additional noise. Using 24 kbits/s and 16 kbits/s compared to 32 kbits/s again offered additional throughput, but with a substantial noise increase. We will show results for 32 kbits/s and 24 kbits/s for both the algorithm by Ito et al. [14] and our algorithm.

For the approach by Miao and Huang [13], we use values of 5 and 13 for N, and a value of 96 for the maximum lambda. 13 was the N value used in [13] when presenting results of their algorithm. Their paper suggested that lowering the value of N would increase hidden throughput at the cost of noise; we chose an N of 5 to see this effect. The maximum lambda serves as a mechanism to prevent overflow – should any embedding operation have the potential to cause a sample to vary from the average magnitude more than the maximum lambda, no bits will be embedded in that sample. Should the maximum lambda be exceeded when adjusting the sample used to restore the average, all embedding operations for the entire group will be canceled, and the audio for the entire group will be unmodified.

5.1 Throughput

Figures 5, 6 and 7 show the average throughput of secret data for no voice with low background noise recordings, the high volume voice with low background noise and the high volume voice with moderate background noise categories, respectively. The Figures assume a no packet loss scenario. Aoki's [16] algorithms perform well under

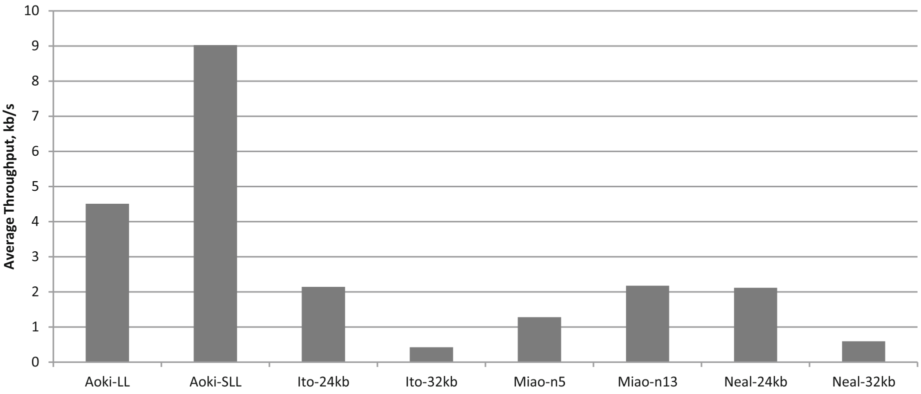


Fig. 5. Average throughput for no voice low background voice recordings

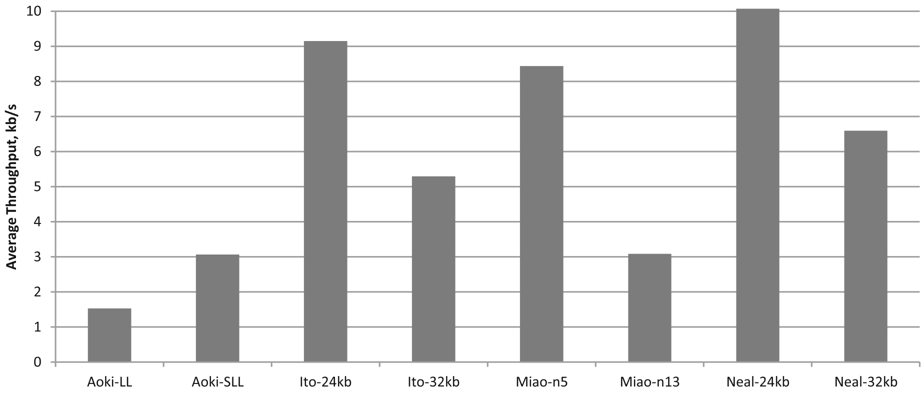


Fig. 6. Average throughput for high voice volume, low background noise recordings

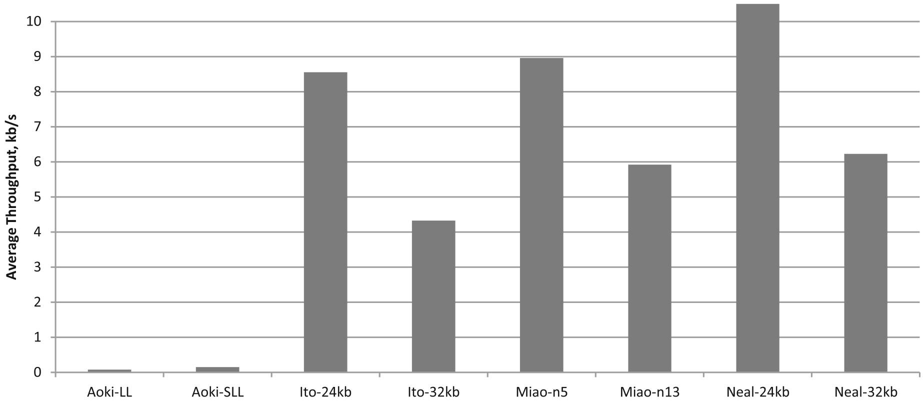


Fig. 7. Average throughput for high voice volume, moderate background noise recordings

no voice and low background noise conditions but poorer under high volume voice conditions. Our algorithm operating at 24 kb/s, outperforms other algorithms under high volume voice conditions, but poorly with generated silence. For Miao and Huang's [13] algorithm, a group of five samples (that is, N = 5) provided better performance than N = 13 for all recordings with the exception of Thunderstorm; for the Thunderstorm recording, N = 13 provided much better performance compared to N = 5.

5.2 Noise to Signal Ratio

In Fig. 8, we show average Noise-to-Signal ratios on modified audio files. The method by Miao and Huang [13] generated the most noise in every instance, with N = 13 generating more noise than N = 5. As N = 5 also provides better throughput in all but one case as well as less noise in all cases, the suggestion by Miao and Huang [13] in their paper to keep N small appears valid. In both recordings from the high volume voice and moderate background noise category (peers and historic), the algorithm by Ito et al. [14] and our algorithm generates more noise than Aoki's [16], but, as shown in Fig. 7, Aoki's generates substantially less throughput. In the high volume voice and low background noise category (VM prompts and ESL), our algorithm generated more noise than Aoki's for VM recording and less noise in the ESL recording, but for both recordings Aoki's [16] algorithm generated less throughput (see Fig. 6). In the no voice with low background noise category, Aoki's [16] algorithm generated more noise than ours, but generated substantially more throughput (see Fig. 5). Neither Ito et al.'s algorithm [14] nor our algorithm made any changes for the silence recording, hence not having a data point in Fig. 8.

5.3 Audio Quality

In Fig. 9, we show PESQ results. Aoki's [16] algorithms score at least 3.0 in all cases and above 4.0 in recordings from the high voice volume with moderate background noise category, suggesting acceptable to good audio quality. Both our and Ito's algorithm consistently score above 4, suggesting consistent good quality. Miao and Huang [13] 's method produced modified audio that scored less than 3.0 for multiple (though not all) recordings, suggesting poor to adequate quality.

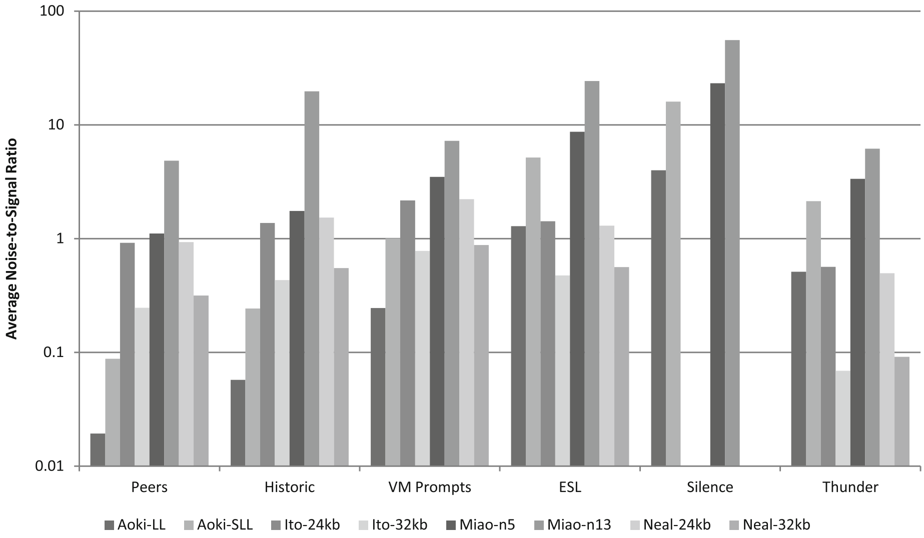


Fig. 8. Average noise-to-signal ratio for recordings

5.4 Effects of Packet Loss

In a more practical scenario where packet loss can occur, the algorithm by Ito et al. [14] quickly levels off and fails to reliably transmit any further data, while our algorithm continue transmitting with graceful degradation as shown in Fig. 10.

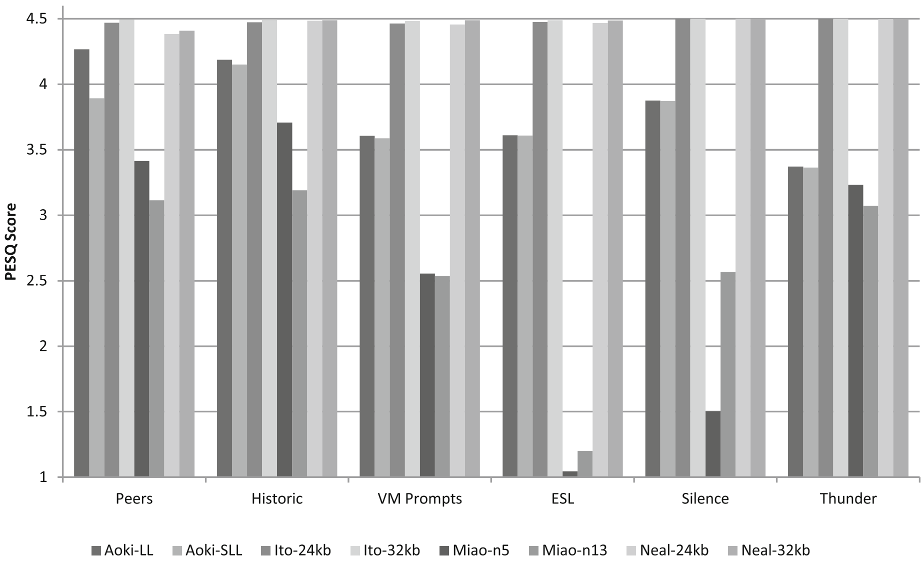


Fig. 9. PESQ scores for recordings

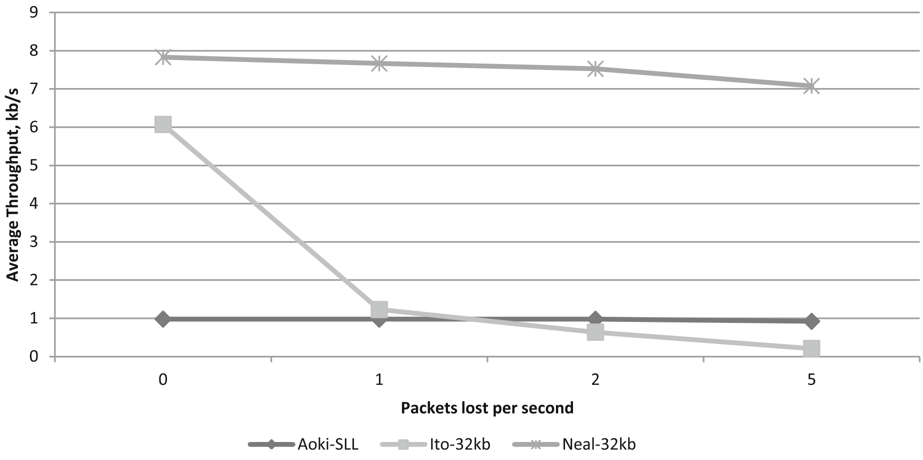


Fig. 10. VM Prompts Sample considering packet loss for 5 s

6 Conclusion and Future Work

In this paper we not only proposed a new algorithm to hide data in a Voice over IP stream but also evaluated several suggested algorithms in this field from the literature, namely those by Aoki [16], Miao and Huang [13] and Ito et al. [14]. As far as our knowledge, this research offers the most comprehensive performance analysis of VoIP steganography algorithms as far as the number of algorithms considered and the performance metrics used. Our evaluation shows that the algorithm by Aoki [16] worked nicely in no voice conditions but far less favorably otherwise compared to other algorithms. On the other hand, other methods do not produce any throughput with artificial silence, while Aoki's [16] method excels. The method by Miao and Huang [13] tends to perform better with a lower N value but generates excessive noise.

We proposed an algorithm based on the work of Ito et al. [14] that dealt with packet loss more appropriately. Our experiments show that our proposed algorithm has several advantages over algorithms found in the literature. The most important advantage is that, unlike other algorithms, it can be used in a practical environment where packet loss is inevitable. It maintains high throughput, low noise levels and high PESQ scores. That is, it maintains a good audio quality on par with or superior to other algorithms found in the literature, in addition, it gracefully degrades as packet loss rate increases.

One drawback of the proposed method is that, similar to all other LSB based steganography techniques, it can be detected by a good steganalysis program.

As VoIP conversations can involve both voice and video, one could extend this research to embed the data not only into the audio stream but also the video stream for a given call, if applicable.

References

1. Artz, D.: Digital steganography: hiding data within data. IEEE Internet Comput. **5**, 75–80 (2001)
2. Murdoch, S.J., Lewis, S.: Embedding covert channels into TCP/IP. In: Barni, M., Herrera-Joancomarti, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 247–261. Springer, Heidelberg (2005)
3. Tian, H., et al.: An adaptive steganography scheme for voice over IP. Huazhong University of Science & Technology, University of Nebraska, Tsinghua University (2009) doi:10.1109/ISCAS.2009.5118414
4. Yongfeng, H., Bo, X., Honghua, X.: Implementation of covert communication based on steganography. Department of Electronic Engineering, Tsinghua University, Beijing (2008). doi:10.1109/IIH-MSP.2008.174
5. Karapantazis, S., Pavlidou, F.-N.: VoIP: a comprehensive survey on a promising technology. Thessaloniki (2009). doi:10.1016/j.comnet.2009.03.010
6. International Telecommunication Union: Packet-based multimedia communications (Recommendation ITU-T H.323) (2009)
7. International Telecommunication Union: Pulse Code Modulation (PCM) of Voice Frequencies (ITU-T Recommendation G.711) (1993)
8. Ahsan, K., Kundur, D.: Practical data hiding in TCP/IP. University of Toronto, Toronto (2002). 1-58113-000-0/00/0000
9. Mazurczyk, W., Lubacz, J.: LACK - a VoIP steganographic method. Institute of Telecommunications, Warsaw University, Warsaw (2009). doi:10.1007/s11235-009-9245-y
10. Mazurczyk, W.: Lost audio packets steganography: the first practical evaluation. Warsaw University of Technology, Institute of Telecommunications, Warsaw, arXiv:1107.4076v1 (2011)
11. Hamdaqa, M., Tahvildari, L.: ReLACK: a reliable VoIP steganography approach. In: IEEE Fifth International Conference on Secure Software Integration and Reliability Improvement, Jeju Island, pp. 189–197 (2011)

12. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
13. Miao, R., Huang, Y.: An approach of covert communication based on the adaptive steganography scheme on voice over IP. Department of Electronic Engineering, Tsinghua University, Beijing (2011). ISBN: 978-1-61284-231-8
14. Ito, A., Abe, S., Suzuki, Y.: Information hiding for G.711 speech based on substitution of least significant bits and estimation of tolerable distortion. Tohoku University, Sendai (2009). ISBN: 978-1-4244-2354-5
15. International Telecommunication Union: Perceptual evaluation of speech quality (Recommendation ITU-T P.862) (2001)
16. Aoki, N.: A band extension technique for G.711 speech using steganography. IEICE Trans. Commun. **E89-B**(6), 1896–1898 (2006)
17. International Telecommunication Union: Perceptual objective listening quality assessment (ITU-T Recommendation P.863) (2011)
18. CenturyLink: CenturyLink IP Network Statistics, December 2011. https://kai02.centurylink.com/PtapRpts/Public/BackboneReport.aspx
19. Neal, H., ElAarag, H.: A packet loss tolerant algorithm for information hiding in voice over IP. In: Proceedings of IEEE Southeast Conference, Orlando, FL, 15–18 March 2012
20. ElAarag, H., Neal, H.: Performance analysis of current data hiding algorithms for VoIP. In: Proceedings of the Communication and Networking Simulation Symposium, Spring Simulation Multiconference, San Diego, CA, 7–10 April 2013
21. Latzenbeisser, S.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, New York (2000)

Adaptive Steganography and Steganalysis with Fixed-Size Embedding

Benjamin Johnson[1,5], Pascal Schöttle[2], Aron Laszka[3(✉)], Jens Grossklags[4], and Rainer Böhme[2]

[1] Cylab, Carnegie Mellon University, Pittsburgh, USA
[2] Department of Information Systems, University of Münster, Münster, Germany
[3] Institute for Software Integrated Systems,
Vanderbilt University, Nashville, USA
aron.laszka@vanderbilt.edu
[4] College of Information Sciences and Technology,
Pennsylvania State University, University Park, USA
[5] School of Information, University of California, Berkeley, Berkeley, USA

Abstract. We analyze a two-player zero-sum game between a steganographer, Alice, and a steganalyst, Eve. In this game, Alice wants to hide a secret message of length k in a binary sequence, and Eve wants to detect whether a secret message is present. The individual positions of all binary sequences are independently distributed, but have different levels of predictability. Using knowledge of this distribution, Alice randomizes over all possible size-k subsets of embedding positions. Eve uses an optimal (possibly randomized) decision rule that considers all positions, and incorporates knowledge of both the sequence distribution and Alice's embedding strategy.

Our model extends prior work by removing restrictions on Eve's detection power. We give defining formulas for each player's best response strategy and minimax strategy; and we present additional structural constraints on the game's equilibria. For the special case of length-two binary sequences, we compute explicit equilibria and provide numerical illustrations.

Keywords: Game theory · Content-adaptive steganography · Security

1 Introduction

In steganography, the objective of a steganographer is to hide a secret message in a communication channel. The objective of her counterpart, the steganalyst, is to detect whether the channel contains a message. Digital multimedia, such as JPEG images, are the most commonly studied communication channels in this context; but the theory can be applied more generally to any data stream having some irrelevant components and an inherent source of randomness [10].

In contrast to random uniform embedding, where the steganographer chooses her message-hiding positions along a pseudo-random path through the

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 69–91, 2015.
DOI: 10.1007/978-3-662-46739-8_5

communication channel, content-adaptive steganography leverages the fact that different parts of a communication channel may have different levels of predictability [2,4]. All content-adaptive embedding schemes have in common that they try to identify less predictable embedding positions. These schemes can be roughly divided into locally calculated criteria and distortion minimizing criteria. An example for the first category is the assumption that areas with a high local variance are more suitable, e.g., [13]. The second category assumes that embedding positions introducing less distortion are preferable, e.g., [15]. The claimed purpose of all adaptivity criteria is to identify a (partial) ordering of all available embedding positions according to their suitability for embedding.

For example, digital images often have areas of homogeneous color where any slight modification would be noticed, whereas other areas are heterogeneous in color so that subtle changes to a few pixels would still appear natural. It follows that if a steganographer wants to modify image pixels to communicate a message, she should prefer to embed in these heterogeneous areas.

Our model abstracts this concept of content-adaptivity, by considering a communication channel as a random variable over binary sequences, where each position in the sequence has a different level of predictability. The predictability of each position is observable by both Alice, a content-adaptive steganographer, and Eve, a computationally-unbounded steganalyst; and we apply a game-theoretic analysis to determine each player's optimal strategy for embedding and detection, respectively.

We show that if Alice changes exactly k bits of a binary cover sequence, then Eve's best-response strategy can be expressed as a multilinear polynomial inequality of degree k in the sequence position variables. In particular, when $k = 1$, this polynomial inequality is a linear aggregation formula similar to what is typically used in practical steganalysis, e.g., [11]. Conversely, given any strategy by Eve to separate cover and stego objects, Alice has a best-response strategy that minimizes a relatively-simple summation over Eve's strategic choices. We give formulas for both players' minimax strategies, and explain why the straightforward linear programming solution for computing these strategies is not efficiently implementable for realistic problem sizes. We give structural constraints to the players' equilibrium strategies; and in the case where there are only two embedding positions, we classify all equilibria, resolving an open question from [30]. Furthermore, we bridge the two research areas of game-theoretic approaches and information-theoretic optimal steganalysis, and conjecture that the main results of earlier works still hold when the steganalyst is conservatively powerful.

The rest of the paper is organized as follows. In Sect. 2, we briefly review related work. In Sect. 3, we describe the details of our game-theoretic model. Section 4 contains our analysis of the general case; and in Sect. 5, we compute and illustrate the game's equilibria for the special case of sequences of length two. We conclude the paper in Sect. 6.

2 Related Work

Game theory is a mathematical framework to investigate competition between strategic players with contrary goals [33]. Game theory gains more and more importance in practically all areas concerned with security ranging from abstract models of security investment decisions [14,17] to diverse applied scenarios such as the scheduling of patrols at airports [29], the modeling of Phishing strategies [6], network defense [23], and team building in the face of a possible insider threat [22].

The application of game theory has also found consideration in the various subdisciplines of information hiding including research on covert channels [16], anonymity [1], watermarking [24] and, of course, steganography.[1] Similarly, game-theoretic approaches can be found in the area of multimedia forensics [3,32].

In content-adaptive steganography [4], where Alice chooses the positions into which she embeds a message and Eve tries to anticipate these positions to better detect the embedding, the situation is naturally modeled using game theory.

Practical content-adaptive steganography schemes, on the other hand, have typically relied primarily on the notion of unpredictability to enhance the security of embedded messages. In fact, the early content-adaptive schemes not only preferred less predictable areas of images, but restricted all embedding changes to the least predictable areas, e.g., [9]. Prior works examining adaptive embedding have dubbed this strategy *naïve adaptive embedding*, and have shown it to be a non-optimal strategy in progressively more general settings [5,18,30]. It was shown in [5] that the steganalyst can leverage her knowledge about the specific adaptive embedding algorithm from [9] to detect it with better accuracy than even random uniform embedding. In [30] it was shown for the first time that, if the steganalyst is strategic, it is never optimal for the steganographer to deterministically embed in the least predictable positions. The game-theoretic analysis in [30] was restricted to a model with two embedding positions, where Eve could only look in one position. A subsequent extension of that model [18] allowed the steganographer to change multiple bits in an arbitrary-sized cover sequence, but maintained limiting restrictions on the power of the steganalyst, by requiring her to make decisions on the basis of only one position. Another extension generalizes the model by introducing a non-uniform cost of steganalysis and models the problem as a quasi-zero-sum game [21].

Another extension of this research stream expanded the power of Eve but required Alice to embed independently in each position [31]. Other authors have studied steganography using game-theoretical models. In 1998, Ettinger [8] proposed a two-player, zero-sum game between a steganographer and an active steganalyst whose purpose it is to interrupt the steganographic communication; Ker [20] uses game theory to find strategies in the special case of batch steganography, where the payload can be spread over many cover objects. The steganalyst anticipates this and tries to detect the existence of any secret message (so-called pooled steganalysis); and Orsdemir et al. [26] frame the competition between

[1] See [27] for an introduction to the area of information hiding.

steganographer and steganalyst with the help of *set theory*. The steganographer has the possibility to use either a naïve or a sophisticated strategy, where in the sophisticated strategy she incorporates statistical indistinguishability constraints. By this they devise a meta-game. The only other game-theoretical approach that is also concerned with content-adaptive embedding, the most common approach in modern steganography, e.g., [12,28], is [7]. Here, the authors examine the embedding operation of LSB matching with a content-adaptive embedding strategy and a multivariate Gaussian cover model.

This work directly extends [19], which first introduced the game theoretic model studied in this paper. Compared to that work, we have added several new results constraining the game's equilibrium strategies. First, we give formal constraints determining when the game admits or does not admit trivial equilibria. We use these constraints to show that under the non-trivial conditions, Alice can affect her payoff by changing her embedding strategy at key positions. Finally, we use these structural results to prove that under relatively general conditions, it is not optimal against an adaptive classifier to naïvely embed in the least biased positions. As an additional contribution, we give a constructive proof that our simplified representation of Eve's mixed strategy is a surjective reduction.

3 Game-Theoretic Model

To describe our game-theoretic model, we specify the set of players, the set of states that the world can be in, the set of choices available to the players, and the set of consequences as a result of these choices. Because our game is a randomized extension of a deterministic game, we first present the structure of the deterministic game, and follow up afterwards with details of the randomization.

3.1 Players

The players are Alice, a steganographer, and Eve, a steganalyst. Alice wants to send a message through a communication channel, and Eve wants to detect whether the channel contains a message. At times, we find it convenient to also mention Nature, the force causing random variables to take realizations, and Bob, the message recipient; although Nature and Bob are not players in a game-theoretic sense because they are not strategic.

3.2 Events

Our event space Ω is the set $\{0,1\}^N \times \{C, S\}$. An event consists of two parts: a binary sequence $x \in \{0,1\}^N$ and a steganographic state $y \in \{C, S\}$, where C stands for *cover* and S for *stego*. The binary sequence represents what Eve observes on the communication channel. The steganographic state tells whether or not a message is embedded in the sequence. In the randomized game, neither of these two states is known by the players until after they make their choices.

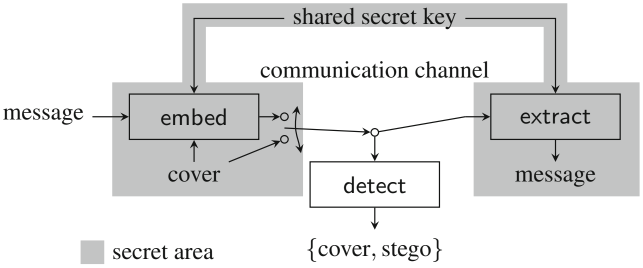


Fig. 1. Block diagram of a steganographic communication system

To define payoffs for the finite game, we simply assume that some event has been chosen by Nature so that the world is in some fixed state (x, y).

Figure 1 illustrates an event with player interaction as a block diagram. Following the diagram, Alice embeds a secret message of length k into the binary sequence x; Nature determines whether the original cover or the modified stego object appears on the communication channel; Eve observes the sequence appearing on the channel and makes a decision as to whether or not it contains a message; and (not relevant to our analysis but useful for narrative closure) Bob extracts the message, if it happened to be there.

3.3 Choices

Alice's (pure strategy) choice is to select a size-k subset I of $\{0, \ldots, N-1\}$, which represents the positions into which she embeds her encoded message, by flipping the value of the given sequence at each of the positions in I.

Eve's (pure strategy) choice is to select a subset E_S of $\{0,1\}^N$, which represents the set of sequences that she classifies as stego objects (i.e., sequences containing a secret message). Objects in $E_C := \{0,1\}^N \setminus E_S$ are classified as cover objects (i.e., sequences not containing a secret message).

3.4 Consequences

Suppose that Alice chooses a pure strategy $I \subseteq \{0, \ldots, N-1\}$, Eve chooses a pure strategy $E_S \subseteq \{0,1\}^N$, and Nature chooses a binary sequence x and a steganographic state y. Then, Eve wins 1 if she classifies x correctly (i.e., either she says stego and Nature chose stego, or she says cover and Nature chose cover), and she loses 1 if her classification is wrong. The game is zero-sum so that Alice's payoff is the negative of Eve's payoff. Table 1 formalizes the possible outcomes as a zero-sum payoff matrix.[2]

[2] The payoff matrix and the zero sum property might be different if false positives and false negatives result in different profits, respectively losses.

Table 1. Payoffs for (Eve, Alice)

Eve's decision for x	Steganographic state	
	C	S
$x \in E_C$	$(1, -1)$	$(-1, 1)$
$x \in E_S$	$(-1, 1)$	$(1, -1)$

3.5 Randomization

In the full randomized game, we have distributions on binary sequences and steganographic states. We also have randomization in the players' strategies. To describe the nature of the randomness, we start by defining two random variables on our event space Ω. Let $X : \Omega \to \{0, 1\}^N$ be the random variable which takes an event to its binary sequence and let $Y : \Omega \to \{C, S\}$ be the random variable which takes an event to its steganographic state. We proceed through the rest of this section by first describing the structure of the distribution on Ω; next describing the two players' mixed strategies; and finally, by giving the players' payoffs as a consequence of their mixed strategies.

Steganographic States. The event $Y = S$ happens when Nature chooses the steganographic state to be stego; and this event occurs with probability p_S. We also define $\Pr_\Omega[Y = C] := p_C = 1 - p_S$. From Eve's perspective, p_S is the prior probability that she observes a stego sequence on the communication channel. A common convention in steganography (following a similar convention in cryptography) is to equate the prior probabilities p_C and p_S of the two steganographic states, so that Eve observes a stego sequence with exactly 50 % probability. Our results describing equilibria for this model carry through with arbitrary prior probabilities; so we retain the notations p_S and p_C in several subsequent formulas. Note however, that with highly unequal priors, the game may trivialize because the prior probabilities can dominate other incentives. For this reason, we do require equal priors for some structural theorems; and we also use equal priors in our numerical illustrations.

Binary Sequences. The distribution on binary sequences depends on the value of the steganographic state. If $Y = C$, then the steganographic state is cover, and X is distributed according to a *cover distribution* $\mathcal{C}$; if $Y = S$, then the steganographic state is stego, and X is distributed according to a *stego distribution* $\mathcal{S}$.

With this notation in hand, we may define, for any event $(X = x, Y = y)$:

$$\begin{aligned} \Pr_\Omega[(x, y)] &= \Pr_\Omega[Y = y] \cdot \Pr_\Omega[X = x | Y = y] \\ &= \begin{cases} p_C \cdot \Pr_\mathcal{C}[X = x] & \text{if } y = C \\ p_S \cdot \Pr_\mathcal{S}[X = x] & \text{if } y = S. \end{cases} \end{aligned} \tag{1}$$

We will define the distributions $\mathcal{C}$ and $\mathcal{S}$ after describing the players' mixed strategies.

Players' Mixed Strategies. We next describe the mixed strategy choices for Alice and Eve. Recall that a mixed strategy is a probability distribution over pure strategies.

In a mixed strategy, Alice can probabilistically embed into any given subset of positions, by choosing a probability distribution over size-k subsets of $\{0, \ldots, N-1\}$. To describe a mixed strategy, for each $I \subseteq \{0, \ldots, N-1\}$, we let a_I denote the probability that Alice embeds into each of the positions in I.

A mixed strategy for Eve is a probability distribution over subsets of $\{0,1\}^N$. Suppose that Eve's mixed strategy assigns probability e_S to each subset $S \subseteq \{0,1\}^N$. Overloading notation slightly, we define $e : \{0,1\}^N \rightarrow [0,1]$ via

$$e(x) = \sum_{S \subseteq \{0,1\}^N : x \in S} e_S \quad . \tag{2}$$

Each $e(x)$ gives the total probability for the binary sequence x that Eve classifies the sequence x as stego. Note that this "projected" representation of Eve's mixed strategy given in Eq. (2) requires specifying 2^N real numbers, whereas the canonical representation of her mixed strategy using the notation e_S would require specifying 2^{2^N} real numbers. For this reason, we prefer to use the projection representation. Fortunately, the projected representation contains enough information to determine both players' payoffs, because it determines the classifier's success rates. In the reverse direction, we may also construct a true mixed strategy from a reduced representation, as evidenced by the subsequent lemma.

Reduced Representation of Eve's Mixed Strategy. The following lemma shows that the mapping from the canonical representation of Eve's mixed strategy to the projected representation is surjective, so we may express results using the simpler representation without loss of generality.

Lemma 1. *For every function $e : \{0,1\}^N \mapsto [0,1]$, there exists a distribution e_S, $S \subseteq \{0,1\}^N$, satisfying Eq. (2).*

Proof. We prove the above lemma using a constructive proof. More specifically, we provide an algorithm that can compute an appropriate distribution e_S, $S \subseteq \{0,1\}^N$, from an arbitrary function $e : \{0,1\}^N \mapsto [0,1]$. First, order the sequences by their $e(x)$ values in a non-increasing order, and denote them $x^1, x^2, \ldots, x^{2^N}$ (i.e., without loss of generality, assume $e(x^1) \geq e(x^2) \geq \ldots \geq e(x^{2^N})$). Second, assign probabilities to subsets of sequences as follows. Let the first subset of sequences be $S^0 = \{\}$, and let its probability be $e_{S^0} = 1 - e(x^1)$. Next, let the second subset be $S^1 = \{x^1\}$, and let its probability be $e_{S^1} = e(x^1) - e(x^2)$. Then, let the third subset be $S^2 = \{x^1, x^2\}$ and its probability be $e_{S^2} = e(x^2) - e(x^3)$. Similarly, let the $(k+1)$th subset be $S^k = \{x^1, x^2, \ldots, x^k\}$, and let its probability be $e_{S^k} = e(x^k) - e(x^{k+1})$. Finally, let the last subset be $S^{2^N} = \{x^1, x^2, ..., x^{2^N}\}$, and let its probability be $e_{S^{2^N}} = e(x^{2^N})$.

We have to show that the output of the algorithm 1) is a distribution (i.e., the probabilities sum up to one) and 2) satisfies Eq. (2). First, the sum of the resulting probabilities is

$$\begin{aligned} & e_{S^0} + e_{S^1} + e_{S^2} + \ \ldots \ + e_{S^{2^N}} && (3)\\ =& 1 - e(x^1) + e(x^1) - e(x^2) + e(x^2) - e(x^3) + \ \ldots \ + e(x^{2^N}) && (4)\\ =& 1. && (5) \end{aligned}$$

Second, for an arbitrary sequence x^k, we have

$$\begin{aligned} \sum_{S \subseteq \{0,1\}^N \ : \ x^k \in S} e_S =& \sum_{l=k}^{2^N} e_{S^l} && (6)\\ =& e(x^k) - e(x^{k+1}) + e(x^{k+1}) - e(x^{k+2}) + \ldots + e(x^{2^N}) && (7)\\ =& e(x^k). && (8) \end{aligned}$$

Therefore, we have that the resulting distribution satisfies Eq. (2), which concludes our proof. □

Note that the resulting distribution is relatively simple, since it assigns a non-zero probability to at most $N^2 + 1$ subsets only (and even less than that if some sequences have zero $e(x)$ values). It is easy to see that we cannot do any better than this generally, in the sense that there exists an infinite number of e functions, for which no distribution with a smaller support can exist.

Cover Distribution. In the cover distribution $\mathcal{C}$, the coordinates of X are independently distributed so that

$$\Pr_{\mathcal{C}}[X = x] = \prod_{i=0}^{N-1} \Pr_{\mathcal{C}}[X_i = x_i]. \tag{9}$$

The bits are not identically distributed however. For each i we have

$$\Pr_{\mathcal{C}}[X_i = 1] = f_i, \tag{10}$$

where $\langle f_i \rangle_{i=0}^{N-1}$ is a monotonically-increasing sequence from $\left(\frac{1}{2}, 1\right)$. Note that this assumption is without loss of generality because, in applying the abstraction of a communication channel into sequences, we can always flip 0 s and 1s to make 1 s more likely; and we can re-order the positions from least to most predictable.

For notational convenience, we define

$$\tilde{f}_i = 2f_i - 1. \tag{11}$$

We construe $\tilde{f}_i$ as a measure of the bias of the predictability at position i. If the bias at some position is close to zero, then the value of that position is not very predictable, while if the bias is close to 1, the value of the position is very predictable.

Putting it all together, the cover distribution is defined by

$$\begin{aligned}\Pr_{\mathcal{C}}[X = x] &= \prod_{x_i=1} f_i \cdot \prod_{x_i=0} (1 - f_i) \\ &= \prod_{i=0}^{N-1} \big(1 - f_i + x_i \tilde{f}_i\big). \qquad (12)\end{aligned}$$

Stego Distribution. The stego distribution $\mathcal{S}$ depends on Alice's choice of an embedding strategy. Let $I \subseteq \{0, \ldots, N-1\}$, and for each $x \in \{0,1\}^N$ let x_I denote the binary sequence obtained from x by flipping the bits at all the positions in I. The stego distribution is obtained from the cover distribution by adjusting the likelihood that each x occurs, assuming that for each I, with probability a_I Alice flips the bits of x in all the positions in I.

More formally, suppose that Alice embeds into each subset $I \subseteq \{0, \ldots, N-1\}$ with probability a_I. We then have

$$\begin{aligned}\Pr_{\mathcal{S}}[X = x] &= \sum_I a_I \cdot \Pr_{\mathcal{C}}[X = x_I] \\ &= \sum_I a_I \cdot \prod_{i \notin I} \Pr_{\mathcal{C}}[X_i = x_i] \cdot \prod_{i \in I} \Pr_{\mathcal{C}}[X_i = 1 - x_i] \\ &= \sum_I a_I \cdot \prod_{i \notin I} \Big(1 - f_i + x_i \tilde{f}_i\Big) \cdot \prod_{i \in I} \Big(f_i - x_i \tilde{f}_i\Big). \qquad (13)\end{aligned}$$

Player Payoffs. In the full game, the expected payoff for Eve can be written as:

$$\begin{aligned}u(Eve) = \quad & \Pr_{\Omega}[X \in E_S \text{ and } Y = S] && \text{(true positive)} \\ + & \Pr_{\Omega}[X \in E_C \text{ and } Y = C] && \text{(true negative)} \\ - & \Pr_{\Omega}[X \in E_S \text{ and } Y = C] && \text{(false positive)} \\ - & \Pr_{\Omega}[X \in E_C \text{ and } Y = S] && \text{(false negative)} \qquad (14)\end{aligned}$$

and this can be further computed as

$$\begin{aligned}u(Eve) &= p_S \Pr_{\mathcal{S}}[X \in E_S] + p_C \Pr_{\mathcal{C}}[X \in E_C] - p_C \Pr_{\mathcal{C}}[X \in E_S] - p_S \Pr_{\mathcal{S}}[X \in E_C] \\ &= \sum_{x \in \{0,1\}^N} \Big[e(x) p_S \Pr_{\mathcal{S}(a)}[X = x] \\ &\qquad + (1 - e(x)) p_C \Pr_{\mathcal{C}}[X = x] \\ &\qquad - (1 - e(x)) p_S \Pr_{\mathcal{S}(a)}[X = x] \\ &\qquad - e(x) p_C \Pr_{\mathcal{C}}[X = x] \Big] \\ &= \sum_{x \in \{0,1\}^N} \big(2e(x) - 1\big) \big(p_S \Pr_{\mathcal{S}(a)}[X = x] - p_C \Pr_{\mathcal{C}}[X = x]\big). \qquad (15)\end{aligned}$$

The terms $\Pr_{\mathcal{C}}[X = x]$ and $\Pr_{\mathcal{S}(a)}[X = x]$ are defined in Eqs. (12) and (13), respectively. Note that we write $\mathcal{S} = \mathcal{S}(a)$ to clarify that the distribution $\mathcal{S}$ depends on Alice's mixed strategy a.

In summary, Eve's payoff is the probability that her classifier is correct minus the probability that it is incorrect; and the game is zero-sum so that Alice's payoff is exactly the negative of Eve's payoff.

4 Model Analysis

In this section, we present our analytical results. We begin by describing best response strategies for each player. Next, we describe in formal notation the minimax strategies for each player. Finally, we present several results which give structural constraints on the game's Nash equilibria.

4.1 Best Responses

To compute best responses for Alice and Eve, we assume that the other player is playing a fixed strategy, and determine the strategy for Alice (or Eve) which minimizes (or maximizes) the payoff in Eq. (15) as appropriate.

Alice's Best Response. Given a fixed strategy e for Eve, Alice's goal is to minimize the payoff in Eq. (15). However, since she has no control over the cover distribution $\mathcal{C}$, this goal can be simplified to that of minimizing

$$\begin{aligned}
&\sum_{x\in\{0,1\}^N}(2e(x)-1)\cdot p_S \Pr_{\mathcal{S}(a)}[X=x]\\
=&p_S \sum_{x\in\{0,1\}^N}(2e(x)-1))\cdot \sum_{I\subseteq\{0,\ldots,N-1\}} a_I \Pr_{\mathcal{C}}[X=x_I]\\
=&p_S \sum_{I\subseteq\{0,\ldots,N-1\}} a_I \sum_{x\in\{0,1\}^N}(2e(x)-1))\cdot \Pr_{\mathcal{C}}[X=x_I].
\end{aligned}$$

This formula is linear in Alice's choice variables, so she can minimize its value by putting all her probability on the sum's least element. A best response for Alice is thus to play a pure strategy I that minimizes

$$\sum_{x\in\{0,1\}^N}(2e(x)-1))\cdot \Pr_{\mathcal{C}}[X=x_I]. \tag{16}$$

Of course, several different I might simultaneously minimize this sum. In this case, Alice's best response strategy space may also include a mixed strategy that distributes her embedding probabilities randomly among such I.

Eve's Best Response. Given a fixed strategy for Alice, Eve's goal is to maximize her payoff as given in Eq. (15). So, for each x, she should choose $e(x)$ to maximize the term of the sum corresponding to x. Specifically, if $p_S \Pr_{S(a)}[X = x] - p_C \Pr_C[X = x] > 0$, then the best choice is $e(x) = 1$; and if the strict inequality is reversed, then the best choice is $e(x) = 0$. If the inequality is an equality, then Eve may choose any value for $e(x) \in [0, 1]$ and still be playing a best response.

Formally, her optimal decision rule is

$$e(x) = \begin{cases} 1 & \text{if } \frac{\Pr_{\Omega}[Y=S|X=x]}{\Pr_{\Omega}[Y=C|X=x]} > 1, \\ 0 & \text{if } \frac{\Pr_{\Omega}[Y=S|X=x]}{\Pr_{\Omega}[Y=C|X=x]} < 1, \\ \text{any } p \in [0,1] & \text{if } \frac{\Pr_{\Omega}[Y=S|X=x]}{\Pr_{\Omega}[Y=C|X=x]} = 1. \end{cases} \tag{17}$$

For a fixed sequence x, the condition for classifying x as stego can be rewritten as:

$$\begin{aligned} 1 <& \frac{\Pr_{\Omega}[Y = S|X = x]}{\Pr_{\Omega}[Y = C|X = x]} \\ =& \frac{\Pr_{\Omega}[X = x]}{\Pr_{\Omega}[X = x]} \cdot \frac{\Pr_{\Omega}[Y = S|X = x]}{\Pr_{\Omega}[Y = C|X = x]} \\ =& \frac{\Pr_{\Omega}[Y = S]}{\Pr_{\Omega}[Y = C]} \cdot \frac{\Pr_{\Omega}[X = x|Y = S]}{\Pr_{\Omega}[X = x|Y = C]} \\ =& \frac{p_S}{p_C} \frac{\Pr_{\mathcal{S}}[X = x]}{\Pr_{\mathcal{C}}[X = x]} \\ =& \frac{p_S}{p_C} \frac{\sum_I a_I \cdot \prod_{i \notin I} \left(1 - f_i + x_i \tilde{f}_i\right) \cdot \prod_{i \in I} \left(f_i - x_i \tilde{f}_i\right)}{\prod_{i=0}^{N-1} \left(1 - f_i + x_i \tilde{f}_i\right)} \\ =& \frac{p_S}{p_C} \sum_I a_I \prod_{i \in I} \left(\frac{f_i - x_i \tilde{f}_i}{1 - f_i + x_i \tilde{f}_i} \right) \\ =& \frac{p_S}{p_C} \sum_I a_I \prod_{i \in I} \left(\frac{f_i}{1 - f_i} - x_i \frac{\tilde{f}_i}{f_i(1 - f_i)} \right). \end{aligned} \tag{18}$$

Note that Eve's decision rule is written as a multilinear polynomial inequality of degree at most k in the binary sequence x, and that the number of terms in the formula is $\binom{N}{k}$. When k is a constant relative to N (as it typically is in practical applications), then $\binom{N}{k}$ is polynomial in N, and Eve's optimal decision rule can be applied for each binary sequence in time that is polynomial in the length of the sequence.

4.2 Minimax Strategies

A minimax strategy in a two-player game is a mixed strategy of one player that maximizes her payoff assuming that the other player is going to respond with an optimal pure strategy [33].

Eve's minimax strategy is given by

$$\operatorname*{argmax}_{e} \left(\min_{I} \left(\sum_{x \in \{0,1\}^N} (2e(x) - 1)(p_S \Pr_{\mathcal{C}}[X = x_I] - p_C \Pr_{\mathcal{C}}[X = x]) \right) \right); \tag{19}$$

while Alice's minimax strategy is given by

$$\operatorname*{argmin}_{a}\left(\max_{E_S}\Big(\sum_{x\in E_S}\big(p_S \Pr_{\mathcal{S}(a)}[X=x]-p_C\Pr_{\mathcal{C}}[X=x]\big)\right.$$
$$\left.+\sum_{x\in E_C}\big(p_C\Pr_{\mathcal{C}}[X=x]-p_S\Pr_{\mathcal{S}(a)}[X=x]\big)\Big)\right). \qquad (20)$$

Each minimax strategy can be determined (recursively) as the solution to a linear program involving the payoff matrix for Alice's and Eve's pure strategies. Unfortunately, Eve's pure strategy space has size 2^{2^N} so it is computationally intractable to find the minimax strategies using this method even for $N = 5$.

4.3 Nash Equilibria

In this subsection, we present structural constraints for Nash equilibria [25]. We begin with a lemma giving natural conditions under which Eve's classifier must respect the canonical partial ordering on binary sequences. It shows that the classifier must essentially divide the set of all binary sequences into low and high, with high sequences classified as cover and low sequences classified as stego. Then, we give specific constraints on the distribution priors relative to the position biases that determine whether or not the game admits trivial equilibria – in which Eve's classifier is constant for all binary sequences. If either the priors are too imbalanced, or the position biases are too small, then the game will admit such trivial equilibria. In more prototypical parameter regions, however, the game does not admit trivial equilibria. Next, we show that when Eve's classifier is non-trivial, Alice can affect the outcome of Eve's detector, and hence her own payoff by changing her embedding probability for one position in the sequence. Finally, we show that in the non-trivial equilibrium setting, it is not optimal for Alice to embed naïvely in only the least biased positions.

Sequence Ordering in Eve's Equilibrium Strategy

Lemma 2. *Define a partial ordering on* $\{0,1\}^N$ *by* $x < z$ *iff* $x_i \le z_i$ *for* $i = 0, \ldots, N-1$ *and* $x_i < z_i$ *for at least one* i*. Then whenever Alice's embedding strategy satisfies the constraint* $\frac{p_S}{p_C}\sum_I a_I \prod_{i\in I}\left(\frac{f_i}{1-f_i} - x_i\frac{\tilde{f}_i}{f_i(1-f_i)}\right) \neq 1$ *for the sequence* x*, the following condition holds:*

- *If Eve classifies* x *as stego and* $z < x$*, then Eve classifies* z *as stego too.*
- *If Eve classifies* x *as cover and* $x < z$*, then Eve classifies* z *as cover too.*

Proof. Suppose Eve classifies x as stego. Then from the conditions on Eve's best response (Eqs. (17) and (18)), we have that $\frac{p_S}{p_C}\sum_I a_I \prod_{i\in I}\left(\frac{f_i}{1-f_i} - x_i\frac{\tilde{f}_i}{f_i(1-f_i)}\right) \ge 1$; and by the hypothesis of the lemma, the inequality is strict. Suppose $z < x$. Then the value of $\frac{p_S}{p_C}\sum_I a_I \prod_{i\in I}\left(\frac{f_i}{1-f_i} - z_i\frac{\tilde{f}_i}{f_i(1-f_i)}\right)$ is at least the value of the same expression with x replacing z. So this value is also greater than 1, and so Eve also classifies z as stego. The proof of the reverse direction is analogous. □

This lemma implies that in any Nash equilibrium, the set of all binary sequences can be divided into three disjoint sets, low sequences which Eve's likelihood test proscribes a clear value of stego, high sequences which Eve's test proscribes as clearly cover, and a small set of mid-level boundary sequences on which Eve's behavior is not obviously constrained. Furthermore, changing 0s to 1s in a clearly-cover sequence keeps it cover, and changing 1s to 0s in a clearly-stego sequence keeps it stego.

Constraints on Parameters to Guarantee Nontrivial Equilibria. Next we give a key parameter constraint on the prior probabilities of cover and stego that determines the complexity of equilibrium strategies for both players. Essentially if the priors are too far apart relative to the sequence position biases, then the game admits trivial equilibria – in which Eve's classifier is constant; while if they are sufficiently close together then it does not.

Lemma 3. *Suppose that*

$$\prod_{i=0}^{k-1} \frac{1-f_i}{f_i} < \frac{p_C}{p_S} < \prod_{i=0}^{k-1} \frac{f_i}{1-f_i}. \tag{21}$$

Then in any equilibrium, Eve classifies 0^N *as stego and* 1^N *as cover.*

Moreover, if either inequality is reversed strictly, then there exists an equilibrium in which Alice plays a pure strategy of the form $I = \{0, \dots, k-1\}$ *(naïve adaptive embedding), and Eve's classifier is constant.*

Proof. Since $\langle f_i \rangle_{i=0}^{N-1}$ is monotonically increasing, we have that for any size-k subset $I \subseteq \{0, \dots N-1\}$,

$$\prod_{i\in I} \frac{1-f_i}{f_i} \le \prod_{i=0}^{k-1} \frac{1-f_i}{f_i} \quad \text{and} \quad \prod_{i=0}^{k-1} \frac{f_i}{1-f_i} \le \prod_{i\in I} \frac{f_i}{1-f_i}. \tag{22}$$

Consequently, for any mixed strategy $\langle a_I \rangle_{I\subseteq\{0,\dots,N-1\}}$ of Alice, we have

$$\sum_I a_I \prod_{i\in I} \frac{1-f_i}{f_i} \le \prod_{i=0}^{k-1} \frac{1-f_i}{f_i} \quad \text{and} \quad \prod_{i=0}^{k-1} \frac{f_i}{1-f_i} \le \sum_I a_I \prod_{i\in I} \frac{f_i}{1-f_i}. \tag{23}$$

The above, together with Eq. (21) now implies that for any $\langle a_I \rangle_{I\subseteq\{0,\dots,N-1\}}$,

$$\frac{p_S}{p_C} \sum_I a_I \prod_{i\in I} \frac{1-f_i}{f_i} < 1 < \frac{p_S}{p_C} \sum_I a_I \prod_{i\in I} \frac{f_i}{1-f_i}.$$

Using Eve's decision rule from Eq. (18), the left inequality above implies that Eve's best response strategy for the sequence 1^N is to classify it as cover. The right inequality implies that Eve's best response to the sequence 0^N is to classify it as stego.

If the first inequality is reversed strictly, then

$$\frac{p_S}{p_C}\prod_{i=0}^{k-1}\frac{1-f_i}{f_i} > 1.$$

So if Alice embeds in exactly the positions $0, \dots, k-1$, then Eve's best response (see Eq. (18)) will classify 1^N as stego; and since her decision inequality is strict, by Lemma 2, she will classify all sequences as stego. In this circumstance, the payoff for Alice is independent of her strategy. Thus both players are playing a best response, and the strategy configuration is an equilibrium.

Similarly, if the second inequality is reversed strictly, then

$$\frac{p_S}{p_C}\prod_{i=0}^{k-1}\frac{f_i}{1-f_i} < 1.$$

So if Alice embeds in exactly the positions $0, \dots, k-1$, then Eve's best response will classify 0^N as cover; and again by Lemma 2 she will classify all sequences as cover. Again Alice has no incentive to change her strategy, and this configuration is an equilibrium. □

Impact of Alice's Strategy on a Nontrivial Classifier. The next result shows explicitly that if Eve's classifier is non-trivial, then there is a sequence x and a position i that witnesses a change from stego to cover depending only on that position. We use this lemma as a tool for allowing Alice to change her payoff by adjusting her strategy in response to a fixed classifier.

Lemma 4. *Suppose that Eve classifies 0^N as stego and 1^N as cover. Then there exists at least one position i and a sequence x such that $x_i = 0$ and Eve classifies x as stego (with some positive probability), but when the value of x at position i is flipped to 1, then Eve classifies the modified sequence as cover.*

Proof. Starting with 0^N, flip the bit in each position sequentially from position 0 to $N-1$ until after N steps, the sequence becomes 1^N. Since Eve says stego at the beginning, and cover by the end, there must be a step at which she changes from (probably) stego to cover. The sequence x at this step, and position i at this step serve as witnesses to the lemma's claim. □

Exclusion of Naïve Adaptive Embedding Strategies. Our last equilibrium result combines the previous lemmas to show that under relatively mild constraints on the game's parameters, there is no equilibrium in which Alice embeds in exactly the k least biased positions. This result compares well with a result from [19] which showed the same property for a steganography game in which Eve's observational power was more restricted.

The first constraint for the theorem says only that the priors for the stego and cover distributions are not too imbalanced, in comparison to the position biases. The second constraint says that the parameters do not naturally make Eve indifferent on sequence classification against pure strategies. This constraint is satisfied if, for example, the position biases are drawn randomly from a continuous distribution; and it is used only to avoid navigating the logic of pathological cases in which Eve's classifier acts arbitrarily when her likelihood test is inconclusive.

Theorem 1. *Suppose that $k < N$ and the following conditions hold:*

1.

$$\prod_{i=0}^{k-1} \frac{1-f_i}{f_i} < \frac{p_C}{p_S} < \prod_{i=0}^{k-1} \frac{f_i}{1-f_i}, \tag{24}$$

2.

$$\forall x \in \{0,1\}^N, \frac{p_S}{p_C} \prod_{i=0}^{k-1} \left(x_i \frac{1-f_i}{f_i} + (1-x_i) \frac{f_i}{1-f_i} \right) \neq 1. \tag{25}$$

Then there does not exist an equilibrium in which Alice embeds in exactly the k least biased positions.

Proof. Suppose by way of contradiction, that Alice plays a pure strategy by embedding in positions $0, \ldots, k-1$, and that the strategy configuration is an equilibrium. Since Eve is playing a best response to Alice, she classifies an input x as stego whenever

$$\frac{p_S}{p_C} \prod_{i=0}^{k-1} \left(x_i \frac{1-f_i}{f_i} + (1-x_i) \frac{f_i}{1-f_i} \right) > 1;$$

and classifies x as cover when the inequality (for x) is reversed. The inequality is never an equality by assumption, so that Eve's decision is necessarily determined by the binary values of x at positions $0, \ldots, k-1$.

Note that since $k < N$, Alice is not embedding in position $N-1$; and Eve's classifier does not depend on position $N-1$.

By Lemma 3, Eve classifies 0^N as stego and 1^N as cover; so by Lemma 4, there is a position $i \in \{0, \ldots, k-1\}$ and sequence $x \in \{0,1\}^N$ such that $x_i = 0$ and Eve classifies x as stego, but when x_i is flipped to 1, Eve classifies the resulting sequence as cover.

Let J be the set $I \setminus \{i\} \cup \{N-1\}$; and suppose Alice changes her pure strategy from I to J. Let $Pr_{S(I)}[X = x]$ denote the probability of the sequence x appearing on the communication channel in the stego distribution under the original strategy I, and let $Pr_{S(J)}[X = x]$ denote the same probability under Alice's new strategy J. Our goal is now to show that

$$\sum_{x:e(x)=1} Pr_{S(J)}[X = x] < \sum_{x:e(x)=1} Pr_{S(I)}[X = x].$$

Let us group all sequences according to their values on positions other than i and $N-1$. For a binary sequence $x \in \{0,1\}^N$, we write x as zw where

$z \in \{0,1\}^{N-2}$ records the $N-2$ binary values of x for positions other than i or $N-1$; and w records the binary values of x at positions i and $N-1$. Let X_z and X_w denote the random variables associated with the respective parts of the sequence x, and let S_z and S_w denote the stego distributions restricted to the parts of the sequence z and w respectively.

Now let x be any sequence that Eve classifies as stego, so that $e(x) = 1$. We assume for now that z (the components of x at positions other than i and $N-1$) is fixed. Since the conditions of Lemma 2 are satisfied, increasing x at position i can only move the classifier from stego to cover, or leave it the same. Moreover, changing x in position $N-1$ does not affect Eve's classifier at all. Given that Eve classifies x as stego, there are only two possible cases. Either

1. Eve classifies all four sequences zw with $w \in \{00, 10, 01, 11\}$, as stego, or
2. Eve classifies exactly the two sequences zw with $w \in \{00, 01\}$ as stego.

In the first case, the change in strategy from I to J does not change the value of

$$\sum_{\{w:e(zw)=1\}} Pr_S[X = zw],$$

since for fixed z,

$$\sum_{w\in\{00,10,01,11\}} Pr_{S(J)}[X = zw] = \sum_{w\in\{00,10,01,11\}} Pr_{S(I)}[X = zw].$$

In the second case, however, the probabilities of stego sequences differ. In the case of the original distribution I, we have

$$\begin{aligned}
&\sum_{w\in\{00,10\}} Pr_{S(I)}[X = zw] \\
&= Pr_{S(I)}[X = z00] + Pr_{S(I)}[X = z10] \\
&= Pr_{S_z(I)}[X_z = z] \cdot (Pr_{S_w(I)}[X_w = 00] + Pr_{S_w(I)}[X_w = 01]) \\
&= Pr_{S_z(I)}[X_z = z] \cdot \left(\frac{f_i}{1-f_i} \cdot \frac{1-f_{N-1}}{f_{N-1}} + \frac{f_i}{1-f_i} \cdot \frac{f_{N-1}}{1-f_{N-1}} \right) \\
&= Pr_{S_z(J)}[X_z = z] \cdot \frac{f_i}{1-f_i} \cdot \frac{f_{N-1}^2 + (1-f_{N-1})^2}{f_{N-1}(1-f_{N-1})};
\end{aligned}$$

while in the case of the modified distribution J, we have

$$\begin{aligned}
&\sum_{w\in\{00,10\}} Pr_{S(J)}[X = zw] \\
&= Pr_{S_z(J)}[X_z = z] \cdot (Pr_{S_w(J)}[X_w = 00] + Pr_{S_w(J)}[X_w = 01]) \\
&= Pr_{S_z(J)}[X_z = z] \cdot \left(\frac{1-f_i}{f_i} \cdot \frac{f_{N-1}}{1-f_{N-1}} + \frac{1-f_i}{f_i} \cdot \frac{1-f_{N-1}}{f_{N-1}} \right) \\
&= Pr_{S_z(J)}[X_z = z] \cdot \frac{1-f_i}{f_i} \cdot \frac{f_{N-1}^2 + (1-f_{N-1})^2}{f_{N-1}(1-f_{N-1})}
\end{aligned}$$

$$\begin{aligned}
&= Pr_{S_z(I)}[X_z = z] \cdot \frac{1-f_i}{f_i} \cdot \frac{f_{N-1}^2 + (1-f_{N-1})^2}{f_{N-1}(1-f_{N-1})} \\
&= \left(\frac{1-f_i}{f_i}\right)^2 \sum_{w\in\{00,10\}} Pr_{S(I)}[X = zw] \\
&< \sum_{w\in\{00,10\}} Pr_{S(I)}[X = zw].
\end{aligned}$$

By Lemma 4, this second case must occur for at least one stego sequence x; therefore, summing over all x with $e(x) = 1$ and grouping these x according to their z components, we see that the total probability of stego sequences

$$\sum_{x:e(x)=1} Pr_S[X = x]$$

is smaller under the distribution $S(J)$ than under the distribution $S(I)$. Thus Alice can strictly increase her payoff in the game by changing her strategy; and so the configuration is not an equilibrium. □

The theorem shows that if the game has non-trivializing parameter conditions, then it is not optimal for Alice to use only the least biased positions. Rather, she should also use additional positions that may not be taken into consideration by Eve. We conjecture an even stronger result holds – namely that Alice must actually use all N of the positions – under additional reasonable and precise parameter constraints. Two avenues for pursuing this conjecture include formulating more restrictive constraints that avoid navigating Eve's indeterminate actions on boundary sequences, or examining Eve's allowable equilibrium actions on boundary sequences more directly. We leave the precise statement and proof of this conjecture for future work.

In the following section, we explicitly compute all equilibria in the case of length-two sequences and an embedding size of $k = 1$.

5 Numerical Illustration

In this section, we instantiate our model with the special case of flipping a single bit ($k = 1$) in sequences of length two ($N = 2$). In this setting, Alice's pure strategy space is $\{\{0\},\{1\}\}$; and since $a_{\{1\}} = 1 - a_{\{0\}}$, her mixed strategy space can be represented by a single value $a_0 = a_{\{0\}} \in [0,1]$. Eve's pure strategy space is represented by the set of all $[0,1]$-valued functions on $\{\binom{0}{0},\binom{0}{1},\binom{1}{0},\binom{1}{1}\}$. Throughout this section we assume that cover and stego objects are equally likely, i.e., $p_C = p_S = \frac{1}{2}$. Notice that the assumption of equal priors implies the conditions from Eq. (21) which guarantee only non-trivial equilibria.

5.1 Alice's Minimax Strategy

To compute Alice's minimax strategy, we first divide Alice's strategy space into three regions based on Eve's best response:

Lemma 5. *The following table gives Eve's best response for each sequence x as a function of a_0.*

Alice's strategy	*Eve's best response*			
	$x =$			
	$\binom{0}{0}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{1}{1}$
$a_0 < \theta_1$	S	C	S	C
$\theta_1 < a_0 < \theta_2$	S	S	S	C
$\theta_2 < a_0$	S	S	C	C

where $\theta_1 = \frac{(1-f_0)\tilde{f}_1}{f_0+f_1-1}$ *and* $\theta_2 = \frac{f_0\tilde{f}_1}{f_0+f_1-1}$.

Proof. We prove Eve's optimal decision for the four realizations separately.

$\binom{0}{0}$: Eve always classifies $\binom{0}{0}$ as stego.

$$\begin{aligned} \Pr_{\mathcal{C}}\left[X = \begin{pmatrix}0\\0\end{pmatrix}\right] &= \\ (1-f_0)(1-f_1) &< a_0 f_0(1-f_1) + (1-a_0)(1-f_0)f_1 \\ &= \Pr_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}0\\0\end{pmatrix}\right], \end{aligned}$$

since $(1-f_0)(1-f_1) < f_0(1-f_1)$ and $(1-f_0)(1-f_1) < (1-f_0)f_1$.

$\binom{0}{1}$: Eve classifies $\binom{0}{1}$ as cover when $a_0 < \frac{(1-f_0)\tilde{f}_1}{f_0+f_1-1} := \theta_1$.

$$\begin{aligned} \Pr_{\mathcal{C}}\left[X = \begin{pmatrix}0\\1\end{pmatrix}\right] &= \\ (1-f_0)f_1 &\overset{!}{>} a_0 f_0 f_1 + (1-a_0)(1-f_0)(1-f_1) \\ &= \Pr_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}0\\1\end{pmatrix}\right] && \Leftrightarrow \\ (1-f_0)(f_1 - 1 + f_1) &> a_0(f_0 f_1 - 1 + f_0 + f_1 - f_0 f_1) && \Leftrightarrow \\ \frac{(1-f_0)\tilde{f}_1}{f_0+f_1-1} &> a_0 \end{aligned}$$

$\binom{1}{0}$: Eve classifies $\binom{1}{0}$ as cover when $a_0 > \frac{f_0\tilde{f}_1}{f_0+f_1-1} := \theta_2$.

$$\begin{aligned} \Pr_{\mathcal{C}}\left[X = \begin{pmatrix}1\\0\end{pmatrix}\right] &= \\ f_0(1-f_1) &\overset{!}{>} a_0(1-f_0)(1-f_1) + (1-a_0)f_0 f_1 \\ &= \Pr_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}1\\0\end{pmatrix}\right] && \Leftrightarrow \\ f_0(1-f_1) - f_0 f_1 &> a_0(1 - f_0 - f_1 + f_0 f_1 - f_0 f_1) && \Leftrightarrow \\ \frac{-f_0\tilde{f}_1}{1-f_0-f_1} &< a_0 \end{aligned}$$

$\binom{1}{1}$: Eve always classifies $\binom{1}{1}$ as cover.

$$\begin{aligned}\Pr_{\mathcal{C}}\left[X = \begin{pmatrix}0\\0\end{pmatrix}\right] &= \\ f_0 f_1 &> a_0(1-f_0)f_1 + (1-a_0)f_0(1-f_1) \\ &= \Pr_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}0\\0\end{pmatrix}\right],\end{aligned}$$

since $f_0 f_1 > (1-f_0)f_1$ and $f_0 f_1 > f_0(1-f_1)$.

Finally, $\theta_1 < \theta_2$ always holds, since $(1-f_0) < f_0$. □

Theorem 2. *The strategy $(\theta_2, 1-\theta_2)$ is a minimax strategy for Alice.*

Proof. First, for each region, we compute the derivative of Alice's payoff as a function of a_0 given that Eve always uses her best response. Then, we have that Alice's payoff is

- strictly increasing when $a_0 < \theta_1$,
- strictly decreasing when $a_0 > \theta_2$,
- and, when $\theta_1 \leq a_0 \leq \theta_2$, it is strictly increasing if $f_0 \neq f_1$, and it is constant if $f_0 = f_1$.

Thus, we have that $a_0 = \theta_2$ always attains the maximum. □

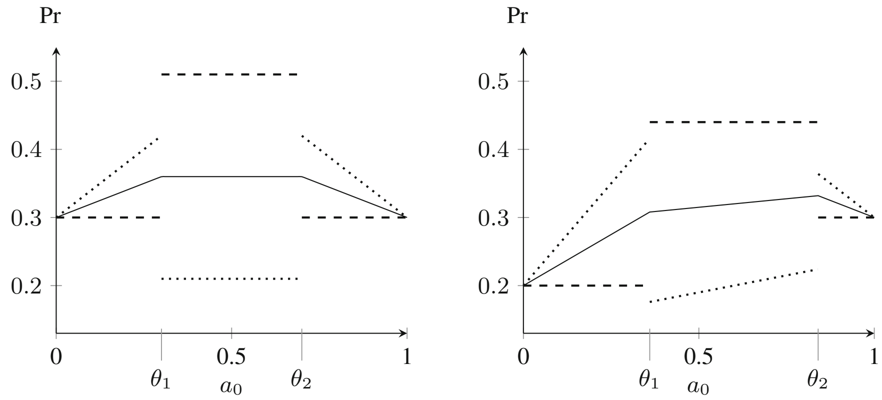
Note that embedding uniformly into both positions ($a_0 = \frac{1}{2}$) is optimal only if the biases are uniform ($f_0 = f_1$); and embedding only in the first position would be optimal only if the bias of the first position were zero ($\tilde{f}_0 = 0$) or if the bias of the second position were one ($\tilde{f}_1 = 1$). This confirms the results from [30], which also considers a two position game but allows Eve to look at only one position.

Figure 2 depicts Eve's error rates and the resulting overall misclassification rate as a function of Alice's strategy $(a_0, 1-a_0)$. Figure 2(a) shows a homogeneous f, while Fig. 2(b) shows a heterogeneous f. It can be seen that neither the false positive rate (dashed line) nor the false negative rate (dotted line) is continuous and that the discontinuities occur at the points θ_1 and θ_2, the points where Eve changes her optimal decision rule. Nonetheless, the overall misclassification rate (solid line) is continuous, which leads to the conclusion that this rate leverages out the discontinuities and thus is a good measure of the overall accuracy of Eve's detector.

5.2 Eve's Minimax Strategy

Theorem 3. *Eve's minimax strategy $e_{minimax}$ is $e_{minimax}\binom{0}{0} = e_{minimax}\binom{0}{1} = 1$, $e_{minimax}\binom{1}{1} = 0$, and*

$$e_{minimax}\begin{pmatrix}1\\0\end{pmatrix} = p = \frac{\tilde{f}_0}{f_0 + f_1 - 1}. \tag{26}$$



(a) $f_0 = 0.7, f_1 = 0.7$

(b) $f_0 = 0.7, f_1 = 0.8$

Fig. 2. Eve's false positive rate (dashed line), false negative rate (dotted line) and her overall misclassification rate (solid line) as a function of a_1, assuming that Eve plays a best response to Alice.

Proof. Since the game is zero sum, Eve's strategy is a minimax strategy if Alice's minimax strategy is a best response to it [33]. Therefore, it suffices to show that Alice has no incentives for deviating from her own minimax strategy when Eve uses $e_{minimax}$. Alice's best response to $e_{minimax}$ is

$$\begin{aligned}
&\operatorname*{argmax}_{a_0 \in [0,1]} \Bigg\{ -\mathrm{Pr}_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}0\\0\end{pmatrix}\right] - \mathrm{Pr}_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}0\\1\end{pmatrix}\right] \\
&\qquad + (1-2p)\mathrm{Pr}_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}1\\0\end{pmatrix}\right] + \mathrm{Pr}_{\mathcal{S}(a_0)}\left[X = \begin{pmatrix}1\\1\end{pmatrix}\right] \Bigg\} \\
= &\operatorname*{argmax}_{a_0 \in [0,1]} \Big\{ -a_0 f_0(1-f_1) - (1-a_0)(1-f_0)f_1 \\
&\qquad - a_0 f_0 f_1 - (1-a_0)(1-f_0)(1-f_1) \\
&\qquad + (1-2p)\big[a_0(1-f_0)(1-f_1) + (1-a_0)f_0 f_1\big] \\
&\qquad + a_0(1-f_0)f_1 + (1-a_0)f_0(1-f_1)\Big\} \\
= &\operatorname*{argmax}_{a_0 \in [0,1]} \Big\{ a_0\left[2 - 4f_0 - 2p\left(1 - f_0 - f_1\right)\right] + \ \mathrm{const}(f,p)\Big\}.
\end{aligned}$$

If $p = \frac{\tilde{f}_0}{f_0+f_1-1}$, then the value of the above optimization problem does not depend on a_0. Consequently, Alice has no incentives for deviating from her minimax strategy. □

It follows immediately from the theorem that Eve's minimax decision function is deterministic if and only if the cover is homogeneous ($f_0 = f_1$). This is interesting from the perspective of practical steganography, as all practical detectors are deterministic although embedding functions are pseudo-random and covers are heterogeneous.

6 Conclusion

We analyzed a two-player game between Alice, a content-adaptive steganographer, and Eve, an unbounded steganalyst. In keeping with a strict application of Kerckhoffs' principle to steganography, we allowed Eve access to Alice's embedding strategy, the cover source distribution, and unbounded computational power. Under these assumptions, we formalized processes both for constructing an optimal content-adaptive embedding strategy under the assumption of an optimal classifier, and for constructing an optimal detector under the assumption of an optimal embedding strategy.

Our formalism applies to arbitrary-sized cover sequences, although implementing the formalism for large covers remains a computational challenge. For the special case of a two-bit cover sequence, we exemplified an optimal classifier/embedding pair, and illustrated its structure in terms of the classification error rates.

For the practical steganalyst, our results give direction to the optimal detection of strategic embedding, and for optimal embedding against a strategic detector. In particular, Eve's optimal classifier should be monotone in the cover's predictability metric; and Alice's optimal adaptive embedding strategy should not naïvely use only the least biased positions. We also showed that a deterministic classifier can be sub-optimal for covers with heterogeneous predictability.

In our detailed analysis of length-two cover sequences, Alice's optimal randomized embedding strategy changed each part of the cover with some positive probability, and with more sophisticated structural constraints on the game's parameters, we expect that an analogous result can be proven for larger covers. It remains for future work to prove this conjecture and more directly address the computational tractability of implementing optimal strategies.

Acknowledgments. We thank the reviewers for their comments on an earlier version of this paper. We gratefully acknowledge support by the Penn State Institute for Cyber-Science. The second author's research visit at Penn State was supported under Visiting Scientists Grant N62909-13-1-V029 by the Office of Naval Research (ONR), and the third author's research visit at Penn State was supported by the Campus Hungary Program.

References

1. Acquisti, A., Dingledine, R., Syverson, P.F.: On the economics of anonymity. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 84–102. Springer, Heidelberg (2003)
2. Anderson, R.: Stretching the limits of steganography. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 39–48. Springer, Heidelberg (1996)

3. Barni, M., Tondi, B.: The source identification game: an information-theoretic perspective. IEEE Trans. Inf. Forensics Secur. **8**(3), 450–463 (2013)
4. Böhme, R.: Advanced Statistical Steganalysis. Springer, Berlin (2010)
5. Böhme, R., Westfeld, A.: Exploiting preserved statistics for steganalysis. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 82–96. Springer, Heidelberg (2004)
6. Chia, P.H., Chuang, J.: Colonel Blotto in the phishing war. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 201–218. Springer, Heidelberg (2011)
7. Denemark, T., Fridrich, J.: Detection of content adaptive LSB matching: a game theory approach. In: Alattar, A., Memon, N., Heitzenrater, C. (eds.) Proceedings SPIE, Media Watermarking, Security, and Forensics, vol. 9028, p. 902804. SPIE and IS&T (2014)
8. Ettinger, J.M.: Steganalysis and game equilibria. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 319–328. Springer, Heidelberg (1998)
9. Franz, E.: Steganography preserving statistical properties. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 278–294. Springer, Heidelberg (2003)
10. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, New York (2009)
11. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Delp, E., Wong, P. (eds.) Proceedings SPIE, Security, Steganography, and Watermarking of Multimedia Contents VI, vol. 5306, pp. 23–34. SPIE (2004)
12. Fridrich, J., Kodovsky, J.: Multivariate Gaussian model for designing additive distortion for steganography. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, pp. 2949–2953, May 2013
13. Fridrich, J., Du, R.: Secure steganographic methods for palette images. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 47–60. Springer, Heidelberg (2000)
14. Grossklags, J., Christin, N., Chuang, J.: Secure or insure?: A game-theoretic analysis of information security games. In: Proceedings of the 17th International World Wide Web Conference (WWW), Beijing, China, pp. 209–218, April 2008
15. Guo, L., Ni, J., Shi, Y.: Uniform embedding for efficient JPEG steganography. IEEE Trans. Inf. Forensics Secur. **9**(5), 814–825 (2014)
16. Hélouët, L., Zeitoun, M., Degorre, A.: Scenarios and covert channels: another game. Electron. Notes Theor. Comput. Sci. **119**(1), 93–116 (2005)
17. Johnson, B., Böhme, R., Grossklags, J.: Security games with market insurance. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 117–130. Springer, Heidelberg (2011)
18. Johnson, B., Schöttle, P., Böhme, R.: Where to hide the bits? In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 1–17. Springer, Heidelberg (2012)
19. Johnson, B., Schöttle, P., Laszka, A., Grossklags, J., Böhme, R.: Bitspotting: detecting optimal adaptive steganography. In: Shi, Y.Q., Kim, H.-J., Pérez-González, F. (eds.) IWDW 2013. LNCS, vol. 8389, pp. 3–18. Springer, Heidelberg (2014)
20. Ker, A.: Batch steganography and the threshold game. In: Delp, E., Wong, P. (eds.) Proceedings SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX, vol. 6505, pp. 0401–0413. SPIE (2007)
21. Laszka, A., Foldes, A.: Modeling content-adaptive steganography with detection costs as a quasi-zero-sum game. Infocomm. J. **5**(4), 33–43 (2013)

22. Laszka, A., Johnson, B., Schöttle, P., Grossklags, J., Böhme, R.: Managing the weakest link. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) ESORICS 2013. LNCS, vol. 8134, pp. 273–290. Springer, Heidelberg (2013)
23. Maillé, P., Reichl, P., Tuffin, B.: Interplay between security providers, consumers, and attackers: a weighted congestion game approach. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 67–86. Springer, Heidelberg (2011)
24. Moulin, P., Ivanovic, A.: The zero-rate spread-spectrum watermarking game. IEEE Trans. Signal Process. **51**(4), 1098–1117 (2003)
25. Nash, J.: Non-cooperative games. Ann. Math. **54**(2), 286–295 (1951)
26. Orsdemir, A., Altun, O., Sharma, G., Bocko, M.: Steganalysis-aware steganography: statistical indistinguishability despite high distortion. In: Delp, E., Wong, P., Dittmann, J., Memon, N. (eds.) Proceedings SPI, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, vol. 6819, p. 681915. SPIE (2008)
27. Petitcolas, F.: Introduction to information hiding. In: Katzenbeisser, S., Petitcolas, F. (eds.) Information Hiding Techniques for Steganography and Digital Watermarking, Recent Titles in the Artech House Computer Security Series, pp. 1–14. Artech House, Boston (2000)
28. 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)
29. Pita, J., Jain, M., Ordónez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., Kraus, S.: Using game theory for Los Angeles airport security. AI Mag. **30**(1), 43–57 (2009)
30. Schöttle, P., Böhme, R.: A game-theoretic approach to content-adaptive steganography. In: Kirchner, M., Ghosal, D. (eds.) IH 2012. LNCS, vol. 7692, pp. 125–141. Springer, Heidelberg (2013)
31. Schöttle, P., Laszka, A., Johnson, B., Grossklags, J., Böhme, R.: A game-theoretic analysis of content-adaptive steganography with independent embedding. In: Proceedings of the 21st European Signal Processing Conference (EUSIPCO), Marrakech, Morocco, September 2013
32. Stamm, M., Lin, W., Liu, K.: Forensics vs. anti-forensics: a decision and game theoretic framework. In: Proceedings of the 2012 IEEE International Conference on Acoustics. Speech and Signal Processing (ICASSP), Kyoto, Japan, pp. 1749–1752, March 2012
33. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, Princeton (1944)

Permutation Steganography in FAT Filesystems

John Aycock[✉] and Daniel Medeiros Nunes de Castro

Department of Computer Science, University of Calgary, 2500 University Drive N.W.,
Calgary, AB T2N 1N4, Canada
{aycock,dmncastr}@ucalgary.ca

Abstract. It is easy to focus on elaborate steganographic schemes and forget that even straightforward ones can have a devastating impact in an enterprise setting, if they allow information to be exfiltrated from the organization.

To this end, we offer a cautionary tale: we show how messages may be hidden in FAT filesystems using the permutation of filenames, a method that allows a hidden message to be embedded using regular file copy commands. A straightforward scheme, but effective. Our experiments on seven different platforms show that the existence of the hidden message is obscured in practice in the vast majority of cases.

1 Introduction

Steganography, the ability to hide messages in a (hopefully) undetectable manner, has a long and storied history – documented examples exist in the 16th century [1] as well as over two thousand years ago [2]. Computer technology has brought with it an explosion of new steganography techniques,[1] the exposition of which can fill entire books (see, for example, [3–5]).

In an enterprise setting, data security with respect to malicious actors is paramount, even if only espionage, sabotage, and insider leaks are considered. Depending on the sector, data security may also be a legal obligation, as with personal health information. Steganography adds a level of complexity to maintaining data security: how can the exfiltration of hidden data from an organization be detected and prevented, and can malicious code be surreptitiously infiltrated into an enterprise?

To underscore the problem, in this work we examine what is, to the best of our knowledge, a novel method for steganography. We show that messages can be hidden in a relatively simple manner in FAT filesystems, a format dating back to MS-DOS and even earlier, to the proto-MS-DOS of 1977 [6]. Probably due to its simplicity in structure and thus implementation, FAT filesystems have become a *de facto* standard for many modern devices and media, including USB flash drives, and FAT is supported by all modern operating systems. Our method has the interesting property that a message can be hidden simply by using regular file copy commands, meaning that it is possible to exfiltrate data without having

[1] And an unhealthy obsession with least-significant bits.

Y.Q. Shi (Ed.): Transactions on DHMS X, LNCS 8948, pp. 92–105, 2015.
DOI: 10.1007/978-3-662-46739-8_6

a telltale signature of special steganography software installed on an enterprise computer.

In the remainder of this paper, we explain our method and its implementation (Sect. 2), present results of the experiments we conducted hiding a message this way (Sect. 3), followed by an analysis and discussion in Sect. 4. Sects. 5 and 6 have related work and conclusions, respectively.

2 Method and Implementation

Our method is based on the fact that FAT filesystem directories store filenames in an order that does not necessarily correspond to how the filenames are displayed to the user. For example, consider a FAT filesystem directory that contains three files. Because of the order in which they were created, they may be actually ordered as follows in the directory:

```
foo.jpg  baz.exe  bar.bat
```

yet when the directory listing is viewed from a GUI, those same files will be sorted and shown in the alphabetical order

```
bar.bat  baz.exe  foo.jpg
```

We can construct a steganography method from this, relying on the fact that an ordering of filenames can exist but yet not be visible, *if* we can somehow use the permutation of filenames to convey data. As it happens, there are ideas dating back to the 1800s [7] (and rediscovered in the 1950s [8]) that allow us to do just that.

To explain the process, we start with base conversion, i.e., converting a number in one base to the equivalent value in another base. For instance, to convert the number 123 in base 10 (denoted 123_{10}) to base 10 – a trivial conversion, admittedly – we can divide 123 repeatedly by the base we want to convert to (10), keeping track of the remainders:

$$\begin{aligned} 123 \div 10 &= 12\ r\mathbf{3} \\ 12 \div 10 &= 1\ r\mathbf{2} \\ 1 \div 10 &= 0\ r\mathbf{1} \end{aligned}$$

Then, reading the remainders backwards reveals the number converted to base 10, **1 2 3**. For this trivial conversion, effectively this is the same as shifting the decimal point of 123 to the left one digit at a time.

The same algorithm works for converting a base 10 number to any arbitrary base b: we divide repeatedly by b until 0 is reached, and read the remainders backwards to find the equivalent number in base b. Converting 123_{10} to base 8:

$$\begin{aligned} 123 \div 8 &= 15\ r\mathbf{3} \\ 12 \div 8 &= 1\ r\mathbf{7} \\ 1 \div 8 &= 0\ r\mathbf{1} \end{aligned}$$

yields the result that $123_{10} = 173_8$.

There is actually no reason, apart from good taste, why the value of b must be the same for each division step. If we begin with $b = 1$ and increment b on each step, we convert into a *factorial number system* [9]. Starting with 43_{10}:

$$\begin{aligned} 43 \div 1 &= 43\ r\mathbf{0} \\ 43 \div 2 &= 21\ r\mathbf{1} \\ 21 \div 3 &= 7\ r\mathbf{0} \\ 7 \div 4 &= 1\ r\mathbf{3} \\ 1 \div 5 &= 0\ r\mathbf{1} \end{aligned}$$

Reading the remainders backwards shows the digits of the equivalent number in the factorial number system are 1 3 0 1 0, or $1_5\,3_4\,0_3\,1_2\,0_1$. It may seem that we are far afield of a filename permutation at this point, but in fact a number in the factorial number system maps very easily into a unique permutation.

Recall that a base b number system requires symbols to uniquely represent values from $0 \ldots b-1$. Base 10, for example, has the digits $0 \ldots 9$. For a five-digit number in the factorial number system as above, this means that the first digit (base 5) must have a value between $0 \ldots 4$, the second digit (base 4) a value between $0 \ldots 3$, and so on, until the last digit (base 1) may only have the value 0.

Remaining list, with indices	*Factorial number digit*	*Next element of permutation*
`aardvark`$_0$ `bat`$_1$ `cat`$_2$ `dog`$_3$ `eagle`$_4$	1	`bat`
`aardvark`$_0$ `cat`$_1$ `dog`$_2$ `eagle`$_3$	3	`eagle`
`aardvark`$_0$ `cat`$_1$ `dog`$_2$	0	`aardvark`
`cat`$_0$ `dog`$_1$	1	`dog`
`cat`$_0$	0	`cat`

Fig. 1. Creating a permutation from the factorial number $1_5\,3_4\,0_3\,1_2\,0_1$

Now, say that we have an ordered list of five filenames we want to permute:

```
aardvark   bat   cat   dog   eagle
```

Treating the individual digits of $1_5\,3_4\,0_3\,1_2\,0_1$ as indices into this list, assuming that `aardvark` is at index 0 and `eagle` is initially at index 4, we can use the digits as a guide to pluck out elements of the permuted sequence one at a time, as shown in Fig. 1. This results in the permutation

```
bat   eagle   aardvark   dog   cat
```

To recover a hidden message from a filename permutation, we recompute the ordered filename list and use the permuted list to reconstruct the factorial number.[2] That, in turn, can be converted back to a base 10 number easily, as illustrated by the running example:

[2] One caveat for recovery is that the filenames must be unique, but that is implied by FAT filesystem semantics.

$$\begin{aligned}1_5\,3_4\,0_3\,1_2\,0_1 &= (1 \times 4!) + (3 \times 3!) + (0 \times 2!) + (1 \times 1!) + (0 \times 0!)\\ &= 24 + 18 + 0 + 1 + 0\\ &= 43_{10}\end{aligned}$$

And, of course, the number can represent any arbitrary data – for example, we can make a string of characters into a single number. We have implemented our method for embedding data with a Python script which, in its simplest form, takes a message to hide along with some filenames to permute; its basic output is an ordered sequence of file copy commands that may be run on a FAT filesystem that result in the appropriate permutation being created. For extraction, the same script is given a set of permuted filenames and outputs the hidden message therein. The script's pseudocode is given in two parts: Fig. 2 contains the message embedding and extraction functions, and Fig. 3 shows the conversion routines to and from the factorial number system. The pseudocode refers to "large" numbers to emphasize use of arbitary precision integers, and we abstract away the conversion between a message and a number because that can be done any number of ways. In the next section we describe our experiments with the script.

3 Experiments

We hid a test message, `Hello, world!`, which is 15 bytes in length with line terminators included; with an ASCII encoding, this message requires 33 files to hide, which were located on a FAT filesystem. For each case below, we accessed the files concealing the message to determine if we could see the real ordering of files, i.e., whether or not the reordering of files would be visible. Details of the test equipment and software may be found in Appendix A.

3.1 Traditional Operating Systems, GUI Interface

From the GUI, the permutation is not visible.

Linux. The default behavior of the file manager, Nemo, is to show files ordered by filename. It is possible to order the files by time, but it only takes into account the time to the second. Files that were stored in the same second are displayed ordered by name, so the original permutation cannot be found, because all the files hiding the message were copied within the same second.

Mac OS and Windows. As with Linux, files are listed ordered by filename. Trying to list by time, the files that were stored at the same second were ordered by their names.

3.2 Traditional Operating Systems, Command-Line Interface

While the command line is alien to most users now, it is interesting to see the contrast with the GUI results. With the exception of Windows' MS-DOS-derived behavior, special measures needed to be taken to see the real file permutation.

```
def embed(message, sourceDirectory):

    value = convertMessageToLargeNumber(message)

    files = list of files in sourceDirectory
    error if not enough files for message size

    sort files lexicographically by filename

    listFactors = numberToFactors(value)

    for each n in listFactors:
        print "copy " + files[n] + " to destinationDirectory"
        delete files[n]

def extract(sourceDirectory):

    files = list of filenames in sourceDirectory in FAT directory order
    sortedFiles = list of filenames in sourceDirectory
    sort sortedFiles lexicographically by filename

    factors = empty list of numbers

    for each filename in files:
        i = index of filename in sortedFiles
        delete sortedFiles[i]
        append i to factors

    n = factorsToNumber(factors)
    message = convertLargeNumberToMessage(n)
    return message
```

Fig. 2. Pseudocode for message embedding and extraction

Linux and Mac OS. `ls` lists the files in alphabetical order by default. Using the `-t` option orders files by time, but only to the second, so the permutation is not recoverable. The `-f` option must be specified to show the files in unsorted order, allowing the permutation to be recovered.

Windows. Using the `dir` command, the files were listed in the order they were saved, by default. The permutation was easily recovered.

3.3 Mobile Devices

For cellphone and tablet, the card reader on a laptop was used to create a directory on the SD card containing the permuted files. Here, the ability to see the real file ordering varied.

Android Cell Phone. If the permuted files are pictures, the "Gallery" app can be used, which shows the files ordered by time, newer files first. This means

```
def numberToFactors(n):

    error if n < 0

    listFactors = empty list of numbers

    while n > 0:
        digit = n mod (length(listFactors) + 1)
        insert digit at the beginning of listFactors
        n = n div length(listFactors)

    return listFactors

def factorsToNumber(listFactors):

    error if last element of listFactors != 0

    n = 0
    returnValue = 0
    i = length(listFactors)-1

    while i >= 0:
            returnValue = returnValue + listFactors[i] * factorial(n)
            n = n + 1
            i = i - 1
    return returnValue
```

Fig. 3. Pseudocode for factorial number system conversion

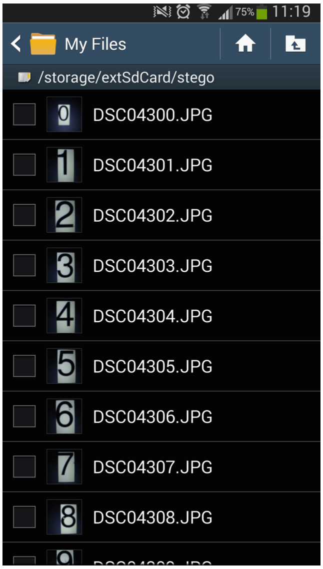
that the inverted permutation can be seen. The filenames are not available in this app, however, so it is also necessary to know the original order of the pictures, and depending on the pictures, the reordering may not be obvious to a casual observer.

Using the "My Files" app, the files are ordered by their names, by default. It is possible to order files by time, however.

Android Tablet. Using the "Gallery" app, as with the cell phone above, we can visually see the inverted permutation. The "Files" app, by default, orders files by name; time ordering behaves like the GUIs above in that the time granularity is only to the second, and the permutation cannot be recovered. We also tried the third-party "Terminal IDE" app, which gives a command line prompt, using a limited shell. Its `ls`, however, does not have an option not to order the files, and no options allow us to recover the permutation.

3.4 Digital Cameras

The cameras we tested were not very flexible at all in terms of the files and directories they would recognize. In the end, out of desperation, we took pictures of a sequence of numbers (see Fig. 4) to generate `.JPG` files the cameras would be happy with, and permuted those in order to ascertain how the cameras were



75% 11:19

My Files

/storage/extSdCard/stego

□ DSC04300.JPG

□ DSC04301.JPG

□ DSC04302.JPG

□ DSC04303.JPG

□ DSC04304.JPG

□ DSC04305.JPG

□ DSC04306.JPG

□ DSC04307.JPG

□ DSC04308.JPG

Fig. 4. Pictures of numbers for camera testing (as shown on a mobile device)

handling files. Obviously, these pictures would be replaced by non-contrived ones in an actual data hiding scenario. Neither camera showed the real file ordering.

Camera 1. The pictures are shown in ordered sequence, which means that it orders the sequence of pictures using the filename. There was no apparent way to change the sorting order, meaning that using just the camera, the permutation cannot be recovered.

Camera 2. Again, the camera shows the pictures in numeric sequence, meaning that it orders by the filename, and there was again no way of changing the order of the files. Overall, though, this camera seemed to be more liberal in terms of the file and directory names it showed.

4 Analysis and Discussion

One important question for any steganographic scheme is how much information may be embedded. Given N filenames, we have $N!$ possible permutations, meaning that we have at most $\lfloor \log_2 N! \rfloor$ bits available. While we are currently just using the files in a single directory, we could extend this straightforwardly to use multiple directories' worth of files in a FAT filesystem in order to increase the number of files at our disposal (and thereby obtain more bits for hiding messages). Another extension would be *within* files in a FAT filesystem, specifically the order of filenames within an archive file like a `.zip` file.[3]

Correct message embedding relies on certain assumptions. Essentially, the file ordering must be controllable, and for that reason our experiments copy

[3] Interestingly, a white paper on steganography in archive files noted the 'arbitrary order' of files in a ZIP archive [10], but failed to make the connection to permutations.

files into an empty directory. Any perturbation in the file ordering would cause embedding to fail, so we assume that the target directory files are written in the order our copy commands specify, and that no other users or processes are manipulating the target directory during embedding.

The experimental results show that there are some methods by which the actual ordering of files in a FAT filesystem may be seen, although these are atypical ways to view a directory's contents. There is a distinction to be made, however, between being able to view the actual file ordering and knowing a message is hidden. Would a human suspect a hidden message, given an unusual file ordering? Would even a computer be able to detect a hidden message, i.e., can a hidden message be found forensically by a strong adversary? Certainly the actual file ordering would be visible to a strong adversary, whereas a weaker adversary would be limited to standard interfaces that do not necessarily reveal file ordering, as our experiments showed.

For further insight into the possibility of forensic detection, we need to understand what the normal appearance of FAT filesystem directories is in order to detect anomalous hidden messages within them. We therefore gathered data from real FAT filesystems to determine their ordering properties. Specifically, we used fresh installations of FreeDOS (version 1.1) and Windows XP (version 2002 with SP2), along with a camera FAT filesystem and a USB key primarily used for photo backup. The latter two have both been used five years or more, so these four FAT filesystems represent a wide spectrum of usage.[4] We measured the Levenshtein distance [11] from the actual FAT ordering for each directory's files to six canonical orderings:

1. lexicographically sorted;
2. lexicographically sorted in reverse;
3. sorted by modification time;
4. sorted by modification time in reverse order;
5. sorted by creation time;
6. sorted by creation time in reverse order.

Each Levenshtein distance was normalized by the number of files in the directory, and the combined results are plotted in Fig. 5. In terms of forensic detection, two things are apparent. First, legitimate FAT directory orderings *without* hidden messages cover almost the entire range of possibilities. Second, adding data from more FAT filesystems, from more devices, will not reduce this range – the ubiquity of FAT makes it very difficult to determine what is (ab)normal.

We then embedded the message from Sect. 3 into those FreeDOS and Windows XP directories that had enough files to contain it. In practice, embedding a message may not use all the files in a source directory, and the message must be padded. We tried four different types of padding: appending random bytes to the message, appending spaces to the message, appending NUL characters to the message, and prepending zeroes to the factorial representation of the message.

[4] Highlighting the problems with real-world devices and FAT filesystems, the camera's clock has never been able to retain the correct time, and 672 of the 752 images claim to be from December 31, 1979.

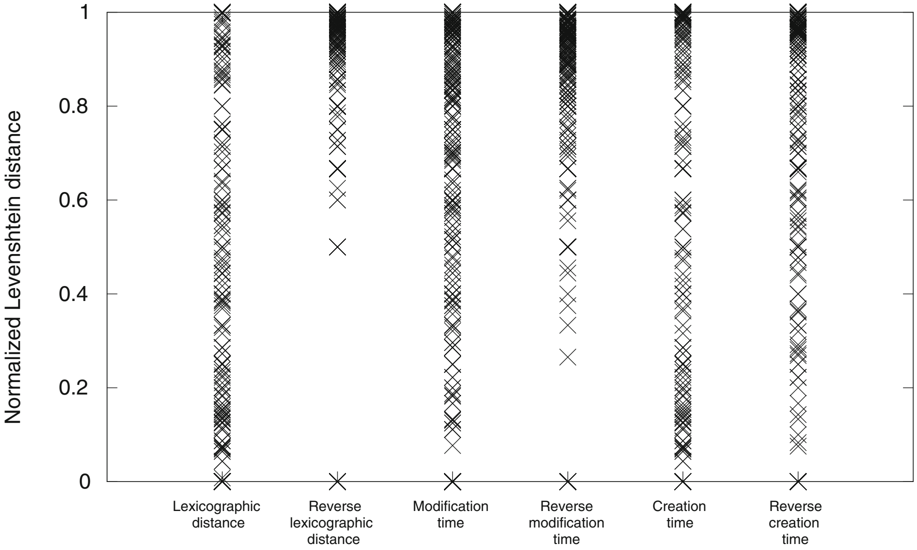


Fig. 5. Normalized Levenshtein distance for FAT filesystems; lower numbers mean greater similarity

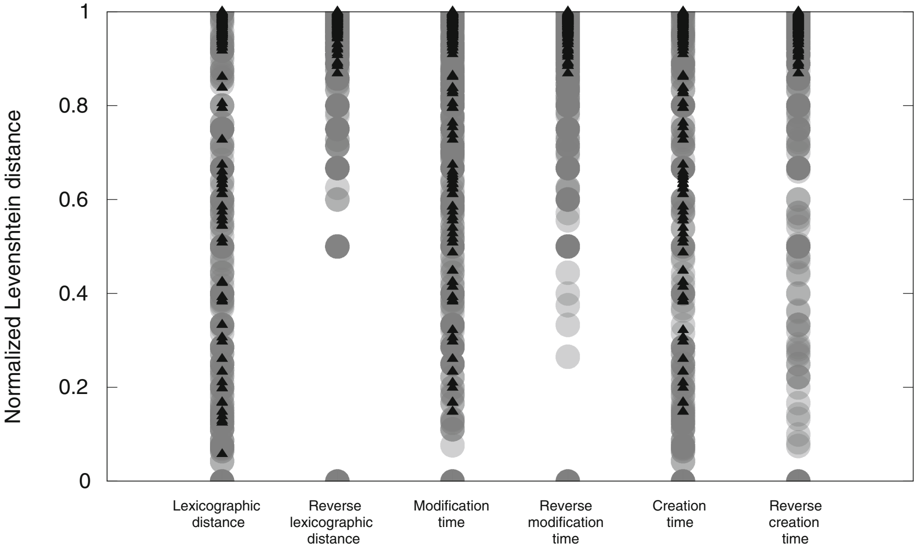


Fig. 6. Normalized Levenshtein distance for embedded messages (triangles) with previous FreeDOS/Windows XP FAT distances for reference (gray circles); lower numbers mean greater similarity

The results are shown in Fig. 6, overlaid over the previous FAT results for FreeDOS and Windows XP (shown as gray circles) for reference. There is no need to separate out the different padding types in the results, as they all fall well within the range of actual FAT filesystem values. There is no way to distinguish directories with embedded messages using this edit distance metric.

Forensic detection by a strong adversary, given these results, seems to be very difficult if not impossible in general. Detection in unusual special cases may still be tractable, where a previous FAT directory ordering is known or can be assumed. Especially from the point of view of an organization trying to prevent data exfiltration, it is far simpler to try and destroy any message hidden with filename permutations.

As can be seen from Sect. 2, message extraction is sensitive to the actual file ordering, and there are several ways that this might be disrupted.

- A file may be added to the directory. The message may still be extracted if the new file's timestamp allows it to be segregated from the files that carry the message.
- A file may be deleted from the directory. FAT filesystems may allow files to be "undeleted", and thus a deleted filename might be recovered. (A deleted filename's directory entry simply has the filename's initial character overwritten with $E5_{16}$ [12].)
- The files may be recopied in some different order, such as alphabetical or random order. This would completely destroy any permutation-based message.

While there are other potential approaches, like combinations of file deletions and additions, the surety of recopying makes it the preferred method. For small induced errors, an error-correcting code (e.g., [13]) might permit recovering the hidden message at the cost of some bits, but even that would fail to withstand recopying.

5 Related Work

From a high level, the related work can be broken into two parts. First, there are uses of permutations in steganography; second, there is work on steganography in filesystems, including but not limited to FAT filesystems. We have not found any work that overlaps the two areas (i.e., applying permutation steganography to filesystems) as ours does.

5.1 Permutation Steganography

Permutations have been used before, albeit not very widely, in steganography. Any cover medium shared between sender and receiver that has, or can be assigned, an ordering is a potential candidate. Permutation steganography schemes have been proposed for TCP packets [14], peer-to-peer networks [15], HTTP requests [16], and Twitter tweets [17]. Outside the network domain, cards in card games have an ordering, and this has been used for permutation steganography too [17,18]. The flip side to hiding messages is finding hidden messages, and there has been some steganalysis work done trying to decide if a hidden message can be distinguished from normal communication [19].

5.2 Steganography in Filesystems

In a general sense, one could imagine that a perfect filesystem incorporating steganography at its core would naturally provide some sort of plausible deniability regarding its contents. This is the idea explored by Anderson *et al.* [20], which inspired an implementation using free space in an `ext2` file system [21] (the work in [22] is similar).

Many filesystem-specific steganography methods consist of (ab)uses of a filesystem's structures and/or unallocated space. One method hides files by deleting all references to them in FAT/NTFS filesystems [23]. Another takes advantage of a quirk of the FAT filesystem in that duplicate filenames can be constructed in a directory [24]. Yet another repurposes portions of NTFS' master file table [25]. Some methods take advantage of clusters in the FAT filesystem: tagging hidden data as being part of a bad cluster [26]; encoding a message using even- and odd-numbered clusters to represent 1s and 0s [27]; making use of unused space in clusters [26], also known as "slack space." A less opportunistic approach to slack space is taken by HideInside [28], which creates its own slack space to hide data in. "The grugq" takes a shine to filesystem metadata, illustrating how to hide data on Unix filesystems in bad block files, directories, and filesystem journals [29].

Some filesystem types are highly standardized, such as those for SIM/USIM cards. Savoldi and Gubian [30,31] describe how to extract files from these filesystems that are "hidden" by virtue of being nonstandard.

Finally, for completeness, some papers catalog methods for steganography in filesystems but do not appear to contribute new methods *per se* [32,33].

6 Conclusion

Using file ordering permutations in FAT filesystems is a viable means for storing short messages, where the message can be embedded using available file copying methods if necessary. The ability to see the actual file ordering is key to being able to both extract and detect a message hidden using our technique. While this ability varies by system, in our testing all GUI interfaces for major commodity operating systems (Windows, Mac OS X, and Linux) were unable to reveal the actual file ordering regardless of their settings. Furthermore, there is a distinction to be made between seeing the real ordering of files, and knowing that a message is hidden using that ordering. These results emphasize that managing data exfiltration is a difficult problem indeed, when even the most innocuous things, like the ordering of files, can be used to hide data.

Acknowledgments. This work was supported by a grant from TELUS Communications. Thanks to the anonymous referees for suggestions that helped improve the paper.

A Test Details

Linux
Linux Mint 16 Petra Cinnamon, Nemo version 2.0.8, `ls` from GNU coreutils 8.20.

Mac OS
Mac OS X 10.9.1 (13B42).

Windows
Windows 7 Home Premium.

Android cell phone
Samsung SIII, model SGH-I747M, Android 4.3. Baseband version I747MVLUEMK5,
kernel 3.0.31-2140838 (from Nov 19, 2013 - 19:35:04), build number JSS15J.I747MVLUEMK5.

Android tablet
Motorola Xoom WiFi, model MZ604 (Canada), Android 4.0.3. Kernel 2.6.39.4-0008-gca76b41, build number I.7.1-34.

Camera 1
Sony Cyber-shot DSC-H10.

Camera 2
Camera Canon EOS Rebel T3i.

References

1. Caraman, P. (trans.): The Hunted Priest: Autobiography of John Gerard. Fontana (1959)
2. Macaulay, G.C. (trans.): The History of Herodotus, vol. 2. Macmillan, London (1890)
3. Johnson, N.F., Duric, Z., Jajodia, S.: Information Hiding: Steganography and Watermarking - Attacks and Countermeasures. Kluwer, Boston (2001)
4. Katzenbeisser, S., Petitcolas, F.A.P. (eds.): Information Hiding: Techniques for Steganography and Digital Watermarking. Artech House, Norwood (2000)
5. Wayner, P.: Disappearing Cryptography, 2nd edn. Morgan Kaufmann, New York (2002)
6. Duncan, R. (ed.): The MS-DOS Encyclopedia. Microsoft Press, Redmond (1988)
7. Laisant, C.A.: Sur la numération factorielle, application aux permutations. Bulletin de la Société Mathématique de France **16**, 176–183 (1888)
8. Lehmer, D.H.: Teaching combinatorial tricks to a computer. In: 10th Symposium in Applied Mathematics of the American Mathematical Society, pp. 179–193 (1960). Symposium was actually held in 1958
9. Knuth, D.E.: The Art of Computer Programming: Seminumerical Algorithms, 3rd edn., vol. 2. Addison Wesley (1998)
10. Reversing Labs: Hiding in the familiar: Steganography and vulnerabilities in popular archives formats. (http://www.reversinglabs.com/sites/default/files/pictures/NyxEngine_BlackH (Accessed 14 March 2014)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics - Doklady **10**, 707–710 (1966). Translation

12. Carrier, B.: File System Forensic Analysis. Addison-Wesley, Reading (2005)
13. Jiang, A., Schwartz, M., Bruck, J.: Error-correcting codes for rank modulation. In: IEEE International Symposium on Information Theory, pp. 1736–1740 (2008)
14. Chakinala, R.C., Kumarasubramanian, A., Manokaran, R., Noubir, G., Rangan, C.P., Sundaram, R.: Steganographic communication in ordered channels. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 42–57. Springer, Heidelberg (2007)
15. Eidenbenz, R., Locher, T., Wattenhofer, R.: Hidden communication in P2P networks steganographic handshake and broadcast. In: Proceedings IEEE INFOCOM 2011, pp. 954–962 (2011)
16. Forest, K., Knight, S.: Permutation-based steganographic channels. In: Fourth International Conference on Risks and Security of Internet and Systems (CRiSIS), pp. 67–73 (2009)
17. Rudebusch, W.G.: Permutation steganography in many systems. Master's thesis, University of Nevada, Reno (2011)
18. Mosunov, A., Sinha, V., Crawford, H., Aycock, J., de Castro, D.M.N., Kumari, R.: Assured supraliminal steganography in computer games. In: Kim, Y., Lee, H., Perrig, A. (eds.) WISA 2013. LNCS, vol. 8267, pp. 245–259. Springer, Heidelberg (2014)
19. Tapiador, J.M., Hernandez-Castro, J.C., Alcaide, A., Ribagorda, A.: On the distinguishability of distance-bounded permutations in ordered channels. Trans. Info. For. Sec. **3**, 166–172 (2008)
20. Anderson, R., Needham, R., Shamir, A.: The steganographic file system. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 73–82. Springer, Heidelberg (1998)
21. McDonald, A.D., Kuhn, M.G.: StegFS: A steganographic file system for Linux. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 463–477. Springer, Heidelberg (2000)
22. Pang, H., Tan, K.L., Zhou, X.: StegFS: a steganographic file system. In: 19th International Conference on Data Engineering 2003, pp. 657–667 (2003)
23. Niu, X., Li, Q., Wang, W., Wang, Y.: G bytes data hiding method based on cluster chain structure. Wuhan University J. Nat. Sci. **18**, 443–448 (2013)
24. Srinivasan, A., Wu, J.: Duplicate file names-a novel steganographic data hiding technique. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. CCIS, vol. 193, pp. 260–268. Springer, Heidelberg (2011)
25. Thompson, I., Monroe, M.: FragFS: An advanced data hiding technique. Presentation at BlackHat Federal (2006)
26. Shu-fen, L., Sheng, P., Xing-yan, H., Lu, T.: File hiding based on FAT file system. In: IEEE International Symposium on IT in Medicine Education, ITIME 2009, vol. 1, pp. 1198–1201 (2009)
27. Khan, H., Javed, M., Khayam, S.A., Mirza, F.: Designing a cluster-based covert channel to evade disk investigation and forensics. Comput. Secur. **30**, 35–49 (2011)
28. Srinivasan, A., Stavrou, A., Nazaraj, S.T.: HideInside - a novel randomized & encrypted antiforensic information hiding. In: Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC), ICNC 2013, pp. 626–631. IEEE Computer Society, Washington, DC (2013)
29. The grugq: The art of defiling - defeating forensic analysis on Unix file systems. Presentation at BlackHat Asia (2003)
30. Savoldi, A., Gubian, P.: Data hiding in SIM/USIM cards: A steganographic approach. In: Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2007, pp. 86–100. IEEE Computer Society, Washington, DC (2007)

31. Savoldi, A., Gubian, P.: SIM and USIM filesystem: A forensics perspective. In: Proceedings of the 2007 ACM Symposium on Applied Computing, SAC 2007, pp. 181–187. ACM, New York (2007)
32. Davis, J., MacLean, J., Dampier, D.: Methods of information hiding and detection in file systems. In: Proceedings of the 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2010, pp. 66–69. IEEE Computer Society, Washington, DC (2010)
33. Huebner, E., Bem, D., Wee, C.K.: Data hiding in the NTFS file system. Digital Invest. **3**, 211–226 (2006)

Author Index