Universitat Politècnica de Catalunya

Department of Telematics Engineering

Ph.D. Dissertation

Transparent Protection of Data

Author: Francesc Sebé Feixas

Advisor: Dr. Josep Domingo Ferrer

Tutor: Dr. Miquel Soriano Ibáñez

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF TELEMATICS ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF UNIVERSITAT POLITÈCNICA DE CATALUNYA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR

December 2002

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor.

Dr. Josep Domingo Ferrer
(Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor.

Dr. Miquel Soriano Ibáñez
(Tutor)

Approved by the University Committee on Graduate Studies:

Preface

This thesis is about protection of data that have to be made available to possibly dishonest users. Data must be protected while keeping its usability. Such protection must be imperceptible, so as not to disrupt correct use of data, and strong against unauthorized uses. The study is divided regarding the two kinds of data whose transparent (imperceptible) protection has been addressed: multimedia content and statistical microdata.

In electronic commerce of multimedia content, merchants sell data to untrusted buyers that may redistribute it. In this respect, intellectual property rights of content providers must be ensured. Different watermarking and fingerprinting schemes are presented focusing on digital images.

Rather often, multimedia content are published in untrusted sites where they may suffer malicious alterations. Invertible watermarking has been studied to provide transparent lossless authentication and integrity to digital images.

When statistical files containing information about individual entities are released, privacy is a major concern. Such files must be masked so that data stay statistically useful but no information about individuals can be inferred. A proposal to enhance the best-performing masking methods is presented in this thesis.

Contents

List of Figures

Chapter 1

Introduction

1.1 Situation and objectives

In the last years, the great development of the Internet has generated an exponential growth of the amount of available data in the network. These data are easily shared and can be accessed from any computer connected to the network.

Obviously, not all data can be freely distributed, so they must be protected against unauthorized uses. The common way of protecting data consists of restricting their accessibility through encryption. In this way, data stay protected when stored in insecure places or transmitted through insecure channels. Unencrypted data can then be recovered by authorized users who know the decryption key. The aforementioned solution cannot be applied when authorized users are not completely trusted as, once data are decrypted, they are not protected anymore. So, other methods must be used to protect data when they have to be made available to possibly dishonest users. These methods protect data in such a way that protection stays imperceptible, *i.e.* *transparent*, to users. Thus such methods aim at

transparent protection of data.

In this thesis we focus on the transparent protection of two kinds of data:

Multimedia content: In electronic commerce of multimedia content, merchants sell their products to possibly dishonest buyers that may redistribute them. In this case, data are imperceptibly protected against illegal redistribution by using *steganography* [1] to embed copyright information in them.

Very often, multimedia contents are published in untrusted places where they could suffer malicious alterations. In an environment where only a few users (not all of them) need to make sure the published contents have not been altered, transparent protection of such contents is a good solution. Steganography can be used to provide authentication while keeping protection imperceptible.

Statistical microdata: Policy makers and researchers often request statistical offices to release data to perform statistical studies. When these data contain information about individual entities (microdata), privacy becomes a top priority issue. Data must be released in a way that combines statistical utility and protection of the privacy of entities concerned.

One of the properties of products sold in electronic format is that they can be copied very cheaply and without quality loss. Although this reduces the production costs, it facilitates illegal redistribution. This illegal redistribution can be rather efficient and fast when taking advantage of the Internet potential. Current research in electronic copyright protection is

[1] *Steganography is the art of hiding a secret message within a larger one in such a way that others cannot discern the presence or contents of the hidden message [KP00].*

focused on copy detection. In it, steganography is used to embed copyright information in the product before being sold. In case of discovery of illegal copies, this additional information will be retrieved and will allow ownership of the content to be proven or the identity of the dishonest buyer who began such distribution to be determined.

An important part of our research is focused on the development of new copyright protection schemes offering new properties.

Traditionally, authentication and integrity of data being published and available through computer networks has been ensured by digital signatures. A digital signature consists of a message attached to the content to be protected that allows to detect any alteration that the product may suffer. It also allows the signature respondent to be authenticated. In the case of multimedia contents, dealing with an attached message is not a feasible option, especially if we want third parties to stay unaware of such protection. Rather than an attached message, a better solution is to imperceptibly embed the authentication message inside the content. In this way, content can be authenticated while overcoming the drawbacks of attached messages and achieving transparency.

Part of our research is focused on the study of reversible steganography for lossless authentication and integrity protection of multimedia contents.

Increased corporate, government and academic demand has prompted official statistics to release individual respondent data (microdata). Obviously, dissemination of such microdata must be done in a way that ensures protection of the confidentiality of survey respondents. Protecting confidentiality necessitates perturbing the data, so that individual respondent

cannot be identified, while preserving the analytical properties of the data file. Methods to pertub data in this way are called *statistical disclosure control* methods (also called *masking* methods when referred to microdata). Methods offering low disclosure risk usually cause a greater data perturbation leading to high information loss. On the other side, low information loss leads to high disclosure risk. Thus, there is a tradeoff between information loss and disclosure risk in masking methods.

Some masking methods presented in the literature offer good masking properties while keeping information loss low. Our research in this area has focused on the development of post-processing methods to post-process the output of current well-performing masking methods in order to decrease information loss while keeping disclosure risk low.

When protecting data in the way addressed in this thesis, the protection method introduces some amount of noise into the data. Generally, this noise is not a matter of concern as it remains imperceptible (or transparent) to users. But when dealing with precision-critical data, this noise may turn data useless for some applications.

To solve the above drawback, we have developed methods that make it possible for trusted users to remove part of the protection to obtain a clearer version of data. In this sense, untrusted users just see the completely protected data while, the more trusted a user is, the more noise she can remove.

1.2 Structure of this thesis

This thesis is organized as follows.

Chapter 2 presents a state of the art on transparent protection of multimedia contents and statistical microdata. It is divided in three main sections. The first two deal with the application of steganography to multimedia contents protection, focusing on digital images. More precisely, the first chapter section is about copyright protection while the second section is about data authentication. The third section deals with statistical disclosure control methods for statistical microdata.

Chapter 3 presents our contributions to watermarking for digital images. It is composed of five sections. The first section presents an algorithm to provide imperceptibility to mark embedding algorithms for images. The next two sections present two new watermarking schemes. Details on the properties of each scheme as well as examples are given. The fourth section proposes mixing watermarked digital objects to increase the robustness of current watermarking schemes by combining their properties. Specific examples involving image watermarking systems are given to prove the effectiveness of the approach. Finally, the last chapter section provides a study on the invertibility of a well known spread-spectrum watermarking scheme which proves suitable for lossless image authentication.

Chapter 4 presents our contributions to binary collusion-secure fingerprinting codes. More precisely, we present a construction that generates codes secure against collusions of up to three dishonest buyers. For a moderate number of possible buyers, our construction results in shorter codewords than the current general proposal by Boneh and Shaw.

In Chapter 5, our contributions to statistical disclosure control of statistical microdata are presented. The first chapter section presents a new score to compare different masking methods that allows consideration of masked data sets with a number of records not equal to the number of records

of the original data set. Next, we propose a post-masking optimization procedure which enhances current best-performing masking methods by decreasing information loss while keeping disclosure risk low.

Chapter 6 presents a novel application of watermarking to providing multilevel access to precision-critical data. This chapter is divided in two sections proposing solutions for statistical microdata and multimedia contents, respectively.

The concluding remarks and a summary of the results presented in this thesis can be found in Chapter 7. Some guidelines for future research are given in this chapter as well.

Chapter 2

State of the Art

The state of the art of transparent protection of multimedia content and statistical microdata is examined in this chapter. For the sake of concreteness, the multimedia content being considered consists of digital images. Open issues later addressed in this thesis are highlighted and dealt with in some detail.

2.1 Steganography for multimedia data copyright protection

In electronic commerce of multimedia contents, merchants sell products in electronic format. These contents can be copied very easily and without quality loss. When multimedia contents are sold to possibly dishonest buyers that may copy and redistribute them, an intellectual property rights problem arises which forces such contents to be protected.

Copy prevention solutions have proven ineffective, so other solutions must be deployed. A failure example of one of such systems can be found in [DVD].

Copy detection is the most promising solution. It is based on hiding an imperceptible mark in the product before selling it. This mark will keep embedded in all copies and future recovery from illegal copies will allow to prove ownership of the product (*watermarking*) or trace the dishonest user who has began redistribution (*fingerprinting*). To imperceptibly embed a mark in a product, copy detection techniques use *steganography* [KP00].

Steganography is the art of hiding a secret message within a larger one in such a way that third parties cannot discern the presence or contents of the hidden message.

There are two kinds of marks, depending on the information they carry: watermarks and fingerprints:

Watermark: The mark contains information about the owner of the content it is embedded in, so all copies carry the same embedded mark. Future retrieval of this mark allows ownership to be proved.

Fingerprint: The mark contains information about the buyer who has bought a certain copy of the product [Wag83]. In this way, all copies sold to different buyers carry a different embedded mark. Later recovery of this mark from illegally redistributed copies allows the dishonest buyer who permitted redistribution of her copy to be identified.

A copy detection scheme consists of two algorithms: *mark embedding* and *mark recovery*.

A general mark embedding procedure is depicted in Figure 2.1. It consists of an algorithm that takes as input the original object X, the mark M to be embedded and a secret key K only known to the merchant, and generates the marked object X' as output.

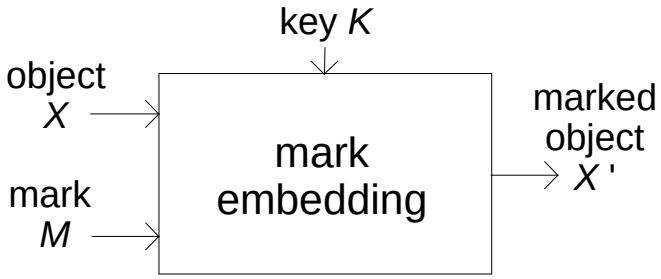

Figure 2.1: Mark embedding procedure.

A general mark recovery procedure is depicted in Figure 2.2. It takes as input a (probably) marked object $\hat{X}$, the secret key K and possibly other information depending on the specific algorithm. The procedure generates as output a Boolean value indicating whether a mark has been found or not, and depending on the scheme, the recovered mark $\hat{M}$ (when found).

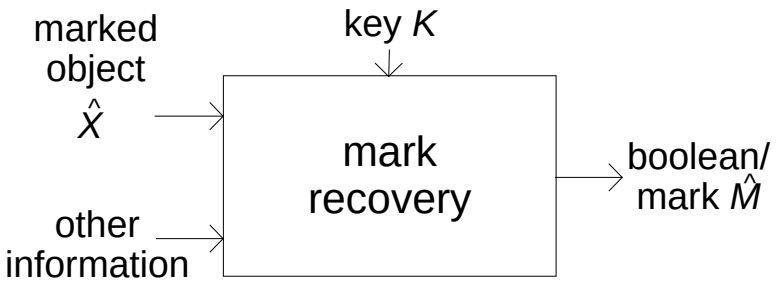

Figure 2.2: Mark recovery procedure.

2.1.1 Watermarking

In a watermarking scheme, the embedded mark contains information about ownership of the content it is embedded in. So, all watermarked copies of a product are identical.

Properties of a watermarking scheme

Next, we present a list of properties a watermarking scheme must meet:

Imperceptibility: The alterations applied to the contents for mark embedding must be imperceptible. This means the quality of content cannot be reduced after watermarking. The most appropiate metric to measure quality of watermarked multimedia contents is not yet clear. In [PA99] the Peak Signal-to-Noise Ratio, PSNR [1], is proposed as a quality metric for image marking systems. It is stated that PSNR between the original and the marked image ought to be greater than 38 dB.

Robustness: This property measures how resistant the mark is against attacks aiming at removing or making it unrecoverable. Ideal robustness is such that alterations necessary to remove the watermark without knowledge of the secret key are so large that the altered content loses its commercial value.

When dealing with multimedia contents, the amount of possible attacks is so great that it is not possible to use formal models of robustness. Instead, robustness is evaluated through benchmarks which take a marked object and apply different alterations to it. In [KP99], a benchmark is proposed to evaluate robustness of image marking systems. Its implementation can be found in [Sti]. Benchmarks like [KP99] evaluate robustness against standard signal processing attacks, but do not deal with *tamper-proofness*, *i.e.* with attacks that exploit knowledge of the algorithm internal operation. If watermarking

[1] $PSNR = 10 \log_{10} \frac{255^2}{(1/L)\sum_{i=1}^{L}(X_i - X_i')^2}$, where X_i and X_i' are the original and the marked pixel values, respectively, and L is the total number of pixels of the image.

is to conform to Kerckhoff's assumption of algorithms being public-domain, tamper-proofness becomes a relevant issue.

A useful feature regarding robustness is *multiple marking* support. In schemes presenting this property, consecutive markings on the same content are possible in such a way that different marks do not interfere. In this way, individual marks can be retrieved by running the mark recovery algorithm with the corresponding key.

In a watermarking scheme, only *single-user attacks* (those performed by a single buyer) make sense. This is because all copies are identical. Later we will see that, using the fact that in fingerprinting every copy is slightly different to others, different buyers can *collude* by comparing their copies to find differences and use this information to try to compose a new copy whose mark does not identify any of them.

Information rate: This property measures how much information can be robustly embedded in a product without degrading its quality. It can be seen as the bitlength of the mark. In [PA99] it is stated that ideally one should be able to embed at least a 70-bit watermark.

Secret information: This is the minimum amount of information that must be kept secret to ensure the robustness of the scheme. Obviously, the original object must be always kept secret together with the secret key.

Imperceptibility and robustness in image watermarking

The imperceptibility property states that alterations applied to the digital object during the marking procedure must not degrade its quality.

In the case of digital images, these alterations are modifications to the color level of their pixels. A mark embedding algorithm that makes small alterations will produce a marked image with a very high degree of imperceptibility but whose mark will be easily disrupted by low intensity (and also imperceptible) attacks. We conclude that mark embedding algorithms producing small alterations to pixels usually produce imperceptible but weak marks while those producing large alterations generate robust marks but higher quality degradation. Thus, there is tradeoff between robustness and imperceptibility inherent to the marking process. To optimize this tradeoff, information is needed about the maximum increment/decrement each pixel can accomodate without generating a negative visual impact; the goal is to determine the strongest mark that can be embedded while keeping an acceptable imperceptibility.

In [Her00], the JPEG [NG96] lossy compression algorithm is used to estimate the maximum modification applicable to each pixel. The original image X is compressed using the JPEG algorithm at a given quality q. Then, the image is decompressed and gives a slightly different image X' as result. The maximum color modification a pixel i can accomodate is given by $\delta_i = |X_i - X'_i|$.

Other proposals embed information in a transformed domain. In these proposals, alteration is applied to the transformed coefficients. In this case, it is necessary to determine which coefficients can better accomodate information without degrading the image that is recovered by applying the inverse transform.

- In [HRPP+98], information is embedded into the 30% middle-frequency

DCT coefficients.

- The scheme presented in [CLMT00] operates in the wavelet transfomed domain and obtains a masking function from sub-band decomposition. This masking function accounts for luminance sensitivity, spatial activity and sub-band orientation.

In Section 3.1, a new algorithm is proposed which is built over the idea that dark pixels and those in non-homogeneous regions can accomodate alterations without generating a negative visual impact.

Classification of watermarking schemes

In [Her00], watermarking schemes are classified depending on the additional information needed by the mark recovery algorithm (see Figure 2.2):

Private watermarking: The mark recovery algorithm of these schemes requires the following inputs: the probably marked object $\hat{X}$, the original object X, the secret key K and the embedded mark M. The output is a Boolean *true/false* value indicating whether the mark M has been found in $\hat{X}$ or not. The schemes presented in [CKLS97, KH97] fall in this category.

Semi-public watermarking: In these schemes, less information is required. There are two possibilities:

- The probably marked object $\hat{X}$, the secret key K and the mark M are needed. The output is Boolean and indicates whether the mark M has been found in $\hat{X}$ or not. [RA00, CLMT00, LM00, LM01] are some examples of image watermarking schemes falling in this category.

- The probably marked object $\hat{X}$, the secret key K and the original object X are needed. If a mark is found, it is given as output; otherwise, a *no mark found* message is given. The schemes proposed in [RP98, LPZ99, Her00] fall in this category.

Public watermarking: These schemes only require the probably marked object $\hat{X}$ and the secret key K. If a mark is found, it is given as output; otherwise, a *no mark found* message is given. Examples of such schemes are [HG98, AM00, Che00].

Section 3.2 presents a new semi-public image watermarking scheme offering new properties.

The concept of *oblivious watermarking* is also found in the literature. Oblivious watermarking schemes are those in which the mark recovery algorithm does not require the original unwatermarked object. Thus, from the above classification, oblivious watermarking includes both public schemes and the subset of semi-public ones that require knowledge of the embedded mark but not of the original object.

Oblivious watermarking

Oblivious watermarking offers a greater organizational flexibility and is better adapted to distributed copy detection than non-oblivious watermarking. For example, it enables the merchant to delegate copy detection to a set of agents distributed over the Internet, who can recover marks from intercepted redistributed content without having been entrusted with the original content. Such an arrangement minimizes disclosure of the original unprotected content (which stays only known to the merchant) and also

minimizes storage requirements (agents do not have to store the original version of all digital content they can come across of).

Commercial oblivious watermarking schemes surviving a broad range of manipulations (*e.g.* Digimarc [Dig]) tend to be based on propietary algorithms not available in the literature.

There are two shortcomings affecting oblivious watermarking systems in the literature:

- Many published oblivious proposals require the embedded sequence to be given as an input to the mark recovery procedure (semi-public schemes).

- To our best knowledge, no oblivious proposal in the literature embeds marks so that they can survive scaling and/or geometric distortion attacks.

Next, we explain the two shortcomings above in more detail.

Many published oblivious schemes require previous knowledge of the embedded sequence for mark detection. This requirement makes mark recovery more robust but less flexible as the merchant needs to know beforehand which sequence she is looking for. Such knowledge is definitely unrealistic if watermarking is used for fingerprinting (where the merchant embeds a different serial number or buyer ID in each copy being sold).

Examples of schemes presenting this problem are [RA00, CLMT00, LM00, LM01]. In these proposals, the watermark takes the form of a Gaussian or binary pseudo random sequence s which is embedded in some transform domain. Let $C = \mathcal{T}(X)$, where $\mathcal{T}$ denotes some transform and C are the transform coefficients of the original unmodified image X. A subset c of C is modified to embed the watermark. Let $C = c \cup \overline{c}$, where $c \cap \overline{c} = \emptyset$. Denoting

by $\mathcal{E}$ the watermark embedding function, the overall embedding operation can be expressed as

$$C = \mathcal{T}(X) \qquad c' = \mathcal{E}(c, s) \qquad C' = c' \cup \overline{c} \qquad X' = \mathcal{T}^{-1}(C')$$

Let $\hat{X} = X' + N$ be the image in which the presence of the watermark is tested, where N is some noise that can appear between mark embedding and mark detection. The detection operation can be expressed as

$$\hat{C} = \mathcal{T}(\hat{X}) \qquad \hat{s} = \mathcal{D}(\hat{c}) \qquad s_d = \frac{s^T \hat{s}}{\mid s \mid\mid \hat{s} \mid}$$

where $\mathcal{D}$ is a detector function and s_d is the detection statistic ($-1 \leq s_d \leq 1$) which is a measure of the normalized correlation of the embedded and detected sequences. In these schemes, an image $\hat{X}$ is considered to contain the watermark s if s_d is greater than a fixed threshold. Of course, computing s_d requires previous knowledge of s.

To our best knowledge, robustness against scaling and geometric distortion attacks is not achieved by any published oblivious scheme. Some previous schemes assume that such attacks can be undone prior to mark recovery [RA00, LM00]. Undoing them requires knowledge of the original image which turns those schemes into non-oblivious ones.

In [AM00] an iterative search technique is used to cope with geometric attacks. The search technique seeks to emulate the inverse operation of the attack. It consists of running the mark recovery algorithm after trying various inverse operations until the bit error rate of the hidden bits drops dramatically or a high correlation with the original watermark is obtained. Thus, even if this system does not generally require knowledge of the

embedded mark for recovery, it definitely requires such knowledge in order to survive geometric attacks. Furthermore, the search technique used to undo such attacks is too expensive and cannot be applied to random distortion attacks where the inverse operation is unknown.

In other oblivious schemes, geometric attacks are left for future research [CLMT00, LM01] or are not even mentioned [HG98, Che00].

Section 3.3 presents a new oblivious image watermarking scheme overcoming both mentioned drawbacks.

2.1.2 Fingerprinting

In a fingerprinting scheme, the merchant embeds a different buyer-identifying mark in each copy being sold [Wag83]. Later recovery of this mark from an illegally redistributed content allows identification of the buyer the original copy was sold to.

Properties of a fingerprinting scheme

Fingerprinting schemes are built over a robust watermarking system. This means that all properties listed in Section 2.1.1 are also necessary in fingerprinting. Some more properties must be met:

Security against collusion: The fact that every buyer receives an object with a different mark makes *collusion attacks* possible. In these attacks, a group of dishonest buyers compare their copies to find differences. This information is then used to try to compose a new copy from which none of their marks can be retrieved. Robustness against these attacks must be provided.

In [BS95], the *marking assumption* is introduced which states that, in a collusion attack, only *detectable positions* of the mark are alterable. Detectable positions are those in which the colluders find some difference when comparing their copies. The underlying watermarking scheme is assumed to be ideally robust against single-user attacks.

Security for the buyer: An honest buyer has to be sure she will not be acused falsely by a dishonest merchant. In the fingerprinting paradigm introduced in [Wag83], the mark is embedded into the content by the merchant who later sells the marked copy to the buyer. In this way, both the merchant and the buyer know the marked copy. Schemes operating in this way are classified as *symmetric schemes*. The main problem of such schemes is that a dishonest merchant can redistribute himself a copy recently sold and accuse the buyer of illegal redistribution. This argument can be used by a dishonest buyer who can claim it was the merchant who redistributed her copy.

To prevent this situation, only the buyer must know her marked copy. Schemes offering this property are called *asymmetric schemes*. In asymmetric schemes, the mark embedding procedure is replaced by a protocol in which both the merchant and the buyer play an active role. As a result of this protocol, the buyer gets a marked object to which no one else, including the merchant, has had access.

Security for the buyer also depends on security against collusion in the sense that a coalition of dishonest buyers must not be able to generate a new marked object whose mark accuses an honest buyer.

Anonymity: In schemes satisfying this property, the identity of honest buyers must be kept secret unless they act dishonestly. This means a

particular buyer's anonymity will only be lost if a copy purchased by that buyer is found to have been illegally redistributed.

In [DH98], an anonymous and asymmetric fingerprinting scheme is presented.

Binary collusion-secure fingerprinting codes

In [BS95], a general construction is given for obtaining binary fingerprinting codes secure against collusions of up to c buyers (c-*secure* codes). For N possible buyers and given $\epsilon > 0$, $L = 2c\log(2N/\epsilon)$ and $d = 8c^2\log(8cL/\epsilon)$, a code with N codewords of length

$$l = 2Ldc = 32c^4\log(2N/\epsilon)\log(8cL/\epsilon)$$

is constructed which allows one of the colluders to be identified with probability $1-\epsilon$. The authors also show that, for $c \geq 2$ and $N \geq 3$, it is not possible to obtain c-secure codes where colluders are identified with probability 1.

In [DH00b], it is shown that, for $c = 2$, collusion security can be obtained using the error-correcting capacity of dual binary Hamming codes. In this way, 2-secure binary fingerprinting codes are obtained which are much shorter than those obtained via the general construction [BS95] for $c = 2$.

In Chapter 4, a new construction to obtain 3-secure binary fingerprinting codes is presented.

2.2 Steganography for multimedia data authentication

The commonest way to authenticate digital contents is through public key cryptography. This is typically achieved by attaching a hash of the content (*i.e.* an image) encrypted under the sender/author's private key [RSA78]. This encrypted message corresponds to the digital signature of the content and can be authenticated by anyone knowing the signer's public key.

The drawback of having to deal with an attached message is obvious. This need can be avoided by using steganography. In this case, instead of appending an authentication message to the content, the sender embeds it as a watermark in such a way that any alteration will be detected by the receiver.

Authentication of digital contents based on watermarking is not intended to replace classical cryptographic authentication protocols. Watermarking cannot provide the same security properties enjoyed when exchanging data using a public key infrastructure. The advantage of watermarking authentication is that it is imperceptible (transparent).

We focus on a simpler scenario where a receiver has to receive authenticated digital content from a sender through an open channel that can be accessed by other third-party entities. We require third parties to stay unaware of protection, *i.e.* protection to be transparent. In this scenario, sender and receiver are supposed to share a secret key.

2.2.1 Fragile watermarking for image authentication

The use of secure *fragile watermarks* has been proposed as a means to verify image integrity without using cryptography. Fragile watermarks are those

whose robustness is very low. Thus, the watermark can be corrupted by a very small distortion of the watermarked image.

The general paradigm assumes that the image can be divided into two disjoint sets: The set that determines the Message Authentication Code (MAC) and the set that will hold the MAC. It is important that these two disjoint sets do not interact, so that MAC embedding does not change the MAC itself.

An example of fragile watermarks can be found in [Won98]. A scheme is proposed where the LSBs (Least Significant Bits) of the original image are erased and replaced with the XOR of the hash of the 7 MSBs (Most Significant Bits) and a binary logo.

In [Fri02], a security analysis of fragile image authentication watermarks that can locate tampered-with areas is presented.

The drawback of authentication based on fragile watermarks is that the image will inevitably be distorted by the noise introduced during mark embedding. When protecting precision-critical images, such distortion may be unaffordable.

2.2.2 Invertible watermarking for image authentication

While most watermarking schemes introduce some small amount of non-invertible distortion to the image, *invertible watermarking* methods are such that, if the watermarked contents are deemed authentic, the distortion due to watermarking can be removed to obtain the original contents. Although

invertible authentication allows recovery of the original undistorted image, it cannot be applied to every image, as shown in [FGD01a].

In this paradigm, the MAC is calculated from the whole image and embedded in an invertible manner in the image.

The first document on invertible watermarking is conjectured to be the patent [HJRS99]; however, this is no public-domain know-how.

In [FGD01b], two invertible watermarking methods for authentication of digital images in the JPEG format are presented. Both methods embed the MAC in the quantized DCT[2] coefficients of the image.

The first method is based on lossless compression. It computes the MAC as the hash $\mathcal{H}$ of the stream of DCT coefficients $D_k(i,j),\ k = 1, \cdots B$ of the JPEG image, where B is the total number of blocks in the image. Then, by seeding a PRNG[3] with a secret key, a random walk is followed through the set E consisting of the middle frequency coefficients of all blocks. While following the random walk, a lossless compression algorithm for the least significant bit of the coefficients is run. This random walk stops when the length difference between the compressed bit stream C and the number of processed coefficients is large enough to accomodate the hash $\mathcal{H}$ in the saved space. Denote the set of visited coefficients as $E_1,\ E_1 \subseteq E$. At this moment, C and $\mathcal{H}$ are concatenated and inserted into the least significant bits of the coefficients from E_1.

[2]DCT=Discrete Cosine Transform
[3]PRNG=Pseudo-Random Number Generator

Authentication is performed by following the same random walk while recovering $\hat{\mathcal{H}}$ and decompressing C. The bit-stream resulting from decompressing C replaces the least significant bits of E_1. Finally, we compare the recovered hash $\hat{\mathcal{H}}$ and the hash of the stream of DCT coefficients of the restored image. If they agree, the image is deemed authentic.

Note that the previous scheme requires the entropy of the least significant bits stream to be low enough to allow substantial compression. Such entropy is low because the random walk is applied over quantized coefficients (quantization can be seen as a pre-processing of the original image which introduces some non-invertible distortion).

The second method presented in [FGD01b] requires using non-standard quantization tables that must be included in the header of the authenticated image. It is based on modifying quantization tables so that all quantized coefficients are even. In this way, any alteration to the LSBs will be trivially invertible.

Additive, non-adaptative watermarking is claimed to be invertible in [FGD01a, GFD01]. In the claim, the existence of an "inverse watermarking operation" is postulated for a generic additive, non-adaptative method, but details are given only on how to derive such inverse operation for the spread-spectrum, frequency-based watermarking algorithm [HRPP+98].

A general algorithm for invertible authentication for images is presented in [FGD01a] (See Figure 2.3):

1. Let X be the original image to be authenticated. Compute its hash $\mathcal{H}(X)$.

2. Choose an additive, non-adaptive robust watermarking technique and

generate a watermark pattern W from a secret key K, so that the payload of W is $\mathcal{H}(X)$.

3. Use a special "invertible addition" to add the watermark pattern W to X to create the authenticated image X'.

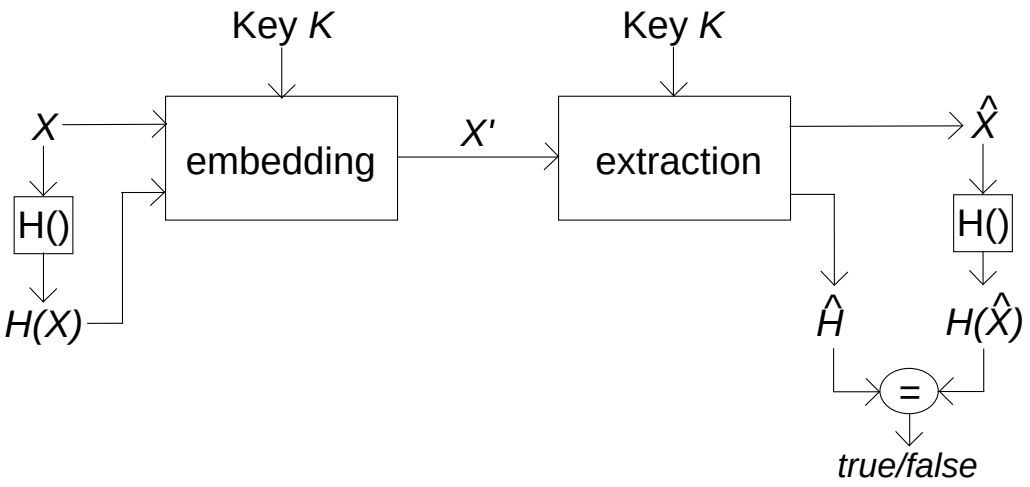

Figure 2.3: Invertible watermarking for image authentication.

The corresponding integrity verification algorithm is:

1. Extract the watermark bit-string $\hat{\mathcal{H}}$ (payload) from X'

2. Generate the watermark pattern $\hat{W}$ from the key K and the extracted bit-string $\hat{\mathcal{H}}$.

3. Using the inverse operation, substract $\hat{W}$ from X' to obtain $\hat{X}$.

4. Compare the hash of $\hat{X}$, $\mathcal{H}(\hat{X})$ with the extracted payload $\hat{\mathcal{H}}$. If they agree, the image is deemed authentic.

Section 3.5 analyzes the invertibility of the spread-spectrum watermarking scheme [HG98] and shows its applicability for image lossless authentication.

2.3 Statistical continuous microdata protection

Statistical offices must guarantee statistical confidentiality when releasing data for public use. *Statistical disclosure control* (SDC) methods are used to that end [WW01]. When data being released consist of individual respondent records, called microdata in the official statistics jargon, confidentiality means avoiding disclosure of the identity of the individual respondent associated with a published record. At the same time, SDC should preserve the informational content to the maximum extent possible. SDC methods are an intermediate option between encryption of the original data set (no disclosure risk but no informational content released) and straightforward release of the original data set (no confidentiality but maximal informational content released). SDC methods for microdata are also known as *masking* methods.

2.3.1 Current masking methods

There is a wide range of masking methods [DT01b]. From the point of view of their operational principles, current masking methods fall into the following two categories:

Perturbative: The microdata set is disturbed before publication. In this way data is altered. The perturbation method used should be such that

statistics computed on the perturbed dataset do not differ significantly from the statistics that would be obtained on the original dataset.

Nonperturbative: These methods do not alter data. They produce partial suppressions or reductions of detail in the original dataset.

From the point of view of the data to which they are applied, a second classification can be established:

Continuous: These methods are suitable for masking continuous variables. A variable is continuous if it is numerical and arithmetic operations can be performed on it. Some examples are age and height.

Categorical: These methods are suitable for masking categorical variables. A variable is categorical if it takes values on a finite set and arithmetic operations on it do not make sense. Some examples are day of the week and hair color.

We do not deal here with categorical variables. So from now on, we will only refer to continuous ones.

Perturbative methods

Perturbative methods are those which mask a data set by giving perturbed values rather than exact ones.

Next we show a list and a short description of current methods:

Additive noise: This method masks data by adding noise. The simplest algorithm consists of adding white noise to the data. More sophisticated methods use more or less complex transformations of the data and more complex error-matrices to improve the results. A description of different methods can be found in [Bra02].

Data distortion by probability distribution: This method replaces values of each variable of the data set by randomly generated values following the same distribution. It consists of three steps:

1. Identification of the density function of each variable of the data set.
2. Generation of random values from each estimated density function.
3. Mapping and replacement of the random generated series in place of the confidential series.

Resampling: Let V be a variable in a data set with n records. Draw with replacement t independent samples $X_1, \ldots, X_t$ of size n from the values of V. Independently rank each sample (using the same criterion for all samples). Finally, for $j = 1$ to n, compute the j-th value v'_j of the masked variable V' as the average of the j-th ranked values in $X_1, \ldots X_t$.

Microaggregation: This method clusters records into small aggregates of size at least k. Then, each value V_i of the original data file is replaced by the average of the values V_i of the records belonging to its aggregate. Univariate methods deal with multivariate datasets by microaggregating one variable at a time. This approach is known as individual ranking. There are multivariate methods that aggregate groups of more than one variable at a time (this is in fact a parameter of the algorithm). A more detailed description of microaggregation can be found in [DM02].

Lossy compression: This method consists of regarding a numerical

microdata file as an image (with records being rows, variables being columns and values being pixels). Lossy compression is then used on the image, and the decompressed image is then interpreted as a masked file. Using lossy compression algorithms with a quality parameter will allow for tunable masking intensity.

Rank swapping: First, values of each variable V_i are ranked in ascending order and information on the permutation π carried out during ranking is stored. Then each value V_i is swapped with another ranked value randomly chosen within a restricted range. Finally, the permutation π^{-1} is applied to the swapped values. This procedure is performed for all variables in the data set [Moo96].

Rounding: Original values of variables are replaced with multiples of a rounding basis b.

Nonperturbative methods

These methods rely on reductions of detail. Next we list the ones suitable for continuous microdata:

Global recoding: It consists of replacing a variable V_i by another variable V_i' which is a discretized version of V_i.

Top and bottom coding: This is a special case of global recoding. The idea is that top values (those above a certain threshold) are lumped together to form a new category. The same is done for bottom values.

2.3.2 Information loss metrics

To successfully compare different masking methods, it is necessary to have some way of measuring how much information has been lost after perturbing data.

In [DMT01], an information loss metric is proposed. Let X, X' be the original and masked data sets (both having n records and d variables). Let V and V' be the covariance matrices of X and X', respectively; similarly, let R and R' be the correlation matrices. The following information loss measures are proposed:

IL_1 describes information loss between X and X'. It computes the mean variation among both data sets:

$$IL_1 = \frac{\sum_{j=1}^{d} \sum_{i=1}^{n} \frac{|x_{ij} - x'_{ij}|}{|x_{ij}|}}{nd}$$

IL_2 is the mean variation between the averages of variables:

$$IL_2 = \frac{\sum_{j=1}^{d} \frac{|\overline{x}_j - \overline{x'}_j|}{|\overline{x}_j|}}{d}$$

IL_3 is the mean variation between covariances of variables:

$$IL_3 = \frac{\sum_{j=1}^{d} \sum_{1 \leq i \leq j} \frac{|v_{ij} - v'_{ij}|}{|v_{ij}|}}{\frac{d(d+1)}{2}}$$

IL_4 is the mean variation between variances of variables:

$$IL_4 = \frac{\sum_{j=1}^{d} \frac{|v_{jj} - v'_{jj}|}{|v_{jj}|}}{d}$$

IL_5 is the mean absolute error between correlation matrices:

$$IL_5 = \frac{\sum_{j=1}^{d} \sum_{1 \leq i < j} \frac{|r_{ij} - r'_{ij}|}{|r_{ij}|}}{\frac{d(d-1)}{2}}$$

All these measures are sumarized in an IL (Information Loss) measure computed as:

$$IL = 100 \cdot \frac{(IL_1 + IL_2 + IL_3 + IL_4 + IL_5)}{5}$$

The higher IL, the higher the information loss.

Computation of IL_1 implicitly assumes that there exists a one-to-one mapping between original and masked records. Corresponding records are assumed to be in the same relative position: the i-th masked record corresponds to the i-th original record.

2.3.3 Disclosure risk metrics

As pointed out in [DMT01], disclosure risk takes into account two aspects:

The first one is the record linkage disclosure risk. It is based on the assumption that an attacker has access to two datasets containing information on individual entities or people. The two datasets have some common variables that can be used to link records. A good masking record must guarantee very few records will be correctly linked.

Example: Suppose two datasets containg data about individual people. The first one contains the following variables: name, age, height, weight and other non confidential data which are publicy shown.

The second one contains medical information. The name of the respondent has been replaced by a code for anonymity's sake. Among other variables, there are height and weight.

Someone who has access to both datasets can use the common variables height and weight to link records from both sets and disclose the names of medical data set respondents (See Figure 2.4).

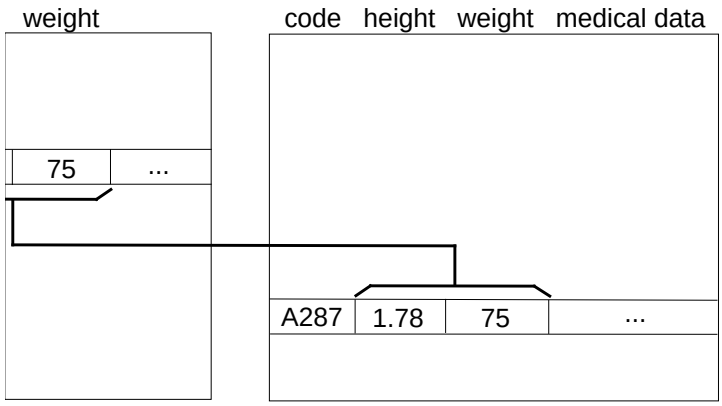

Figure 2.4: Record matching between two data sets.

The second aspect regards the information about original values an attacker can deduce once she gains access to masked values. It is necessary that, in a masked dataset, original values are not too close to original ones; otherwise they will be approximately known.

Distance-based record linkage

Let the original and masked data sets consist both of d variables (it is assumed that both data sets contain the same variables).

Assume further that the intruder can only access i key variables of the original data set and tries to link original and masked records based on these

i variables. Linkage then proceeds by computing i-dimensional Euclidean distances between records in the original and the masked data sets (using only i key variables). The variables are standardized to avoid scaling problems. A record in the masked data set is labelled as 'correctly linked' when the nearest record using i-dimensional distance is the correponding one (i-th masked record corresponds to the i-th original record).

From the record linkage method explained above, we define the DLD-i measure as the percent of records correctly linked using distance-based record linkage with Euclidean distance when the intruder knows i key variables of the original file.

The DLD (distance-based record linkage disclosure risk) measure is computed as the average of DLD-1,...,DLD-7. If data sets have a number d of variables less than 7, DLD will be computed as the average of DLD-1,...,DLD-d.

Probabilistic record linkage

This method defined in [Jar89] uses a matching algorithm to pair records in the masked and original data sets. The matching algorithm is based on the linear sum assignment model. The definition of "correctly linked" records is the same as in distance-based record linkage. This method is attractive because it only requires the user to provide two probabilities as input: one is an upper bound of the probability of a false match and the other an upper bound of the probability of a false non-match. Unlike distance-based record linkage, probabilistic record linkage does not require rescaling variables nor makes any assumption on their relative weight (by default, distance-based record linkage assumes that all variables have the same weight).

The PLD (probabilistic record linkage disclosure risk) measure is

computed in the same way as DLD but using probabilistic record linkage.

Interval Disclosure

Each variable is independently ranked and a rank interval is defined around the value the variable takes on each record. This interval has size p percent of the total number of records. Then, the proportion of original values that fall into the interval centered around their corresponding masked value is a measure of disclosure risk.

ID is then computed as the average percent of values falling in the intervals around their corresponding masked values. The average is over interval widths from $p = 1\%$ to $p = 10\%$ to each side of the masked value.

2.3.4 A score for method comparison

In [DMT01], a score is proposed to measure the tradeoff between information loss and disclosure risk based on the previous measures.

The score is computed as follows:

$$Score = 0.5 \cdot IL + 0.125 \cdot DLD + 0.125 \cdot PLD + 0.25 \cdot ID$$

The lower the $Score$, the better a masking method is.

This score assumes the i-th masked record corresponds to the i-th original record and does not deal with masked sets containing a differing number of records with respect to the original one. Section 5.1 presents a modification to overcome both mentioned drawbacks.

2.3.5 Best performing masking methods

In the comparison of [DMT01, DT01a], two masking methods were singled out as particularly well-performing to protect numerical microdata:

- Rank swapping
- Multivariate microaggregation

Section 5.2 presents a post-masking procedure to enhance performance of masking methods.

Chapter 3

Watermarking for Digital Images

In Sections 2.1 and 2.2 we pointed out the need to protect multimedia data against illegal redistribution and malicious alterations. In this chapter, we present our contributions to watermarking for digital images for copyright protection and authentication.

3.1 Image visual components for imperceptible mark embedding

Next, we propose an algorithm that computes pixel *visual components*, that is, the perceptual value of pixels. This value is an estimate of the maximum subperceptual increment/decrement that each pixel can accomodate without causing visual degradation.

The idea underlying Algorithm 1 is that dark pixels and those pixels in non-homogeneous regions are the ones that can best accomodate embedded information while minimizing the perceptual impact. This algorithm is used

to provide imperceptibility to the new watermarking schemes proposed in Sections 3.2 and 3.3 which we have published in [SDH00, SD01] respectively.

Without loss of generality, we will assume a monochrome image in what follows; for RGB color images, watermarking is independently done for each color plane. Let the original image be $X = \{x_i : 1 \leq i \leq n\}$, where x_i is the color level of the i-th pixel and n is the number of pixels in the image. Let x_i take integer values between 0 and $MAXCOLOR$, so that the lower x_i, the darker is the color level. Let parameter $dt \in [0, MAXCOLOR]$ be a threshold such that all color levels x_i below dt visually appear as dark.
Let lb_1 and ub_1 be integer values $lb_1 < ub_1$ that are used as parameters to bound the variation of pixel color values. For a given $MAXCOLOR$, suitable values for dt, lb_1 and ub_1 are empirically chosen.

Algorithm 1 (Visual components(dt, lb_1, ub_1))

1. *For $i = 1$ to n do:*

   (a) *Compute $m_i := \max_j |x_i - x_j|/2$, for all pixels j which are neighbors of pixel i in the image (there are up to eight neighbors); m_i can be regarded as a kind of discrete derivative at pixel i. To bound the value of m_i between lb_1 and ub_1, perform the following corrections:*

      i. *If $m_i > ub_1$ then $m_i := ub_1$.*

      ii. *If $m_i < lb_1$ then $m_i := lb_1$.*

   (b) *Compute the darkness of the i-th pixel as $d_i := (dt - x_i) * ub_1/dt$ if $x_i < dt$ and $d_i := 0$ otherwise. A pixel is considered as dark if its color level is below dt. The value of d_i lies between 0 and ub_1.*

(c) *Compute the preliminary visual component of the i-th pixel as* $v_i := \max(m_i, d_i)$.

2. *For* $i = 1$ *to* n *compute the final visual component of the i-th pixel as* $V_i := \max_j v_j$, *for all pixels* j *which are neighbors of* i *in the image plus the pixel* i *itself.*

The higher V_i for a pixel, the less perceptible are changes in that pixel.

Figure 3.1: Visual components of the image Lena.

Figure 3.1 shows the result of applying the visual component algorithm to the image Lena (parameters are $dt = 70$, $lb_1 = 2$ and $ub_1 = 11$). The lighter a pixel, the larger is its visual component value. Note that large visual component values are located in non-homogeneous or dark areas where color level alterations will not be easily perceived.

3.2 Scale-proof semi-public image watermarking

This section describes a new robust semi-public[1] watermarking scheme for images which we have published in [SDH00]. Bits are embedded into the image using the same underlying idea of the scheme described in [Her00], but are distributed so as to replace robustness against cropping attacks by robustness against scaling attacks.
Imperceptibility is achieved by using the visual components algorithm described in previous section.

The scheme consists of the mark embedding and mark recovery algorithms.

3.2.1 Mark embedding

A set of parameters must be specified. These are:

- dt, lb_1, ub_1 are required by the visual components algorithm (see Section 3.1) run before mark embedding to compute values V_j;
- k is a secret key only known to the merchant and used to generate a pseudo-random bit sequence $\{s_i\}_{i\geq 1}$. This sequence is used to encrypt the mark bits before embedding;
- p and r are two parameters used to locate the pixels into which mark bits will be embedded.

[1] Knowledge of the original image is needed for mark recovery.

Algorithm 2 (Mark embedding(p,r))

1. *Divide the image into the maximum possible number of square tiles of p pixels side, so that there is a r pixels wide band between neighboring tiles (the band separates tiles). Let q be the number of resulting tiles. Each tile will be used to embed one bit, so q is the capacity of this watermarking scheme.*

2. *Call ε the mark to be embedded. Encode ε using an error-correcting code (ECC) to obtain the encoded mark E. If $|E|$ is the bit-length of E, we must have $|E| \leq q$. Replicate the mark E to obtain a sequence E' with q bits.*

3. *For $i = 1$ to q compute $s'_i = e'_i \oplus s_i$, where e'_i is the i-th bit of E'.*

4. *To embed the i-th encrypted mark bit s'_i into the i-th tile do:*

   (a) *If $s'_i = 0$ then $x'_j := x_j - V_j$ for all pixels x_j in the i-th tile.*

   (b) *If $s'_i = 1$ then $x'_j := x_j + V_j$ for all pixels x_j in the i-th tile.*

5. *For every pixel j not lying into any tile, $x'_j := x_j$.*

$X' = \{x'_i : 1 \leq i \leq w \times h\}$ is the marked image.

3.2.2 Mark recovery

The assumptions for mark recovery are knowledge of the original image X, the secret key k (in order to regenerate the pseudo-random sequence $\{s_i\}_{i \geq 1}$) and parameters p and r. The only required knowledge on the original mark ε is its length, so that the proposed scheme is also usable for fingerprinting. Let $\hat{X}$ be the redistributed image, and let $\hat{w}$ and $\hat{h}$ be its width and height.

Algorithm 3 (Mark recovery(p,r))

1. *Let* $ones[\cdot]$ *and* $zeroes[\cdot]$ *be two vectors with* $|E|$ *integer positions initially all set to 0.*

2. *From the length* p *of the tile side, the width* r *of the intertile band and* X*, compute the number of tiles* q*.*

3. *For* $t = 1$ *to* q *do:*

   (a) *Let* $u := 1 + ((t-1) \bmod |E|)$

   (b) *For each pixel in the* t*-th tile of the original image* X *do:*

      i. *Let* i *and* j *be the row and column of the considered original pixel, which will be denoted by* x_{ij}*.*

      ii. *Locate the pixel* $\hat{x}_{ab}$ *in the marked image* $\hat{X}$ *corresponding to* x_{ij}*. To do this, let* $a := i \times \hat{h}/h$ *and* $b := j \times \hat{w}/w$*.*

      iii. *Compute* $\hat{\delta}_{ij} := \hat{x}_{ab} - x_{ij}$*.*

      iv. *If* $\hat{\delta}_{ij} > 0$ *then*

         A. *If* $s_t = 0$ *then* $ones[u] := ones[u] + 1$*.*

         B. *If* $s_t = 1$ *then* $zeroes[u] := zeroes[u] + 1$*.*

      v. *If* $\hat{\delta}_{ij} < 0$ *then*

         A. *If* $s_t = 0$ *then* $zeroes[u] := zeroes[u] + 1$*.*

         B. *If* $s_t = 1$ *then* $ones[u] := ones[u] + 1$*.*

4. *For* $u = 1$ *to* $|E|$ *do:*

   (a) *If* $ones_u > zeroes_u$ *then* $\hat{e}_u := 1$*, where* $\hat{e}_u$ *is the recovered version of the* u*-th embedded bit.*

(b) If $ones_u < zeroes_u$ *then* $\hat{e}_u := 0$.

(c) If $ones_u = zeroes_u$ *then* $\hat{e}_u := \#$, *where* $\#$ *denotes erasure.*

5. *Decode* $\hat{E}$ *with the same ECC used for embedding to obtain* $\hat{\varepsilon}$.

3.2.3 Parameter choice

The scheme was implemented with parameter values $MAXCOLOR = 255$, $dt = 70$, $lb_1 = 1$, $ub_1 = 4$. These values were empirically chosen to achieve a satisfactory tradeoff between robustness and imperceptibility.

Regarding parameters p and r, we recommend to use $p = 5$ and $r = 3$ as a tradeoff between capacity —which would favor tiles as small as possible and intertile bands as narrow as possible—, robustness —the larger a tile, the more redundancy in bit embedding and the more likely is correct bit recovery— and imperceptibility —the wider a band, the less chances for artifacts. The band between tiles is never modified and it helps avoiding perceptual artifacts that could appear as a result of using two adjacent tiles to embed a 0 and a 1.

3.2.4 Robustness assessment

The scheme, with parameters described above, was implemented using a dual binary Hamming code $DH(31, 5)$ as ECC (which provides a correction capacity of 7 errors per codeword). The base test of the StirMark 3.1 benchmark [PAK98] was used to evaluate its robustness. The following images from [Bas] were tried: Lena, Bear, Baboon and Peppers. A 70-bit long mark ε was used (as stated in [PA99]), which resulted in an encoded E with $|E| = 434$. Figure 3.2 shows the original and the marked Lena after

embedding a 70 bit length mark; the peak signal-to-noise ratio between both images is 41.13 dB.

Figure 3.2: Semi-public scheme: Left, original Lena. Right, Lena after embedding a 70 bit long mark.

The following StirMark 3.1 manipulations were survived by the embedded mark:

1. Color quantization.
2. Most filtering manipulations. More specifically:
   (a) Gaussian filter (blur).
   (b) Median filter (2×2 and 3×3).
   (c) Frequency mode Laplacian removal [BP98].
   (d) Simple sharpening.
3. JPEG compression for qualities 90% down to 20% (for some images down to 10%).

4. Rotations with and without scaling of -0.25 up to 0.25 degrees.

5. Shearing up to 1% in the X and Y directions.

6. Cropping up to 1%.

7. Row and column removal.

8. All StirMark scaling attacks (scale factors from 0.5 to 2).

Scaling is resisted because a mark bit is embedded in each pixel of a tile; even if the tile becomes smaller or larger, the correct bit can still be recovered. Extreme compression and scaling attacks for which the mark still survives are presented in Figure 3.3.

Figure 3.3: Semi-public scheme: Left, marked Lena after JPEG 15% compression. Right, marked Lena after 50% scaling.

3.3 Robust oblivious image watermarking

This section describes a new robust oblivious watermarking scheme for images which we have published in [SD01]. It overcomes the two main shortcomings of current oblivious schemes (see Section 2.1.1); it survives scaling and moderate geometric distortion attacks and does not need previous knowledge of the embedded mark for mark recovery.
Imperceptibility is achieved by using the visual components algorithm described in Section 3.1.

The scheme is composed of the mark embedding and mark recovery algorithms.

3.3.1 Mark embedding

A set of parameters must be specified. These are:

- dt, lb_1, ub_1 are required by the visual components algorithm (see Section 3.1) run before mark embedding to compute values V_j;
- lb_2, ub_2 are used when determining how color level encodes mark bits;
- k is a secret key only known to the merchant M and used to pseudo-randomly locate the pixels into which mark bits will be embedded and the way color levels determine the value of the embedded bits.

Algorithm 4 (Mark embedding(k, lb_2, ub_2))

1. *Let ε be the binary sequence to be embedded. Encode ε using an error-correcting code (ECC) to obtain the encoded mark E.*

2. *Using the key k as a seed, pseudo-randomly place $|E|$ non-overlapped square tiles R_i over the image, where $|E|$ is the bitlength of E. Tile size is also determined by k.*

3. *Using k as a seed, pseudo-randomly assign a value a_i between lb_2 and ub_2 to tile R_i, for $i = 1$ to $|E|$.*

4. *To embed the i-th bit e_i of the mark E in R_i:*

   (a) *Divide the color level interval $[0, MAXCOLOR]$ into subintervals of size a_i.*

   (b) *Label consecutive subintervals alternately as "0" or "1".*

   (c) *For each pixel x_j in R_i:*

      i. *If x_j lies in a subinterval labeled e_i, bring it as close as possible to the interval center by increasing or decreasing x_j no more than V_j.*

      ii. *If x_j lies in a subinterval labeled $\bar{e}_i$, bring it as close as possible to the nearest neighbor interval center (neighbor intervals are labeled e_i) by increasing or decreasing x_j no more than V_j.*

3.3.2 Mark recovery

Upon detecting a redistributed image $\hat{X}$, $\hat{\varepsilon}$ can be recovered as follows, provided that the length $|E|$ of the embedded mark and the secret key k used for embedding are known (the merchant should know these parameters).

Algorithm 5 (Mark recovery(k, lb_2, ub_2))

1. *Using the key k as a seed, pseudo-randomly place $|E|$ non-overlapped square tiles R_i over the image (again, tile size is also determined by k). Also using k as a seed, pseudo-randomly assign a value a_i between lb_2 and ub_2 to tile R_i, for $i = 1$ to $|E|$. (Tiling done in this step is analogous to tiling done during mark embedding).*

2. *To recover the i-th bit $\hat{e}_i$ of $\hat{E}$ from R_i:*

   (a) *Divide the color level interval $[0, MAXCOLOR]$ into subintervals of size a_i.*

   (b) *Label consecutive subintervals alternately as "0" or "1".*

   (c) *Let $ones := 0$ and $zeroes := 0$.*

   (d) *For each pixel x_j in R_i:*

      i. *If x_j lies in a subinterval labeled "1", then $ones := ones + 1$*

      ii. *If x_j lies in a subinterval labeled "0", then $zeroes := zeroes + 1$*

   (e) *If $ones > zeroes$ then $\hat{e}_i := 1$; if $ones < zeroes$ then $\hat{e}_i := 0$; otherwise $\hat{e}_i$ is an erasure.*

3. *Decode $\hat{E}$ with the same ECC used for embedding to obtain $\hat{\varepsilon}$.*

3.3.3 Parameter choice

In Algorithm 1 (visual components), suitable values for parameters dt, lb_1 and ub_1 should be empirically chosen for a given $MAXCOLOR$. Suitability depends on visual perception and robustness considerations. We have suggested a good choice for $MAXCOLOR = 255$, namely $dt = 70$, $lb_1 = 2$

and $ub_1 = 11$; for other values of $MAXCOLOR$, a rule of thumb of is to scale that choice by $MAXCOLOR/255$. We discuss below other parameters related to Algorithm 4 (embedding) and Algorithm 5 (recovery).

On the size of tiles

At Step 2 of the mark embedding algorithm, $|E|$ square tiles are randomly placed over the image. From the point of view of robustness, the tile size must be large enough so that each bit is embedded in a sufficient number of pixels. However, the requirement that all $|E|$ tiles should not overlap limits the maximum tile size, which decreases as $|E|$ increases.

An additional consideration is imperceptibility. Better imperceptibility is gained if neighboring tiles are separated by a band of unmodified pixels. This further limits the tile size.

On the width of color level subintervals

The size a_i in which we divide the color level interval is a tradeoff between robustness and imperceptibility:

- Making such intervals narrow means that, during the mark embedding algorithm, the variation applied to pixels to mark them is low, which leads to better imperceptibility. The drawback of narrow intervals is a loss of robustness, because even small noise can easily shift a color level to a neighboring subinterval, which can cause an incorrect bit to be recovered.

- Larger values a_i yield higher robustness, but moving the color level of a pixel to a neighboring subinterval is more perceptible and sometimes it cannot be achieved as the maximum increment/decrement of a pixel

x_j is limited by its visual component value V_j.

Thus, given $MAXCOLOR$, the interval $[lb_2, ub_2]$ where a_i randomly takes values has its lower bound lb_2 limited by robustness and its upper bound ub_2 limited by imperceptibility.

3.3.4 Robustness assessment

The scheme was implemented and tested using parameter values suggested in Section 3.3.3. The error-correcting code used was a dual binary Hamming code $DH(31, 5)$. The following images from [Bas] were tried: Lena, Bear, Baboon and Peppers. A 30-bit long mark ε was used, which needed six codewords of the dual Hamming code and resulted in an encoded E with $|E| = 31 \times 6 = 186$ bits. For Lena, a version of size 512×512 pixels was used and the length of the tile side was randomly chosen between 11 and 31; for the other images, a similar proportion between image size and tile size was maintained. For all images, tiles were placed so that neighboring tiles were separated by a band of unmodified pixels at least one pixel wide.

Figure 3.4 shows the original and the marked Lena after embedding a 30 bit long mark; the peak signal-to-noise ratio (PSNR) between both images is as high as 41.16 dB.

Some general considerations follow regarding robustness in front of the various kinds of attacks:

- After an attack, a bit is correctly recovered if a majority of correct mark bits are still inside the corresponding tile.

- Scaling attacks are survived by placing tiles in positions relative to the image size. In this way, even if the image size varies, each tile still contains original mark bits.

Figure 3.4: Oblivious scheme: Left, original Lena. Right, Lena after embedding a 30 bit long mark.

- Unless tiles are very small, other attacks like row and column removal, shearing, cropping and rotation will only succeed if most pixels in the tile suffer a variation so large that it leads to visual degradation.

The base test of the StirMark 3.1 benchmark [Sti] was used to evaluate robustness on the marked versions of the four test images. The following manipulations were survived:

1. Color quantization
2. Most filtering manipulations. More specifically:
   (a) Gaussian filter (blur)
   (b) Median filter (2×2, 3×3 and 4×4).
   (c) Linear filter
3. JPEG compression for qualities 90 down to 30.

4. All StirMark scaling attacks (scale factors from 0.5 to 2).

5. All StirMark aspect ratio modification attacks.

6. All StirMark row and column removal attacks.

7. All StirMark shearing attacks.

8. Small rotations with and without scaling from -2 to 2 degrees.

9. Small cropping up to 2%.

10. StirMark random bend.

3.3.5 Multiple marking

A useful feature of the presented algorithm is that multiple marking is supported. Up to three consecutive markings on the same image are possible without substantial perceptual degradation nor loss of robustness. For example, the content creator M_1 can mark an image and sell the marked image to a distributing company M_2 which re-marks the image with its own mark, re-sells it to a retailer M_3, who re-marks the image again before selling it to the end consumer. Each of M_1, M_2, M_3 can recover their embedded watermark by using the same key they used at embedding time.

To illustrate the effects on imperceptibility, Figure 3.5 shows Lena after three consecutive markings.

Figure 3.5: Oblivious scheme: Lena after three consecutive markings.

3.4 Enhancing watermark robustness through mixture of watermarked digital objects

Coming up with a watermarking method surviving all conceivable attacks may indeed be a difficult task. We explore in this section ways to obtain increased robustness by mixing the outputs of several watermarking methods. We have published the material in this section in [DS02a].

We will first discuss prior mixture, whereby a digital object is watermarked with different methods and a mixture of the watermarked objects is released. Posterior mixture will then be presented, which consists of mixing several attacked versions of the same watermarked digital object. It will be shown that prior mixture may result in a combination of the robustness properties of the watermarking methods being used. It will

also be shown that posterior mixture may allow recovery of the embedded watermark, even if this watermark can no longer be recovered from each individual attacked version of the watermarked object. Note that prior or posterior mixtures are non-exclusive.

3.4.1 Prior mixture

Prior mixture is a general technique that allows a watermarked object to be obtained that combines the robustness properties of several watermarking schemes. No knowledge on the specific embedding and recovery algorithms is needed as they are used as a black box.

Mark embedding and prior mixture

Let $E_1, \cdots, E_n$ be n different watermark embedding algorithms which can be used to embed a watermark M into the original digital object X. It is assumed in what follows that M contains some kind of redundancy (checksum, cyclic redundancy check, etc.), that allows its correctness or integrity to be checked. We then proceed as follows:

Algorithm 6 (Prior mixed embedding)

1. *The watermark M is embedded into X using algorithms $E_1, \cdots, E_n$ to obtain $X'_1, \ldots, X'_n$, where X'_i is the output of E_i.*
2. *A weight α_i is selected for each object X'_i, such that $0 \leq \alpha_i \leq 1$ and $\sum \alpha_i = 1$.*
3. *The watermarked mixed object is computed as*

$$X'_{premix} = f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) \tag{3.1}$$

where f is a mixture function (see below).

Figure 3.6 illustrates Algorithm 6.

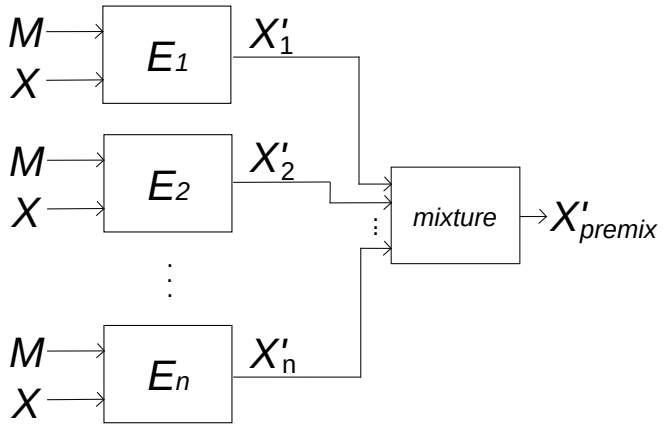

Figure 3.6: Prior mixture mark embedding procedure.

Any mixture function can be used in Algorithm 6. However, sensible choices are an additive mixture

$$f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) = \alpha_1 X'_1 + \cdots + \alpha_n X'_n$$

or a multiplicative mixture

$$f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) = X'^{\alpha_1}_1 X'^{\alpha_2}_2 \cdots X'^{\alpha_n}_n$$

The above mixtures are componentwise between the semantically corresponding components of objects: for example, if the object is an image, components are pixels and the mixture amounts to averaging the color levels of corresponding pixels.

Mark recovery from a mixed object

Denote by $R_1, \cdots, R_n$ the watermark recovery algorithms corresponding to embedding algorithms $E_1, \cdots, E_n$ respectively. Let $\hat{X}$ be the object we want to recover the watermark from; if it has been attacked, $\hat{X}$ will not exactly match any watermarked object X'. The recovery procedure is as follows:

Algorithm 7 (Recovery from a mixed object)

1. *Run algorithms $R_1, \cdots, R_n$ on $\hat{X}$ and record the output of those algorithms, if any. Depending on the attacks suffered by $\hat{X}$ some algorithms may give no output.*

2. *Look for a correct watermark among the outputs of the recovery algorithms (the redundancy included in marks is checked for correctness). If all correct watermarks found have the same value, then recovery is successful. If there is no correct watermark or if there are several correct watermarks with different values, recovery fails.*

Figure 3.7 illustrates Algorithm 7.

Note that mixing watermarked objects entails some amount of noise for each invidual watermarking method (E_i, R_i). In other words, when running recovery algorithm R_i, the effect of embedding algorithms E_j for $j \neq i$ is perceived as noise. Therefore, for prior mixture to be practical noise-robust watermarking methods must be used.

Applying prior mixture

Next, prior mixture is demonstrated for combining the crop-proof [Her00] and the scale-proof [SDH00] schemes for image watermarking. The resulting mixture stands both cropping and scaling attacks.

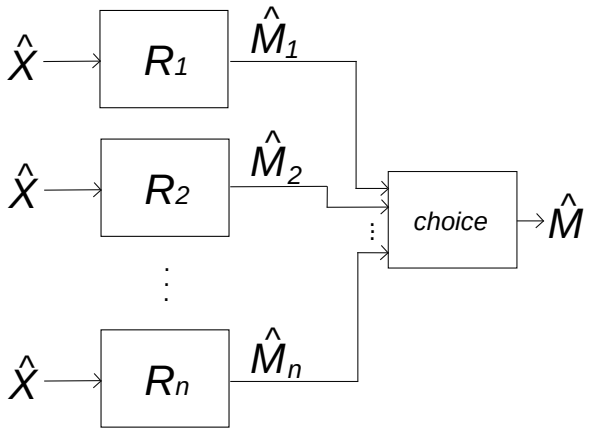

Figure 3.7: Prior mixture mark recovery procedure.

The benchmark image Lena [Bas] was watermarked using the two aforementioned schemes, so that two watermarked versions were obtained. In both cases, the embedded watermark was the same 45-bit binary sequence. Prior image mixture was applied to mix the two watermarked versions of Lena. Additive and multiplicative mixtures with weights $\alpha_1 = \alpha_2 = 0.5$ were tried; in what follows, we report results only for additive mixture, which turned out to outperform multiplicative mixture for this particular example. Additive mixture with the above weights is actually the arithmetic average of color levels of pairs of pixels in the same position within images to be mixed.

The error correcting code (ECC) used in this experiment was a $(31, 5)$ dual Hamming binary code (with correcting capacity 7 errors). When attempting mark recovery from an attacked watermarked image, the average number of corrected errors per codeword at the decoding stage gives an indication of the vulnerability of the scheme against the attack. If the number of errors that must be corrected to reconstruct the watermark is low, then the scheme easily survives the attack; the higher the number of corrected errors after an

attack	crop-proof	mix crop-proof
JPEG 30	2.1	not survived
gaussian	0	2.3
sharpening	0	0
FMLR	1.9	not survived
median 3x3	0	1
cropping	0	0

Table 3.1: Average no. of corrected errors at mark recovery for Lena (crop-proof method).

attack	scale-proof	mix scale-proof
JPEG 30	0	2.8
gaussian	0	1
sharpening	0	3.2
FMLR	5.5	not survived
median 3x3	1.7	not survived
scaling	0	1.5

Table 3.2: Average no. of corrected errors at mark recovery for Lena (scale-proof method).

attack, the more vulnerable is the scheme against the attack.

The following tables show the average number of errors corrected when recovering the watermark from the image Lena. Table 3.1 shows the average number errors corrected by the crop-proof recovery algorithm: the second column accounts for recovery from the crop-proof watermarked Lena (before mixing), while the third column refers to recovery from the mixed crop-proof and scale-proof Lena. Table 3.2 corresponds to errors corrected by the scale-proof recovery algorithm: its second column displays the average number of errors corrected when recovering a mark from the scale-proof watermarked Lena (before mixing); the third column refers to corrected errors in the recovery from the mixed crop-proof and scale-proof Lena.

It can be seen from Tables 3.1 and 3.2 that the result of mixing both

schemes is a semi-public image watermarking scheme robust against color quantization, filtering, JPEG compression, cropping and scaling. Thus, we have succeeded in combining resistance against cropping attacks with resistance against scaling attacks. Of course, the amount of noise tolerated by the mixture of both schemes is lower than the amount that would be tolerated by each scheme individually and some filters like FMLR are no longer survived.

The experiment above was repeated with other benchmark images in [Bas], and the results were similar to those obtained with Lena.

Imperceptibility is a very important feature of a watermarking scheme. It refers to the extent to which the image quality is preserved after the mark has been embedded. The Peak Signal-to-Noise Ratio (PSNR) between the original and the watermarked images is one common way to measure imperceptibility.

Table 3.3 shows how, after mixture, image quality does not decrease but stays similar or even higher than quality of watermarked images input to mixture. Table rows correspond to images Lena and Baboon [Bas]. Table columns correspond to the three watermarking possibilities: crop-proof only, scale-proof only or additive mixture of both methods. For each image, the PSNR of the three watermarked versions vs the original image is given. It is noteworthy that the PSNR of the mixed image can even be higher than the PSNR of images watermarked with a single method.

	crop-proof	scale-proof	mixed
Lena	38	41.12	40.59
Baboon	36.7	36.52	36.76

Table 3.3: PSNR of watermarked vs original images

3.4.2 Posterior mixture

Posterior mixture is a technique usable if the following assumptions hold:

A1. Several attacked versions $\hat{X}_1, \cdots, \hat{X}_m$ originating from the same watermarked digital object X' are available, where the watermarking method used and the embedded watermark are the same for all attacked versions. The difference between versions is only caused by the attacks they have undergone.

A2. None of $\hat{X}_1, \cdots, \hat{X}_m$ separately allows recovery of the common embedded watermark.

A3. It must be possible to find a one-to-one mapping between semantically corresponding components of $\hat{X}_i$ and $\hat{X}_{i+1}$, for $i = 1$ to $m - 1$. Note that some attacks may render fulfilling this assumption difficult or even infeasible. For example, let objects be images; then components are pixels and mapping semantically equivalent pixels may require undoing scaling attacks, rotation attacks, mapping cropped images with the corresponding parts of uncropped images, etc.

The recovery procedure based on a posterior mixture can be illustrated as shown in Figure 3.8 and be described as follows:

Algorithm 8 (Posterior mixed recovery)

1. *Mix the attacked watermarked objects, by computing*

$$\hat{X}_{postmix} = f(\beta_1, \cdots, \beta_m, \hat{X}_1, \cdots, \hat{X}_m)$$

where f is a componentwise mixture function (mixing semantically corresponding components, see Assumption A3 above) and β_j, for $j = 1, \cdots, m$ are weights such that $0 \leq \beta_j \leq 1$ and $\sum \beta_j = 1$.

2. *Use the recovery algorithm of the common watermarking method to recover the embedded watermark from $\hat{X}_{postmix}$.*

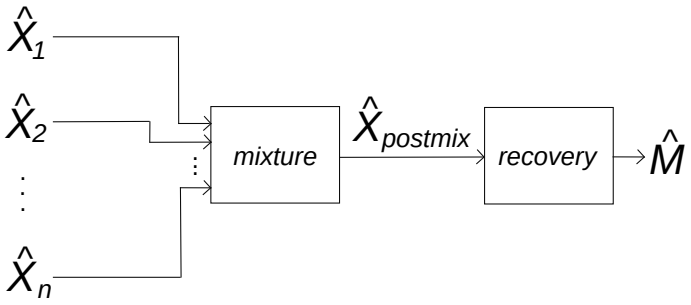

Figure 3.8: Posterior mixture mark recovery procedure.

Algorithm 8 must be regarded as a last chance to repair an otherwise unrecoverable attacked watermark. Posterior mixture can be used as a second line of defense in combination with prior mixture, *i.e.* prior mixture can be used before the attacks happen and posterior mixture after the attacks have happened: in this case, the attacked $\hat{X}_1, \cdots, \hat{X}_m$ would originate from the same prior mixed object X'_{premix} (see Expression (3.1)).

Applying posterior mixture

Using the oblivious scheme described in Section 3.3 a sequence of 35 bits was embedded into the benchmark image Lena. The embedded sequence was encoded using the $(31, 5)$ dual binary Hamming code.

Two attacks were performed on the watermarked image: the first one consisted of JPEG compression with quality 20, and the second one was a sharpening filter. From none of both attacked images could the watermark be recovered.

Posterior additive mixture with weights $\beta_1 = \beta_2 = 0.5$ was used to mix both attacked images. In other words, the arithmetic average of color levels for semantically corresponding pixels in the attacked images was computed; since neither compression nor sharpening attacks alter the size nor the orientation of images, semantically corresponding pixels are those occupying the same position in both images. The watermark was recoverable from the posterior mixed image, with an average number of 2.5 corrected errors per codeword, well below the correcting capacity of the $(31, 5)$ dual binary Hamming code (7 errors).

Exactly the same experiment was successfully repeated with other benchmark images, like Skyline_arch and Bear [Bas]. For those images, the average number of corrected errors per codeword were, respectively, 6 and 5.7, which are already closer to the correcting capacity of the code. Thus, the effectiveness of posterior mixture depends on the particular image, method and attacks being dealt with.

3.5 Invertible spread-spectrum watermarking for image authentication

In Section 2.2.2 invertible watermarking for image authentication is introduced. Next, we show how to invert under certain conditions one of the most widely known robust oblivious watermarking methods, namely the Hartung-Girod [HG98] spread-spectrum spatial-domain watermarking algorithm. This invertibility can be used to construct an image authentication scheme, as shown in this section. Results presented here have been published in [DS02b].

3.5.1 Hartung-Girod spread-spectrum watermarking

In [HG98], a spread-spectrum technique is used to obtain an oblivious watermarking method in the spatial domain. Oblivious watermarking does not require the original image to recover the watermark embedded in the watermarked image. We will first recall the fundamentals of this method and then we will show that, under certain conditions, this kind of watermarking is invertible.

The embedding and recovery procedures of [HG98] are as follows:

Embedding The copyright information to be embedded is a binary sequence $a_j, \quad a_j \in \{-1, 1\}$. This discrete signal is spread by a large factor cr, called chip-rate, to obtain the sequence $b_i = a_j, \quad j \cdot cr \leq i < (j+1) \cdot cr$. The spread sequence b_i is amplified by a locally adjustable amplitude factor $\alpha_i \geq 0$ and is then modulated by a binary pseudo-noise sequence $p_i, \quad p_i \in \{-1, 1\}$ generated from a seed s (which acts as the secret key). Let x_i be the original signal to be marked. The resulting watermarked

signal is

$$x'_i = x_i + \alpha_i \cdot b_i \cdot p_i \tag{3.2}$$

Recovery Mark recovery is performed by demodulating the watermarked signal with the same pseudo-noise signal p_i that was used for embedding, followed by summation over the window for each embedded bit, which yields the correlation sum c_j for the j-th information bit $c_j = \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i \cdot x'_i \approx \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i^2 \cdot \alpha_i \cdot b_i$. The sign of $c_j \approx cr \cdot \overline{\alpha}_j \cdot b_i = cr \cdot \overline{\alpha}_j \cdot a_j$, where $\overline{\alpha}_j = \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} \alpha_i / cr$, is interpreted as the embedded bit $\hat{a}_j$.

With the above method, several watermarks can be superimposed (multiple marking) if different pseudo-noise sequences are used for modulation. This is due to the fact that different pseudo-noise sequences are in general orthogonal to each other and do not significantly interfere [Nic88].

3.5.2 Inverting spread-spectrum watermarks

In order for the above watermarking scheme to be totally invertible, the following three conditions must be met:

1. The seed s used to generate the pseudo-noise signal p_i must be known. Being able to re-create p_i is needed to recover the embedded bits (see Algorithm 9 below).

2. The locally adjustable amplitude factor α_i used at each sample of the watermarked signal during the embedding phase must be known. α_i is needed to invert Equation (3.2) as shown in Equation (3.3). This requirement can be easily met by using a constant value α for all samples.

3. For every sample x_i to be modulated, its modulated value $x'_i = x_i + \alpha_i \cdot b_i \cdot p_i$ must fall within the same range of the original values x_i (otherwise truncation would be needed, which would hamper invertibility).

Assuming that the above three conditions are met, Algorithm 9 shows how the original unwatermarked signal x_i can be recovered from x'_i:

Algorithm 9 (Spread-spectrum watermark inversion)

1. *Recover all embedded bits, where the j-th embedded bit $\hat{a}_j \in \{-1, 1\}$ is obtained as the sign of the correlation sum $c_j = \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} p_i \cdot x'_i$.*

2. *Spread the recovered sequence $\hat{a}_j$ by the chip-rate cr value, to obtain the sequence $\hat{b}_i = \hat{a}_j, \quad j \cdot cr \leq i < (j+1) \cdot cr$.*

3. *Recover the original $\hat{x}_i$ sequence by computing*

$$\hat{x}_i = x'_i - \alpha_i \cdot \hat{b}_i \cdot p_i \tag{3.3}$$

Note that $\hat{x}_i = x_i, \ \forall i$ if $\hat{a}_j = a_j, \ \forall j$, *i.e.* if the embedded bits are correctly recovered, the unwatermarked image will match the original one. This can be readily seen by comparing Equations (3.2) and (3.3).

3.5.3 Image authentication using invertible spread-spectrum spatial-domain watermarking

Based on the spatial-domain spread-spectrum watermarking described above, we next adapt the ideas of [FGD01a] (reproduced in Section 2.2.2) to give a construction that, given an image, allows the hash of the image to be embedded in its pixels; the hash is used as a MAC (Message Authentication

Code). Anyone knowing the embedding key and the amplitude factor α is able to recover the embedded MAC, undo the watermark, get the original image and check for MAC validity.

Without loss of generality, we will assume a monochrome image in what follows. Let the original image be $X = \{x_i : 1 \leq i \leq n\}$, where x_i is the color level of the i-th pixel and n is the number of pixels in the image. Let x_i be the grayscale level of the pixel, which is assumed to take integer values between 0 and $MAXCOLOR$.

Invertible addition

One of the three conditions stated above for a watermark to be invertible is that the value of a modulated pixel must fall into the grayscale range of original pixels. In [HJRS99], modular addition modulo $MAXCOLOR$ is proposed as another way to keep modulated pixel values within $[0, MAXCOLOR]$. In [FGD01a], this operation is criticized because of possible visual artifacts in the watermarked image resulting from grayscale values close to 0 being flipped to grayscale values close to $MAXCOLOR$, and grayscale values close to $MAXCOLOR$ being flipped to values close to 0 (nearly white pixels become nearly black and conversely). We claim that, in addition to visual artifacts, modular addition may lead to incorrect watermark recovery when inverting the Hartung-Girod watermarking. This is illustrated by the following example.

Example 1 *In the watermarking procedure described in Section 3.5.1, assume that* $MAXCOLOR = 255$, $\alpha = 3$ *and that we want to embed* $a = 1$ *in the first four pixels of the original image. If the values of those four pixels are* $(v_0, v_1, v_2, v_3) = (1, 2, 3, 2)$ *and the first four bits of the pseudorandom sequence are* $(p_0, p_1, p_2, p_3) = (-1, 1, 1, -1)$, *we spread a over four pixels to*

obtain $(b_0, b_1, b_2, b_3) = (1, 1, 1, 1)$ *and compute*

$$x'_i = x_i + \alpha \cdot b_i \cdot p_i, \quad \textit{for } i = 0 \textit{ to } 3$$

This yields $(x'_0, x'_1, x'_2, x'_3) = (254, 5, 6, 255)$. *Values 254 and 255 result from modular reduction of* -2 *and* -1 *(which were out of range). Now, when trying to recover the embedded bit, we compute*

$$c = \sum_{i=0}^{3} p_i \cdot x'_i = -254 + 5 + 6 - 255 = -498$$

Since the sign of c *is negative, we reach the erroneous conclusion that the embedded bit was* $\hat{a} = -1$.

A better way to keep modulated pixels within range is to pre-process the image in the following simple way:

Algorithm 10 (Gray-level pre-processing(α))

For $i = 1$ *to* n *do:*

1. *If* $x_i < \alpha$ *then* $x_i := \alpha$
2. *If* $x_i > MAXCOLOR - \alpha$ *then* $x_i := MAXCOLOR - \alpha$

Algorithm 10 does indeed result in some non-invertible distortion, so when watermarking is inverted, there may be some slight difference between the grayscale values of some pixels of the original and the watermarked images. However, the advantages over modular addition are clear:

- There are no visual artifacts in the watermarked image, because the magnitude of grayscale changes is at most α levels.
- Erroneous bit recovery illustrated in Example 1 is avoided.

Hash embedding and verification

As shown in Section 2.2.2, invertible watermarking for image authentication consists of computing a hash of the original image and embedding the hash bits in the image. Our embedding algorithm takes as input the pre-processed image resulting from Algorithm 10 and depends on two parameters: a seed s for pseudo-random number generation and the amplitude factor α.

Algorithm 11 (Hash embedding(s,α))

1. *Compute the hash $\mathcal{H}$ of the pre-processed image X.*
2. *Construct the sequence a_i to be embedded by doing, for $i = 1$ to $|\mathcal{H}|$:*
   - *If $\mathcal{H}_i = 0$ then $a_i := -1$*
   - *If $\mathcal{H}_i = 1$ then $a_i := 1$*
3. *Using the spread-spectrum Hartung-Girod technique with parameter α and seed s, embed the sequence $\{a_i : i = 1, \cdots, |\mathcal{H}|\}$ into X. Let X' be the resulting watermarked image.*

The algorithm for image authentication, *i.e.* for verification of image integrity, is now straightforward:

Algorithm 12 (Integrity verification(s,α))

1. *Use Algorithm 9 to recover the embedded sequence $\hat{\mathcal{H}}$ and the unwatermarked image $\hat{X}$ from X'.*
2. *Compute the hash of $\hat{X}$, $\mathcal{H}(\hat{X})$*

3. *Compare* $\mathcal{H}(\hat{X})$ *with the hash* $\hat{\mathcal{H}}$*. If they agree, then* $\hat{X} = X$ *and the image is deemed authentic. If they do not,* $\hat{X}$ *is deemed non-authentic.*

Chapter 4

Collusion 3-Secure Fingerprinting Codes

In this chapter we present a construction to come up with collusion-secure fingerprinting codes for collusions of up to 3 colluders. Results presented here have been published in [SD02b]. For a not too large number of buyers, our construction generates much shorter codes than those obtained from the general construction [BS95] for $c = 3$. The basic idea is to compose a new kind of code, which we call *scattering code*, with a *dual binary Hamming code*.

Section 4.1 presents some definitions and properties on dual Hamming codes. Section 4.2 presents a set of lemmas on the probability of successful collusion as a function of colluders'strategy. The construction and decoding of scattering codes are introduced in Section 4.3. Then, Section 4.4 explains how to generate fingerprinting codes secure against collusions of up to three buyers by composing a scattering code code with a dual binary Hamming code together with some numerical results comparing the length of codes from our construction with the length of codes from [BS95].

4.1 Dual binary Hamming codes

The dual code of a binary Hamming code (denoted by $DH(n)$) is a binary code with 2^n codewords of length $N = 2^n - 1$ such that the distance between any two codewords is 2^{n-1}. A few definitions and useful properties related to such codes are presented next.

Definition 1 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code,* i.e. *$a^i = a^i_1 a^i_2 \cdots a^i_N$. Define $inv(a^1, a^2, a^3)$ to be the set of invariant positions between all three codewords, that is, those bit positions in which all three codewords have the same bit value. Formally speaking,*

$$inv(a^1, a^2, a^3) = \{i,\ 1 \leq i \leq N,\ a^1_i = a^2_i = a^3_i\}$$

Definition 2 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Define $minor(a^1; a^2, a^3)$ to be the set of bit positions in which a^1 has a value different from the values in a^2 and a^3 (for such positions, $a^2_i = a^3_i$). Formally speaking,*

$$minor(a^1; a^2, a^3) = \{i,\ 1 \leq i \leq N,\ a^1_i \neq a^2_i,\ a^1_i \neq a^3_i\}$$

Lemma 1 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code and let $|\cdot|$ denote the bitlength operator. Then it holds that $|inv(a^1, a^2, a^3)| = 2^{n-2} - 1$, $|minor(a^1; a^2, a^3)| = 2^{n-2}$, $|minor(a^2; a^1, a^3)| = 2^{n-2}$ and $|minor(a^3; a^1, a^2)| = 2^{n-2}$.*

Proof: Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Define $I = inv(a^1, a^2)$ and $\overline{I}$ to be the positions not in I. Since $d(a^i, a^j)_{i \neq j} = 2^{n-1}$, then $|I| = 2^{n-1} - 1$.

Let $x = |inv(a^1, a^2, a^3)|$ (obviously, $inv(a^1, a^2, a^3) \subset I$) and let y be the total number of positions $i \in \overline{I}$ where $a^2_i = a^3_i$ (these are the positions that

form $minor(a^1; a^2, a^3)$). As $d(a^2, a^3) = 2^{n-1}$, then $x + y = 2^{n-1} - 1$.

There are $2^{n-1} - 1 - x$ positions $i \in I$ where $a_i^3 \neq a_i^1$ (these are the positions that form $minor(a^3; a^1, a^2)$) and y positions $i \in \overline{I}$ where $a_i^3 \neq a_i^1$. As $d(a^3, a^1) = 2^{n-1}$ then $2^{n-1} - 1 - x + y = 2^{n-1}$.

Solving the following equations for x and y

$$\begin{cases} x + y = 2^{n-1} - 1 \\ 2^{n-1} - 1 - x + y = 2^{n-1} \end{cases}$$

we get $x = 2^{n-2} - 1$ and $y = 2^{n-2}$. Finally, we conclude

$|inv(a^1, a^2, a^3)| = x = 2^{n-2} - 1$,

$|minor(a^1; a^2, a^3)| = y = 2^{n-2}$,

$|minor(a^2; a^1, a^3)| = 2^{n-1} - y = 2^{n-2}$,

$|minor(a^3; a^1, a^2)| = 2^{n-1} - 1 - x = 2^{n-2}$

□

Example: The following are three codewords of a $DH(5)$ code.

	$inv(a^1, a^2, a^3)$	$minor(a^1; a^2, a^3)$	$minor(a^2; a^1, a^3)$	$minor(a^3; a^1, a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000

The codeword length is $2^5 - 1 = 31$. Now $|inv(a^1, a^2, a^3)| = 2^{5-2} - 1 = 7$, $|minor(a^1; a^2, a^3)| = |minor(a^2; a^1, a^3)| = |minor(a^3; a^1, a^2)| = 2^{5-2} = 8$.

□

Lemma 2 *Let a^1, a^2, a^3 be three codewords of a $DH(n)$ code. Then it holds that:*

- *There exists one and only one codeword $a^z \in DH(n) \backslash \{a^1, a^2, a^3\}$*

such that $a_i^z = a_i^1 = a_i^2 = a_i^3,\ \forall i \in inv(a^1, a^2, a^3)$. *Furthermore,* $a_i^z = a_i^1,\ \forall i \in minor(a^1; a^2, a^3)$, $a_i^z = a_i^2,\ \forall i \in minor(a^2; a^1, a^3)$ *and* $a_i^z = a_i^3,\ \forall i \in minor(a^3; a^1, a^2)$.

- *The remaining codewords satisfy that* $\forall a^j \in DH(n) \backslash \{a^1, a^2, a^3, a^z\}$, $d_{inv(a^1,a^2,a^3)}(a^j, a^1) = d_{minor(a^1;a^2,a^3)}(a^j, a^1) = d_{minor(a^2;a^1,a^3)}(a^j, a^1) = d_{minor(a^3,a^1,a^2)}(a^j, a^1) = 2^{n-3}$, *where* $d_P(x, y)$ *denotes Hamming distance between codewords* x *and* y *restricted to bit positions in* P. *The same distances hold with respect to* a^2 *and* a^3.

Proof: First, the existence and properties of a^z will be proven. As a $DH(n)$ code is a linear code, any linear combination of codewords results in another codeword. Then, we get $a^z = a^1 \oplus a^2 \oplus a^3$, where $\oplus$ denotes the component-wise modulo 2 addition.

We prove that $a_i^z = a_i^1 = a_i^2 = a_i^3,\ \forall i \in inv(a^1, a^2, a^3)$. This is true because if $a_i^1 = a_i^2 = a_i^3 = 1$, then $a_i^1 \oplus a_i^2 \oplus a_i^3 = 1$, and if $a_i^1 = a_i^2 = a_i^3 = 0$, then $a_i^1 \oplus a_i^2 \oplus a_i^3 = 0$.

Then, we prove $a_i^z = a_i^1,\ \forall i \in minor(a^1; a^2, a^3)$. This is true because $a_i^z = a_i^1 \oplus a_i^2 \oplus a_i^3$ and as $a_i^2 = a_i^3$, then $a_i^z = a_i^1$.

Using the same idea, we can prove $a_i^z = a_i^2,\ \forall i \in minor(a^2; a^1, a^3)$ and $a_i^z = a_i^3,\ \forall i \in minor(a^3; a^1, a^2)$.

Next, the second part of the Lemma will be proven. Consider $a^j \in DH(n) \backslash \{a^1, a^2, a^3, a^z\}$

Call x the number of positions in $inv(a^1, a^2, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $inv(a^1, a^2, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - 1 - x$ (see Lemma 1).

Call y the number of positions in $minor(a^1; a^2, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^1; a^2, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - y$.

Call z the number of positions in $minor(a^2; a^1, a^3)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^2; a^1, a^3)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - z$.

Call t the number of positions in $minor(a^3; a^1, a^2)$ where $a_i^j = a_i^1$. Then the number of positions in $minor(a^3; a^1, a^2)$ where $a_i^j \neq a_i^1$ is $2^{n-2} - t$.

Since $d(a^j, a^1) = d_{inv(a^1,a^2,a^3)}(a^j, a^1) + d_{minor(a^1;a^2,a^3)}(a^j, a^1) + d_{minor(a^2;a^1,a^3)}(a^j, a^1) + d_{minor(a^3,a^1,a^2)}(a^j, a^1) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + (2^{n-2} - y) + (2^{n-2} - z) + (2^{n-2} - t) = 2^{n-1}$$

Since $d(a^j, a^2) = d_{inv(a^1,a^2,a^3)}(a^j, a^2) + d_{minor(a^1;a^2,a^3)}(a^j, a^2) + d_{minor(a^2;a^1,a^3)}(a^j, a^2) + d_{minor(a^3,a^1,a^2)}(a^j, a^2) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + y + z + (2^{n-2} - t) = 2^{n-1}$$

Since $d(a^j, a^3) = d_{inv(a^1,a^2,a^3)}(a^j, a^3) + d_{minor(a^1;a^2,a^3)}(a^j, a^3) + d_{minor(a^2;a^1,a^3)}(a^j, a^3) + d_{minor(a^3,a^1,a^2)}(a^j, a^3) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + y + (2^{n-2} - z) + t = 2^{n-1}$$

Since $d(a^j, a^z) = d_{inv(a^1,a^2,a^3)}(a^j, a^z) + d_{minor(a^1;a^2,a^3)}(a^j, a^z) + d_{minor(a^2;a^1,a^3)}(a^j, a^z) + d_{minor(a^3,a^1,a^2)}(a^j, a^z) = 2^{n-1}$, we have

$$(2^{n-2} - 1 - x) + (2^{n-2} - y) + z + t = 2^{n-1}$$

From the expressions above, the following equation system can be derived:

$$\begin{cases} x + y + z + t = 2^{n-1} - 1 \\ -x + y + z - t = 1 \\ -x + y - z + t = 1 \\ -x - y + z + t = 1 \end{cases}$$

By solving it, we get $x = 2^{n-3} - 1$ and $y = z = t = 2^{n-3}$.

Finally, we conclude,

$d_{inv(a^1,a^2,a^3)}(a^j, a^1) = 2^{n-2} - 1 - x = 2^{n-3}$

$d_{minor(a^1;a^2,a^3)}(a^j, a^1) = 2^{n-2} - y = 2^{n-3}$

$d_{minor(a^2;a^1,a^3)}(a^j, a^1) = 2^{n-2} - z = 2^{n-3}$

$d_{minor(a^3;a^1,a^2)}(a^j, a^1) = 2^{n-2} - t = 2^{n-3}$

In the same way, we can prove these distances hold between a^j and a^2, a^3.

□

Example: The table below displays the unique codeword a^z corresponding to three particular codewords a^1, a^2, a^3 of a $DH(5)$ code.

	$inv(a^1, a^2, a^3)$	$minor(a^1; a^2, a^3)$	$minor(a^2; a^1, a^3)$	$minor(a^3; a^1, a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000
a^z	0000000	11111111	11111111	00000000

It can be seen that $a^z_i = a^1_i = a^2_i = a^3_i,\ \forall i \in inv(a^1, a^2, a^3)$. Also, $a^z_i = a^1_i,\ \forall i \in minor(a^1; a^2, a^3)$, $a^z_i = a^2_i,\ \forall i \in minor(a^2; a^1, a^3)$ and $a^z_i = a^3_i,\ \forall i \in minor(a^3; a^1, a^2)$.

□

Example: The table below displays three codewords a^1, a^2, a^3 of a $DH(5)$ code and another codeword $a^i \in DH(5)\backslash\{a^1, a^2, a^3, a^z\}$.

	$inv(a^1, a^2, a^3)$	$minor(a^1; a^2, a^3)$	$minor(a^2; a^1, a^3)$	$minor(a^3; a^1, a^2)$
a^1	0000000	11111111	00000000	11111111
a^2	0000000	00000000	11111111	11111111
a^3	0000000	00000000	00000000	00000000
a^i	0001111	00001111	00001111	00001111

It can be seen that $d_{inv(a^1,a^2,a^3)}(a^i, a^1) = d_{minor(a^1;a^2,a^3)}(a^i, a^1) = d_{minor(a^2;a^1,a^3)}(a^i, a^1) = d_{minor(a^3,a^1,a^2)}(a^i, a^1) = 2^{n-3} = 4$. The same distances hold between a^i and a^2, a^3. □

4.2 3-Collusions over $DH(n)$

4.2.1 Detectable positions

Let us assume three dishonest buyers c^1, c^2, c^3 compare their copies of the same multimedia content. According to the marking assumption [BS95], they can only modify the embedded marks in those *detectable* positions where not all three marks take the same bit value. In those positions, colluders can set the corresponding bit to '0', '1' or "unreadable". In this way, we conclude that, if three different buyers are assigned codewords a^1, a^2 and a^3 of a $DH(n)$ code, the result of their collusion will be a word a^{coll} where no bit has been modified in the $2^{n-2} - 1$ positions in $inv(a^1, a^2, a^3)$. On the other hand, colluders will be able to detect and identify positions in $minor(a^1; a^2, a^3)$ as the bit positions of those content fragments which are identical between the copies of c^2 and c^3 and different from the copy of c^1. In a similar way,

$minor(a^2; a^1, a^3)$ and $minor(a^3; a^1, a^2)$ can be detected and identified as well.

4.2.2 Decoding by minimum distance

As it has been said, colluders can generate a new object whose embedded codeword may have been altered in detectable positions. In this way, it is possible that the word retrieved from a collusion-generated object does not correspond to any $DH(n)$ codeword. In such cases, the recovered word will be error-corrected by minimum distance.

Thus, in order for a collusion to be successful, colluders c^1, c^2, c^3 with assigned codewords a^1, a^2, a^3, respectively, must generate, by mixing fragments of their copies, a word such that the closest codeword in the $DH(n)$ code is not in $\{a^1, a^2, a^3\}$ (see Figure 4.1). A successful collusion will cause an innocent buyer to be accused in lieu of the colluders. Note that we are assuming that colluders do not generate "unreadable" positions when colluding over $DH(n)$ codewords. It will be shown later that our construction actually prevents unreadable positions from being fed by colluders to the dual Hamming decoder.

4.2.3 The aim of colluders

As decoding is done by minimum distance, the aim of colluders is to come up with an object whose embedded word is as distant as possible from their assigned codewords.

Intuitively, it can be realized that all colluders must contribute with the same number of bits from their corresponding codewords. Otherwise, the collusion-generated word would be closer to the codewords of those colluders having contributed more bits.

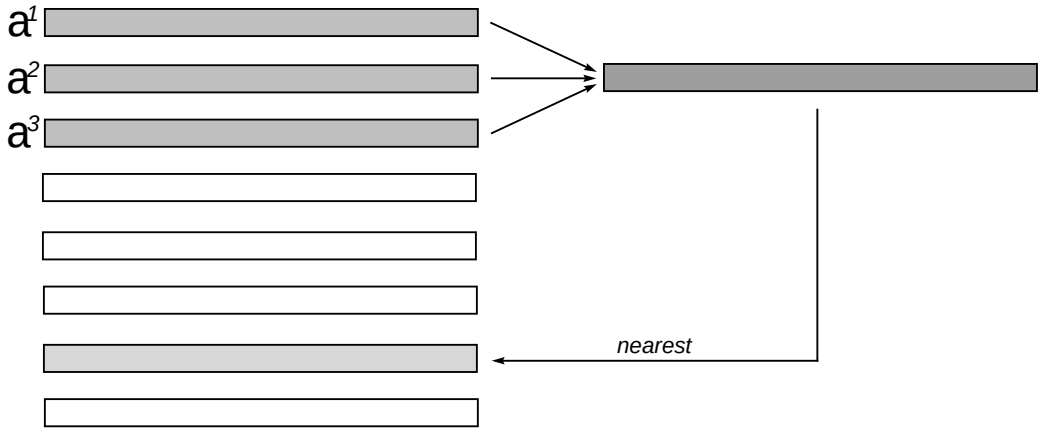

Figure 4.1: A successful collusion.

Definition 3 *A* p-majority *collusion strategy is one in which colluders choose with probability* p *the majority bit value in positions* $minor(a^i; a^j, a^k)$ *(that is, the bit values in* a^j *or* a^k*) (See Figure 4.2).*

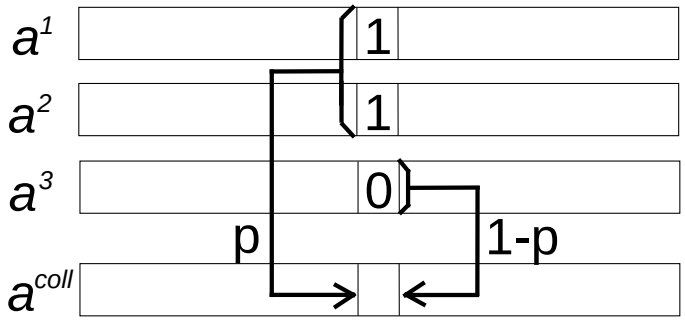

Figure 4.2: p-majority collusion strategy.

It can be seen that a word generated using a p-majority strategy from $a^1, a^2, a^3 \in DH(n)$ is expected to have the same distance to a^1, a^2 and a^3.

4.2.4 Distance from a collusion-generated word to colluders'codewords

Lemma 3 *Let a^{coll} be a word that has been generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_1 = d(a^{coll}, a^i) = K_1, \forall i = 1, 2, 3$ with*

$$p_1(k) = p(K_1 = k) = \sum_{t=\max(0,k-2^{n-1})}^{\min(k,2^{n-2})} b(t; 2^{n-2}, p) b(k-t; 2^{n-1}, 1-p)$$

where $b(x_1; x_2, x_3)$ is the binomial probability function (x_2 is the number of trials, x_3 the success probability per trial and x_1 es the number of successful trials).

Proof: Without loss of generality, take $i = 1$. We have that, for bit positions in $inv(a^1, a^2, a^3)$, there is no difference between a^1 and a^{coll} since bits in those positions are undetectable. Also, each of the 2^{n-2} bits in $minor(a^1; a^2, a^3)$ differs between a^1 and a^{coll} with probability p; therefore, the probability of there being t differing bits in those positions is given by a binomial probability function $b(t; 2^{n-2}, p)$. Also, each of the $2 \cdot 2^{n-2}$ bits in $minor(a^2; a^1, a^3)$ and $minor(a^3; a^1, a^2)$ differs between a^1 and a^{coll} with probability $(1-p)$; therefore, the probability of there being $k-t$ differing bits in those positions is given by a binomial probability function $b(k-t; 2^{n-1}, 1-p)$. In this way, the expression in the lemma corresponds to the probability of there being a total of $t + (k-t) = k$ differing bits between a^1 and a^{coll}. $\square$

Remarks: The total amount of differing bits is the addition of two binomially distributed random variables. We use this fact to compute its

expected value as

$$E(d_1) = p \cdot 2^{n-2} + (1-p)2^{n-1} = 2^{n-1} - p \cdot 2^{n-2}$$

As can be seen in Table 4.1, the expected number of differing bits between the word (a^{coll}) generated by collusion and any of the colluders' codewords ($a^1, a^2, a^3 \in DH(6)$) decreases as the value p gets closer to 1 (a^{coll} gets closer to a^1, a^2, a^3).

p	0	0.2	0.4	0.6	0.8	1
$E(d_1)$	32	28.8	25.6	22.4	19.2	16

Table 4.1: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and any of the colluders'codewords. The code is a $DH(6)$.

Lemma 4 *Let a^{coll} be a word generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_2 = \min_{i=1,2,3} d(a^{coll}, a^i) = K_2$ with*

$$p_2 = p(K_2 = k) = \sum_{i=1}^{3} \binom{3}{i} p_1(k)^i \Big[\sum_{k'>k} p_1(k') \Big]^{3-i}$$

Proof: The expression in the lemma corresponds to the probability of one, two or three codewords in $\{a^1, a^2, a^3\}$ being at distance k from a^{coll} and the remaining codewords being at a greater distance. □

Lemma 5 *Let a^{coll} be a word generated using a p-majority collusion strategy between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_3 =$*

$\max_{i=1,2,3} d(a^{coll}, a^i) = K_3$ *with*

$$p_3 = p(K_3 = k) = \sum_{i=1}^{3} \left(\begin{array}{c} 3 \\ i \end{array} \right) p_1(k)^i \Big[\sum_{k'<k} p_1(k') \Big]^{3-i}$$

Proof: The expression in the lemma corresponds to the probability of one, two or three codewords in $\{a^1, a^2, a^3\}$ being at distance k from a^{coll} and the remaining codewords being at a minor distance. □

Table 4.2 presents expected values of d_2 and d_3 for different values of p over a $DH(6)$ code.

p	0	0.2	0.4	0.6	0.8	1
$E(d_2)$	32	26.5	22.7	19.5	16.9	16
$E(d_3)$	32	31.12	28.46	25.27	21.54	16

Table 4.2: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and the nearest ($E(d_2)$) and the farthest ($E(d_3)$) among the colluders' codewords. The code is a $DH(6)$.

4.2.5 Distance from a collusion-generated word to codewords not in the collusion

Lemma 6 *Let a^{coll} be a word generated using a p-majority strategy between three codewords $a^1, a^2, a^3 \in DH(n)$ and let a^z be the only codeword in $DH(n) \backslash \{a^1, a^2, a^3\}$ with $a^z_i = a^1_i = a^2_i = a^3_i$, $\forall i \in inv(a^1, a^2, a^3)$ (existence and uniqueness of a^z are guaranteed by Lemma 2). Then, $d_4 = d(a^z, a^{coll}) = K_4$ with*

$$p_4(k) = p(K_4 = k) = b(k; 3 \cdot 2^{n-2}, p)$$

Proof: Lemma 2 says that bits of a^z are identical to bits of a^i in the positions in $minor(a^i; a^j, a^k)$ for $(i,j,k) \in \{(1,2,3),(2,1,3),(3,1,2)\}$. Therefore, the probability of there being k different bits in those $3 \cdot 2^{n-2}$ positions is given by a binomial probability function $b(k; 3 \cdot 2^{n-2}, p)$. $\square$

Remarks: The expected number of differing bits between a^z and a^{coll} is

$$E(d_4) = p \cdot 3 \cdot 2^{n-2}$$

Lemma 7 *Let a^{coll} be a word generated using a p-majority strategy between three codewords $a^1, a^2, a^3 \in DH(n)$ and let a^z be the only codeword in $DH(n) \backslash \{a^1, a^2, a^3\}$ with $a_i^z = a_i^1 = a_i^2 = a_i^3, \ \forall i \in inv(a^1, a^2, a^3)$. Then, for any codeword $a \in DH(n) \backslash \{a^1, a^2, a^3, a^z\}$ it holds that $d_5 = d(a, a^{coll}) = 2^{n-3} + K_5$ with*

$$p_5(k) = p(K_5 = k) = \sum_{t=\max(0,k-3\cdot 2^{n-3})}^{\min\{k,3\cdot 2^{n-3}\}} b(t; 3 \cdot 2^{n-3}, 1-p) b(k-t; 3 \cdot 2^{n-3}, p)$$

Proof: According to Lemma 2, a^{coll} and a have 2^{n-3} differing bits in positions in $inv(a^1, a^2, a^3)$. In each $minor(a^i; a^j, a^k)$, for $(i,j,k) \in \{(1,2,3),(2,1,3),(3,1,2)\}$, a^{coll} has all 2^{n-3} bits each of which is different with probability p and 2^{n-3} bits each of which is different with probability $(1-p)$. Therefore, we have $3 \cdot 2^{n-3}$ bits with probability p of being different, and thus the probability that t of such bits are different is $b(t; 3 \cdot 2^{n-3}, p)$. On the other hand, we have $3 \cdot 2^{n-3}$ bits with probability $1-p$ of being different, and thus the probability that $k-t$ of such bits are different is $b(k-t; 3 \cdot 2^{n-3}, 1-p)$. In this way, the expression in the lemma computes the probability of there being $t + (k-t) = k$ differing bits between a and

a^{coll}. $\square$

Remarks: The expected number of differing bits between a and a^{coll} is

$$E(d_5) = 2^{n-3} + 3 \cdot 2^{n-3}(1-p) + 3 \cdot 2^{n-3}p = 2^{n-1}$$

Table 4.3 presents expected values of d_4 and d_5 for different values of p over a $DH(6)$ code.

p	0	0.2	0.4	0.6	$0.\hat{6}$	0.8	1
$E(d_4)$	0	9.6	19.2	28.8	32	38.4	48
$E(d_5)$	32	32	32	32	32	32	32

Table 4.3: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy and a^z ($E(d_4)$) and the remaining codewords in $DH(6)\backslash\{a^1, a^2, a^3, a^z\}$ ($E(d_5)$).

For the sake of simplicity, let us assume in what follows that d_4 is distributed like d_5. Since for $p > 0.\hat{6}$ the number of differing bits expected for d_4 is greater than the number of different bits expected for d_5 ($E(d_4) > E(d_5) \Leftrightarrow p \cdot 3 \cdot 2^{n-2} > 2^{n-1} \Leftrightarrow p > 0.\hat{6}$), such a distributional assumption will cause actual security to be even slightly higher than computed in what follows.

Lemma 8 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. It holds that $d_6 = \min_{i \notin \{1,2,3\}}\{d(a^{coll}, a^i)\} = 2^{n-3} + K_6$, with*

$$p_6(k) = p(K_6 = k) = \sum_{i=1}^{2^n-3} \binom{2^n - 3}{i} p_5(k)^i \Big[\sum_{k'>k} p_5(k')\Big]^{2^n-3-i}$$

Proof: The expression in the lemma computes the probability that at least one out of the $2^n - 3$ codewords in $DH(n)\backslash\{a^1, a^2, a^3\}$ is at distance k of a^{coll}, with the remaining codewords at a longer distance. □

Table 4.4 presents expected values of d_6 for different values of p over a $DH(6)$ code.

p	$0.\hat{6}$	0.8	1
$E(d_6)$	24.5	25.6	32

Table 4.4: Expected number of differing bits between a word a^{coll} generated using a p-majority strategy ($p > 0.\hat{6}$) and the nearest of the codewords not in the collusion. The code is a $DH(6)$.

As it can be seen in Figure 4.3, when $p > 0.\hat{6}$, d_2 tends to take smaller values than d_6. This means that, with high probability, the codeword in $DH(n)$ nearest to the collusion generated word is a colluder codeword.

4.2.6 Identifying colluders'codewords

Lemma 9 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. The probability that the codeword in $DH(n)$ closest to a^{coll} is* not *in $\{a^1, a^2, a^3\}$ is expressed by*

$$\epsilon = \sum_{k=0}^{2^n-1} p(d_2 = k)p(d_6 \leq k)$$

ϵ is the probability that decoding a^{coll} yields as a result a codeword different from any of the colluders' codewords, that is, the probability of a honest buyer being unjustly accused instead of the colluders.

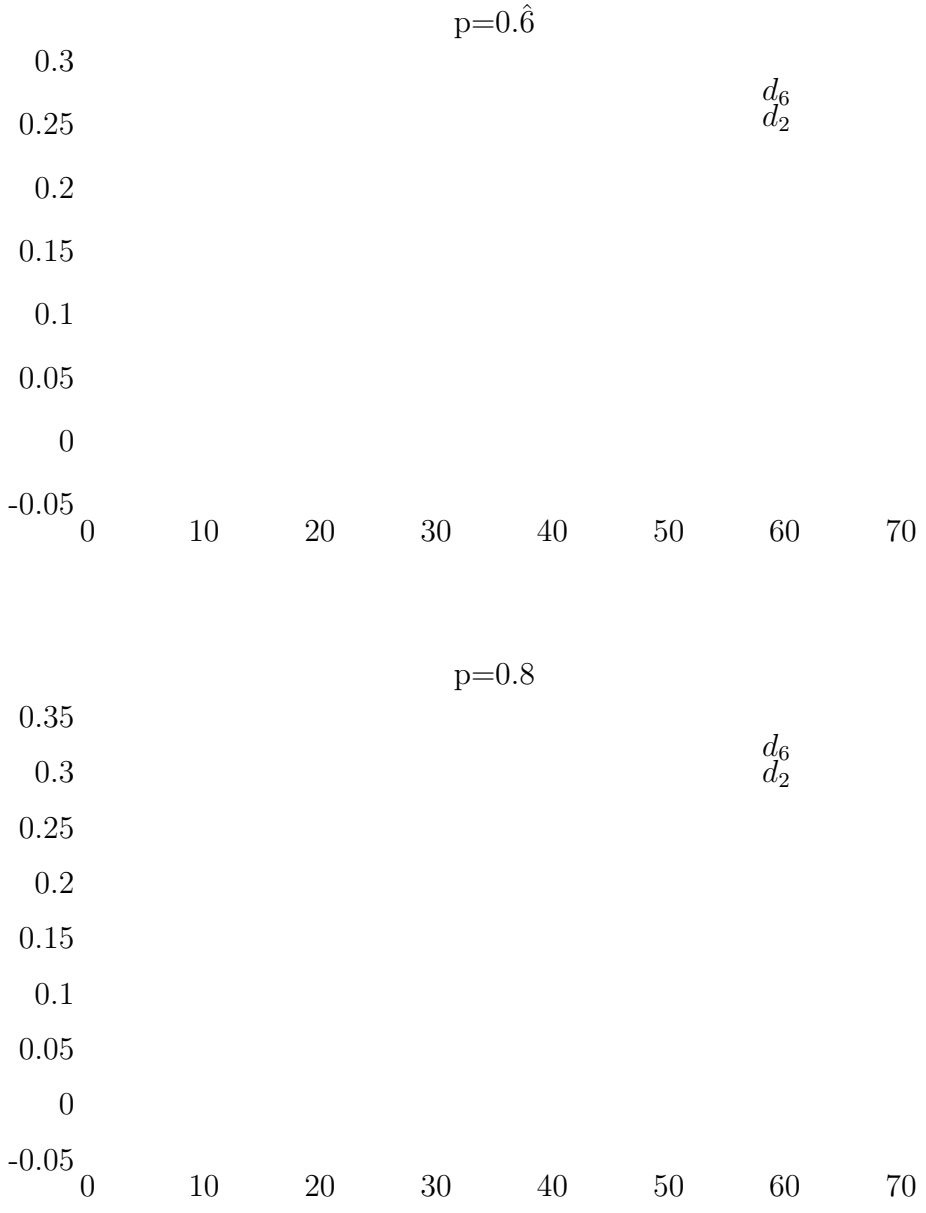

Figure 4.3: Distribution of d_2 and d_6 for $p = 0.\hat{6}$ and $p = 0.8$. The code is a $DH(6)$.

	p					
	0.0	$0.\hat{6}$	0.7	0.8	0.9	1.0
$DH(7)$	1.0	$0.59 \cdot 10^{-3}$	$0.14 \cdot 10^{-3}$	$0.14 \cdot 10^{-6}$	$0.77 \cdot 10^{-14}$	0.0
$DH(8)$	1.0	$0.17 \cdot 10^{-7}$	$0.10 \cdot 10^{-7}$	$0.15 \cdot 10^{-13}$	$0.70 \cdot 10^{-28}$	0.0

Table 4.5: Probability ϵ of success of a 3-collusion in $DH(7)$ and $DH(8)$ for several values of p

Remarks: It can be observed from Table 4.5 that, as n increases and p approaches 1, the probability ϵ of accusing an innocent buyer can be made arbitrarily close to 0.

Lemma 10 *Let a^{coll} be a word generated using a p-majority strategy ($p > 0.\hat{6}$) between three codewords $a^1, a^2, a^3 \in DH(n)$. The probability that the three closest codewords in $DH(n)$ to a^{coll} are $\{a^1, a^2, a^3\}$ is expressed by*

$$1 - \epsilon_2 = \sum_{k=0}^{2^n-1} p(d_3 = k)p(d_6 > k)$$

	p					
	0.0	$0.\hat{6}$	0.7	0.8	0.9	1.0
$DH(7)$	1.0	0.1	$0.5 \cdot 10^{-1}$	$0.1 \cdot 10^{-2}$	$0.25 \cdot 10^{-7}$	0.0
$DH(8)$	1.0	$0.14 \cdot 10^{-2}$	$0.26 \cdot 10^{-3}$	$0.6 \cdot 10^{-7}$	$0.2 \cdot 10^{-16}$	0.0

Table 4.6: Probability ϵ_2 of *not* identifying all three colluders in $DH(7)$ and $DH(8)$ for several values of p

Remarks: It can be observed from table 4.6 that as n increases and p approaches 1, the probability of not identifying all three colluders can be made arbitrarily close to 0.

The problem is that the parameter p defining the collusion strategy is chosen by the colluders, which implies they can take $p = 0$ to make sure they

are not identified!

In Section 4.3, a new kind of codes named *scattering codes* are presented. These codes are used in Section 4.4 to prevent colluders from avoiding identification in this way.

4.3 Scattering codes

In this section, we present a new kind of codes named *scattering codes*. Their construction, decoding and properties have been published in [SD02a].

4.3.1 Construction

A *scattering code* $SC(d,t)$ with parameters (d,t) is defined as a binary code consisting of $2t$ codewords of length $(2t+1)d$ constructed as follows:

1. The construction starts with generation of $SC(1,t)$:
   (a) The i-th codeword for $1 \leq i \leq t$ is constructed by setting the first and the $(i+1)$-th bits of the codeword to '1'. The remaining bits are set to '0'.
   (b) The i-th codeword for $t+1 \leq i \leq 2t$ is constructed by setting the $(i+1)$-th bit of the codeword to '1'. The remaining bits are set to '0'.
2. The code $SC(d,t)$ is generated by replicating d times every bit of $SC(1,t)$. Define a *block* to be a group of d replicated bits.
3. By convention, the first t codewords of $SC(d,t)$ are defined to encode a '1' and the last t codewords are defined to encode a '0'. The first block of the code is called 'Zone-A', the next t blocks are called 'Zone-B' and the last t blocks are called 'Zone-C'.

Example: The following are the codewords of a scattering code $SC(4, 3)$.

Encodes	Zone-A	Zone-B			Zone-C		
'1'	1111	1111	0000	0000	0000	0000	0000
	1111	0000	1111	0000	0000	0000	0000
	1111	0000	0000	1111	0000	0000	0000
'0'	0000	0000	0000	0000	1111	0000	0000
	0000	0000	0000	0000	0000	1111	0000
	0000	0000	0000	0000	0000	0000	1111

Using a scattering code, a '1' is encoded by randomly choosing one of the first t codewords and a '0' is encoded by randomly choosing one of the last t codewords.

4.3.2 Decoding

In a scattering code, a word is decoded by using the first applicable rule among the following ordered list:

1. If all bits in 'Zone-A' are '1' and all bits in 'Zone-C' are '0', decode as '1'.

2. If all bits in 'Zone-A' are '0' and all bits in 'Zone-B' are '0', decode as '0'.

3. If in two blocks of 'Zone-B' there is at least one bit in each with value '1', decode as '1'.

4. If in two blocks of 'Zone-C' there is at least one bit in each with value '1', decode as '0'.

5. If there are more '1' bits than '0' bits in 'Zone-A', decode as '1'.

6. If there are more '0' bits than '1' bits in 'Zone-A', decode as '0'.

7. Decode as 'Unreadable'

Note: It is easy to see that an odd value for d makes Rule 7 unreachable, making a '0' or '1' to be always returned.

4.3.3 Collusions over $SC(d,t)$

Lemma 11 *Let b^{coll} be a word generated using a p-majority strategy between three codewords $b^1, b^2, b^3 \in SC(d,t)$ encoding the same bit value v. Then, b^{coll} decodes as v with probability 1.*

Proof: It can be seen that, if v ='1', bits in 'Zone-A' and in 'Zone-C' stay undetectable and thus decoding will be through Rule 1 and return a value '1'.

If v ='0', bits in 'Zone-A' and in 'Zone-B' also stay undetectable. Thus, decoding will be through Rule 2 and return a value '0'. □

Lemma 12 *Let b^{coll} be a word generated using a p-majority strategy between three codewords $b^1, b^2, b^3 \in SC(d,t)$, with two of them ($b^1$ and b^2) encoding a value v and the other (b^3) the value $\overline{v}$. Then, the probability that b^{coll} decodes as v is given by*

$$p(v) = (1 - \frac{1}{t})p_{dif}(v) + \frac{1}{t}p_{coi}(v)$$

where $p_{dif}(v)$ is the probability of decoding as v when $b^1 \neq b^2$ and is computed as $p_{dif}(v) = 1 - p_{dif}(\overline{v})$ (we assume d to have an odd value) and

$$\begin{aligned} p_{dif}(\overline{v}) &= (1-p)^d p^{2d} + \\ &+ 2 \cdot p^d(1-p^d) \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p) + \\ &+ p^{2d} \sum_{k=1}^{\lfloor \frac{d-1}{2} \rfloor} b(k;d,p) \end{aligned}$$

and $p_{coi}(v)$ *is the probability of decoding as* v *when* $b^1 = b^2$ *and is computed as*

$$\begin{aligned} p_{coi}(v) &= p^{2d} + \\ &+ (1-p^d)\sum_{k=\lfloor \frac{d+2}{2} \rfloor}^{d} b(k;d,p) + \\ &+ p^d \sum_{k=\lfloor \frac{d+2}{2} \rfloor}^{d-1} b(k;d,p) \end{aligned}$$

Proof: In a collusion between three codewords $b^1, b^2, b^3 \in SC(d,t)$ with two of them (b^1 and b^2) encoding a value v (without loss of generality, assume $v = 1$ and $\overline{v} = 0$), we have $b^1 = b^2$ with probability $\frac{1}{t}$ and $b^1 \neq b^2$ with probability $1 - \frac{1}{t}$.

a) In the case $b^1 \neq b^2$, we compute $p_{dif}(v) = 1 - p_{dif}(\overline{v})$ (we assume d to have an odd value), where $p_{dif}(\overline{v})$ corresponds to the probability of decoding $\overline{v}$ after a collusion based on a p-majority strategy. $p_{dif}(\overline{v})$ is actually the probability of decoding using Rules 2 or 6.

   - Rule 2 will be applied if all bits in 'Zone-A' and 'Zone-B' are '0'. Since we are assuming a p-majority strategy, all bits in 'Zone-A' will be '0' with probability $b(d;d,1-p) = (1-p)^d$, because the majority bit in these positions is '1'. Since $b1 \neq b2$, there will be two detectable blocks in 'Zone-B' where the majority bit is '0'. Bits in 'Zone-B' will be all '0' with probability $b(2d;2d,p) = p^{2d}$. So, the probability of applying Rule 2 is $(1-p)^d p^{2d}$.
   - Since only one out of the three colluding codewords has value '0', it is not possible to have more than one block of 'Zone-C' with bit values different from '0'. So Rule 4 cannot be applied.
   - The next possibility for decoding as '0' is to apply Rule 6. This happens if there are more '0' bits than '1' bits in 'Zone-A' and no other rule between 1 and 5 has been applied before. In order to

render Rule 3 not applicable, we need one of the two detectable blocks of 'Zone-B' to be all zeros. Let us assume it is the leftmost one. This happens with probability $b(d; d, p) = p^d$.

Then we need more than half of the d bits of 'Zone-A' with value '0' (or less than one half with value '1'), which happens with probability $\sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k; d, p))$. We also need that one of the two detectable blocks of 'Zone-B' is all zeros (with probability p^d) and the other with at least one '1' bit to (which causes Rule 2 not to be applied), and happens with probability $1 - b(d; d, p) = 1 - p^d$. As this can happen twice, one with each of the blocks of 'Zone-B' forced to have all bits to '0', the total probability is $2 \cdot p^d(1 - p^d) \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} b(k; d, p)$.

The same rule is also executed if both blocks of 'Zone-B' have all bits to '0' (with probability p^{2d}) and the number of '1' bits in 'Zone-A' is greater than 0 (to make Rule 2 not applicable) and less than one half of the block length d. The total probability is $p^{2d} \sum_{k=1}^{\lfloor \frac{d-1}{2} \rfloor} b(k; d, p)$.

b) In the case $b^1 = b^2$, the probability of decoding value '1' corresponds to the probability of decoding after applying Rule 1 or Rule 5 (note that Rule 3 is not applicable).

   - Rule 1 will be applied if all bits of 'Zone-A' are '1' and all bits of 'Zone-C' are '0'. In both cases, we need all bits to take the majority value, which happens with probability $b(2d; 2d, p) = p^{2d}$.
   - The other possibility is to apply Rule 5 conditioned to not having applied Rule 1 before. There are two possible scenarios.

     In the first scenario, we need at least one bit of 'Zone-C' and more

than one half of the bits of 'Zone-A' with value '1'. This happens with probability $(1-p^d)\sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d} b(k;d,p)$.

In the other scenario, we need all bits of 'Zone-C' to be '0' and the number of ones in 'Zone-A' to be more than a half of the zone but less than d (otherwise Rule 1 would have been applied before).This happens with probability $p^d\sum_{k=\lfloor\frac{d+2}{2}\rfloor}^{d-1} b(k;d,p)$. □

Example: A possible collusion of three codewords of a $SC(4,3)$ code with $b^1 \neq b^2$ both encoding a '1' and b^3 encoding a '0'.

	Zone-A	Zone-B			Zone-C		
b^1	1111	1111	0000	0000	0000	0000	0000
b^2	1111	0000	1111	0000	0000	0000	0000
b^3	0000	0000	0000	0000	1111	0000	0000

Example: A possible collusion of three codewords of a $SC(4,3)$ code with $b^1 = b^2$ encoding a '1' and b^3 encoding a '0'.

	Zone-A	Zone-B			Zone-C		
b^1	1111	1111	0000	0000	0000	0000	0000
b^2	1111	1111	0000	0000	0000	0000	0000
b^3	0000	0000	0000	0000	1111	0000	0000

Figure 4.4 shows graphically the probability of decoding the majority value $p(v)$ as a function of the p-majority strategy applied over a $SC(d,t)$.

Table 4.7 present some numerical results on the lowest probability $p(v)$ of decoding the majority value v in a collusion of three buyers, for several scattering codes.

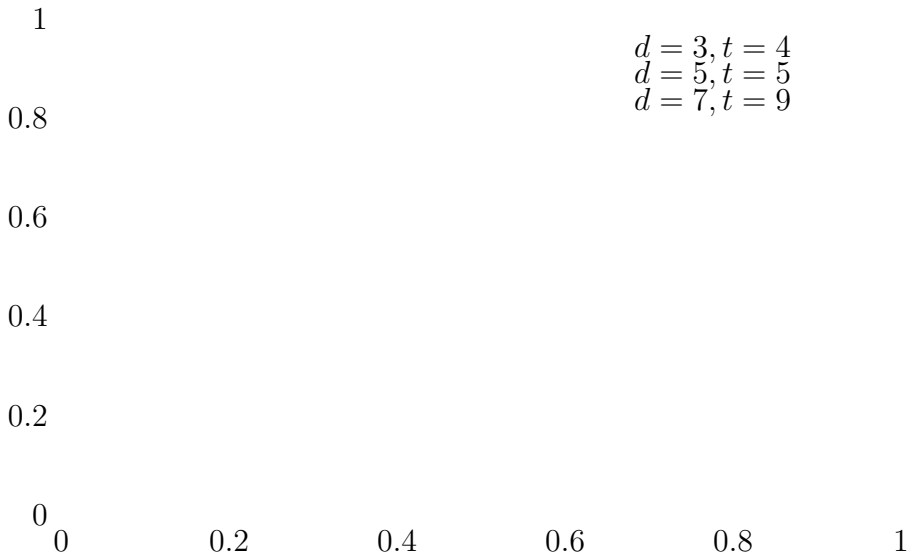

Figure 4.4: For different values of d and t, probability of decoding the majority value $p(v)$ as a function of the p-majority strategy applied over a $SC(d, t)$.

4.4 3-Secure codes

4.4.1 Construction

For $N = 2^n$ buyers, each buyer c^i is assigned a different codeword $a^i \in DH(n)$. Rather than directly embedding a^i in the content to be sold, the merchant generates a codeword A^i by composing a scattering code $SC(d, t)$ with a^i (See Figure 4.5). Such a composition is performed by replacing each bit of a^i with a codeword in $SC(d, t)$ which encodes the value of the bit of a^i. In this way, the codeword A^i will have bitlength

$$l = (N - 1)(2t + 1)d$$

d	t	$\min p(v)$
3	4	0.68
5	5	0.8
7	9	0.89
31	100	0.99

Table 4.7: Lowest probability $p(v)$ of decoding as the majority bit v in a collusion of three buyers, for several parameter choices (d, t).

The merchant then permutes the bits in A^i using a pseudo-random permutation seeded by a secret key known only to the merchant. The same permutation is applied to all codewords A^i. Figure 4.5 graphically depicts the construction described in this section. Finally, the merchant embeds the permuted version of A^i in the content being sold.

4.4.2 3-Collusions

Let us assume three dishonest buyers c^1, c^2, c^3 are assigned three codewords A^1, A^2, A^3 which have been built by:

1. Composing a scattering code with three different codewords $a^1, a^2, a^3 \in DH(n)$

2. Permuting the bits of the composed codewords

By comparison of their copies, the colluding dishonest buyers can identify $minor(A^1; A^2, A^3)$, $minor(A^2; A^1, A^3)$ and $minor(A^3; A^1, A^2)$. But as the bits of A^i have been secretly permuted, colluders cannot find out which bit of A^i corresponds to which bit of a^i. Thus, the colluders cannot identify $minor(a^1; a^2, a^3)$, $minor(a^2; a^1, a^3)$ nor $minor(a^3; a^1, a^2)$. Therefore, the only way for colluders to generate A^{coll} is to use a p-majority strategy.

According to Lema 9, all bits at positions $inv(a^1, a^2, a^3)$ remain unmodified after decoding each of the $2^n - 1$ components of A^{coll} to obtain a^{coll}. Also, according to Lema 10, all bits at positions $minor(a^i; a^j, a^k)$ for $(i, j, k) \in \{(1, 2, 3), (2, 1, 3), (3, 1, 2)\}$ will keep the majority value v (the one of a^j and a^k) with probability at least $p(v)$.

What is achieved with the above composition is that, regardless of the p-majority strategy used by colluders to generate words A^{coll}, the word a^{coll} resulting from decoding A^{coll} is a word generated by a $p(v)$-majority strategy collusion between a^1, a^2, a^3, where the value $p(v)$ is controlled by the merchant by choosing appropriate values for parameters d and t (see Table 4.7). It can be seen from Table 4.5 that controlling $p(v)$ is necessary to the keep low the probability ϵ of successful collusion. If A^i has some bits with value "unreadable", those bits are randomly set to '0' or '1'.

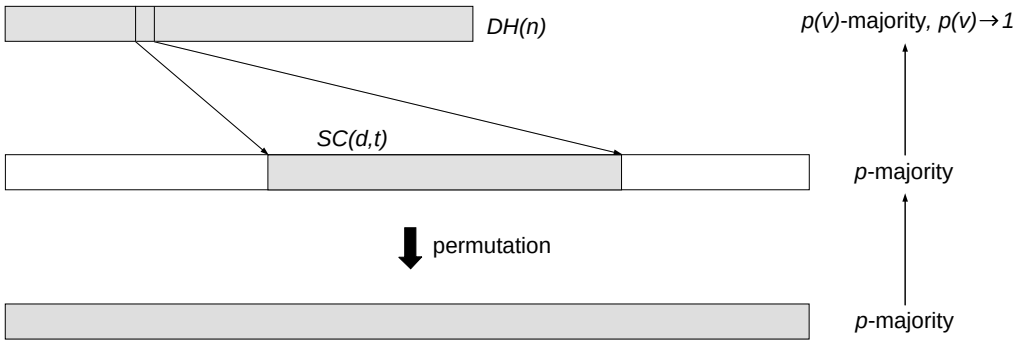

Figure 4.5: Construction of 3-secure codes.

4.4.3 Numerical results

Once parameters d and t have been fixed, the number of buyers can be increased by increasing n. For $d = 5$ and $t = 5$, Table 4.8 shows the size of the

code (number of buyers), the codeword length of our proposal, the probability of a successful collusion ϵ and the length of Boneh-Shaw's proposal for the same n and ϵ.

n	buyers	ϵ	Our length	Boneh-Shaw's length
7	128	$0.14 \cdot 10^{-6}$	6985	$2,788,320$
8	256	$0.15 \cdot 10^{-13}$	14025	$8,393,220$
9	512	$0.19 \cdot 10^{-27}$	28105	$28,340,928$

Table 4.8: Comparison between our codeword length and Boneh-Shaw's for the same number of buyers and security level (scattering code parameters: $d = 5$, $t = 5$)

It can be seen that Boneh-Shaw's construction results in much longer codewords than our proposal. Further, as n increases, their codeword length increases faster than ours.

In our proposal, once d and t have been fixed, the value ϵ decreases exponentially as n increases, which yields security levels higher than needed.

Thus, a better comparison is to use a fixed ϵ and assume that, for our security requirements, $\forall \epsilon' < \epsilon$ one has $\epsilon' \approx 0$. We take a value $\epsilon = 10^{-10}$ and use it as security level for Boneh-Shaw's construction. Results are presented in Table 4.9.

For a fixed $\epsilon = 10^{-10}$, we can observe that our proposal yields shorter codeword lengths up to $n = 16$ (number of buyers N is $65,536$). For values of $n > 16$ Boneh-Shaw's proposal offers a shorter codeword length. The explanation is that our codeword length increases as $O(N)$ while Boneh-Shaw's increases as $O(\log N)$ with a large constant factor; this large constant factor prevents Boneh-Shaw's scheme from comparing favorably unless N is very large.

buyers	Our length	Boneh-Shaw's length
512	$28,105$	$5,148,000$
$1,024$	$56,265$	$5,269,992$
$\ldots$	$\ldots$	$\ldots$
$32,768$	$1,802,185$	$5,883,888$
$65,536$	$3,604,425$	$6,006,780$
$131,072$	$7,208,905$	$6,129,816$

Table 4.9: Comparison of codeword length between our proposal and Boneh-Shaw's for the same number of buyers and assuming $\epsilon = 10^{-10}$ (scattering code parameters: $d = 5$, $t = 5$)

Chapter 5

Statistical Microdata Protection

In Section 2.3 we pointed out the need to protect statistical microdata when released to possibly dishonest users that may infer about individual respondents. In this chapter, we present our contributions to statistical disclosure control (SDC) for continuous microdata.

5.1 A modified score

In Subsections 2.3.2 and 2.3.3 above, measures to compute information loss and disclosure risk were shown. Those measures assume that the i-th masked record corresponds to the i-th original record.

Such one-to-one mapping cannot be assumed when the original and masked files have a different number of records or when masked records have been permuted. In this case, a new way to compute IL_1 must be defined. A natural way is to map each published masked record i to the nearest original record $c(i)$ using the d-dimensional Euclidean distance between standardized

records (where d is the number of variables in the data sets). Then a new IL'_1 is computed as the mean variation between masked records and the original records to which they are mapped. Denoting by n' the number of masked records, we have

$$IL'_1 = \frac{\sum_{j=1}^{d} \sum_{i=1}^{n'} \frac{|x_{c(i),j} - x'_{ij}|}{|x_{c(i),j}|}}{n'd}$$

where x_{ij}, x'_{ij} are the values taken by the j-th variable for the i-th record of the original and masked data sets, respectively.

Replacing IL_1 by IL'_1 leads to a modified information loss measure IL':

$$IL' = 100 \cdot \frac{(IL'_1 + IL_2 + IL_3 + IL_4 + IL_5)}{5}$$

Also, the lack of a one-to-one mapping between original and masked records forces a redefinition of disclosure risk measures DLD and PLD. As in the definition of IL'_1, we will say that a masked record is correctly linked to an original record if they are at the shortest possible d-dimensional Euclidean distance. Additionally, we redefine ID so that "corresponding values" mean values in records at shortest d-dimensional Euclidean distance. Distance is always measured over standardized records. Call DLD', PLD' and ID' the resulting redefined disclosure risk measures.

The new $Score'$ is computed by replacing IL, DLD and ID with IL', DLD' and ID' as well as dropping PLD for computational reasons. This yields:

$$Score' = 0.5 \cdot IL' + 0.25 \cdot DLD' + 0.25 \cdot ID' \tag{5.1}$$

This new $Score'$ was first used to evaluate the performance of a recently proposed method for synthetic microdata generation. We published our results in [DDS02].

5.2 Post-masking optimization

In the previous section, a measure $Score'$ has been proposed to measure how good is a masking method in terms of information loss and disclosure risk.

In this section, a post-masking optimization approach is presented which seeks to modify the masked data set to minimize information loss without increasing disclosure risk, leading to a better $Score'$ and thus to a better masking. Results presented here have been published in [SDMT02].

Once an original data set X has been masked as X', post-masking optimization aims at modifying X' into X'' so that the first and second-order moments of X are preserved as much as possible by X'' while keeping IL'_1 around a prescribed value. Near preservation of first and second-order moments results in (constrained) minimization of IL_2, IL_3, IL_4 and IL_5, which implies near preservation of multivariate statistics. Regarding IL'_1, a slight reduction is reasonable and desirable, whereas minimization is not; too small an IL'_1 would most likely result in a dramatic disclosure risk increase, because post-masking optimized data would look too much like the original data.

5.2.1 Mathematical background

Next, we explain the mathematical background on which our post-masking procedure is based.

Preserving averages

Let X_1 and X_2 be two data sets with d common variables and with n_1 and n_2 records, respectively. Then, it is easy to see that, if

$$\frac{\sum_{i=1}^{n_1} x_{1ij}}{n_1} = \frac{\sum_{i=1}^{n_2} x_{2ij}}{n_2}, \quad \forall j \in (1 \cdots d)$$

where x_{1ij}, x_{2ij} are the values taken by the j-th variable for the i-th record of X_1 and X_2, respectively, then first-order moments of X_2 match those of X_1 (thus causing IL_2 between X_1 and X_2 to be 0).

Preserving variances

Let X_1 and X_2 be two data sets with d common variables and with n_1 and n_2 records, respectively. Let x_{1ij}, x_{2ij} be the values taken by the j-th variable for the i-th record and $\overline{x}_{1j}$, $\overline{x}_{2j}$ be the averages of the j-th variables of X_1 and X_2 respectively. Preserving the variances of the j-th variables in both data sets can be written as

$$\frac{\sum_{i=1}^{n_1}(x_{1ij} - \overline{x}_{1j})^2}{n_1} = \frac{\sum_{i=1}^{n_2}(x_{2ij} - \overline{x}_{2j})^2}{n_2}, \quad \forall j \in (1 \cdots d)$$

The above is equivalent to

$$\frac{\sum_{i=1}^{n_1}(x_{1ij}^2 - 2x_{1ij}\overline{x}_{1j} + \overline{x}_{1j}^2)}{n_1} = \frac{\sum_{i=1}^{n_2}(x_{2ij}^2 - 2x_{2ij}\overline{x}_{2j} + \overline{x}_{2j}^2)}{n_2}$$

and

$$\frac{\sum_{i=1}^{n_1} x_{1ij}^2}{n_1} - 2\overline{x}_{1j}\frac{\sum_{i=1}^{n_1} x_{1ij}}{n_1} + \overline{x}_{1j}^2 = \frac{\sum_{i=1}^{n_2} x_{2ij}^2}{n_2} - 2\overline{x}_{2j}\frac{\sum_{i=1}^{n_2} x_{2ij}}{n_2} + \overline{x}_{2j}^2$$

Finally, the previous expression is equivalent to

$$\frac{\sum_{i=1}^{n_1} {x_1}_{ij}^2}{n_1} - {\overline{x}_1}_j^2 = \frac{\sum_{i=1}^{n_2} {x_2}_{ij}^2}{n_2} - {\overline{x}_2}_j^2$$

From the expression above, we can see that, if $\overline{x}_{1j} = \overline{x}_{2j}$ (first-order moments are preserved), and $\frac{\sum_{i=1}^{n_1} {x_1}_{ij}^2}{n_1} = \frac{\sum_{i=1}^{n_2} {x_2}_{ij}^2}{n_2}$, $\forall 1 \leq j \leq d$, then the variance of corresponding j-th variables of X_1 and X_2 will be the same (which will result in IL_4 between X_1 and X_2 being 0).

Preserving covariances

In a similar way, let X_1 and X_2 be two data sets (with n_1 and n_2 registers respectively and d variables). Preserving the covariance between any pair $1 \leq j < k \leq d$ of variables can be written as:

$$\frac{\sum_{i=1}^{n_1}(x_{1ij} - \overline{x}_{1j})(x_{1ik} - \overline{x}_{1k})}{n_1} = \frac{\sum_{i=1}^{n_2}(x_{2ij} - \overline{x}_{2j})(x_{2ik} - \overline{x}_{2k})}{n_2}, \quad \forall j,k \in (1 \cdots d),\ j < k$$

This is equivalent to

$$\frac{\sum_{i=1}^{n_1}(x_{1ij}x_{1ik} - x_{1ij}\overline{x}_{1k} - \overline{x}_{1j}x_{1ik} + \overline{x}_{1j}\overline{x}_{1k})}{n_1} = \frac{\sum_{i=1}^{n_2}(x_{2ij}x_{2ik} - x_{2ij}\overline{x}_{2k} - \overline{x}_{2j}x_{2ik} + \overline{x}_{2j}\overline{x}_{2k})}{n_2}$$

Finally, the above is equivalent to

$$\frac{\sum_{i=1}^{n_1} x_{1ij}x_{1ik}}{n_1} - \overline{x}_{1j}\overline{x}_{1k} = \frac{\sum_{i=1}^{n_2} x_{2ij}x_{2ik}}{n_2} - \overline{x}_{2j}\overline{x}_{2k}$$

From the expression above, we can see that if $\overline{x}_{1j} = \overline{x}_{2j}, \quad \forall 1 \leq j \leq d$ (first-order moments are preserved) and $\frac{\sum_{i=1}^{n_1} x_{1ij}x_{1ik}}{n_1} = \frac{\sum_{i=1}^{n_2} x_{2ij}x_{2ik}}{n_2}, \quad \forall 1 \leq j < k \leq d$, then the covariance between the j-th and k-th variables of X_1 will match with the corresponding one of X_2 (if variances are also preserved,

this causes IL_3 between X_1 and X_2 to take a value of 0).

Preserving correlations

From the definition of correlation, it is trivial to see that, when two data sets X_1 and X_2 preserve variances and covariances, correlations are also preserved (causing IL_5 between X_1 and X_2 to be 0).

5.2.2 The model

As it has been shown in the previous section, the first-order moments of a data set X depend on the sums

$$\frac{\sum_{i=1}^{n} x_{ij}}{n} \text{ for } j = 1, \cdots, d$$

where x_{ij} is the value taken by the j-th variable for the i-th record. The second-order moments of X depend on the sums

$$\frac{\sum_{i=1}^{n} x_{ij}^2}{n} \text{ for } j = 1, \cdots, d$$

$$\frac{\sum_{i=1}^{n} x_{ij}x_{ik}}{n} \text{ for } j, k = 1, \cdots, d \text{ and } j < k$$

Therefore, our goal is to modify X' (masked data set) to obtain a X'' (optimized masked data set) so that the above $2d + d(d-1)/2$ sums are nearly preserved between X (original data set) and X'', IL_1' is reduced to a desired value and disclosure risk stays similar in X' and X''. First, let us compute IL_1' of X' vs X as

$$IL_1' := \frac{\sum_{i=1}^{n'} \sum_{j=1}^{d} \frac{|x'_{ij} - x_{c(i),j}|}{|x_{c(i),j}|}}{dn'} \tag{5.2}$$

where $c(i)$ is the original record nearest to the i-th masked record of X' (d-dimensional Euclidean distance [1] is used). Now let $0 < q \leq 1$ be a parameter and let M be the set formed by the $100q\%$ records of X' contributing most to IL_1' above. Then let us compute the values x_{ij}'' of X'' as follows. For $x_{ij}' \notin M$ then $x_{ij}'' := x_{ij}'$. For $x_{ij}' \in M$, the corresponding x_{ij}'' are solutions of the following minimization problem:

$$\min_{\{x_{ij}''|x_{ij}'\in M\}} \sum_{j=1}^{d}\left(\frac{\sum_{i=1}^{n'} x_{ij}''}{n'} - \frac{\sum_{i=1}^{n} x_{ij}}{n}\right)^2 + \sum_{j=1}^{d}\left(\frac{\sum_{i=1}^{n'} x_{ij}''^2}{n'} - \frac{\sum_{i=1}^{n} x_{ij}^2}{n}\right)^2$$

$$+ \sum_{1\leq j<k\leq d}\left(\frac{\sum_{i=1}^{n'} x_{ij}'' x_{ik}''}{n'} - \frac{\sum_{i=1}^{n} x_{ij}x_{ik}}{n}\right)^2 \tag{5.3}$$

subject to

$$0.99 \cdot p \cdot IL_1' \leq \frac{\sum_{j=1}^{d}\sum_{i=1}^{n'} \frac{|x_{ij}''-x_{C(i),j}|}{|x_{C(i),j}|}}{dn'} \leq 1.01 \cdot p \cdot IL_1' \tag{5.4}$$

where $p > 0$ is a parameter and $C(i)$ is the original record nearest to the i-th masked record of X'' after optimization. Note that, in general, $C(i) \neq c(i)$, because in general $X'' \neq X'$.

5.2.3 A heuristic optimization procedure

To solve the minimization problem (5.3) subject to constraint (5.4), the following hill-climbing heuristic procedure has been devised:

[1] Distance is always measured over standardized variables.

Algorithm 13 (PostMaskOptim(X,X',p,q,TargetE))

1. *Standardize all variables in X and X' by using for both data sets the averages and standard deviations of variables in X.*
2. *Compute IL_1' between X and X' according to expression (5.2).*
3. *Let $TargetIL_1' := p \cdot IL_1'$.*
4. *Let $X'' := X'$.*
5. *Rank records in X'' according to their contribution to IL_1'. Let M be the subset of the $100q\%$ records in X'' contributing most to IL_1.*
6. *For each record i in X'', determine its nearest record $C(i)$ in X (use d-dimensional Euclidean distance).*
7. *Compute E, where E denotes the objective function in Expression (5.3).*
8. *While $E \geq TargetE$*
   (a) *Randomly select one value v of a record i_v in $M \subset X''$ and randomly perturb it to get v'. Replace v with v' in record i_v.*
   (b) *Recompute the nearest record $C(i_v)$ in X nearest to the updated i_v.*
   (c) *Let $PreviousIL_1' := IL_1'$.*
   (d) *Compute IL_1' between X and X''. To do this, use Expression (5.2) while replacing x_{ij}' by x_{ij}'' and $c(i)$ by $C(i)$.*
   (e) *Let $PreviousE := E$.*
   (f) *Recompute E (X'' has been modified).*

(g) *If* $E \geq previousE$ *then* $undo := true$.

(h) *If* $IL'_1 \notin [0.99 \cdot TargetIL'_1, 1.01 \cdot TargetIL'_1]$ *and* $|IL'_1 - TargetIL'_1| \geq |PreviousIL'_1 - TargetIL'_1|$ *then* $undo := true$.

(i) *If* $undo = true$ *then restore the original value* v *of record* i_v *and recompute the nearest record* $C(i_v)$ *in* X *nearest to* i_v.

9. *Destandardize all variables in* X *and* X'' *by using the same averages and standard deviations used in Step 1.*

Note that, by minimizing E, the algorithm above attempts to minimize the information loss IL'. No direct action is taken to reduce or control disclosure risk measures DLD' and ID', beyond forcing that IL'_1 should be in a pre-specified interval to prevent the optimized data set from being dangerously close to the original one. The performance of Algorithm 13 is evaluated *a posteriori*: once E reaches $TargetE$, the algorithm stops and yields an optimized data set for which IL', DLD' and ID' must be measured.

5.2.4 Computational results

The test microdata set no. 1 of [DDS02] was used. This microdata set was constructed using the Data Extraction System (DES) of the U.S. Census Bureau (`http://www.census.gov/DES`). $d = 13$ continuous variables were chosen and 1080 records were selected so that there were not many repeated values for any of the attributes (in principle, one would not expect repeated values for a continuous attribute, but there were repetitions in the data set).

In the comparison of [DMT01, DT01a], two masking methods were singled out as particularly well-performing to protect numerical microdata: rank swapping [Moo96] and multivariate microaggregation [DM02]. For both methods, the number of masked records is the same as the number of

p	q	$Score'$	IL'	DLD'	ID'	E
None	None	25.66	23.83	14.74	40.23	0.419
0.5	0.5	24.45	14.73	20.30	48.03	0.04
0.5	0.3	22.15	13.65	16.30	44.98	0.04
0.5	0.1	21.71	15.26	14.81	41.51	0.09

Table 5.1: Rank-swapping with parameter 14. First row, best $Score'$ without optimization; next rows, scores after optimization.

original records ($n = n' = 1080$). Several experiments have been conducted to demonstrate the usefulness of post-masking optimization to improve on the best (lowest) scores reached by rank swapping and multivariate microaggregation.

The first row of Table 5.1 shows the lowest $Score'$ reached by rank swapping for the test microdata set: the $Score'$ is 25.66 and is reached for parameter value 14 (see [DDS02]). The next rows of the table show $Score'$ reached when Algorithm 13 is used with several different values of parameters p (proportion between target IL'_1 and initial IL'_1) and q (proportion of records in M). The last column shows the value of the objective function E reached (for all rows but the first one, this is the $TargetE$ parameter of Algorithm 13). The $Score'$ is computed using Expression (5.1) and the values of IL', DLD' and ID' reached are also given in Table 5.1.

The first row of Table 5.2 shows the lowest $Score'$ reached by multivariate microaggregation for the test data set: the score is 31.86 and is reached for parameter values 4 and 10, that is, when four variables are microaggregated at a time and a minimal group size of 10 is considered (see [DDS02]). The next rows of the table show $Score'$ reached when Algorithm 13 is used with several different values of parameters p and q.

When looking at the results on rankswapped data (Table 5.1), we can

p	q	$Score'$	IL'	DLD'	ID'	E
None	None	31.86	22.48	22.14	60.34	0.122
0.5	0.5	26.96	14.16	21.06	58.54	0.008
0.5	0.3	27.39	14.74	21.29	58.80	0.008
0.5	0.1	28.03	14.94	21.83	60.38	0.008

Table 5.2: Multivariate microaggregation with parameters 4 and 10. First row, best $Score'$ without optimization; next rows, scores after optimization.

observe the following:

- There is substantial improvement of the $Score'$: 21.71 for post-masking optimization with $p = 0.5$ and $q = 0.1$ in front of 25.66 for the initial rankswapped data set.

- The lower q (*i.e.* the smaller the number of records altered by post-masking optimization), the better is $Score'$. In fact, $Score'$ for $q = 0.1$ is lower than for $q = 0.3, 0.5$ even if the target E for $q = 0.1$ is less stringent (higher) than for the other values of q.

- Post-masking optimization improves the score by reducing information loss IL' and hoping that disclosure risks DLD' and ID' will not grow. In fact, Table 5.1 shows that DLD' and ID' increase in the optimized data set with respect to the rankswapped initial data set. The lower q, the lower is the impact on the rankswapped initial data set, which results in a smaller increase in the disclosure risk. This small increase in disclosure risk is dominated by the decrease in information loss, hence the improved $Score'$.

The results on microaggregated data (Table 5.2) are somewhat different. The following comments are in order:

- Like for rankswapping, there is substantial improvement of $Score'$: 26.96 for post-masking optimization with $p = 0.5$ and $q = 0.5$ in front of 31.86 for the initial microaggregated data set.

- The higher q, the better is $Score'$. This can be explained by looking at the variation of IL', DLD' and ID'. Microaggregated data are such that there is room for decreasing IL' while keeping DLD' and ID' at the same level they had in the initial microaggregated data set. In this respect, we could interpret that, multivariate microaggregation being "less optimal" than rank swapping, we should not be afraid of changing a substantial number of values because this can still lead to improvement.

Chapter 6

Multilevel Access to Precision-Critical Data

When protecting data in the sense addressed in this thesis (transparent protection), the protection method introduces some amount of noise into the data. For statistical microdata, this noise corresponds to the distortion caused by a statistical disclosure control (SDC) method; for multimedia content, noise is caused by the modifications performed on the multimedia content in order to embed a watermark (or fingerprint).

Generally, the noise injected by transparent protection is quite tolerable. A masking method applied over a microdata file must not substantially alter the statistical properties of the file. In the same sense, if the mark embedding algorithm used for watermarking satisfies the property of imperceptibility, the marked multimedia content has the same commercial value as the original one. However, when dealing with precision-critical data, any small amount noise may render the data useless for some applications.

In this chapter we address this problem and propose a solution for *multilevel access to precision-critical data*. The goal of our solution is to

provide multilevel access to precision-critical data, in such a way that:

- Non-privileged users just see the protected data
- The higher the clearance of the user, the more protection she can remove:
  - In SDC protected microdata, such protection removal translates to partial removal of the distortion, and thus to data more similar to the original.
  - In watermarked multimedia contents, such protection removal allows the unwatermarked version of some parts of the content to be recovered.

6.1 Watermarking for multilevel access to statistical databases

A novel application of watermarking is presented in this section which allows multilevel access to numerical microdata: depending on her clearance, the data user can remove more or less of the masking. Non-privileged users just see the published data, but, as the clearance of a user increases, she can get a data set which is closer and closer to the original one. Results presented in this section have been published in [DMS01].

6.1.1 Partially removable masking

We assume that the information of a microdata file is represented as a two-dimensional table where one dimension corresponds to the set of objects (*i.e.* elements, individuals, persons) and the other is the set of attributes (*i.e.*

variables). The microdata file contains a value for each object-attribute pair, so that it can be modelled as a function

$$X : O \rightarrow D(X_1) \times D(X_2) \times \cdots \times D(X_d)$$

where O denotes the set of objects, $X_1, X_2, \cdots X_d$ denote the attributes and $D(X_i)$ refers to the domain of attribute X_i. Without loss of generality, the d-dimensional function X can be assumed to be of the form:

$$X(\cdot) = (X_1(\cdot), X_2(\cdot), \cdots, X_d(\cdot))$$

where $X_i(\cdot) : O \rightarrow D(X_i)$ is a one-dimensional function assigning a value for attribute X_i to a given object.

Assumption 1 We assume that a perturbative masking method can be expressed as a masking algorithm F which takes as inputs the original microdata file X and the outputs of r pseudorandom number generators $PRNG_i$ seeded by s_i, for $i = 1, \cdots, r$. The output of F is the masked microdata file X'. Formally speaking,

$$X' = F(X, \{s_1, \cdots, s_r\})$$

F and $PRNG_i$, for $i = 1, \cdots, r$ are assumed to be public, so the only secret parameters of masking are s_i, for $i = 1, \cdots, r$.

Assumption 2 We assume that each $PRNG_i$ is used to independently mask a part of the microdata file, so that *knowledge of the random numbers generated by $PRNG_i$ should allow to retrieve the original values from the corresponding masked values in that part of the microdata file.* Formally speaking, given a subset $S \subset \{s_1, \cdots, s_r\}$, we can compute

$$X'(S) = F^{-1}(X', S)$$

where $X'(S)$ is a microdata file resulting from removing the masking of X' that was produced using generators seeded by elements in S. In particular $X'(\{s_1, \cdots, s_r\}) = X$.

For some masking methods, it is easy to meet Assumptions 1 and 2, because they make explicit use of random number generation and knowledge of the generated random numbers suffices to undo the masking. Such is the case for additive noise.

For methods which do not directly meet both assumptions above, consider the sequence of differences between the masked and the original data

$$X'_i(o_j) - X_i(o_j) \text{ for } i = 1, \cdots, d \text{ and } j = 1, \cdots, n \tag{6.1}$$

where d is the number of attributes and n is the number of objects.

Now, the Berlekamp-Massey algorithm [Mas69] can be used to synthesize a Linear Feedback Shift Register (LFSR) generating the sequence (6.1). More generally, r LFSRs can be synthesized such that their interleaved outputs yield the sequence (6.1) (the i-th LFSR generates integers in positions j of the sequence such that $j \bmod r = i$). This construction reduces any perturbative method to a variant of additive noise (X' can be computed by adding the Sequence (6.1) to X), which thus meets Assumptions 1 and 2.

Now, if a user is revealed a subset S of the seeds, by Assumption 2 she can remove the masking in those parts of X' masked using generators seeded by values in S, so that the user obtains a partially unmasked file $X'(S)$. In particular, if the user is revealed all seeds, she can retrieve the original file

$X'(\{s_1, \cdots, s_r\}) = X$.

6.1.2 Watermarking solutions for multilevel access

From the previous section, we can see that, the larger the subset S of seeds known by a user, the more masking the user can remove, *i.e.* the closer is the unmasked file $X'(S)$ to the original X. This suggests the following algorithm to implement multilevel access to the masked file X':

Algorithm 14 *1. Let H be a clearance hierarchy comprising u user categories (for example, "statistician", "researcher", "civil servant", "other users"). For each category j, let k_j be a secret key known only to users in that category (the user does not actually need to know k_j, which can reside in her smart card).*

2. For $i = 1, \cdots, r$ and $j = 1, \cdots, u$, encrypt s_i with some redundancy R_i under k_j to get $E_{k_j}(s_i||R_i)$ if s_i should be revealed to user category j.

3. Use a watermarking algorithm to embed $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$, into the masked file X' to get a watermarked file X''.

From X'', a user can retrieve the subset S of seeds her category is entitled to know, and thus retrieve $X'(S)$. Redundancy R_i encrypted with s_i allows the user to check that s_i was correctly decrypted. We next discuss which features the watermarking algorithm should offer.

6.1.3 Watermarking requirements

Unlike in most usual watermarking applications (see [CMB00]), watermarking in the application described here is positive for the user. In the worst case, the user with no clearance just gets X'', but the user with some

clearance gets a better file. Therefore, there is no reason to expect a malicious behaviour by the user to destroy the watermark. Robustness should provide for those normal accidental alterations that may occur during the life cycle of a numerical file. These are basically rounding errors, mostly due to the software used to manipulate the data (*e.g.* when importing an ASCII version of X'' into a spreadsheet which rounds to two decimal positions). The rest of processing manipulations an image watermark should resist (see [PAK98]) do not make much sense on a microdata file, because nobody is really interested in a cropped, scaled or compressed version of X''.

The capacity of the watermarking scheme should be sufficient to allow embedding of $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$. It must be noticed that a numerical microdata file is usually smaller than multimedia files: just in one color 512×512 RGB image, we have 3×2^{18} pixel values, which is more than the number of values in a typical microdata file. Thus capacity should be medium to high in comparison to standard multimedia watermarking schemes.

Regarding obliviousness and imperceptibility, there is an interesting tradeoff in this multilevel access application:

- An oblivious watermarking scheme does not require X' to recover the watermark from X''. In principle, distribution of X' is thus unnecessary, which saves storage and communication. However, this means that the user will remove masking from X'' rather than from X'; so unless both files are very similar (*i.e.* the watermark is very imperceptible), unmasking X'' will not yield analytically valid results, because masking was performed on X'.

- A non-oblivious watermarking scheme assumes that X' is available when recovering the mark from X''. So there is no problem if X''

differs significantly from X', because the user will be able to perform the unmasking on X'. Thus, for a non-oblivious watermarking scheme, imperceptibility is not a requirement.

6.1.4 Choice of a watermarking algorithm

As noted in Section 2.3.1 when discussing SDC methods based on lossy compression, a numerical microdata file can be regarded as an image. Therefore, all image watermarking algorithms are potentially usable. However, from the requirements analysis of Section 6.1.3, we can conclude the following:

- Robust oblivious schemes like [HG96] cannot yield enough capacity while preserving a good level of imperceptibility. For example, for a microdata file with 1080 records and 13 variables, 10% distortions are needed to embed a 60-bit mark using [HG96]; it can be empirically seen that the amount of bits that can be embedded grows linearly with the percent distortion being used. In spite of 10% being already a distinctly perceptible distortion, a 60-bit mark can hardly accomodate a single encrypted seed (whereas embedding several encrypted seeds would be desirable).

- Robust non-oblivious schemes like [Her00, SDH00] offer a good level imperceptibility and good capacity, but require distributing X' along with X''.

- Oblivious methods, like Least Significant Bit (LSB) embedding, which are not robust enough for image watermarking, may be successfully adapted for the purposes of the application discussed here. Basically, the only manipulation LSB methods should survive in our case is

quantization (due to rounding errors): this can be achieved by embedding one bit in a *group* of least significant bits rather than in the least significant bit. LSB methods offer high imperceptibility and allow embedding one bit in each numerical value of the microdata file, so they offer high capacity as well.

6.2 Multilevel access to precision-critical images

A novel application of invertible watermarking is presented in this section which allows multilevel access to precision-critical images. Our goal is to devise a mechanism for user access to precision-critical images whereby the user can invert the embedded watermarks (and thus the distortion they cause) to an extent proportional to her clearance. Users with no clearance at all only see the watermarked image, which is copyright-protected and perceptually good but not suitable for high-precision processing. At the other end, users with full clearance can completely invert the watermarking process so as to obtain the original precision-critical image from the watermarked one. Between both extreme user types, users with intermediate clearances can invert the watermarking for some parts of the image. Results presented in this section have been published in [DS02b].

6.2.1 Mark embedding for multilevel access

Next we present an algorithm to implement multilevel access to precision-critical images.

Assumption 3 *We assume that the image to be protected can be divided*

into r semantically significant disjoint subimages. The i-th subimage is watermarked using an invertible method keyed with a different seed s_i. Formally speaking, if we denote the original image by X, the copyright sequence by M, the watermarked image by X' and the watermarking transformation by F, we have

$$X' = F(X, M, \{s_1, \cdots, s_r\})$$

Under the above assumption, knowledge of s_i allows the original i-th subimage to be retrieved from its watermarked version. More formally, given a subset $S \subset \{s_1, \cdots, s_r\}$, we can compute $X'(S) = F^{-1}(X', S)$, where $X'(S)$ is a partially unwatermarked image resulting from inverting the watermarks in the subimages of X' that were watermarked with seeds in S.

From the previous discussion, we can see that, the larger the subset S of seeds known by a user, the more watermarks the user can invert, *i.e.* the closer is the unwatermarked image $X'(S)$ to the original X. This suggests the following algorithm to implement multilevel access to the watermarked file X':

Algorithm 15

1. *Let CH be a clearance hierarchy comprising u user categories (for example, for medical images, we could think of "doctor", "nurse", "other users"). For each category j, let k_j be a secret key known only to users in that category (the user does not actually need to know k_j, which can reside in her smart card).*

2. *For $i = 1, \cdots, r$ and $j = 1, \cdots, u$, encrypt s_i with some redundancy R_i under k_j to get $E_{k_j}(s_i||R_i)$ if s_i should be revealed to user category j.*

Note that different seeds may be used for each image, whereas the key k_j corresponding to category j is assumed to stay stable.

3. *Assuming that the invertible watermarking algorithm used allows multiple marking without significant increase of the non-invertible distortion, embed $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$, into the watermarked file X' to get a rewatermarked file X''. A public seed is used for this second watermarking round.*

6.2.2 Partial mark removal

From X'', a user can recover and decrypt the subset S of seeds her category is entitled to know together with X', and thus retrieve $X'(S)$. Redundancy R_i encrypted with s_i allows the user to check that s_i was correctly recovered and decrypted.

As pointed out in Section 3.5, the Hartung-Girod method is an example of invertible watermarking which supports multiple marking. The only shortcoming of using n marking rounds (as required by Step 3 of Algorithm 15) is that α in the pre-processing algorithm (Algorithm 10) must be replaced with $n\alpha$. Thus, two marking rounds result in an increase of the non-invertible distortion introduced at the pre-processing stage. An alternative to avoid embedding $E_{k_j}(s_i||R_i)$ in the image is to keep those encrypted values in a freely accessible public repository.

Example: Figure 6.1 shows the original image "Chips" [1] and its division into 12 subimages. The Hartung-Girod method has been used to embed the same watermark in each subimage.

[1] `http://www.microscopy.fsu.edu/micro/gallery/chips/chipshots.html`

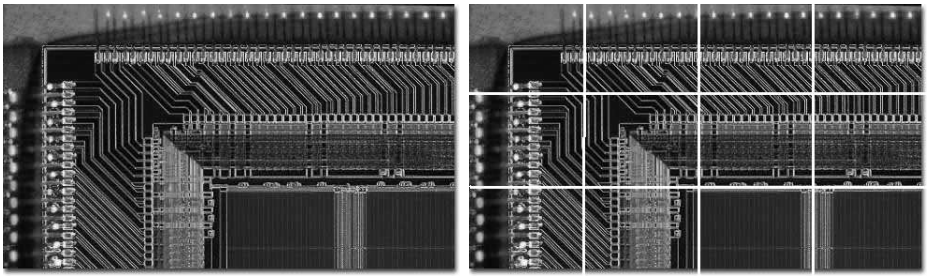

Figure 6.1: Left, original Chips image. Right, subimage division of Chips (12 tiles).

A partially unwatermarked version offers maximum precision and integrity verification to a user wishing to inspect the chip contact area depicted in the unwatermarked subimages; the remaining subimages showing the rest of the chip still carry watermarks whose copyright messages can be used to prove ownership in case of unlawful redistribution.

Chapter 7

Conclusions

7.1 Concluding remarks

In this thesis, we have covered different aspects of the field of protection of data that have to be made available to non completely trusted users. Data must be protected against unauthorized uses while preserving their utility. Therefore, protection must stay imperceptible, *i.e.* transparent.

The primary concern has been to offer a broad overview of current techniques for transparent data protection. Such techniques depend on the kind of data and the kind of risks we wish to protect data against. We have focused on transparent protection for two kinds of data: multimedia contents and statistical microdata.

Regarding multimedia contents, we first have studied the use of steganography for protecting data against unauthorized redistribution. More precisely, we have presented proposals corresponding to the two main subfields of copyright protection: watermarking and fingerprinting. A second achievement has been to develop steganographic techniques for providing lossless authentication of multimedia contents. In both cases, the multimedia

contents being considered consist of digital images.

Statistical disclosure control methods have been studied for microdata protection together with measures to compare performance of different methods in terms of information loss and disclosure risk.

7.2 Results of this thesis

Several results have been presented in this thesis.

In Chapter 3, our contributions to watermarking for digital images have been presented. First, a new visual components algorithm to guide image watermarking in achieving imperceptibility has been proposed. The algorithm provides information on the maximum alteration each pixel of an image can suffer without damaging the visual quality of the image. This algorithm can be plugged into watermarking algorithms that operate in the spatial domain and embed the information by directly incrementing/decrementing the color level of pixels.

Next, two new image watermarking schemes have been proposed. Both of them achieve imperceptibility using the aforementioned visual components algorithm. The first one is semi-public and offers high embedding capacity and robustness against compression, filtering and scaling attacks. The second one is oblivious and offers medium embedding capacity; its mark recovery algorithm does not require previous knowledge on the embedded watermark and offers robustness against compression, filtering, scaling and moderate geometric distortion attacks.

Mixture of watermarked digital objects has been presented as a way to increase robustness of current proposals. Prior mixture has been shown to be effective as a technique to combine the robustness properties of a set of image

watermarking algorithms. Posterior mixture has been proposed as a techique which aims at making the mark recoverable again when several attacked versions of the same content are found and the mark is not recoverable from any of them.

In the field of watermarking for image authentication, we have studied the invertibility of the well known spatial-domain spread-spectrum algorithm. Our study shows the suitability of this algorithm for lossless image authentication.

In the field of fingerprinting, Chapter 4 proposes a new construction for obtaining collusion-secure fingerprinting codes robust against collusions of up to three buyers. This construction provides, for a moderate number of possible buyers, shorter codewords than those offered by the general construction of [BS95] for $c = 3$.

In the protection of statistical microdata, a modification to a current performance metric to evaluate masking methods has been proposed in Chapter 5. The modified metric is more general in that it can deal with methods resulting in masked files whose number of records is not the same of the original file.

Also in statistical data protection, a procedure has been presented which enhances the performance of masking methods in terms of the previously mentioned metric. It is a post-masking optimization procedure which postprocesses masked files in order to decrease information loss while leaving disclosure risk practically unchanged. In this way, a better tradeoff between information loss and disclosure risk is obtained. This procedure has been shown to enhance the performance of the two best-performing masking methods: rankswapping and multivariate microaggregation.

The last chapter of this thesis has presented the novel concept of

multilevel access to precision-critical data. In this way, protected data are made available to different users, who, depending on their clearance, can remove part of the noise introduced by protection, thus obtaining better data quality. Two methods have been described to provide multilevel access to masked microdata files and watermarked digital images.

7.3 Future research

We sketch here some open problems that remain to be solved and possible extensions to some of the presented proposals that will be addressed in the future.

In the field of watermarking, our future research will be directed to:

- Achieving tamper-proofness in watermarking. That is, design watermarks that cannot be removed even if the intruder knows the particular watermarking algorithm used to embed them.
- Study the invertibility of other robust watermarking schemes for lossless image authentication.

Research on short binary fingerprinting codes robust against collusions of size greater than three should be done. Our future research in this field will be directed to the construction of codes robust against collusions of size greater than 3.

In statistical microdata protection, the presented post-masking optimization procedure can be extended in at least two directions:

- Study its applicability as a generator of *synthetic microdata*.
- Extend the procedure to preserve all moments up to m-th order.

Our Contributions

[DDS02] R.A. Dandekar, J. Domingo-Ferrer and F. Sebé, "LHS-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection", in *Inference Control in Statistical Databases*, LNCS 2316, Ed. J. Domingo-Ferrer, Berlin: Springer-Verlag, pp. 153-162, 2002.

[DMS01] Josep Domingo-Ferrer, Josep M. Mateo-Sanz and Francesc Sebé. "Watermarking for Multilevel Access to Statistical Databases", In *Proceedings of Information Technology: Coding and Computing - ITCC'2001*. Los Alamitos CA: IEEE Computer Society, pp. 243-247, 2001.

[DS02a] Josep Domingo-Ferrer and Francesc Sebé, "Enhancing watermark robustness through mixture of watermarked digital objects", In *Proceedings of Information Technology: Coding and Computing - ITCC'2002*. Los Alamitos CA: IEEE Computer Society, pp. 85-89, 2002.

[DS02b] Josep Domingo-Ferrer and Francesc Sebé, "Invertible spread-spectrum watermarking for image authentication and multilevel access to precision-critical watermarked images", In *Proceedings of Information Technology: Coding and Computing - ITCC'2002*. Los Alamitos CA: IEEE Computer Society, pp. 152-157, 2002.

[SD01] Francesc Sebé and Josep Domingo-Ferrer. "Oblivious image watermarking robust against scaling and geometric distortions", In *Information Security Conference - ISC'2001*, LNCS 2200. Berlin: Springer-Verlag, pp. 420-432, 2001.

[SD02a] F. Sebé and J. Domingo-Ferrer. "Scattering codes to implement short 3-secure fingerprinting for copyright protection", In *Electronics Letters*, vol. 38, no. 17, pp. 958-959, 2002.

[SD02b] Francesc Sebé and Josep Domingo-Ferrer. "Short 3-Secure Fingerprinting Codes for Copyright Protection", in *7th Australasian Conference on Information Security and Privacy - ACISP'2002*, LNCS 2384 . Berlin: Springer-Verlag, pp. 316-327, 2002.

[SDH00] F. Sebé, J. Domingo-Ferrer and J. Herrera. "Spatial-domain image watermarking robust against compression, filtering, cropping and scaling". In *Information Security Workshop - ISW'2000*, LNCS 1975. Berlin: Springer-Verlag, pp. 44-53, 2000.

[SDMT02] Francesc Sebé, Josep Domingo-Ferrer, Josep Maria Mateo-Sanz and Vicenç Torra. "Post-Masking Optimization of the Tradeoff between Information Loss and Disclosure Risk in Masked Microdata Sets", in *Inference Control in Statistical Databases*, LNCS 2316, Ed. J. Domingo-Ferrer, Berlin: Springer-Verlag, pp. 163-171, 2002.

Bibliography

[AM00]

F. Alturki and R. Mersereau, "Robust oblivious digital watermarking using image transform phase modulation", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.

[Bas] `http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/image_database.html`

[BP98] R. Barnett and D. E. Pearson, "Frequency mode L.R. attack operator for digitally watermarked images", *Electronics Letters*, vol. 34, pp. 1837-1839, Sep. 1998.

[Bra02] R. Brand, "Microdata Protection through Noise Addition", in *Inference Control in Statistical Databases*, LNCS 2316, Berlin: Springer-Verlag, Ed. J. Domingo-Ferrer, pp. 97-116, 2002.

[BS95] Boneh, D. and Shaw, J. "Collusion-secure fingerprinting for digital data", en *Advances in Cryptology-CRYPTO'95*, LNCS 963, Berlin: Springer-Verlag, pp. 452-465, 1995.

[Che00] P.-M. Chen, "A robust digital watermarking based on a statistical approach", in *International Conference on Information Technology:*

Coding and Computing - ITCC'2000, Los Alamitos CA: IEEE Computer Society, pp. 116-121, 2000.

[CKLS97] I.Cox, J.Kilian, T.Leighton and T.Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", in *IEEE Transactions on Image Processing*, vol.6, num.12, pp.1673-1687, 1997.

[CLMT00] M. Caramma, R. Lancini, F. Mapelli and S. Tubaro, "A blind & readable watermarking scheme for color images", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.

[CMB00] I. J. Cox, M. L. Miller and J. A. Bloom. Watermarking applications and their properties. In *International Conference on Information Technology: Coding and Computing - ITCC'2000*. Los Alamitos CA: IEEE Computer Society, pp. 6-10, 2000.

[Dig] Digimarc, `http://www.digimarc.com`

[DH00a] J. Domingo-Ferrer and J. Herrera. "Simple collusion-secure fingerprinting schemes for images". In *International Conference on Information Technology: Coding and Computing - ITCC'2000*. Los Alamitos CA: IEEE Computer Society, pp. 128-132, 2000.

[DH00b] J. Domingo-Ferrer and J. Herrera-Joancomartí, "Short collusion-secure fingerprints based on dual binary Hamming codes", *Electronics Letters*, vol. 36, no. 20, pp. 1697-1699, 2000.

[DH98] J. Domingo-Ferrer and J. Herrera. "Efficient smart-card based anonymous fingerprinting". In *Smart Card Research and Advanced Application - CARDIS'98*, LNCS 1820, Berlin: Springer-Verlag, pp. 231-238, 1998.

[DM02] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control", *IEEE Transactions on Knowledge and Data Engineering,* vol. 14, no. 1, pp. 189-201, 2002.

[DMT01] J. Domingo-Ferrer, J.M. Mateo-Sanz and V. Torra, "Comparing SDC methods for microdata on the basis of information loss and disclosure risk", *Proc. of ETK-NTTS'2001.* Luxembourg: Eurostat, pp. 807-825, 2001.

[DT01a] J. Domingo-Ferrer and V. Torra, "A quantitative comparison of disclosure control methods for microdata", in *Confidentiality, Disclosure and Data Access*, eds. P. Doyle, J. Lane, J. Theeuwes and L. Zayatz. Amsterdam: Elsevier, pp. 111-133, 2001.

[DT01b] Domingo-Ferrer and V.Torra, "Disclosure Control Methods and Information Loss for Microdata", in *Confidentiality, Disclosure and Data Access,* eds. P. Doyle, J. Lane, J. Theeuwes and L. Zayatz. Amsterdam: North-Holland, pp. 91-110, 2001.

[DVD] `http://www.lemuria.org/DeCSS`

[FGD01a] J. Fridrich, M. Goljan and R. Du, "Invertible authentication", in *Proc. SPIE Security and Watermarking of Multimedia Contents*, San José CA, Jan. 23-26, 2001.

[FGD01b] J. Fridrich, M. Goljan and R. Du, "Invertible authentication watermark for JPEG images", in *International Conference on Information Technology: Coding and Computing - ITCC'2001*, Los Alamitos CA: IEEE Computer Society, pp. 223-227, 2001.

[Fri02] J. Fridrich, "Security of Fragile Authentication Watermarks with Localization", in *Proc. SPIE Photonic West, Electronic Imaging 2002,*

Security and Watermarking of Multimedia Contents, San Jose, California, January, 2002.

[GFD01] M. Goljan, J. Fridrich and R. Du, "Distortion-free data embedding", in *4th Information Hiding Workshop*, Pittsburgh PA, April 2001.

[Her00] Jordi Herrera Joancomartí, *Secure Electronic Commerce Of Multimedia Contents Over Open Networks*, PhD. Thesis, UPC, 2000.

[HG96] F. Hartung and B. Girod. Digital watermarking of raw and compressed video. In *Proceedings of SPIE*, vol. 2952, pp. 205-213, 1996.

[HG98] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video", *Signal Processing*, 66(3): 283-301, May 1998.

[HJRS99] C. W. Honsinger, P. Jones, M. Rabbani and J. C. Stoffel, "Lossless recovery of an original image containing embedded data", US Patent Application, Docket No: 77102/E-D, 1999.

[HRPP+98] A. Herrigel, J. Ó Ruanaidh, H. Petersen, S. Pereira and T. Pun, "Secure copyright protection techniques for digital images", in *2nd Information Hiding Workshop*, Portland, Oregon, 1998.

[Jar89] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida", *Journal of the American Statistical Association*, vol. 84, pp. 414-420, 1989.

[KH97] D.Kundur and D.Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion", in *International Conference on Image Processing*, pp. 544-547, 1997.

[KP00] S. Katzenbeisser and Fabien A.P. Petitcolas. Information Hiding: techniques for steganography and digital watermarking. Artech House. ISBN 1-58053-035-4. 2000.

[KP99] M. Kutter and F.A.P. Petitcolas. "A fair benchmark for image watermarking systems", in *Proc. of SPIE*, Vol. 3657, January 1999.

[LM00] C.-S. Lu and H.-Y. Mark Liao, "Oblivious cocktail watermarking by sparse code shrinkage: a regional- and global-based scheme", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.

[LM01]
C.S. Lu and H.-Y. Mark Liao, "An oblivious and robust watermarking scheme using communications-with-side-information mechanism", in *International Conference on Information Technology: Coding and Computing - ITCC'2001*, Los Alamitos CA: IEEE Computer Society, pp. 103-107, 2001.

[LPZ99] H.J.Lee, J.H.Park and Y.Zheng, "Digital watermarking robust against JPEG compression", in *Information Security*, LNCS 1729. Berlin: Springer-Verlag, pp.167-177, 1999.

[Mas69] J. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, vol. IT-15, pp. 122-127, 1969.

[Moo96] R. Moore, "Controlled data swapping techniques for masking public use microdata sets", U. S. Bureau of the Census, 1996 (unpublished manuscript).

[NG96] Mark Nelson, Jean-Loup Gailly. The Data Compression Book. Ed. M&T Books. ISBN 1-55851-434-1. 1996.

[Nic88] D. Nicholson, *Spread Spectrum Signal Design - Low Probability of Exploitation and Anti-Jam Systems*, Computer Science Press, 1988.

[PA99] Fabien A. P. Petitcolas and Ross J. Anderson. "Evaluation of copyright marking systems". In *Proc. of IEEE Multimedia Systems'99*, vol. 1, Florence, Italy, pp. 574-579, June 1999.

[PAK98] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. "Attacks on copyright marking systems". In *2nd International Workshop on Information Hiding*, LNCS 1525. Berlin: Springer-Verlag, pp. 219-239, 1998.

[RA00] M. Ramkumar and A. N. Akansu, "A robust oblivious watermarking scheme", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.

[RP98] J.J.K. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking", in *Signal Processing*, 66(3), pp. 303-317, May 1998.

[RSA78] R.L.Rivest, A.Shamir and L.Adleman, "A method for obtaining digital signatures and public-key cryptosystems.", in *Communications of the ACM*, 21(2):120-126, February 1978.

[Sti] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, *StirMark v3.1* `http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/`

[Wag83] N. R. Wagner, "Fingerprinting", in *1983 IEEE Symposium on Security and Privacy*, Oakland CA: IEEE, pp. 18-22, 1983.

[Won98] P.W.Wong, "A watermark for image integrity and ownership verification", in *Proc IS&T PIC*, Portland, Oregon, 1998.

[WW01] L. Willenborg and T. de Waal, *Elements of Statistical Disclosure Control*. New York: Springer-Verlag, 2001.

Francesc Sebé Feixas

December 2002