Data-Hiding Codes

PIERRE MOULIN, FELLOW, IEEE, AND RALF KOETTER, MEMBER, IEEE

Invited Paper

This tutorial paper reviews the theory and design of codes for hiding or embedding information in signals such as images, video, audio, graphics, and text. Such codes have also been called watermarking codes; they can be used in a variety of applications, including copyright protection for digital media, content authentication, media forensics, data binding, and covert communications. Some of these applications imply the presence of an adversary attempting to disrupt the transmission of information to the receiver; other applications involve a noisy, generally unknown, communication channel. Our focus is on the mathematical models, fundamental principles, and code design techniques that are applicable to data hiding. The approach draws from basic concepts in information theory, coding theory, game theory, and signal processing, and is illustrated with applications to the problem of hiding data in images.

Keywords**—***Coding theory, data hiding, game theory, image processing, information theory, security, signal processing, watermarking.*

I. INTRODUCTION

For thousands of years, people have sought secure ways to communicate. Today secure communication is often identified with cryptography. However, some aspects of security are not at all addressed by cryptographic techniques. For instance, how can we conceal the very fact that we are communicating secretly? How can we guarantee that the information we are communicating will be decoded reliably by the intended receiver? What can the receiver learn about the communication channel?

The problems that form the subject of this paper consist of hiding data in a cover object, such as image, video, audio, or text. There are many applications, ranging from copyright protection to content authentication and to steganography, in which data-hiding methods play an important role. In fact new applications keep emerging, prompted by new societal needs, by the rapid development of information networks, and by the need for enhanced security mechanisms. For an overview of such applications, we refer the reader to the recent IEEE TRANSACTIONS ON SIGNAL PROCESSING supplements on secure media (October 2004, February 2005, and October 2005), the PROCEEDINGS OF THE IEEE special issue on digital rights management (June 2004), the *IEEE Signal Processing Magazine* (September and November 2003), and special issues of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (April 2003), the *IEEE Communications Magazine* (August 2001), *Signal Processing* (June 2001), and the *IEEE Signal Processing Magazine* (September 2000). The state of the art before 2000 is surveyed in the papers by Swanson *et al.* [1] and by Petitcolas [2]. The recent books by Barni and Bartolini [3], Cox, Miller, and Bloom [4], Eggers and Girod [5], Johnson, Duric, and Jajodia [6], and Katzenbeisser and Petitcolas [7] are also valuable resources.

The goal of this paper is to provide an overview of this field, focusing on the core principles and the mathematical methods that can be used for data hiding. We do not attempt to provide a comprehensive overview of the many techniques that have been developed (indeed, a whole book would be needed to cover research from the last ten years alone); instead we have tried to develop a systematic presentation of the fundamental ideas, emphasizing the connection with first principles from information theory, coding theory, game theory, and signal processing. Most of these ideas have been developed in the last seven years and presented in various research papers and short courses.

A. A Brief History

The *Histories* of Herodotus relate the following story which took place around 480 B.C. Histiaeus wanted to secretly notify the regent of the Greek city of Miletus to start a revolt against the Persian occupier. Histaeus chose an ingenious, albeit rather slow, secret communication method: shave the head of a slave, tattoo the message on his skull, allow the hair to grow back, and finally dispatch the slave to Miletus. There the slave was shaved again to reveal the secret message.

Manuscript received October 24, 2004; revised August 29, 2005. This work was supported by the National Science Foundation under Grants CCR 00-81268, CCR 03-25924, and CDA 96-24396.

The authors are with the University of Illinois, Urbana, IL 61801 USA (e-mail: moulin@ifp.uiuc.edu; koetter@comm.csl.uiuc.edu).

Digital Object Identifier 10.1109/JPROC.2005.859599

In this story, the physical communication medium is kept out of plain sight. One can also carry secret communication in plain sight, using a method developed in ancient China. The message sender and recipient share identical copies of a paper mask with holes cut out at random locations. The sender places the mask over a sheet of paper, writes the secret message through the holes, removes the mask, and fills in the blanks with an arbitrary composed message, giving the appearance of an innocuous text. This method was reinvented 500 years ago by the Italian mathematician Cardan and has become known as the Cardan grille.

A commercial application in which information is camouflaged in a visible physical medium is logarithmic tables in the 17th and 18th centuries. Errors were deliberately introduced in the least significant digits in order to assert intellectual property rights.

In both examples above, casual inspection of the message carrier fails to detect the presence of hidden information. Moreover, a secret code is used to embed the information: the location of the holes in the paper mask and the location of the numerical errors, respectively.

B. Modern Applications

The advent of the Internet and other public communication networks has given rise to a multitude of applications in which information hiding plays (or has the potential to play) an important role. Let us review some of these applications.

Copyright Protection [8]. This is arguably the most popular, yet also most controversial application of information hiding [9], [10]. The goal is to embed secret digital signatures in valuable digital documents such as text, audio, image, or video files. These digital signatures play the role of copyright notices which cannot be removed by an adversary without destroying (or severely damaging) the document itself. Copyright protection led to the emergence of digital watermarking[1] at the beginning of the Internet revolution, in the early 1990s.

Fingerprinting and Traitor Tracing [11]. This is analogous to the copyright protection problem, with a twist: the distribution list for the digital document is limited, and a distinct digital signature is embedded in each document, making it possible to trace back unauthorized use of a document to its original recipient. The 17th-century logarithmic tables are an example of this application; modern examples include distribution of digitized movies to theatres, distribution of audiovisual (A/V) material over restricted private networks, and distribution of sensitive company and government documents. Fingerprinting is considered to be a difficult problem due to possible *collusion* between users, making it easier for them to partially identify and degrade the fingerprints.

Content Authentication and Signature Verification (Forgery Detection). While standard cryptographic protocols may be used to authenticate message originators, authentication of A/V *content* (rather than the electronic file *per se*) presents unique challenges. For instance, the transmission medium may introduce errors, in which case conventional authentication protocols are inadequate. Applications include automatic video surveillance [12] and authentication of drivers' licenses [13].

Media Forensics. The goal here is to extract information about any processing that may have been applied to a signal [14]. For instance, authentication methods would reveal that an image has been tampered with, but not *how*. Forensic methods would take the analysis one step further, e.g., by indicating which part(s) of the image were modified, identifying new objects that may have been inserted into the image, etc.

Steganography. This ancient application is alive and well today. It may be used by people wishing to secretly communicate over public networks, including military and intelligence personnel, people living under oppressive governments, and terrorists [15].

The steganography application suggests that the party wishing to secretly communicate is sometimes the "good guy" and sometimes the "bad guy." The adversary trying to detect or prevent the secret communication can similarly be either the "bad guy" or the "good guy."

The problems listed above usually involve an intelligent adversary, whose objectives conflict with those of the sender. The nature of these objectives depends on the application. For steganography, the objective is undetectable communication: the presence of hidden data should be undetectable to the adversary. For watermarking, the objective is reliable transmission of a message or signature embedded in the host signal. The message itself need not be secret (e.g., a copyright notice), nor is the presence of an embedded message. For traitor tracing, the objective is to reliably extract the signature of a traitor from an intercepted document.

Other applications in which a message is embedded in a cover signal are nonadversarial in nature. A requirement is that the embedding survive common, nonmalicious signal degradations such as image compression and channel noise in the communication system. We list some of these applications because of the mathematical similarity between information embedding problems with and without an adversary. In particular, it is usually desired that the embedding be perceptually transparent (invisible or inaudible).

Database Annotation. Some large A/V databases contain various types of captions (e.g., text or speech). It is sometimes preferable to integrate the captions with the A/V file. This may be done using information-embedding algorithms, with the advantage that the embedded captions resist common signal processing manipulations.

Upgrade of Legacy Systems. It is sometimes possible to upgrade conventional signal transmission systems by embedding an "enhancement layer" into the transmitted data. Examples include digital audio broadcasting in the FM band [16] and embedding of stereo disparity maps into mono images [17].

Content Identification. Embedding scene/song identifiers in commercial TV and radio signals would enable applications such as automatic content monitoring and usage surveys (e.g., how many times was this commercial or

[1] So called by analogy with watermarks embedded in banknotes. Digital watermarks may also be visible, but in most applications they are required to be invisible.

this song played on radio station XYZ; how often was this political candidate shown on national TV.)

Device Control. Various synchronization and control signals may be embedded in radio and television signals. An example reported in [4] is the Dolby FM noise reduction technique, which was used by some commercial FM stations and required the use of an appropriate decoder. A signal embedded in the radio signal was used to trigger the receiver's Dolby decoder.

In-Band Captioning. Various types of data may be embedded in television and video programs: e.g., movie subtitles, financial information, and other data available for premium customers. Similarly, data for various services can be embedded in commercial radio signals.

Transaction Tracking. Video is usually produced, edited, distributed, and reedited multiple times. Embedding of a digital stamp makes it possible to retrace these steps, with optional security features.

C. *Basic Technical Issues*

Despite the bewildering variety of applications, each of them features a relatively small number of key attributes:

Transparency (Fidelity). In most applications, embedding of information should not cause perceptual degradation of the host signal. Embedded information should be invisible in images and text, and inaudible in speech and audio. For a given application there is a tolerable distortion level, generically denoted as D_1.

Payload. This refers to the number of information bits that are embedded in the host signal. This can vary from megabytes of information (for secret communication applications) to as little as a few bits (for copyright protection applications). For instance, DVD players have been proposed that verify the status of only four information bits before recognizing the file as legitimate and playing it. The payload is often normalized by the number of samples of the host signal, resulting in a bit rate R per sample of the host.

Robustness. This refers to the ability of the embedding algorithm to survive common signal processing operations such as compression, filtering, noise addition, desynchronization, cropping, insertions, mosaicing, and collage. The algorithm is commonly designed to survive a certain level of distortion, generically denoted as D_2.

Security. This refers to the ability of an adversary to crack the information-hiding code and design a devastating attack wiping out the hidden information, with little or no effect on perceptual quality. An example of ideal attack would be the recovery of the original host signal, which contains no trace of the message of interest.

Detectability. In most data-hiding applications, no secret is made of the fact that information is embedded in the host signal. In applications such as steganography, though, the very existence of secret communication must not be revealed. This introduces a constraint on the type of data-hiding algorithm that may be used. Detectability may be measured in a statistical sense, or in a computational-complexity sense.

The distinction between robustness and security is somewhat fuzzy. One may always think of cracking a code and applying the appropriate attack as an intelligent signal processing operation, and of standard signal processing operations as conventional attacks. If a code cannot be cracked, conventional attacks are the adversary's only option.

Security in the sense defined above is not the same as conventional cryptographic security, in which the primary goal is to make a message unreadable to unauthorized parties. It it worth noting that in a data-hiding problem, one may always encrypt the message before embedding to prevent unauthorized decryption. Message encryption may slightly increase the payload to be embedded but has otherwise no effect on transparency, robustness, or detectability.

There exist fundamental tradeoffs between transparency (D_1), payload or bit rate (R), robustness and security (D_2), and detectability. Much of the mathematical work on information hiding consists of analyzing these tradeoffs, identifying fundamental limits, and developing practical algorithms that approach those limits.

D. *System Issues*

In any application, selection of the data-hiding method depends on a number of practical considerations.

- Does the decoder have full, partial, or no knowledge of the host signal? The corresponding systems are sometimes called *public*, *semiprivate*, and *private* data-hiding systems. A slightly more commonly terminology, which is adopted in this paper, is *nonblind*, *semiblind*, and *blind*, respectively. Availability of side information about the host signal at the decoder generally improves detection or decoding performance but introduces a communication and storage burden.
- What kind of decision does the decoder have to make? For high-payload applications, the decision space of the decoder may be very large (return the message which the decoder believes to be embedded in the received signal). In other applications, such as signature verification, the decoder's task is simply to make a binary decision. The latter task is fundamentally much simpler than the former [18].
- Does the system rely on a cryptographic method, either public or private? It is widely believed that secure communication of hidden data requires the use of cryptographic methods. A private-key protocol requiring a prior key exchange between message sender and receiver is often impractical. Instead, a public-key protocol (say, RSA-based) may provide adequate security against an adversary with limited computational resources.
- What kind of communication protocol is desirable? Higher performance is expected if the decoder has access to side information about the host signal, is able to communicate with a central repository to acquire useful information, etc. However, such features increase storage and/or communication costs and introduce new potential failure modes.
- What security level is needed for the application at hand? In most applications this level is quite low, because the information being protected has relatively

low value. An example is protection of cable TV programs against pirates: a certain percentage of pirates are successful, but this does not put the cable companies out of business. In some applications the security level could be much higher, e.g., military grade.

- What detection/decoding accuracy is needed for the application at hand? This depends on the cost of making incorrect decisions. If the system is to be used in a court of law, false allegations of illegal behavior may be more damaging than letting the occasional cheat escape. In a video watermarking system, viewers would have little tolerance for devices that stop playing upon incorrect detection of a copyright violation. It has been suggested that probabilities of false alarms should be of the order of 10^{-9} or below for such applications. If the data-hiding channel is to be used as a regular communications channel (e.g., for upgrade of a legacy system), one may require probabilities of decoding errors of the order of 10^{-6}. In other applications, higher error probabilities may be acceptable.
- What are the attacker's computational resources? If the "attacker" is a government agency monitoring Internet traffic, real-time signal processing requirements preclude the systematic application of computationally complex detection tools; if the attacker is an amateur hacker trying to cheat the TV company, one may also expect a relatively low level of technical sophistication and computational resources.
- How easily can the system be reconfigured in the event of a major security failure? The worst case scenario is that of a hacker discovering secret keys and posting them on the Internet. Proposed solutions include the use of dynamic, signal-dependent keys as an alternative to the more conventional static keys.
- Does the attacker have access to multiple signals (data streams) produced by the same message sender? If so, this may requires frequent update of the keys used for data hiding, otherwise the attacker will eventually manage to learn these keys [19], [20].
- Does the attacker have repeated access to the decoder? This problem is analogous to the chosen-plaintext attacks in cryptography.

Regarding security of the data-hiding codes, two options are possible. The first one is *security through obscurity*: the algorithm used for data hiding is not publicly revealed. This option is regarded as theoretically unsafe because such secrets are hard to keep. Nevertheless this approach may be practically acceptable if: 1) the required security level is low; 2) it takes substantial time for adversary to discover which algorithm was used; 3) the marked data become less valuable as time goes by; and 4) the algorithm is changed relatively frequently. This approach has been used by the Disney Corporation to embed fingerprints in digitized movies [21].

The second option is the one favored by cryptographers and is based on *Kerckoffs' law* [22]: the algorithm is made public, but secret cryptographic keys are not. All established cryptographic methods, such as RSA, satisfy this condition. An important advantage of making the algorithm public is that the research community can test it and uncover potential flaws.

E. Benchmarking and Standards

So far there exists no foolproof watermarking, fingerprinting, or steganography algorithm. In our view this is due in good part to the lag between theory and practice: theory is still under development and, while specialized, practical codes have been developed based on the current theory, they have some weaknesses.

A few years ago, the music recording industry selected a particular watermarking code to protect digital music, and challenged the research community to break this code. (This became known as the SDMI[2] challenge). The SDMI approach was "security through obscurity." Sure enough, the SDMI code was broken shortly afterwards by a team from Princeton University [23] and a team from France [24]. There were plans to make provisions for watermarking as part of the international MPEG-4 video standard, but these plans did not materialize.

In order to rigorously test watermarking algorithms, several research groups have developed benchmarking tools. Programs such as *Stirmark* can be used to select an attack (or a cascade of attacks) from a comprehensive list and apply this attack to the marked data. Other benchmarking tools have been developed as part of the European *Certimark* program, which began in 1999 [25], and the WET project at Purdue University [26].

F. Basic Theoretical Concepts

Our brief overview of data hiding suggests this is a highly multidisciplinary field, pooling concepts and techniques from signal processing, cryptography, coding theory, detection and estimation theory, information theory, and computer science. An additional feature of these problems is that they involve parties with competing interests; for instance, the message sender and receiver do collaborate against the attacker. More complex applications such as fingerprinting may involve a team of attackers; one may also envision applications with a team of message senders and receivers. While it may be useful to think of such problems as involving attacks, countermeasures, and counterattacks on these countermeasures, a more fundamental and elegant framework for analyzing such problems and deriving appropriate strategies is *game theory* [27]. Randomized strategies for the message sender and attacker are obtained as the natural optimal solutions to a variety of data-hiding problems.

Mathematical analyses of data hiding are based on a number of simplifying assumptions. The goal is to more clearly understand the fundamental concepts and derive tangible mathematical results. The theory, which is relatively mature now, provides completely new insights and methods for data hiding. It also provides a precise framework for evaluating any data-hiding algorithm and can therefore be used to benchmark new algorithms. Finally, while different applications have different requirements, the fundamental

[2] Secure Digital Music Initiative.

principles uncovered by the theory cut across all these applications.

One of the most remarkable aspects of the theory is that very high communication performance can often be achieved if one views the host signal in which data are to be embedded as an interference that is *known to the encoder* and develops special codes that optimally adapt to this known interference. Indeed, there are clear connections between data hiding and information-theoretic problems of communication with side information at the encoder and/or decoder [28]–[30]. These connections have been identified independently by several researchers in 1999 [31]–[34] and subsequently developed in great detail. Other researchers, perhaps most notably Cox and Miller [4], [35], have contributed to bridging the gap between the theory and practice, which is still fairly large at the time of this writing.

G. *Outline of This Paper*

Our emphasis is on the fundamental aspects of data hiding. The detailed analysis relies on an arsenal of mathematical and statistical methods which is generally available only to a limited readership. We have therefore decided to organize the material in this paper to help readers with limited time or limited background in these specialized areas to easily access and digest the information of primary interest to them. Section II introduces a mathematical model for data hiding. Section III provides an overview of early data-hiding codes. Section IV introduces binning schemes, which play a central role in the design of good data-hiding codes. While binning schemes are fairly abstract information-theoretic constructions, we have emphasized the core ideas and illustrated them with several examples. Section V introduces quantization-based codes, which are good binning schemes and have been successfully used in recent years. Section VI requires a more advanced probability background and shows how one can analyze and design quantization codes which minimize decoding probability of error—or bounds thereon. Section VII requires some knowledge of information theory and derives the connections between quantization codes and some classical work in information theory. Section VIII complements the previous section by deriving results for practical systems such as those based on scalar quantization. In the following two sections, we address the forefront of current research: Section IX deals with the design of data-hiding codes that survive fairly complex attacks such as signal warping, and Section X deals with the design of codes that can resist cryptanalysis. Section XI outlines the application of basic principles to encompass problems of system-level attacks, steganography, authentication, fingerprinting, media forensics, and some theoretical issues. Application of data-hiding codes to images is illustrated in Section XII. The paper concludes with a discussion in Section XIII. Three short appendixes summarizing relevant notions of coding theory [36], vector quantization (VQ) [37], and detection theory [38] have been included.

H. *Notation*

We use uppercase letters to denote random variables, lowercase letters for their individual values, calligraphic fonts for sets, and boldface fonts for sequences, e.g., $\mathbf{x} = (x_1, x_2, \cdots, x_N)$. The length of the vector will be clear from the context. We denote by $p(x)$, $x \in \mathcal{X}$, the probability mass function (pmf) of a random variable X taking its values in the set $\mathcal{X}$; we use the same notation if $\mathcal{X}$ is a continuum, in which case $p(x)$ is referred to as the probability density function (pdf) of X. The symbol $\mathbb{E}$ denotes mathematical expectation. If X is a Gaussian random vector with mean μ and covariance matrix R, its pdf is denoted by $\mathcal{N}(\mu, R)$. Acronyms and specific notation for quantities frequently encountered in this paper are summarized below.

QIM	Quantizer-index modulation.
SSM	Spread-spectrum modulation.
STDM	Spread-transform dither modulation.
WNR	Watermark-to-noise ratio.
WHR	Watermark-to-host ratio.
GSNR	Generalized signal-to-noise ratio.
VQ	Vector quantizer.
DCT	Discrete cosine transform.
i.i.d.	Independent and identically distributed.
pdf	Probability density function.
pmf	Probability mass function.
$\mathbf{s}$	Host signal.
$\mathbf{x}$	Marked signal.
$\mathbf{w}$	Perturbation of $\mathbf{x}$ due to an attacker.
$\mathbf{y}$	Degraded (attacked) marked signal.
N	Length of host signal sequence.
m	Embedded message.
$\mathcal{M}$	Set of possible messages.
$\mathcal{Y}_m$	Decoding region for message m.
k	Cryptographic key.
$\mathbf{x} = f(\mathbf{s}, m, k)$	Encoding function.
$\hat{m} = g(\mathbf{y}, k)$	Decoding function.
$\mathbf{d}$	Dither sequence.
e	Self-noise.
Δ	Quantizer scale parameter.
Λ	Lattice.
G	Generator matrix for lattice and for linear code.
$Q(\cdot)$	Lattice quantization function.
J	Subsampling matrix.
L	Lattice dimension.
$\mathcal{C}$	Codebook.
α	QIM code scale parameter.

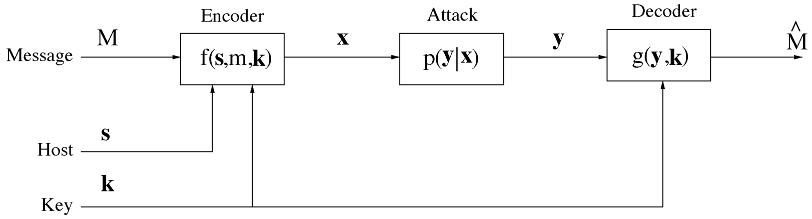


Fig. 1. Basic communication model for data hiding.

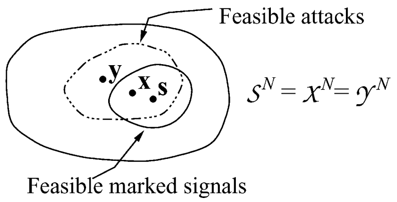


Fig. 2. Constraints on perceptual closeness of $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{y}$ can be used to define a class of admissible encoders and a class of admissible attacks.

D_1	Per-sample distortion due to embedding.
D_2	Per-sample distortion due to attacker.
C	Capacity.
P_e	Probability of error.
$B()$	Bhattacharyya distance between two pdf's.
n_P	Number of parallel channels.

II. Mathematical Models

In Section II–IX, we focus on a generic data-hiding problem in which a message m is to be communicated through the attack channel to a receiver. The basic communication model is depicted in Fig. 1.

A. *Encoders and Decoders*

The encoder has three inputs: the host sequence $\mathbf{s} \in \mathcal{S}^N$, the message $m \in \mathcal{M}$, and the key $k \in \mathcal{K}$ shared with the decoder. The encoder produces a marked sequence $\mathbf{x} \in \mathcal{X}^N$ using an *encoding function*

$$\mathbf{x} = f(\mathbf{s}, m, k). \tag{2.1}$$

Often k is a cryptographic key, independent of the host $\mathbf{s}$. In some applications, though, k is signal-dependent [19], [20], [39], [40]. In fact, (2.1) is general enough to include all nonblind and semiblind setups, in which k conveys information about $\mathbf{s}$ to the decoder. In general, k also provides a higher level of security against some system attacks, as mentioned in Section I-D. The sequences $\mathbf{s}$ and $\mathbf{x}$ should be *perceptually close* in a sense to be made precise. This relation is represented conceptually in Fig. 2.

The *payload* of the code is defined as the number $|\mathcal{M}|$ of messages that the encoder is designed to transmit. The payload could be just a few bits. In some applications, the payload is much larger, possibly exponentially large in the length N of the host sequence. A more convenient measure in this case is the *code rate*, which is expressed in number of bits per host signal sample

$$R = \frac{1}{N} \log_2 |\mathcal{M}|. \tag{2.2}$$

In the watermarking literature, R is occasionally referred to as the "capacity" of the encoder f. If one views data hiding as a communication problem, the above terminology is misleading because it can be confused with capacity in the Shannontheoretic sense. Shannon capacity is the maximum rate of reliable transmission *over all* f in a given class of encoders, *with respect to a given class of attacks*; this topic is covered in Section VII.

A decoder is a function $\hat{m} = g(\mathbf{y}, k)$ where $\mathbf{y} \in \mathcal{Y}^N$ is the received (attacked) signal, k is the key shared with the encoder, and $\hat{m} \in \mathcal{M}$ is the decoded message.

B. *Attacks*

The attacker takes the marked sequence $\mathbf{x}$ and creates a modified sequence $\mathbf{y}$ such that $\mathbf{y}$ is *perceptually close* to $\mathbf{x}$ and the *communication performance* between the encoder and decoder is reduced.

For the time being, we postpone the discussion of "perceptual closeness" and "communication performance" and address the problem of modeling attacks. Referring to Fig. 2, we could say that for each $\mathbf{x}$ there is an admissible set of degraded signals $\mathbf{y}$ that satisfy the perceptual closeness requirement. An intelligent attacker would select $\mathbf{y}$ according to some optimal strategy, i.e., minimizing communication performance.

Some typical (not necessarily optimal) choices are listed in Table 1. The choices include deterministic attacks (e.g., it is known that $\mathbf{x}$ will be subject to JPEG compression at a given quality factor) or more realistically, randomized attacks. By randomized we mean that the attacker selects one of several deterministic attacks with a certain probability distribution. One such strategy would be for the attacker to choose between a JPEG and a JPEG2000 compression attack, so both the encoder and the decoder are uncertain about the attacker's choice. Clearly this makes system design more complex, both from a theoretical and a practical standpoint.

In the example above, the attacker chooses between only two deterministic attacks, but the set of possible attacks is potentially vast. The attacker could select any $\mathbf{y}$ in the feasible set determined by the perceptual fidelity constraint, by

Table 1
Attacks

Attack Type	*Examples*
Memoryless	independent noise, random pixel replacement
Blockwise memoryless	JPEG compression
Attacks with "statistical regularity"	stationary noise, spatially invariant filtering, some estimation attacks
Deterministic	compression, format changes
Arbitrary attacks	cropping, permutations, desynchronization, nonstationary noise

randomly altering $\mathbf{x}$ according to an appropriate conditional probability distribution $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$. This will be termed an "arbitrary attack."

Some of the essential concepts and methods for data hiding are obtained if we restrict our attention to attacks with "statistical regularity." Loosely speaking, such attacks introduce a maximum amount of randomness; they include familiar operations such as addition of white noise or colored noise.

C. *Distortion*

To characterize perceptual closeness, it is convenient to introduce distortion functions. The distortion between two signals $\mathbf{s}$ and $\mathbf{x}$ is denoted by $d(\mathbf{s},\mathbf{x})$. A rudimentary but common choice is the squared Euclidean metric

$$d_E(\mathbf{s},\mathbf{x}) = \|\mathbf{s}-\mathbf{x}\|^2, \quad \text{if } \mathcal{S}=\mathcal{X}=\mathbb{R}.$$

Another choice is Hamming distance

$$d_H(\mathbf{s},\mathbf{x}) = |\{n : s_n \neq x_n\}|, \quad \text{if } \mathcal{S}=\mathcal{X}=\{0,1\}.$$

The first may be used to measure distortion between audio signals, between grayscale images, etc. The second is applicable to binary images, text files, and other binary data files. While tractable, such distortion measures fail to capture the complexities of human perception, including masking and threshold effects. The reader is referred to [41] for an excellent overview of this subject. Detailed perceptual models for images and speech have been constructed and refined over time. A popular example in image processing is Watson's metric [42], which is based on the concept of *just noticeable differences* and captures both threshold effects and spatial-frequency sensitivity of the human visual system. Psychovisual studies by Julesz [43] suggest that image textures with the same second-order statistics are perceived as identical by the human visual system. More accurate models have been developed later; given a natural texture one can extract a set of features and generate synthetic textures that look like the original one [44]. Advances in computer graphics have likewise made it possible to generate synthetic images that look like natural ones [45]. The relevance of this work to image watermarking, for instance, is that a sophisticated embedder or attacker could replace a textured portion of an image (say a grass field) with a similar-looking synthetic texture, introducing negligible perceptual degradation. A distortion function based on such texture perception models would take the form

$$d_F(\mathbf{s},\mathbf{x}) = \tilde{d}\,(F(\mathbf{s}), F(\mathbf{x}))$$

where $F(\cdot)$ is a feature mapping and $\tilde{d}(\cdot,\cdot)$ is a distance between features.

Most perceptual studies involve signals that are synchronized, e.g., they quantify the visibility of local image manipulations. To capture format changes and *desynchronization effects* such as temporal or spatial shifts, which have limited or no impact on perceptual quality, some modifications of classical distortion measures are needed. For instance, if a class of transformations T_θ parameterized by $\theta \in \Theta$ has no effect on signal quality, our distortion function should satisfy $d(\mathbf{s}, T_\theta \mathbf{s}) = 0$ for all $\theta \in \Theta$. An example of distortion function that satisfies this condition is [46]

$$d(\mathbf{s},\mathbf{x}) = \min_{\theta\in\Theta} \|T_\theta \mathbf{s} - \mathbf{x}\|. \tag{2.3}$$

Examples of transformations T_θ include the following.

- Amplitude scaling: $(T_\theta \mathbf{s})_n = \theta s_n$, where $\theta_{\min} \leq \theta \leq \theta_{\max}$.
- Temporal shifts: $(T_\theta \mathbf{s})_n = s_{n-\theta}$. If θ is not an integer, $s_{n-\theta}$ denotes a resampled version of the shifted, interpolated signal $\mathbf{s}$.

Based on the psychovisual studies by Julesz and others [43], [44], a meaningful distortion metric between image textures $\mathbf{s}$ and $\mathbf{x}$ with statistics $\Sigma_\mathbf{S}$ and $\Sigma_\mathbf{X}$ would be a function of $\Sigma_\mathbf{S}$ and $\Sigma_\mathbf{X}$ only.

Having defined a distortion function, we can precisely define a set of feasible data-hiding codes and a set of feasible attacks, each satisfying a distortion constraint. The distortion constraint may be "hard" or "soft," as discussed below.

A hard distortion constraint for the data-hiding code is the *maximum-distortion* constraint

$$d\,(\mathbf{s}, f(\mathbf{s},m,k)) \leq D_1, \quad \forall \mathbf{s}, m, k. \tag{2.4}$$

A softer constraint is the *average-distortion* constraint

$$\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\sum_{\mathbf{s}\in\mathcal{S}^N} p_\mathbf{S}(\mathbf{s}) d\,(\mathbf{s}, f(\mathbf{s},m,k)) \leq D_1. \tag{2.5}$$

where the averaging is over k, m, and $\mathbf{s}$. Here $p_\mathbf{S}(\mathbf{s})$ is some averaging measure on $\mathbf{s}$, e.g., a probability distribution on $\mathbf{s}$. There exist fairly good statistical models for host signals such as images, speech, etc. that can be used to select an appropriate $p_\mathbf{S}$ [47], [48].

The distortion introduced by the attacker can be measured in terms of $d(\mathbf{x}, \mathbf{y})$ or in terms of $d(\mathbf{s}, \mathbf{y})$. A natural requirement for the attacker is that this distortion, measured in a suitable average or maximum sense, does not exceed some level D_2. If the distortion is measured with respect to $\mathbf{x}$, we have

$$d(\mathbf{x}, \mathbf{y}) \leq D_2, \quad \forall \mathbf{x}, \mathbf{y} \tag{2.6}$$

and

$$\sum_{\mathbf{x} \in \mathcal{X}^N} p_{\mathbf{X}}(\mathbf{x}) d(\mathbf{x}, \mathbf{y}) p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) \leq D_2 \tag{2.7}$$

respectively. The averaging measure on $\mathbf{X}$ is given by

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \sum_{\mathbf{s} \in \mathcal{S}^N} p_{\mathbf{S}}(\mathbf{s}) 1_{\{\mathbf{x} = f(\mathbf{s}, m, k)\}}.$$

The use of averaging measures also makes it possible to define the average distortion with respect to the host

$$\frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \sum_{\mathbf{s} \in \mathcal{S}^N} p_{\mathbf{S}}(\mathbf{s}) d(\mathbf{s}, \mathbf{y}) p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|f(\mathbf{s}, m, k)) \leq D_2. \tag{2.8}$$

It is useful to keep some simple quantities in mind when designing a data-hiding system. Assume $\mathcal{S} = \mathcal{X} = \mathcal{Y} = \mathbb{R}$ and the distortion function is the squared-error metric. One can define the WNR as

$$\mathrm{WNR} = \frac{D_1}{D_2} \tag{2.9}$$

and the WHR as

$$\mathrm{WHR} = \frac{D_1}{||\mathbf{s}||^2} \tag{2.10}$$

(sometimes referred to as watermark-to-document ratio in the literature). An alternative definition with $\mathbb{E}||\mathbf{s}||^2$ in place of $||\mathbf{s}||^2$ in the denominator is sometimes used.

III. Early Work

The first papers on data hiding appeared in the early 1990s [49]. The ideas proposed during that period include least significant bit (LSB) embedding techniques, which are elementary and nonrobust against noise. They are however closely related to more advanced binning techniques. The period 1995–1998 saw the development of SSM codes, which are more robust [50] and have been used in several commercial products; see [1] and references therein. Both SSM and LSB methods are reviewed next. We also comment briefly on system performance methods that were often used in the 1990s.

A. Spread-Spectrum Codes

The watermarking problem is analogous to a communication problem with a jammer. This has motivated many researchers to apply techniques from this branch of the communications literature—especially SSM techniques, which have been successfully used against jammers. We first briefly review these techniques and then show how they can be applied to watermarking and data hiding.

The jamming problem. In a standard radio or TV communication system, the transmitter sends a signal in a relatively narrow frequency band. This technique would be inappropriate in a communication problem with a jammer, because the jammer would allocate all his power to that particular band of frequencies. An SSM system therefore allocates secret sequences (with a broad frequency spectrum) to the transmitter, which sends data by modulating these sequences. The receiver demodulates the data using a filter matched to the secret sequences. Essentially, the transmitter is communicating information over a secret low-dimensional subspace; only noise components in that subspace may affect communication performance. The jammer must spread his power over a broad frequency range, but only a small fraction of that power will have an effect on communication performance.

The application of SSM to data hiding is illustrated in Fig. 3. Associated with each message m and secret key k is a pattern $\mathbf{p}^{(m,k)}$ which is "mixed" with the host $\mathbf{s}$ to form the marked signal $\mathbf{x}$. Each pattern is typically a pseudorandom noise (PRN) sequence. The mixing could be as simple as a weighted addition

$$x_n = s_n + \gamma p_n^{(m,k)}, \quad 1 \leq n \leq N \tag{3.1}$$

where γ is a strength parameter, which depends on the embedding distortion allowed. The mean-square embedding distortion is $\gamma^2 ||\mathbf{p}^{(m,k)}||^2$ and is usually the same for all m and k. The marked signal $\mathbf{x}$ is possibly corrupted by the attacker's noise, which produces a degraded signal

$$\mathbf{y} = \mathbf{x} + \mathbf{w}. \tag{3.2}$$

The receiver knows the secret key k and can match y with the $|\mathcal{M}|$ possible waveforms $\mathbf{p}^{(m,k)}$. If the host is not available to the receiver, the matching could be a simple correlation

$$\hat{m} = \operatorname*{argmax}_{m \in \mathcal{M}} t_m(\mathbf{y}, k) \tag{3.3}$$

where

$$t_m(\mathbf{y}, k) = \sum_{n=1}^{N} y_n p_n^{(m,k)}, \quad m \in \mathcal{M} \tag{3.4}$$

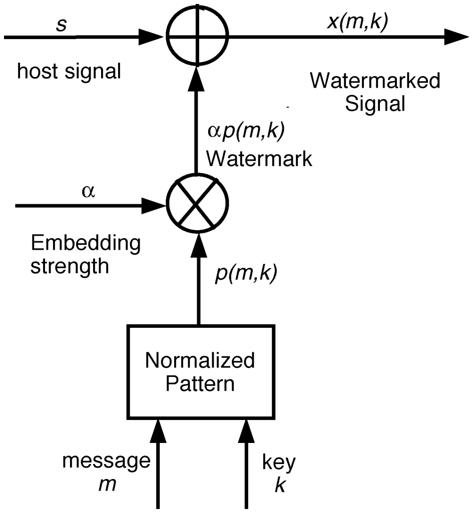


Fig. 3. SSM.

are the correlation statistics. If the host is available to the receiver, performance can be improved (see discussion at the end of this section) by subtracting the host from the data before correlating with the watermark patterns

$$t_m(\mathbf{y}, \mathbf{s}, k) = \sum_{n=1}^{N} (y_n - s_n) p_n^{(m,k)}, \quad m \in \mathcal{M}.$$

For the blind data-hiding case, due to (3.1) and (3.2), we can write the received data as the sum of the watermark $\gamma \mathbf{p}^{(m,k)}$ and total noise $\mathbf{s} + \mathbf{w}$

$$\mathbf{y} = \gamma \mathbf{p}^{(m,k)} + (\mathbf{s} + \mathbf{w}). \tag{3.5}$$

Typically the host signal $\mathbf{s}$ has high energy relative to the embedding and attack distortions. As we shall see in Section VI, the performance of the decoder is limited by the high total noise level. For nonblind data hiding, the decoder knows $\mathbf{s}$, so the noise at the decoder is just $\mathbf{w}$.

Several important refinements of the basic system of Fig. 3 have been developed over the years.

1) The embedding strength parameter γ can be locally adapted to host signal characteristics, e.g., (3.1) can be replaced with

$$x_n = s_n + \gamma_n(\mathbf{s}) p_n^{(m,k)} \tag{3.6}$$

where γ_n depends on the local characteristics of the host (e.g., frequency and temporal characteristics) [4], [51].

2) To reduce decoder's noise in (3.5), one can *preprocess* the host $\mathbf{s}$ prior to embedding the watermark [4, Sec. 5.1]. This can be done using a *causal* preprocessor, leveraging information-theoretic results by Shannon on the capacity of communication systems with side information available causally to the encoder [31]. In data hiding, however, the encoder need not be restricted to causal strategies. Good results have been obtained using linear preprocessing [52], [53]. The embedding rule in [52] is of the form

$$\mathbf{x} = \Gamma \mathbf{s} + \gamma \mathbf{p}^{(m,k)} \tag{3.7}$$

where Γ is a matrix that depends on the second-order statistics of $\mathbf{S}$ and can be optimized against worst case filtering and colored noise attacks. The embedding rule in [53] is of the form

$$\mathbf{x} = \mathbf{s} + \gamma(\mathbf{s}) \mathbf{p}^{(m,k)} \tag{3.8}$$

where $\gamma(\mathbf{s})$ is an optimized linear function of $\mathbf{s}$. The contribution of $\mathbf{s}$ in the decoder's noise can be greatly reduced (and even eliminated) if the attacker adds signal-independent noise and the code rate is very low [53], [54].

3) The basic correlator decoder (3.3) is generally not well matched to noise statistics.[3] For colored Gaussian noise, a weighted correlation statistic is ideal. With non-Gaussian noise such as impulsive noise, the performance of any correlator decoder can be quite poor.

B. LSB Codes

An early form of data hiding for grayscale images is based on LSB embedding techniques. In Section IV, we will see that these schemes may be interpreted as rudimentary binning schemes.

The method is applicable to host signals of the form $\mathbf{s} = \{s_1, s_2, \cdots, s_N\}$, where each sample s_i is encoded using b bits representing the natural binary decomposition of an integer between zero and $2^b - 1$. For instance, s_i could represent one of the 256 intensity levels of a monochrome image, such as $69 = (01\,000\,101)$; the LSB is one in this case. The LSB plane is the length-N binary sequence made of all the LSBs. The LSBs can be changed without adversely affecting signal quality, and so LSB embedding methods simply replace the LSB plane with an information sequence; the information rate is 1 bit per sample of $\mathbf{s}$. The payload could be increased by replacing the second LSB with an information sequence as well, but this would increase embedding distortion.

Note that the value of b (i.e., the range of host signal amplitudes) is immaterial here. The LSB embedding scheme is capable of rejecting host-signal interference. Unfortunately, LSB embedding does not survive modest amounts of noise. For instance, an attacker could simply randomize the LSB

[3] An exception arises when the noise is white and Gaussian. Then the correlation statistic is a *sufficient statistic* [38], and the correlator decoder is ideal.

plane, effectively destroying the hidden information that was originally embedded there.

C. *Performance Evaluation*

Various methods have been used to evaluate performance of watermarking and data-hiding algorithms. Many of these methods are simple but heuristic, e.g., quantify the similarity of an "extracted watermark" with the actual watermark that was embedded. However, there generally exists a well-defined, natural measure of system performance such as probability of error. The weakness of the heuristic methods is that they do not provide a reliable indication of the actual performance index of interest.

IV. Binning Schemes: General Principles

Binning is an important information-theoretic technique used in many different scenarios ranging from distributed source coding [55] and the problem of encoding data with side information at the transmitter only [28], to the classical problem of decoding data with side information that is available at the receiver only [30], [56]. Since blind data hiding is especially related to the problem of transmission with side information at the transmitter, we provide an overview of binning in this section. We start with a simple illustrative example.

Assume we want to embed one bit of information into an image, given in a raw, uncompressed format. At the same time we would like to compress the image. We can use one of several compression formats, for example we may choose to either use JPEG or JPEG2000 for this task. In fact, by simply making a choice of compression standard, we can embed one bit of information into the compressed image. An intended receiver for this one bit of information can identify which compression standard was used and, hence, could associate a JPEG compression with this bit being one while a JPEG2000 compression would assign this embedded bit the value zero. Now using information-theoretic jargon, we say that we have compressed the image using one of two bins (JPEG and JPEG2000). Note that the compression techniques both constitute different ways to represent a sequence of numbers (the original image file) as a string of bits. The latter process is also summarily referred to as vector quantization (VQ) [37]; see Appendix B for more details.

In data embedding applications, the essential idea of binning may be described as a VQ task using a family of *distinct* VQ mappings (for example JPEG or JPEG2000). In an information-theoretic context[4] VQ may be described as a generic name for any lossy data compression method. While the problems of general VQ are difficult and manifold, we next abstract the notions in an information-theoretic setting.

Let a source be given that produces random sequences $\mathbf{S}$ of length N over some alphabet $\mathcal{S}$. Assume we are given a collection $\mathcal{C}$ of length-N vectors $\mathbf{U}_j \in \mathcal{S}^N$ which play the role of a quantization codebook together with a distortion function D that measures distortion between vector in $\mathcal{S}^N$. The *VQ problem* consists of finding the vector $\mathbf{U}_j$ within the codebook $\mathcal{C}$ that minimizes the distortion between the observed $\mathbf{S}$ and the so called reconstruction vector $\mathbf{U}_j$.

Table 2
A Simple Binning Scheme: Embedding Length-2 Binary Message m Into Length-3 Binary Sequence $\mathbf{S}$, in a Way That Modifies at Most One Bit of $\mathbf{S}$

	$m = 00$	$m = 01$	$m = 10$	$m = 11$
$\mathbf{x} =$	$\boxed{000}$	001	$\boxed{010}$	$\boxed{011}$
	111	$\boxed{110}$	101	100

Next, assume that rather than only one codebook $\mathcal{C}$ we are given M different codebooks $\mathcal{C}_m$,[5] each consisting of a number of length-N vectors $\mathbf{U}_{m,j}$. Once we have a collection of codebooks $\mathcal{C}_m$ we may *choose* which codebook we want to use for the VQ task. In fact, given M codebooks we can embed $k = \log_2 M$ bits by this choice. Thus, given any observed source sequence $\mathbf{S}$ we can choose to quantize $\mathbf{S}$ to a vector $\mathbf{U}_{m,j}$ where m is chosen by the quantizer in order to embed k bits of information and j is chosen so as to minimize the distortion between $\mathbf{U}_{m,j}$ and $\mathbf{S}$ given the quantizer index m.

The following examples will clarify the basic ideas behind binning.

Example 1: Let $\mathbf{S}$ be a binary sequence of length $N = 3$. There are 8 such sequences: 000, 001, $\cdots$, 111, all assumed equally likely. We want to embed information into $\mathbf{S}$, producing a new sequence $\mathbf{X}$. Simultaneously, we require that the embedding method must satisfy the distortion constraint that $\mathbf{S}$ and $\mathbf{X}$ may differ in at most one position. We transmit $\mathbf{X}$ to a receiver which must decode the embedded information without knowing the original host data $\mathbf{S}$.[6]

Question 1: How many bits of information can we embed in $\mathbf{S}$?

Question 2: How can we design an appropriate encoding/decoding scheme?

Answer. Under the distortion constraint, the original $\mathbf{S}$ can be modified in at most four ways: $\mathbf{S} \oplus \mathbf{X} \in \{000, 001, 010, 100\}$, so at most two bits of information can be embedded. Straightforward spread-spectrum ideas do not work in this case: simply adding (modulo 2) one of the four patterns above to $\mathbf{S}$, which itself can assume any of the eight binary strings of length three, conveys no information to the receiver. Instead, consider a partition of the eight possible sequences $\mathbf{X}$ into four bins (columns of the 2 × 4 array), as shown in Table 2. Each bin corresponds to one of the 2-bit information sequences we want to communicate. Given an arbitrary sequence $\mathbf{S}$ and an arbitrary index $i \in \mathcal{M} = \{0, 1, 2, 3\}$ that we want to embed in the quantized version of $\mathbf{S}$, we look in bin i for the sequence $\mathbf{U}_{i,j}$ closest to $\mathbf{S}$ in the sense of Hamming distance, and declare that sequence to be $\mathbf{X}$. For instance, if $\mathbf{S} = 010$ and $i = 1$, corresponding to the second column in Table 4, we have to choose between the two sequences 001 and 110 in bin 1. The latter is closest to $\mathbf{S}$ and is thus declared to be $\mathbf{X}$. In Table 2, the four choices of $\mathbf{X}$ corresponding to the four

[4] In contrast to classification problems where VQ is also used to denote pattern classification problems.

[5] These different codebooks are referred to as the bins in binning schemes.

[6] The name host data refers to the role of $\mathbf{S}$ of hosting the embedded information.

Table 3
A Simple Binning Scheme: Embedding Message m (1 bit) Into Length-7 Sequences $\mathbf{S}$, in a Way That Modifies at Most Three Bits of $\mathbf{S}$. The Modified Sequence $\mathbf{X}$ is Later Degraded by Noise With Hamming Weight at Most 1

	$m=0$	$m=1$
$\mathbf{x}=$	0000000	1110000
	0001111	1111111
	0111100	1001100
	0110011	1000011
	1010101	0100101
	1011010	0101010
	1101001	0011001
	1100110	0010110

possible messages (with $\mathbf{S} = 010$) have been boxed. The decoder observes $\mathbf{X}$ and simply outputs the corresponding bin index m. Observe the following.

1) In any given bin, the two candidates $\mathbf{X}$ are maximally distant (Hamming distance $= 3$) as should be expected for a good vector quantization codebook.[7]
2) In any given bin, there is always one sequence that satisfies the embedding distortion constraint.
3) The receiver can decode the information bits *without error*.

Example 2: Let $\mathcal{S} = \{0, 1, \cdots, 2^b - 1\}$, and partition this set into the subset $\mathcal{S}_e = \{0, 2, \cdots, 2^b - 2\}$ of even integers and the subset $\mathcal{S}_o = \{1, 3, \cdots, 2^b - 1\}$ of odd integers. Let $\mathbf{S}$ be a host data length-N sequence in $\mathbf{S}$. Here the marked sequence $\mathbf{X}$ should satisfy $|X_i - S_i| \le 1$ (addition is modulo 2^b) for $1 \le i \le N$. Denote by $m = \{m_1, \cdots, m_N\}$ a binary sequence to be embedded into $\mathbf{S}$. Consider the LSB code of Section III-B which can be written as

$$x_i = m_i + 2\left\lfloor \frac{s_i}{2} \right\rfloor, \quad 1 \le i \le N.$$

So we choose $x_i \in \mathcal{S}_e$ if $m_i = 0$, and $x_i \in \mathcal{S}_o$ if $m_i = 1$. In terms of binning schemes we can interpret this LSB embedding as a binning scheme where $\mathcal{S}_e$ and $\mathcal{S}_o$ are the two bins from which we select x_i depending on the value of m_i.

Example 3: Consider Example 1 again, with the modification that observed sequences $\mathbf{S}$ of length seven are considered. Now we want to embed one bit of information into these sequences, which should incur a Hamming distance between $\mathbf{S}$ and the transmitted sequence $\mathbf{X}$ of at most two. Moreover, we allow for the additional modification that the decoder now does not have access to the marked sequence $\mathbf{X}$, but to a degraded sequence $\mathbf{Y} = \mathbf{X} + \mathbf{W}$. At most one bit of $\mathbf{X}$ is corrupted by noise, so there are eight possible noise sequences, $\mathbf{W} \in \{0\,000\,000, 0\,000\,001, \ldots, 10\,000\,000\}$. We ask the same questions Q1 and Q2 as in Example 1.

It turns out we can embed one bit of information using the binning scheme of Table 3. Here we need *two* different bins in order to embed $1 = \log_2 2$ bit of information. Each bin contains eight possible quantization words.[8] It can be verified that the distortion requirements are satisfied for both bins, i.e. to each of the $2^7 = 128$ possible sequences $\mathbf{S}$ there exists a quantization word at Hamming distance at most two. Moreover the union of the two bins constitutes an error correction code (7,4,3) with minimum Hamming distance of three,[9] which allows the correction of the error that is potentially introduced by $\mathbf{W}$.

The decoder observes $\mathbf{Y}$ and simply outputs the index of the bin which contains a word at distance at most one from the received word.

Example 3 casts light on the tradeoffs that will be a major topic in the remainder of this paper. In particular, the distortion that is allowable in the embedding process is offset against the amount of information that can be embedded as well as the distortion that a channel may incur between the transmitted word $\mathbf{X}$ and a received word $\mathbf{Y}$. For example, while we only can embed one bit of information in the setup of Example 3 (mostly due to the noisy channel) it is possible to embed up to four bits of information at a cost of at most two bits of embedding distortion if the channel would not incur any further distortion.

Let us make a few comments about terminology before concluding this section.

- In the watermarking literature, the encoding function $\mathbf{x} = f(\mathbf{s}, m, k)$ of (2.1) is often viewed as the cascade of two blocks. The first one produces a *watermark* $\mathbf{p}(\mathbf{s}, m, k)$, the second one "adds" it to the host to produce $\mathbf{x} = \mathbf{s} + \mathbf{p}$. These two steps are respectively termed *watermark encoding* and *watermark casting*. While there cannot be any fundamental advantage to this representation of the encoding function, there is nothing wrong with it either,[10] and several practical codes are based on it. The watermark casting step may also be replaced by a more general *mixing* step: $\mathbf{x} = \tilde{f}(\mathbf{s}, \mathbf{p}, k)$, where $\tilde{f}$ is the mixing function. See Section III-A for examples involving spread-spectrum codes.
- Binning and related methods are frequently termed "informed embedding" schemes in the watermarking literature, presumably to distinguish them from more elementary methods such as spread-spectrum which are termed "blind embedding" schemes. However, the encoder *always* has access to the host, and in this sense the distinction "informed embedding" versus "blind embedding" appears artificial. In contrast, the decoder does not necessarily have access to the original host, and therefore the "blind decoding" versus "nonblind decoding" terminology captures two fundamentally different scenarios.

V. Quantization-Based Codes

In 1999, Chen and Wornell introduced a class of data-hiding codes known as dither modulation codes,

[7] In fact, the different bins or VQ codebooks here are chosen as cosets of the linear repetition code of length three, an algebraic construction that will feature prominently in the next section.

[8] Both bins corresponds to a coset of the so-called simplex code.

[9] This is the one-error correcting Hamming code of length seven.

[10] Provided $\mathcal{S}$ is a field, so that addition can be properly defined.

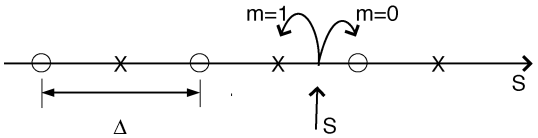


Fig. 4. Embedding one bit into one sample using original QIM. Here Λ_0 and Λ_1 are the sets of circles and crosses, respectively.

also referred to as quantization-index modulation (QIM) codes[11] [33], [57]. These methods embed signal-dependent watermarks using quantization techniques. It turns out that QIM is a binning scheme, in the sense of Section IV. The main objective of the embedding schemes in this section is, however, embedding in real-valued host data. These schemes are, furthermore, related to work from the early 1980s in information theory (see Section VII). Interestingly, based on this theory, Willems in 1988 had already formulated a setup for quantization-based codes [58], but his ideas remained undeveloped for about ten years. Meanwhile, Swanson *et al.* [59] and Yeung and Mintzer [60] invented quantization codes that are based on sound ideas but introduce excessive distortion relative to QIM. To introduce QIM, we start out the simplest case of embedding one information bit in a single real-valued sample.

A. Scalar-Quantizer Index Modulation

The basic idea of QIM can be explained by looking at the simple problem of embedding one bit in a real-valued sample. Here we have $m \in \{0,1\}$ (1-bit message), $s \in \mathbb{R}$ (1 sample), and no key k. A scalar, uniform quantizer $Q(s)$ with step size Δ is defined as $Q(s) = \Delta\lfloor s/\Delta \rfloor$. We may use the function $Q(s)$ to generate two new, *dithered* quantizers[12]

$$Q_i(s) = Q(s - d_i) + d_i, \quad i = 0, 1 \tag{5.1}$$

where

$$d_0 = -\frac{\Delta}{4}, \quad d_1 = \frac{\Delta}{4}. \tag{5.2}$$

The reproduction levels of quantizers Q_0 and Q_1 are shown as circles and crosses on the real line in Fig. 4. They form two lattices[13]

$$\Lambda_0 = -\frac{\Delta}{4} + \Delta\mathbb{Z}, \quad \Lambda_1 = \frac{\Delta}{4} + \Delta\mathbb{Z}. \tag{5.3}$$

1) Original QIM: In [57], the marked signal is defined as

$$x = \begin{cases} Q_0(s): & m = 0 \\ Q_1(s): & m = 1. \end{cases} \tag{5.4}$$

See Fig. 5. The maximum error due to embedding is $\Delta/2$. If the quantization errors are uniformly distributed over $[-(\Delta/2), (\Delta/2)]$ (more details in Section VI), the mean-squared distortion due to embedding is $D_1 = \Delta^2/12$.

Assume the marked signal x is corrupted by the attacker, resulting in a noisy signal $y = x + w$. The QIM decoder is a *minimum-distance decoder*. It finds the quantizer point closest to y and outputs the estimated message

$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} \operatorname{dist}(y, \Lambda_m) \tag{5.5}$$

where $\operatorname{dist}(y, \Lambda) \triangleq \min_{s \in \Lambda} |y - s|$. Clearly this scheme works perfectly (no decoding error) if $|w| < \Delta/4$. Observe that QIM may be thought of as a binning scheme with some error protection against noise (analogously to Example 3 in Section IV). The two bins are the lattices Λ_0 and Λ_1.

2) Distortion-Compensated Scalar QIM: The above QIM embedding scheme works poorly if the noise level exceeds $\Delta/4$. However, the scheme can be modified to increase resistance to noise [33], [63]. Given a host-sample s, the distortion-compensated scalar QIM embedding function is defined as

$$x = \begin{cases} Q_0(\alpha s) + (1-\alpha)s: & m = 0 \\ Q_1(\alpha s) + (1-\alpha)s: & m = 1 \end{cases} \tag{5.6}$$

(see Fig. 6), where $\alpha \in [0,1]$ is a parameter to be optimized. Observe that (5.6) coincides with the original scheme for $\alpha = 1$. Also, if $\alpha = 0$, (5.6) yields $x = s$, i.e., the embedding is degenerate and introduces no distortion. More generally, adjusting the value of α in the range [0,1] allows us to compensate the distortion introduced by the quantizer.

The embedding formula (5.6) may also be rewritten as the sum of s and a perturbation due to quantization of αs

$$x = \begin{cases} s + (Q_0(\alpha s) - \alpha s): & m = 0 \\ s + (Q_1(\alpha s) - \alpha s): & m = 1. \end{cases} \tag{5.7}$$

A third expression for the embedding function is

$$x = \begin{cases} \frac{d_0}{\alpha} + X_{\text{proto}}\left(s - \frac{d_0}{\alpha}\right): & m = 0 \\ \frac{d_1}{\alpha} + X_{\text{proto}}\left(s - \frac{d_1}{\alpha}\right): & m = 1 \end{cases} \tag{5.8}$$

where

$$X_{\text{proto}}(s) = Q(\alpha s) + (1-\alpha)s \tag{5.9}$$

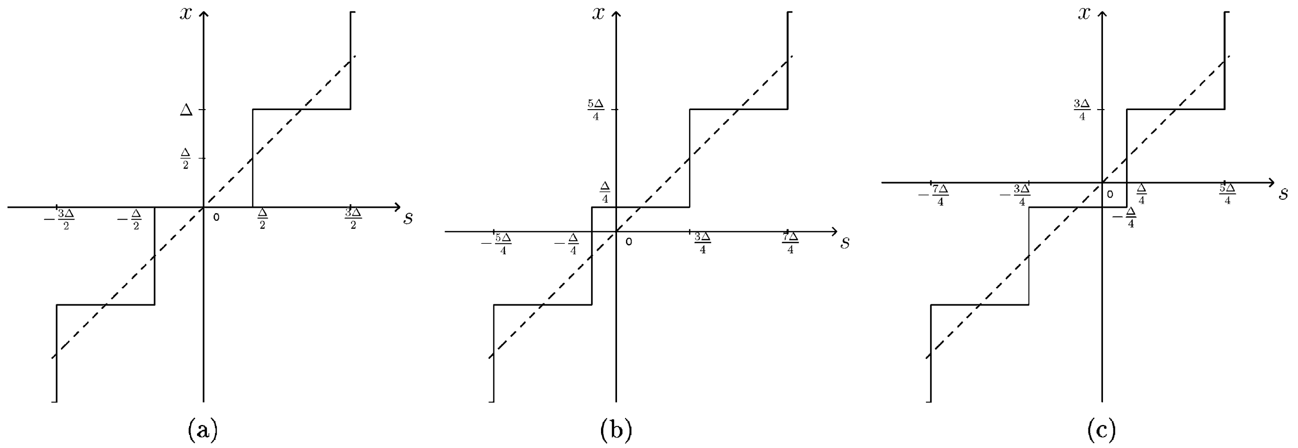
is the prototype sloped-staircase function shown in Fig. 6. This function is symmetric around $s = 0$, is made of linear segments with slope $1 - \alpha$, and takes its values in a union of intervals of width $(1-\alpha)\Delta/\alpha$

$$\mathcal{X}_{\text{proto}} := \frac{\Delta}{2\alpha} \bigcup_{n \in \mathbb{Z}} [n - (1-\alpha), n + (1-\alpha)]. \tag{5.10}$$

[11] Later termed “scalar Costa scheme” when scalar quantizers are used [61], [62]. We retain the original QIM terminology in this paper.

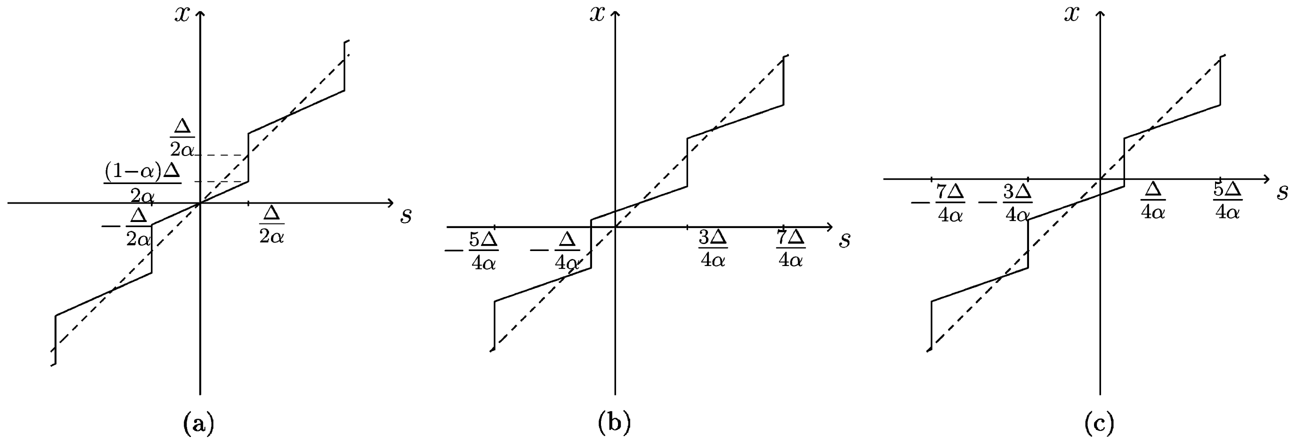
[12] Dithering is classical technique used in signal compression for improving the perceptual aspect of quantized signals.

[13] Strictly speaking, two cosets of a lattice $\Delta\mathbb{Z}$. Lattices are formally defined in Section V-C1.



(a) (b) (c)

Fig. 5. Selection of marked sample X given S and $m \in \{0, 1\}$, using original QIM method. (a) Prototype symmetric function. (b) Embedding function for $m = 0$. (c) Embedding function for $m = 1$.



(a) (b) (c)

Fig. 6. Selection of marked sample x given s and $m \in \{0, 1\}$, using distortion-compensated QIM. (a) Prototype $X_{\text{proto}}(s)$. (b) $m = 0$. (c) $m = 1$.

The actual marked value X takes its values in the offset domain $(d_m/\alpha) + \mathcal{X}_{\text{proto}}$. The maximal quantization error is $|x - s| = \Delta/2$ and occurs when $x = (\Delta/\alpha)((1/2) + n)$, $n \in \mathbb{Z}$. The decoder implements

$$\hat{m} = \operatorname*{argmin}_{m\in\{0,1\}} \text{dist}(\alpha y, \Lambda_m).$$

The advantages of this generalized scheme are not obvious now but will become clear in Section VI when a statistical model for the attack noise w is considered. So compelling are these advantages, in fact, that the distortion-compensated QIM scheme has replaced the original QIM scheme in practice, and the qualifier "distortion-compensated" is often omitted for the sake of brevity. It is interesting to note that, while the distortion compensation technique outlined here is widely used, it does not come with any claim of optimal distortion compensation. In fact it is possible to find functions other than $X_{\text{proto}}(s)$ which exhibit a slightly better performance than the function of (5.9). However, the gains offered are fairly small and the complexity of solving the nonlinear optimization problem to find the best function in place of (5.9) goes beyond the scope of this paper.

B. *Sparse QIM*

Chen and Wornell showed how to extend the scalar QIM scheme above to embed one bit in a length-N host sequence. They considered two basic methods.

The first method, which they called spread transform dither modulation (STDM), consists of quantizing the projection of the host vector along a given direction $\mathbf{p}$. Specifically, given a host vector $\mathbf{s}$ and a unit-length vector $\mathbf{p}$, they define the marked signal as

$$\mathbf{x} = \begin{cases} \mathbf{s} + \left(Q_0(\mathbf{s}^T\mathbf{p}) - \mathbf{s}^T\mathbf{p}\right)\mathbf{p}: & m = 0 \\ \mathbf{s} + \left(Q_1(\mathbf{s}^T\mathbf{p}) - \mathbf{s}^T\mathbf{p}\right)\mathbf{p}: & m = 1 \end{cases} \tag{5.11}$$

where the superscript T denotes vector transpose. See Fig. 7. The decoder projects the received data onto direction $\mathbf{p}$ and decides whether quantizer Q_0 or Q_1 was used

$$\hat{m} = \operatorname*{argmin}_{m\in\{0,1\}} \text{dist}(\mathbf{y}^T\mathbf{p}, \Lambda_m). \tag{5.12}$$

Observe that the distortion due to embedding takes place in direction $\mathbf{p}$ only; no other component of $\mathbf{s}$ is modified. Therefore the embedder can allocate the entire distortion budget in

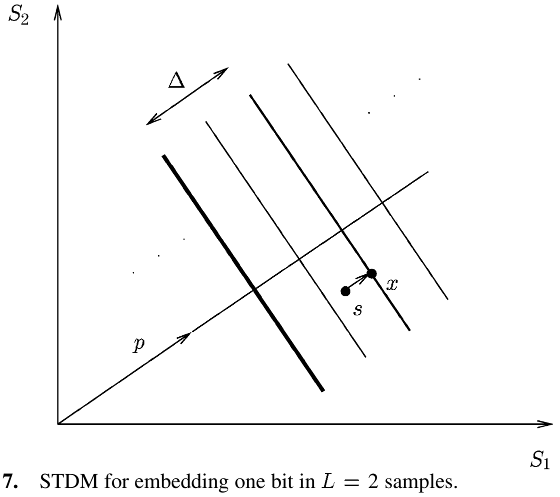


Fig. 7. STDM for embedding one bit in $L = 2$ samples.

direction $\mathbf{p}$, enabling the use of a large quantizer step size. For instance, if $\mathbf{p}$ is chosen at random, choosing $\Delta = \sqrt{12ND_1}$ results in an expected per-sample mean-square error equal to D_1.[14] The large quantizer step size (relative to the case $N = 1$) offers an increased protection against noise. The distance between the lattices Λ_0 and Λ_1 is $d_{\min} = \Delta/2 = \sqrt{3ND_1}$.

In our view, the name "STDM" is somewhat misleading because the method does not involve a transform of the host signal—just a projection onto a small-dimensional space. For this reason we think of it as a sparse QIM coding method.

Various extensions and refinements of the basic STDM method are possible. In particular, one can use distortion-compensated STDM (as will be seen later, the optimal choice for α is close to one in that case, i.e., the scheme is very similar to basic STDM). Another idea is to quantize a few components of the host signal and not just one. All these codes may be thought of as *sparse* QIM codes. The number of signal components used for embedding, divided by N, is the *sparsity factor* τ of the code; N/τ is sometimes called spreading factor. If one bit is embedded per signal component, the code rate R is equal to τ. We note that for a very sparse code, R approaches zero. This observation will be of interest in Section VIII-B.

C. Lattice-Quantizer Index Modulation

Chen and Wornell [33] presented a second extension of the scalar QIM scheme to the vector case. The idea is to replace the scalar quantizer of (5.6) with a L-dimensional VQ quantizer. Fig. 8 illustrates this concept when $L = 2$ and the VQ is obtained by independently quantizing each coordinate of $\alpha\mathbf{s}$ with the scalar quantizer of (5.6). In effect $\alpha\mathbf{s}$ is quantized using one of the two lattices

$$\Lambda_0 = \left(-\frac{\Delta}{4}, \cdots, -\frac{\Delta}{4}\right) + \Delta\mathbb{Z}^L$$
$$\Lambda_1 = \left(\frac{\Delta}{4}, \cdots, \frac{\Delta}{4}\right) + \Delta\mathbb{Z}^L. \qquad (5.13)$$

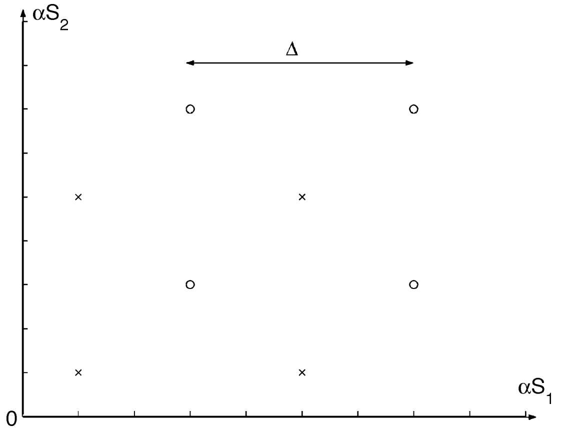


Fig. 8. QIM for embedding one bit in $L = 2$ samples using cosets Λ_0 (circles) and Λ_1 (crosses) of a cubic lattice.

Observe that the mean-squared distortion due to embedding is still $D_1 = \Delta^2/12$. The rate of the code is $R = \tau = 1/L$. The distance between the sets Λ_0 and Λ_1 is now

$$d_{\min} = \frac{1}{2}\Delta\sqrt{L} = \sqrt{3LD_1}.$$

The decoder's output is

$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} \operatorname{dist}(\alpha\mathbf{y}, \Lambda_m), \qquad (5.14)$$

defining $\operatorname{dist}(\mathbf{y}, \Lambda) := \min_{\mathbf{p}\in\Lambda} \|\mathbf{y} - \mathbf{p}\|$. The quantity $\operatorname{dist}(\alpha\mathbf{y}, \Lambda_m)$ is a coordinatewise sum of squared quantization errors.

1) General Construction: The papers [64] and [65] presented a general approach for constructing structured binning schemes to approach capacity. The approach is based on nested lattices.

A lattice Λ in N-dimensional Euclidean space is defined as a set of points in $\mathbb{R}^N$ such that $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda$ implies $\mathbf{x} + \mathbf{y} \in \Lambda$ and $\mathbf{x} - \mathbf{y} \in \Lambda$, which equips Λ with the structure of an additive subgroup of $\mathbb{R}^N$. A lattice may be defined by a set of row-vectors $g_1, \cdots, g_N \in \mathbb{R}^N$. These vectors are stacked in a $N \times N$ matrix called *generator matrix*. The lattice is the set of all integral combinations of the basis vectors: $\Lambda = \{\mathbf{x} = \mathbf{v}G, \mathbf{v} \in \mathbb{Z}^N\}$. Given Λ, the choice of G is nonunique.

Next consider a sublattice Λ_c of Λ. Since Λ_c is a subgroup of Λ, the cosets[15] of Λ_c form a partition of Λ. A *nested* lattice then consists of an N-dimensional *lattice partition* Λ_f/Λ_c where Λ_f and $\Lambda_c \subset \Lambda_f$ are respectively referred to as the *fine* lattice and the *coarse* lattice.

For a pair of nested lattices (Λ_c, Λ_f) there exist corresponding generator matrices G_c and G_f such that

$$G_c = JG_f, \qquad (5.15)$$

[14] It is worthwhile to point out that the same performance would be achieved by, for example, just choosing one element in a length-N vector $\mathbf{s}$ in order to embed information with a distortion budget of $\Delta = \sqrt{12ND_1}$. While the per-sample mean square error again equals D_1 such a scheme would incur a visually noticable distortion in the chosen element.

[15] A coset of a group G' with respect to G is defined as $G' + g$ for $g \in G$.

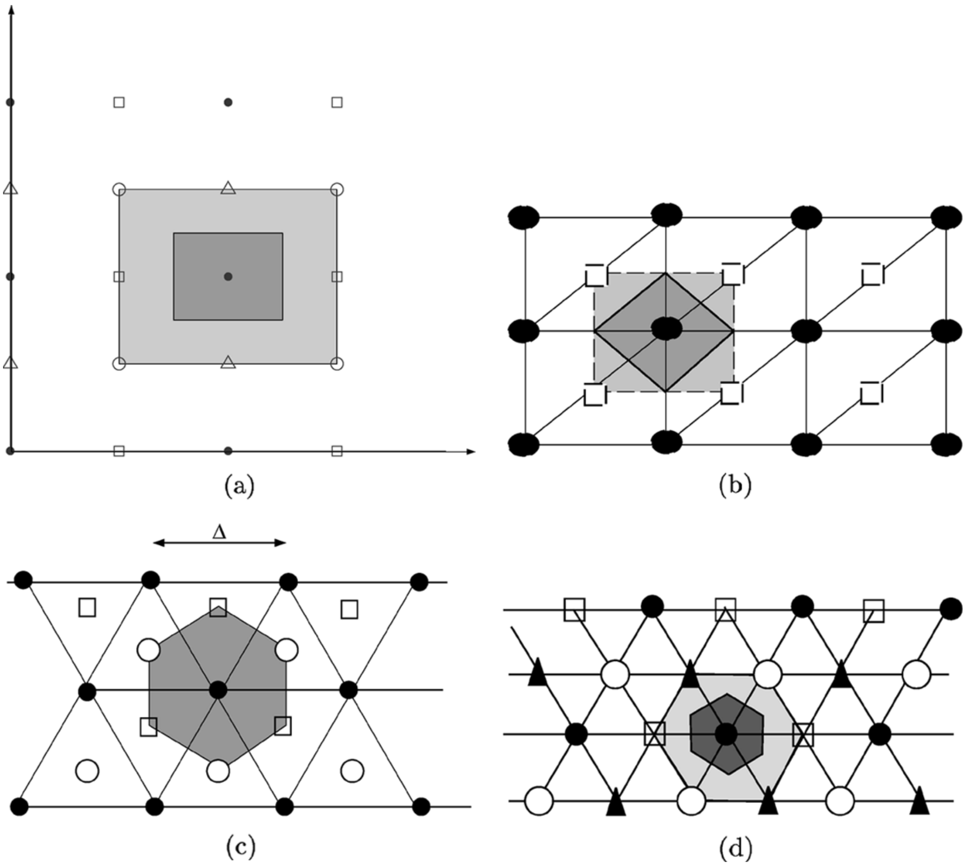


Fig. 9. Nested two-dimensional lattices. The coarse lattice Λ_c is the set of heavy dots, and its cosets are represented by squares, circles, and triangles. Each lightly shaded region is $\mathcal{V}$, the Voronoi cell of Λ_c. The darker regions are Voronoi cells for Λ_f. (a) Cubic lattice. (b) Quincunx lattice. (c) Hexagonal lattice #1. (d) Hexagonal lattice #2.

where J is an $N \times N$ integer matrix, referred to as *subsampling matrix*, whose determinant satisfies $|\det J| > 1$. Then $\Lambda_c \subset \Lambda_f$. The density of Λ_f relative to Λ_c is equal to $|\det J|$. Thus, the lattice Λ_f may be decomposed as the union of $|\det J|$ cosets $\{\Lambda_m,\ 0 \le m < |\det J|\}$ of Λ_c

$$\Lambda_f = \cup_{m=0}^{|\det J|-1} \Lambda_m.$$

For each coset Λ_m of Λ_c in Λ_f, we can find an element $\mathbf{v}_{[m]} \in \Lambda_f$ of shortest norm such that $\Lambda_m = \Lambda_c + \mathbf{v}_{[m]}$. Such an element $\mathbf{v}_{[m]}$ is called the coset leader of Λ_m.

The set

$$\mathcal{C} = \Lambda_f \setminus \Lambda_c = \{\Lambda_m,\ 0 \le m < |\det J|\} \tag{5.16}$$

carries itself a group structure and is termed the *quotient group* of Λ_f by Λ_c. $\mathcal{C}$ may be efficiently represented by the coset leaders of the respective cosets.

Finally, we define

Q quantization function mapping each point $\mathbf{x} \in \mathbb{R}^N$ to the nearest lattice point in Λ_c;

$\mathcal{V}$ $\{\mathbf{x} \in \mathbb{R}^N : Q(\mathbf{x}) = 0\}$ = Voronoi cell of Λ_c.

Example: Let $\Lambda_c = \Delta\mathbb{Z}^N$ and $\Lambda_f = D_N^* = \Delta\mathbb{Z}^N \cup ((\Delta/2), \cdots, (\Delta/2)) + \Delta\mathbb{Z}^N$. We obtain $\mathcal{C} = \{(0, \ldots, 0), ((\Delta/2), \ldots, (\Delta/2))\}$. Then $\mathcal{V}$ is the N-dimensional cube $[-(\Delta/2), (\Delta/2)]^N$; its normalized second-order moment is equal to $\Delta^2/12$. Fig. 9(b) illustrates this design when $N = 2$; $\Lambda_f = D_2^*$ is then called the quincunx lattice.

If $\mathcal{M}$ grows exponentially with N (i.e, the code rate $R > 0$), the lattice partition Λ_f/Λ_c should have the following properties.

(P1) Q should be a *good* vector quantizer with mean-squared distortion D_1; $\mathcal{V}$ should thus be, loosely speaking, nearly spherical.

(P2) $\mathcal{C}$ should be a good channel code with respect to Gaussian noise: loosely speaking, the codewords in $\mathcal{C}$ should be far away from each other.

To each $m \in \mathcal{M}$ corresponds a codeword $\mathbf{d}_m \in \mathcal{C}$ and a translated coarse lattice $\Lambda_m = \mathbf{d}_m + \Lambda$. The fine lattice is the union of all these translated lattices.

Given m and $\mathbf{s}$, the encoder quantizes $\alpha\mathbf{s}$ to the nearest point in Λ_m, obtaining

$$\mathbf{u}(m) = Q_m(\alpha\mathbf{s}) \triangleq Q(\alpha\mathbf{s}-\mathbf{d}_m)+\mathbf{d}_m \qquad \in \Lambda_m \tag{5.17}$$

by quantizing $\alpha\mathbf{s}$ to the nearest point in Λ_m. The difference $\mathbf{u}(m) - \alpha\mathbf{s}$ represents a quantization error. Finally, the marked sequence is given by

$$\mathbf{x} = (1-\alpha)\mathbf{s} + \mathbf{u}(m) \tag{5.18}$$
$$= (1-\alpha)\mathbf{s} + Q_m(\alpha\mathbf{s}) \tag{5.19}$$

Table 4
Examples of Nested Lattice Pairs (G_f, G_c)

construction	G_f	G_c	$J = G_c G_f^{-1}$	det J
cubic lattice #1	$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$2I_2$	$2I_2$	4
cubic lattice #2	I_2	$\begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$	2
cubic lattice #3	$\begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 0 & \epsilon \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$	2
quincunx lattice	$\begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$	I_2	$\begin{pmatrix} 1 & 0 \\ -1 & 2 \end{pmatrix}$	2
hexagonal lattice #1	$\begin{pmatrix} 1 & -\frac{1}{\sqrt{3}} \\ 1 & \frac{1}{\sqrt{3}} \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 1 & \sqrt{3} \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$	3
hexagonal lattice #2	$\begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 1 & \sqrt{3} \end{pmatrix}$	$2I_2$	4

which is a generalization of (5.6). The payload of the code is $|\det J|$ and its rate is $R = (1/N)\log|\det J|$.

The decoder quantizes $\alpha\mathbf{y}$ to the nearest point in the fine lattice $\Lambda_f = \cup_{m\in\mathcal{M}}\Lambda_m$. It then outputs the corresponding index

$$\hat{m} = \underset{m\in\mathcal{M}}{\operatorname{argmin}}\, \operatorname{dist}(\alpha\mathbf{y}, \Lambda_m). \qquad (5.20)$$

Table 4 and Fig. 9 depict this construction for several examples in which $N = 2$. The first and third cases are examples in which the coarse and fine lattices are *self-similar*. The $|\det J|$ coset leaders are labeled by squares, circles, and triangles in Fig. 9. The third case yields the STDM technique when $\epsilon \to 0$, as discussed in Section V-B.

2) Practical Codes: To satisfy properties (P1) and (P2) above, we need Λ_f and Λ_c to be high-dimensional. In practice, one cannot afford using arbitrary high-dimensional lattices, because quantization operations become prohibitively expensive. Instead one can would use lattices that have a special structure, e.g., products of low-dimensional lattices.[16] Another powerful idea is to use recursive quantization techniques such as trellis-coded quantization [37], [66] to (implicitly) define the coarse lattice Λ_c. Similarly, one can use classical error-correction codes such as Hamming codes and turbo codes to (implicitly) define the fine lattice Λ_f. The latter idea is illustrated in Fig. 10, where the actual message $m \in \mathcal{M}$ is first encoded into a longer (redundant) sequence $\tilde{m}$, which is used as an input to the nested lattice code. These two codes are termed outer code and inner code, respectively. Chou and Ramchandran [67] recently proposed the use of an outer erasure code; their scheme is intended to resist erasures, insertions, and deletions, in addition to the Gaussian-type attacks that the inner code is designed to survive. Solanki *et al.* [68] studied a closely related system and applied it to data hiding in images.

It should be emphasized that the cascade of linear outer and inner codes as depicted in Fig. 10 is done solely for computational convenience and is a special case of the general construction of Section V-C1. Any linear code may be thought of as a lattice code.

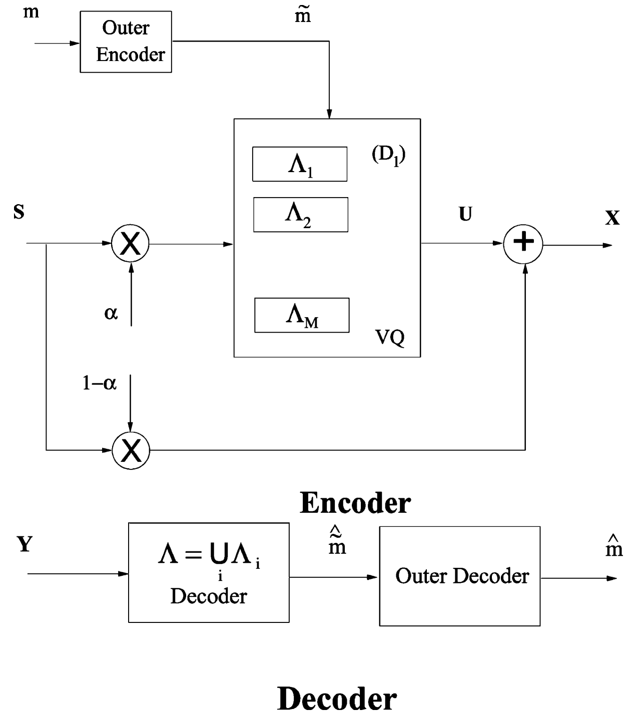


Fig. 10. Lattice-based encoder and decoder for data hiding, using the encoding function (5.18) and the decoding function (5.14).

3) External Dithering: Working on different problems, Eggers *et al.* [62] and Zamir *et al.* [65] studied lattice QIM schemes in which the traditional quantization function $Q : \mathbb{R}^N \to \Lambda_c$ is replaced with a *dithered quantization function*. Given any $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{d} \in \mathcal{V}$, a dithered quantizer produces the output

$$\hat{\mathbf{x}} = Q(\mathbf{x} - \mathbf{d}) + \mathbf{d} \quad \in \Lambda_c + \mathbf{d}.$$

If the external dither sequence $\mathbf{d}$ is independent of $\mathbf{x}$ and uniformly distributed over $\mathcal{V}$, it turns out that the quantization

[16] The cubic lattice is the simplest example of a product lattice.

在细格构造dither更好

error $\hat{\mathbf{x}} - \mathbf{x}$ is also independent of $\mathbf{x}$ and uniformly distributed over $\mathcal{V}$ [76], [77]. This property considerably simplifies the analysis and understanding of QIM schemes and has therefore been popular in theoretical analyses. Additionally, $\mathbf{d}$ is shared with the decoder; $\mathbf{d}$ can thus be used to randomize the lattice QIM code and provide some level of protection against attacks on the code.

When dithered QIM is used in place of nondithered QIM, the basic equations (5.17), (5.19), and (5.20) are replaced by the following expressions. Given m and $\mathbf{s}$, the encoder computes

$$\mathbf{u}(m) = Q(\alpha\mathbf{s} - \mathbf{d}_m - \mathbf{d}) + \mathbf{d}_m + \mathbf{d} \quad \in \Lambda_m + \mathbf{d} \tag{5.21}$$

and outputs the marked sequence

$$\mathbf{x} = (1-\alpha)\mathbf{s} + \mathbf{u}(m). \tag{5.22}$$

The decoder's output is

$$\hat{m} = \operatorname*{argmin}_{m \in \mathcal{M}} \operatorname{dist}(\alpha\mathbf{y} - \mathbf{d}, \Lambda_m). \tag{5.23}$$

VI. Probability of Error

The natural metric for quantifying decoding performance is probability of decoding error. This type of analysis can be rather complicated but useful results can be obtained using appropriate asymptotic methods (as $N \to \infty$).

Refer to Fig. 1 and for simplicity of the exposition, assume that the only data available to the decoder is the degraded signal $\mathbf{Y}$ (i.e., no side information K). The decoding rule partitions the received data space into decoding regions $\mathcal{Y}_m$, $m \in \mathcal{M}$. The decoder outputs message m for all sequences that belong to $\mathcal{Y}_m$. The probability that message m is not decoded correctly is $P_{e|m} = Pr[\mathbf{Y} \notin \mathcal{Y}_m | m \text{ sent}]$. It depends on $\{\mathcal{Y}_m\}$ and the statistics of the host signal and the randomized code. To analyze this problem, it is convenient to study the case of two codewords first. The reader is invited to review detection-theoretic notions in Appendix C. For simplicity of the exposition we shall assume that the attacker's noise is Gaussian. The same type of analysis can be performed when the noise is non-Gaussian [69], [70].

A. Binary Detection—Scalar Case

Consider the case of binary detection first: $\mathcal{M} = \{0, 1\}$. The decoding problem takes the form of the following equation [(C.1) from Appendix C]:

$$\begin{cases} H_0 : \mathbf{Y} \sim p_0 \\ H_1 : \mathbf{Y} \sim p_1. \end{cases}$$

Some detection rules are relatively simple, e.g., the correlators and nearest-neighbor decoders encountered in SSM and QIM watermarking. A statistical model such as (C.1) is not even required in this case.

Improved detection rules can often be derived by exploiting knowledge of the statistics of $\mathbf{Y}$. For instance, if both messages are equally likely, the detector that minimizes probability of error is the maximum likelihood (ML) detector [(C.2) from Appendix C, restated below for convenience]

$$L(\mathbf{y}) = \frac{p_1(\mathbf{y})}{p_0(\mathbf{y})} \underset{H_0}{\overset{H_1}{\gtrless}} 1.$$

The probability of error for this test is given by (C.3) from Appendix C.

To achieve low P_e, we need to create a substantial disparity between the pdf's p_0 and p_1. Let us see how some basic data-hiding codes perform in this respect. We use a simple model to illustrate the ideas: embed 1 bit into $N = 1$ sample.

Example: Consider real-valued s, x and y. The host signal sample S is distributed as $\mathcal{N}(0, \sigma_s^2)$. The attack is

$$y = x + w \tag{6.1}$$

where W is Gaussian noise, distributed as $\mathcal{N}(0, \sigma_w^2)$, and independent of S. The performance of SSM and QIM systems is derived below.

1) SSM: The spread-spectrum scheme is given by

$$x = \begin{cases} s + a : & m = 0 \\ s - a : & m = 1 \end{cases} \tag{6.2}$$

in which the original s is unknown to the detector. Equation (6.2) is a special case of (3.1). From (2.9) and (2.10), we obtain

$$\text{WNR} = \frac{a^2}{\sigma_w^2}, \quad \text{WHR} = \frac{a^2}{\sigma_s^2}.$$

The rival pdf's in (C.1) are given by

$$p_0 = \mathcal{N}\left(a, \sigma_s^2 + \sigma_w^2\right) \quad \text{and} \quad p_1 = \mathcal{N}\left(-a, \sigma_s^2 + \sigma_w^2\right)$$

and are shown in Fig. 11. They are hard to distinguish when $a^2 \ll \sigma_s^2 + \sigma_w^2$. This corresponds to the common case of a strong host-to-watermark ratio; detection performance is poor. More precisely, $P_e = Q(d/2)$, where

$$d = \sqrt{\frac{(2a)^2}{\sigma_s^2 + \sigma_w^2}} = 2\sqrt{\left(\text{WNR}^{-1} + \text{WHR}^{-1}\right)^{-1}}$$

is a normalized distance between the two pdf's, and the Q function was defined in Appendix C. Note that detection becomes completely unreliable when WHR → 0.

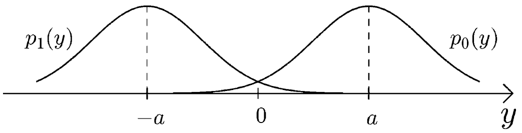


Fig. 11. Rival pdf's for detection of $m \in \{0, 1\}$, using SSM.

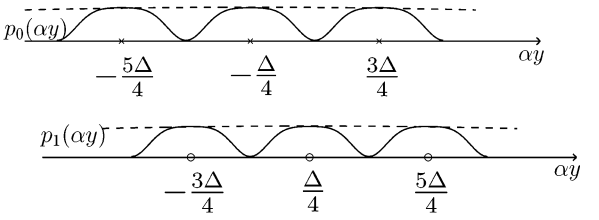


Fig. 12. Rival pdf's for detection of $m \in \{0, 1\}$ based on scaled data αY, using scalar QIM with WNR $= 0.1$ and $\alpha = \text{WNR}/(1 + \text{WNR})$.

For detection with the host signal known to the detector (nonblind watermarking), we have $P_e = Q(d/2)$ again, where $d = \sqrt{(2a)^2/\sigma_w^2} = 2\sqrt{\text{WNR}}$. This performance is achieved independently of the value of WHR, and so detection performance is much improved when WHR $\ll$ WNR. In fact SSM is an ideal modulation scheme for this problem.

2) Scalar QIM: Assume again that S is unknown to the detector. Consider the distortion-compensated scalar QIM scheme (5.7). The rival distributions of Y are shown in Fig. 12. Observe that

1) The perturbation due to embedding (quantization noise) is limited between $-\Delta/2$ and $\Delta/2$. Under Bennett's high-rate model for quantization noise, this perturbation is *approximately* uniformly distributed between $-\Delta/2$ and $\Delta/2$, and the distortion due to embedding is $D_1 := \mathbb{E}(X - S)^2 \approx \Delta^2/12$. In fact, the uniform quantization model is *exact* for any value of Δ if a dither quantizer is used, as discussed in Section V-C3. For the problem at hand, this means that d_0 is randomized uniformly over $[-\Delta/2, \Delta/2]$ and that we keep $|d_1 - d_0| = \Delta/2$. Equivalently, given D_1, we select
$$\Delta = \sqrt{12 D_1}. \tag{6.3}$$
Also WNR $= (\Delta^2/12)/\sigma_w^2$.
2) For large σ_s^2, we can view the pdf's p_0 and p_1 as quasi-periodic, with period equal to Δ/α. Roughly speaking, the ability to discriminate between p_0 and p_1 depends on the overlap between the support sets of p_0 and p_1, and fairly little on σ_s^2.
3) As mentioned below (5.10), X takes its values in the set $d_m/\alpha + \mathcal{X}_{\text{proto}}$. Since W is independent of X, the "rounded pulses" that make up the pdf's p_0 and p_1 are given by the convolution of a rectangular pulse of width $(1 - \alpha)\Delta/\alpha$, with the $\mathcal{N}(0, \sigma_w^2)$ pdf.
4) For good discrimination between p_0 and p_1, the pulses should have relatively small overlap.
5) In the absence of attacker's noise ($\sigma_w^2 = 0$), the best choice for α would be one, in which case we obtain error-free detection.
6) For $\sigma_w^2 > 0$, the choice of α results from a tradeoff between embedding distortion and detection performance. The tradeoff is determined by the value of the parameters Δ and α of the embedding function (5.6).
7) For large σ_s^2, little information is lost by reducing Y to the test statistic
$$\tilde{Y} := \alpha Y \bmod \Delta \quad \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right]. \tag{6.4}$$
The pdf's of $\tilde{Y}$ under H_0 and H_1 are shown in Fig. 13 for two values of α. The minimum-distance decoding rule (5.14) is replaced by
$$\hat{m} = \operatorname*{argmin}_{m \in \{0,1\}} |\tilde{y} - d_m| \tag{6.5}$$
where $|d_1 - d_0| = \Delta/2$.

B. Modulo Additive Noise Channel

The advantage of the processing (6.4) of the data Y is that it yields approximations to the optimal ML test (C.2) and to the probability of error (C.3) that are simple, good, and independent of the exact statistics of S. From (3.2), (5.8), and (6.4), note that
$$\tilde{y} = (d_m + \tilde{e} + \tilde{w}) \bmod \Delta \tag{6.6}$$
where
$$\tilde{E} := \alpha X_{\text{proto}}\left(S - \frac{d_m}{\alpha}\right) \bmod \Delta \tag{6.7}$$
is termed *self-noise*, and
$$\tilde{W} := \alpha W \bmod \Delta \tag{6.8}$$
is the *aliased attacker's noise*. Indeed the pdf of $\tilde{W}$ is an aliased version of $p_{\alpha W}$
$$p_{\tilde{W}}(\tilde{w}) = \sum_{k=-\infty}^{\infty} p_{\alpha W}(\alpha \tilde{w} + k\Delta), \quad 0 \le \tilde{w} \le \Delta. \tag{6.9}$$
Note that $\tilde{E} = 0$ for $(1 - \alpha)(\Delta/2) < |\tilde{E}| < (1 + \alpha)(\Delta/2)$. Under the high-rate quantization model, $\tilde{E}$ is independent of

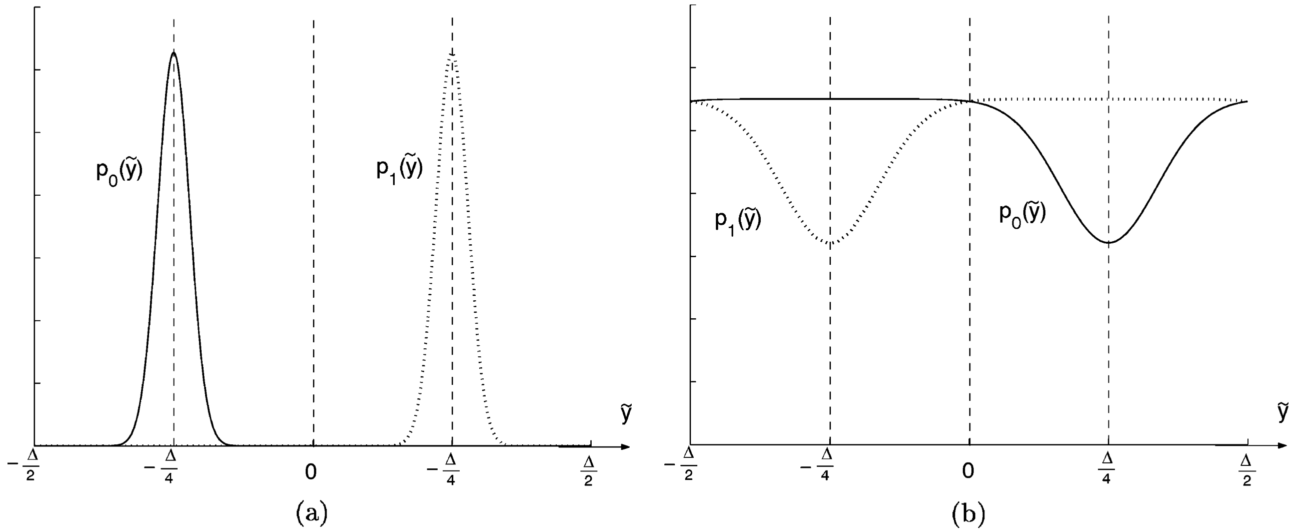


Fig. 13. Rival pdf's for detection of $m \in \{0, 1\}$ based on $\tilde{Y}$, using scalar QIM with $\alpha = \text{WNR}/(1 + \text{WNR})$. (a) WNR = 100. (b) WNR = 0.1.

m, and $p_{\tilde{E}}$ may be approximated with a rectangular pulse of width $(1-\alpha)\Delta$ centered at zero

$$p_{\tilde{E}}(\tilde{e}) = \frac{1}{\Delta(1-\alpha)} 1_{\{|\tilde{e}| \le \frac{\Delta}{2}(1-\alpha)\}}.$$

This statistical model is *exact* if dithered QIM is used, as described in Section V-C3.

Under hypothesis $H_i, i = 0, 1$, the data $\tilde{Y}$ may be viewed as the sum of an offset d_i and a noise V equal to the sum of the self-noise and the aliased attacker's noise

$$V = \tilde{E} + \tilde{W} \bmod \Delta. \tag{6.10}$$

Since $\tilde{E}$ and $\tilde{W}$ are statistically independent, the pdf of V is the circular convolution of the pdf's of $\tilde{E}$ and $\tilde{W}$

$$p_{\mathrm{V}}(v) = (p_{\tilde{E}} \star p_{\tilde{W}})(v), \quad 0 \le v \le \Delta. \tag{6.11}$$

Therefore, the pdf of $\tilde{Y}$ under H_i takes the form

$$q_i(\tilde{y}) = p_{\mathrm{V}}(\tilde{y} - d_i), \quad i = 0, 1. \tag{6.12}$$

The rival pdf's $q_i(\tilde{y})$, $i = 0{,}1$ are simply translates of p_{V}. The detector must decide between the two hypotheses

$$\begin{cases} H_0 : \tilde{Y} = d_0 + V \\ H_1 : \tilde{Y} = d_1 + V. \end{cases} \tag{6.13}$$

The role of α as a tradeoff between self-noise and attacker's noise appears clearly in this formulation of the detection problem. For small α, the self-noise $\tilde{E}$ dominates the attacker's aliased noise $\tilde{W}$. For $\alpha = 1$, the self-noise is zero, and the attacker's noise dominates. Equation (6.13) defines a *modulo additive noise* (MAN) channel, diagrammed in Fig. 14.

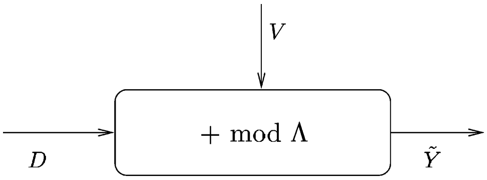


Fig. 14. Modulo additive noise channel.

As an alternative to the simple minimum-distance detector (6.5), we study the theoretically optimal ML detector (C.2). The ML detector based on the transformed data $\tilde{Y}$ and the statistical model above is

$$\frac{p_{\mathrm{V}}(\tilde{y} - d_1)}{p_{\mathrm{V}}(\tilde{y} - d_0)} \underset{H_0}{\overset{H_1}{\gtrless}} 1. \tag{6.14}$$

It coincides with the nearest-neighbor detection rule (5.14) if the attacker's noise distribution p_W is unimodal and symmetric.

The probability of error for the optimal test (6.14) is

$$\tilde{P}_e = \frac{1}{2} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \min\left(q_0(\tilde{y}), q_1(\tilde{y})\right) d\tilde{y}. \tag{6.15}$$

If the noise distribution $p_W(w)$ is symmetric around $w = 0$, so is $p_{\tilde{W}}(\tilde{w})$. The two rival pdf's, $q_0(\tilde{y})$ and $q_1(\tilde{y})$, have means $\mu_0 = d_0$ and $\mu_1 = d_1$ respectively, and common variance σ_v^2. For moderate-to-large WNR, we have

$$\sigma_v^2 \approx (1-\alpha)^2 \frac{\Delta^2}{12} + \alpha^2 \sigma_{\tilde{w}}^2.$$

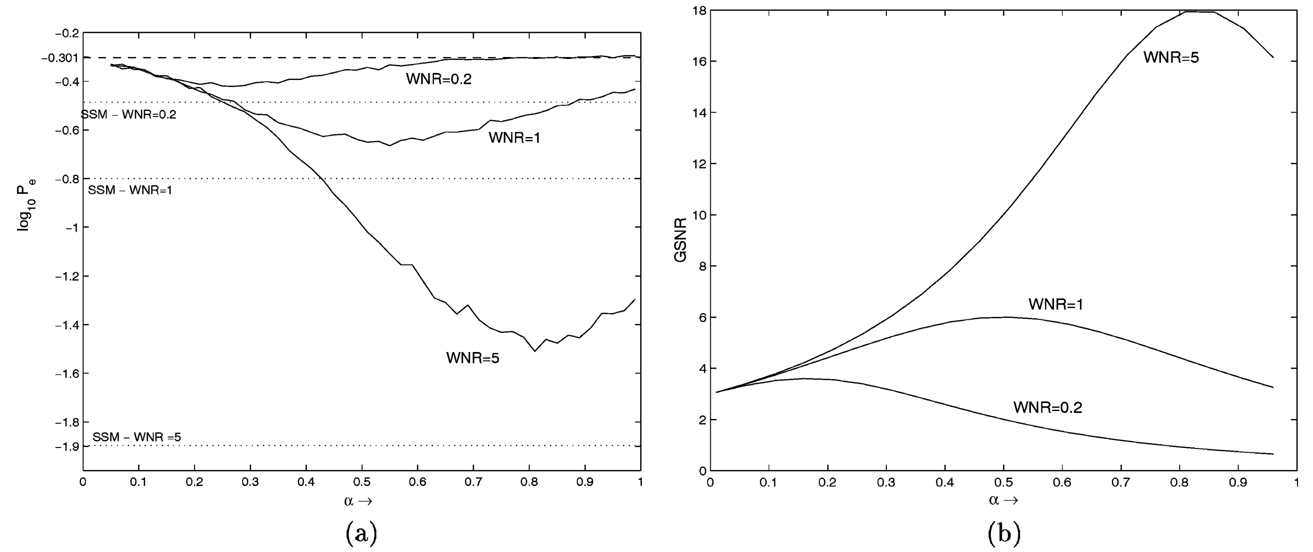


Fig. 15. Generalized SNR and probability of error P_e for binary detection based on one single sample. The variable on the horizontal axis is the tradeoff parameter α for QIM. For comparison, P_e for the nonblind and blind SSM schemes is given by the ordinate of the dotted horizontal lines. (a) P_e. (b) GSNR.

So the GSNR for detection is given by

$$\begin{aligned}\text{GSNR} &:= \frac{(\mu_1-\mu_0)^2}{\sigma_v^2}\\ &\approx \frac{(d_1-d_0)^2}{\frac{1}{12}(1-\alpha)^2\Delta^2+\alpha^2\sigma_{\tilde{w}}^2}\end{aligned} \qquad (6.16)$$

where $|d_1 - d_0| = \Delta/2$. The value of α that maximizes GSNR is given by a nonlinear equation. (Note that $\sigma_{\tilde{w}}^2$ is a decreasing function of Δ and tends to σ_w^2 if $\Delta \gg \alpha\sigma_w$.) A reasonable approximation for α that maximizes GSNR is

$$\alpha_{\max}\text{-GSNR} \approx \frac{\frac{\Delta^2}{12}}{\frac{\Delta^2}{12}+\sigma_w^2} = \frac{\text{WNR}}{\text{WNR}+1} \qquad (6.17)$$

whence $\max_\alpha \text{GSNR} \approx 3(\text{WNR}+1)$. The actual maximizing α is slightly lower than the right side of (6.17) because $\sigma_{\tilde{w}}^2 \geq \sigma_w^2$.

While GSNR is often useful as a rough measure of separation of the pdf's q_0 and q_1, it does not necessarily serve as an accurate predictor of detection performance. Fig. 15 plots GSNR and $\tilde{P}_e$ as a function of α, for three different values of WNR. Note that the optimal α is slightly different under the GSNR and $\tilde{P}_e$ criteria.

Quite interesting is the performance gap relative to nonblind watermarking, which bounds the performance of any blind watermarking scheme [71]. In this case the spread-spectrum scheme (6.2) yields an error probability $P_e = Q(\sqrt{\text{WNR}})$ which is typically smaller than the QIM error probabilities by a factor of two to three when WNR ranges from 0.2 to 5; see Fig. 15. The performance loss is quite small, considering that the QIM detector does not known the host signal.

C. *Binary Detection—Vector Case*

The previous two subsections have described the basic principle of a binning scheme and its benefits in terms of probability of error for binary detection based on a single observation. This subsection considers the more realistic case of N observations and studies two approximations to the probability of error.

Assume we have a host data vector $\mathbf{S} = \{S_1, S_2, \cdots, S_N\}$ and we mark each component S_i using the spread-spectrum and QIM techniques. Moreover,

(a) $\mathbf{S}$ is Gaussian with mean zero and covariance matrix $R_\mathbf{S}$;

(b) the marked signal $\mathbf{X}$ is corrupted by additive white Gaussian noise $\mathbf{W}$ with mean zero and variance σ_w^2.

1) SSM: For the spread-spectrum scheme, (6.2) generalizes to

$$\mathbf{x} = \begin{cases} \mathbf{s}+\mathbf{a}: & m=0\\ \mathbf{s}-\mathbf{a}: & m=1\end{cases} \qquad (6.18)$$

where the spread sequence $\mathbf{a}$ is known to the detector. For blind watermarking we have

$$p_0 = \mathcal{N}\left(\mathbf{a}, R_\mathbf{S}+\sigma_w^2 I_N\right), \quad p_1 = \mathcal{N}\left(-\mathbf{a}, R_\mathbf{s}+\sigma_w^2 I_N\right).$$

The LRT takes the form

$$\mathbf{a}^T\left(R_\mathbf{S}+\sigma_w^2 I_N\right)^{-1}\mathbf{y} - \frac{1}{2}\mathbf{a}^T\left(R_\mathbf{S}+\sigma_w^2 I_N\right)^{-1}\mathbf{a} \underset{H_0}{\overset{H_1}{\gtrless}} 0 \qquad (6.19)$$

and the probability of error of the test (6.19) is $P_e = Q(d/2)$, where $d^2 = 4\mathbf{a}^T(R_\mathbf{S}+\sigma_w^2 I_N)^{-1}\mathbf{a}$ is the GSNR for the detector.

For nonblind watermarking we have

$$p_0 = \mathcal{N}\left(\mathbf{a}, \sigma_w^2 I_N\right), \quad p_1 = \mathcal{N}\left(-\mathbf{a}, \sigma_w^2 I_N\right).$$

Then $P_e = Q(d/2)$ where $d^2 = \|2\mathbf{a}\|^2/\sigma_w^2 = 4\text{WNR}$.

2) Scalar QIM: For the scalar QIM scheme, let $\mathbf{d}_i = \{d_{i,1}, d_{i,2}, \cdots, d_{i,N}\}$ for $i = 0, 1$. We assume again that $d_{i,n} \in \{\pm(\Delta/4)\}$ and that the noise pdf $p_W(w)$ is symmetric around $w = 0$. Equation (5.6) generalizes to

$$\mathbf{x} = \begin{cases} Q_0(\alpha\mathbf{s}) + (1-\alpha)\mathbf{s} : & m = 0 \\ Q_1(\alpha\mathbf{s}) + (1-\alpha)\mathbf{s} : & m = 1 \end{cases} \tag{6.20}$$

where each Q_i is viewed as a vector quantizer, in this case simply a product of scalar quantizers

$$(Q_i(\mathbf{s}))_n = Q(s_n - d_{i,n}) + d_{i,n}, \quad 1 \le n \le N, i = 0, 1.$$

Without loss of generality, we shall assume $d_{0,n} \equiv \Delta/4$ and $d_{1,n} \equiv 3\Delta/4$.

The first step at the receiver is to compute the transformed data

$$\tilde{Y}_n = \alpha Y_n \bmod \Delta, \quad 1 \le n \le N. \tag{6.21}$$

Under the uniform quantization noise model, the preprocessed data $\{\tilde{Y}_n, 1 \le n \le N\}$ are mutually independent, even though there may be dependencies between the host signal samples $\{S_n\}$. The detector must decide between the two hypotheses

$$\begin{cases} H_0 : \tilde{\mathbf{Y}} = \mathbf{d}_0 + \mathbf{V} \\ H_1 : \tilde{\mathbf{Y}} = \mathbf{d}_1 + \mathbf{V} \end{cases} \tag{6.22}$$

where the samples V_n, $1 \le n \le N$, are i.i.d. with pdf p_{V} given in (6.11). The addition is modΔ (componentwise). The ML detector based on $\tilde{\mathbf{Y}}$ and the statistical model above is

$$\tilde{L}(\tilde{\mathbf{y}}) = \prod_{n=1}^{N} \frac{p_{\text{V}}(\tilde{y}_n - d_{1,n})}{p_{\text{V}}(\tilde{y}_n - d_{0,n})} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{6.23}$$

which coincides with the nearest-neighbor detector (6.24) in some cases.

Similarly to (6.5), the minimum-distance detection rule may be written in the form

$$\hat{m} = \underset{m \in \{0,1\}}{\operatorname{argmin}} \|\tilde{\mathbf{y}} - \mathbf{d}_m\|. \tag{6.24}$$

The probability of error is given by

$$\tilde{P}_e = \frac{1}{2} \int_{[-\frac{\Delta}{2}, \frac{\Delta}{2}]^N} \min\left(q_0^N(\tilde{\mathbf{y}}), q_1^N(\tilde{\mathbf{y}})\right) d\tilde{\mathbf{y}}. \tag{6.25}$$

It may in principle be computed numerically, using integration over the N-dimensional cube $[0, \Delta]^N$. Unfortunately such methods are impractical even for relatively small N. Monte-Carlo simulations are an alternative, but are time-consuming and do not necessarily provide analytical insights. Two analytic methods for approximating $\tilde{P}_e$ are considered next.

3) Gaussian Approximation: One may easily derive the GSNR at the detector, as was done in Section VI-B. Formula (6.16) generalizes to

$$\begin{aligned} \text{GSNR} &\approx \frac{\|\mathbf{d}_1 - \mathbf{d}_0\|^2}{\frac{1}{12}(1-\alpha)^2\Delta^2 + \alpha^2\sigma_{\tilde{w}}^2} \\ &= \frac{\frac{N\Delta^2}{4}}{\frac{1}{12}(1-\alpha)^2\Delta^2 + \alpha^2\sigma_{\tilde{w}}^2} \end{aligned} \tag{6.26}$$

If the noise V was Gaussian, the probability of error would be given by

$$\tilde{P}_e = Q\left(\frac{\sqrt{\text{GSNR}}}{2}\right). \tag{6.27}$$

However V is non-Gaussian, and (6.27) is generally a poor approximation to the actual $\tilde{P}_e$.

4) Large Deviations: If GSNR is large (as is always the case for sufficiently large N), the performance of the detection test is dominated by rare events (as described by the tails of the pdf's q_0^N and q_1^N) and Gaussian approximations of these tails are usually severely inaccurate. The usual approach to such problems in the detection literature is based on large deviations theory, as discussed in Appendix C. For any N, we have $\tilde{P}_e \le (1/2)e^{-NB(q_0,q_1)}$, where

$$B(q_0, q_1) = -\ln \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \sqrt{q_0(\tilde{y})q_1(\tilde{y})}d\tilde{y} \tag{6.28}$$

is the Bhattacharyya distance between the pdf's q_0 and q_1. The bound is tight in the exponent:[17]

$$\lim_{N \to \infty} \left[-\frac{1}{N} \ln \tilde{P}_e\right] = B(q_0, q_1).$$

Hence $B(q_0, q_1)$ is a more useful predictor of detection performance than is GSNR, and is simple to compute as well.

The Bhattacharyya coefficient $B(q_0, q_1)$ depends on the QIM parameter α via q_0 and q_1. The log probability of error when $N = 15$ is shown in Fig. 16 as a function of α, along with the Bhattacharyya and Gaussian approximations.

[17] In general, a Chernoff bound with optimal Chernoff exponent is tight. However, due to the symmetry of p_{v} and the fact that q_0 and q_1 are translates of p_{v}, the optimal Chernoff exponent is 1/2, and thus the optimal bound is the Bhattacharyya bound.

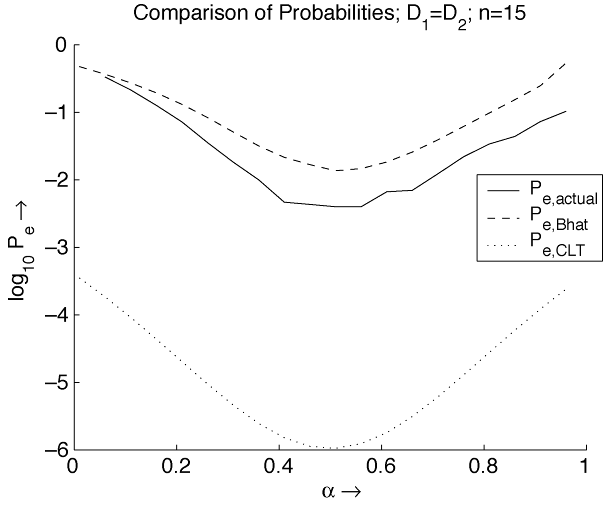


Fig. 16. $\tilde{P}_e$ and its upper bound based on the Bhattacharyya coefficient $B(q_0, q_1)$ for binary detection based on $N = 15$ samples. Also shown is the Gaussian approximation to $\tilde{P}_e$, which is overoptimistic by several orders of magnitude. The variable on the horizontal axis is the QIM tradeoff parameter α.

The Bhattacharyya approximation is quite good, unlike the Gaussian approximation which is off by several orders of magnitude.

D. Multiple Codewords and Lattice QIM

In the case of $|\mathcal{M}| > 2$, calculation of the probability of error

$$P_e = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} Pr[\mathbf{Y} \notin \mathcal{Y}_m | m \text{ sent}]$$

presents difficulties if $\mathcal{M}$ is large. Fortunately, useful bounds on P_e can be derived. We consider the general case of lattice QIM; note that scalar QIM is optimal when $|\mathcal{M}| = 2$ [71].

Assume equally likely codewords. For linear codes, the conditional error probability $Pr[\mathbf{Y} \notin \mathcal{Y}_m | m \text{ sent}]$ is independent of the message m that was sent. Thus we may arbitrarily select message $m = 0$ and write

$$P_e = Pr[\mathbf{Y} \notin \mathcal{Y}_0 | m = 0].$$

A useful upper bound on P_e can sometimes be obtained using the union bound [38]

$$P_e \leq (|\mathcal{M}| - 1) \max_{i \neq j \in \mathcal{M}} P_{e|i,j} \qquad (6.29)$$

where

$$P_{e|i,j} = \frac{1}{2} \int \min\left[p(\mathbf{y}|i \text{ sent}), p(\mathbf{y}|j \text{ sent})\right] d\mathbf{y}$$

is the probability of error for a binary test between hypotheses $m = i$ and $m = j$. The union bound is typically useful at low bit rates.

Example: Consider a scalar QIM system in which the codewords $\mathbf{d}_i$, $i \in \mathcal{M}$ are designed with letters $d_{i,n} \in \{\pm(\Delta/4)\}$ for all $i \in \mathcal{M}$ and $n \in \{1, 2, \cdots, N\}$. Let the message set have cardinality $|\mathcal{M}| = 2^k$, where $k < N$, and the code be a (N, k, d_H) linear code. Thus, any two codewords differ in at least d_H positions. The worst codeword pairs are the ones that differ only in d_H positions. The Bhattacharyya distance between such pairs is $B_{\min} = d_H B^*$, where $B^* \triangleq B(q_0, q_1)$ is given in (6.28). We obtain

$$P_e \leq (2^k - 1) e^{-d_H B^*}.$$

Given N and B^*, this upper bound quantifies the tradeoff between rate $R = k/N$ and achievable probability of error; given k, codes with large d_H are clearly desirable. Fig. 17 displays the Bhattacharyya bound on P_e as a function of k for the best known codes of length $N = 256$.

In the case where $|\mathcal{M}|$ grows exponentially with N, the union bound (6.29) may be loose; if

$$R = \frac{k}{N} \geq \frac{d_H}{N} B^*$$

the union bound becomes trivial (≥ 1), and the notion of minimum distance is less relevant. Finding better bounds in this case is a topic of current research [72]–[74].

Consider the L-dimensional nested lattice code $\mathcal{C} = \Lambda_f / \Lambda_c$ in (5.16). Recall that $\mathcal{V}$ and Q are respectively the Voronoi cell and lattice quantizer associated with the coarse lattice Λ_c, and that Λ_i, $0 \leq i < |\det J|$ are the cosets of Λ_c, with associated coset leaders $\mathbf{d}_i \in \mathcal{V}$ playing the role of L-dimensional dither vectors. A different dither vector is potentially selected for each length-L host-data block. For simplicity we first consider the case $N = L$ (one single data block).

Case $N = L$. Write

$$Q_i(\mathbf{s}) = Q(\mathbf{s} - \mathbf{d}_i) + \mathbf{d}_i, \quad 0 \leq i < |\det J| = |\mathcal{M}|.$$

Under high-rate lattice quantization theory [75], the quantization noise $Q(\mathbf{s}) - \mathbf{s}$ may be modeled as random, independent of $\mathbf{S}$, and uniformly distributed over $\mathcal{V}$; moreover, that model is exact if a dithered lattice quantizer is used [76], [77]. The embedding distortion per sample is given by

$$D_1 = \frac{1}{L} \frac{1}{\text{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}. \qquad (6.30)$$

For the hexagonal A_2 lattice, the minimum distance between lattice points (twice the inradius of $\mathcal{V}$) is given by

$$\Delta = \sqrt{\frac{72}{5} D_1}. \qquad (6.31)$$

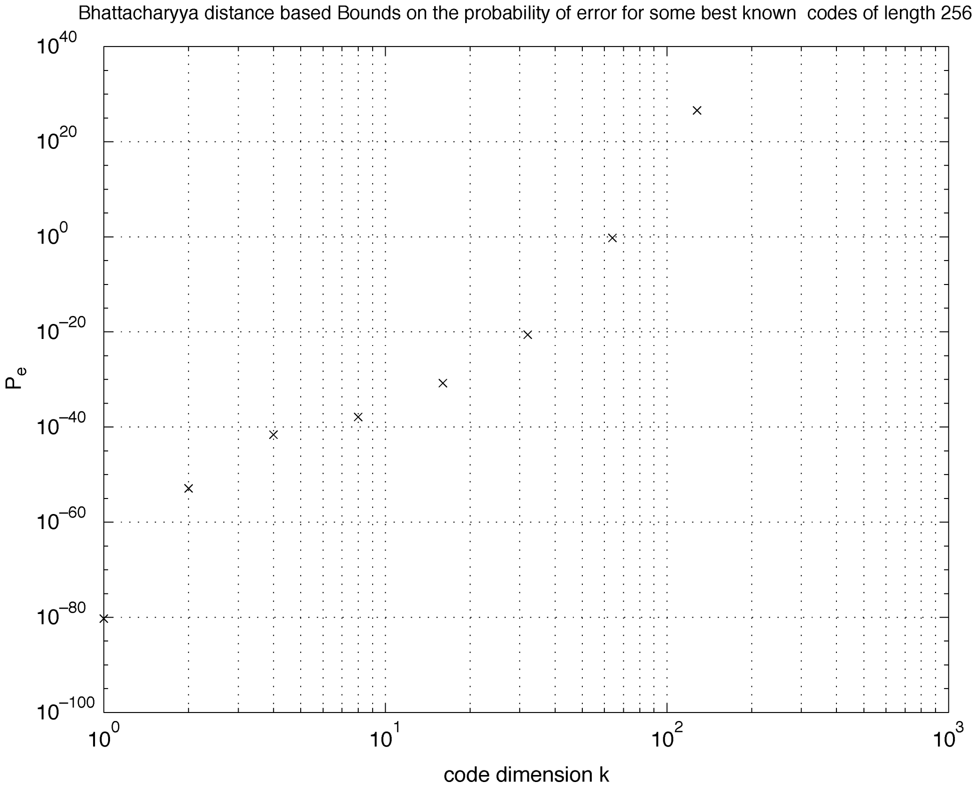


Fig. 17. Bhattacharyya bound on log probability of error, $\log_{10} \tilde{P}_e$ versus number of information bits, $k = \log_2 |\mathcal{M}|$. In this experiment, $D_1/D_2 = 4$ dB and $N = 256$. The figure shows representative points for the best known codes [256,128,38], [256,64,62], [256,32,96], [256,16,113], [256,8,128], [256,4,136], [256,2,170], [256,1,256].

The receiver first implements the modulo lattice operation

$$\tilde{\mathbf{Y}} = \alpha \mathbf{Y} \bmod \Lambda \triangleq \alpha \mathbf{Y} - Q(\alpha \mathbf{Y}).$$

Assuming that message i was sent, the processed vector $\tilde{\mathbf{Y}}$ may be viewed as the output of a MAN channel (Fig. 14) with input $\mathbf{d}_i$ and noise

$$\mathbf{V} = \tilde{\mathbf{E}} + \tilde{\mathbf{W}} \bmod \Lambda, \tag{6.32}$$

analogously to (6.10) in the scalar QIM case. Here $\mathbf{V} \in \mathcal{V}$, the self-noise $\tilde{\mathbf{E}}$ is uniformly distributed over the scaled Voronoi cell $(1-\alpha)\mathcal{V}$, and the aliased attacker's noise is given by $\tilde{\mathbf{W}} = \alpha \mathbf{W} \bmod \Lambda$. If dither vector $\mathbf{d}_i$ is embedded, $\tilde{\mathbf{Y}}$ follows the distribution

$$q_i(\tilde{\mathbf{y}}) = p_{\mathbf{V}}(\tilde{\mathbf{y}} - \mathbf{d}_i).$$

The receiver decides between the statistical hypotheses

$$H_i : \tilde{\mathbf{Y}} \sim q_i, \quad 0 \le i < |\det J| = |\mathcal{M}|. \tag{6.33}$$

Letting $B(q_i, q_j) = -\ln \int_{\mathcal{V}} \sqrt{q_i q_j}$ be the Bhattacharyya distance between q_i and q_j, it follows from the union bound that

$$P_e \le (|\mathcal{M}| - 1)\, e^{-B^*}$$

where $B^* \triangleq \min_{i \neq j} B(q_i, q_j)$.

Case $N > L$. Denoting by $\mathbf{Y}_n \in \mathbb{R}^L$ the n-th block of received data, the receiver first implements the modulo lattice operation

$$\tilde{\mathbf{Y}}_n = \alpha \mathbf{Y}_n \bmod \Lambda, \quad 1 \le n \le \frac{N}{L}. \tag{6.34}$$

The vectors $\tilde{\mathbf{Y}}_n$ are mutually independent because the noise process W is assumed to be white. Message $m \in \mathcal{M}$ is represented using dither vectors $\mathbf{d}_{i(m,n)}$, with associated pdf's $q_{i(m,n)}$ for $\tilde{\mathbf{Y}}_n$ at the receiver, where $0 \le i < |\det J|$ and $1 \le n \le N/L$. The receiver decides between the $|\mathcal{M}|$ hypotheses

$$H_m : \tilde{\mathbf{Y}}_n \sim q_{i(m,n)}, \quad 1 \le n \le \frac{N}{L}, \quad m \in \mathcal{M}. \tag{6.35}$$

The Bhattacharyya distance between the pdf's associated with hypotheses m and m' is given by

$$B(m, m') = \sum_{n=1}^{\frac{N}{L}} B\left(q_{i(m,n)}, q_{i(m',n)}\right).$$

Equivalently, if we let N_{ij} be the number of n's such that $i(m,n) = i$ and $i(m',n) = j$, we can write

$$B(m, m') = \sum_{i,j=0}^{|\det J|-1} N_{ij} B(q_i, q_j)$$

where $\sum_{i,j} N_{ij} = N/L$.

A possible code construction is the following. Select a $((N/L), k, d)|\det J|$-ary code. Then

$$B(m, m') \geq dB^*$$

where $B^* = \min_{i \neq j} B(q_i, q_j)$. We obtain

$$P_e \leq \left(|\det J|^k - 1\right) e^{-dB^*}.$$

Example: Consider the case $|\mathcal{M}| = |\det J| = 3$ and $L = 2$ using the hexagonal lattice of Fig. 9(e). To encode message m, we choose $i(m, n) = m$, i.e., we use a repetition code and embed the same dither vector $\mathbf{d}_m$ in each length-2 block. Let $D_1 = D_2 = 1$. From (6.31), we obtain $\Delta = \sqrt{72/5}$. Choose

$$\mathbf{d}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{d}_1 = \begin{pmatrix} \frac{\Delta}{2} \\ \frac{\Delta}{(2\sqrt{3})} \end{pmatrix}, \quad \mathbf{d}_2 = \begin{pmatrix} 0 \\ \frac{\Delta}{\sqrt{3}} \end{pmatrix}.$$

These dither vectors are equidistant: $\|\mathbf{d}_i - \mathbf{d}_j\| = \Delta/\sqrt{3}$ for all $i \neq j$. We also have symmetry between the Bhattacharyya distances: $B(q_i, q_j) = B^*$ for all $i \neq j$.

E. Shaping Gain

The traditional tradeoff in source coding is rate versus distortion. For high-rate lattice quantization, distortion is the second-order moment of the Voronoi cell $\mathcal{V}$, and rate is a linear function of $\log|\mathcal{V}|$. The optimal tradeoffs are obtained using nearly spherical lattices, in the sense that the normalized second moment

$$G(\Lambda) = D_1 |\mathcal{V}|^{-\frac{2}{L}}. \tag{6.36}$$

of the lattice approaches the lower bound, $1/2\pi e \approx 0.0586$ [78].[18]

In data hiding, distortion is measured the same way as in source coding, but the rate of interest is $R = (1/N)\log|\det J|$. The second-order moment of the coarse lattice is determined by the embedding distortion D_1. The attacker's noise pdf is assumed to be spherically symmetric, in which case the ideal decoding regions are spherical. Assuming $R > 0$, the ideal shape for the Voronoi cells of the fine and coarse lattices is spherical because this geometry maximizes the density of decoding regions in $\mathcal{V}$, the Voronoi cell for the coarse lattice. Hence this geometry maximizes rate as well.

Practical Codes. A folk theorem in coding theory is that *almost all random linear codes are good, but only a few nonrandom codes are good.* For large dimensions N, random linear codes provide (in a probabilistic sense) the ideal spherical geometry discussed above; unfortunately such codes lack structure and are prohibitively hard to decode. Structured linear codes are practical, but it is hard to find good ones.

VII. Capacity

After analyzing probability of decoding error for binning schemes, we turn our attention to a closely related problem, namely what is the maximal rate of a code that allows reliable transmission ($P_e \to 0$ as $N \to \infty$). In other words, we wish to determine a Shannon capacity for data hiding [34].

We assume that the key is a sequence $\mathbf{k}$ of random variables defined over an alphabet $\mathcal{K}$. Furthermore, (S_i, K_i), $1 \leq i \leq N$ are i.i.d. with pmf $p(s, k)$. This model accounts for the possibility of signal-dependent keys. In nonblind data hiding, S is a function of K.

The rate of the data-hiding code $(\mathcal{M}, f, g)$ is $R = (1/N)\log|\mathcal{M}|$, and the average probability of error is

$$P_{e,N} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} Pr\left[g(\mathbf{Y}, \mathbf{K}) \neq m | M = m\right]. \tag{7.1}$$

A rate R is said to be achievable for distortion D_1 and for a class of attack channels $\mathcal{P}_{\mathbf{Y}|\mathbf{X}}$, $N \geq 1$, if there is a sequence of codes subject to distortion D_1, with rate R, such that $P_{e,N} \to 0$ as $N \to \infty$, for any sequence of attacks in $\mathcal{P}_{\mathbf{Y}|\mathbf{X}}$. The *data-hiding capacity* $C(D_1, \{\mathcal{P}_{\mathbf{Y}|\mathbf{X}}\})$ is then defined as the supremum of all achievable rates for distortion D_1 and attacks in the class $\mathcal{P}_{\mathbf{Y}|\mathbf{X}}$, $N \geq 1$.

Gel'fand-Pinsker. The data-hiding problem is closely related to a fundamental problem of communication with side information studied by Gel'fand and Pinsker [28] in 1980. They derived the capacity of a memoryless channel whose state is known to the encoder but not to the decoder. The encoder may exploit the state information using a binning technique, as discussed below. The role of the channel state is analogous to the role of the host signal in blind data hiding. Key differences with the Gel'fand-Pinsker problem include the existence of distortion constraints, the availability of different amounts of side information to the encoder, attacker, and decoder, and the fact that the attack channel is unknown to the encoder.

First we state the fundamental capacity result for discrete alphabets $\mathcal{S}$, $\mathcal{X}$ and $\mathcal{Y}$ and relate it to the Gel'fand-Pinsker result. Then we consider the case of continuous alphabets (where S, X and Y are real-valued.)

A. Finite Alphabets

For simplicity of the exposition, consider the average distortion constraints (2.5) and (2.7), and assume the host signal and the attack channel are memoryless. Then

$$p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N} A(y_i|x_i). \tag{7.2}$$

The data-hiding capacity defined above turns out to be the solution of a certain mutual-information game and is given in the theorem below. Let $U \in \mathcal{U}$ be an auxiliary random

[18] Cubic and hexagonal lattices respectively achieve $G(\Lambda) = 1/12 \approx 0.0833$ and $G(\Lambda) = 5/36\sqrt{3} \approx 0.0802$ [78, Sec. 3.3].

variable such that $(U, S) \to X \to Y$ forms a Markov chain. Let $\mathcal{Q}(D_1)$ be the set of *covert channels* Q that satisfy the constraint

$$\sum_{x,s,k,u} d(s,x)Q(x,u|s,k)p(s,k) \leq D_1, \tag{7.3}$$

$\mathcal{A}(D_2)$ be the set of attack channels A that satisfy the constraint

$$\sum_{s,x,k,y} d(x,y)A(y|x)p(x|s,k)p(s,k) \leq D_2, \tag{7.4}$$

and $\mathcal{A}$ be an arbitrary subset of $\mathcal{A}(D_2)$.

Theorem 7.1: [34] Assume the attacker knows the encoding function f and the decoder knows f and the attack channel A. A rate R is achievable for distortion D_1 and attacks in the class $\mathcal{A}$ if and only if $R < C$, where C is given by

$$C := C(D_1, \mathcal{A}) = \max_{Q(x,u|s,k) \in \mathcal{Q}(D_1)} \min_{A(y|x) \in \mathcal{A}} J(Q, A) \tag{7.5}$$

where $|\mathcal{U}| \leq |\mathcal{X}||\Omega| + 1$, Ω is the support set of $p(s,k)$, and

$$J(Q,A) = I(U;Y|K) - I(U;S|K) \tag{7.6}$$

where

$$I(X;Y|Z) \triangleq \sum_{x,y,z} p(x,y,z) \log \frac{(p(x,y|z)}{(p(x|z)p(y|z)))}$$

denotes conditional mutual information [30].

Key differences between the capacity result (7.5) and the Gel'fand-Pinsker problem include the existence of distortion constraints, the availability of $\mathbf{K}$ at both the encoder and decoder, and the fact that the attack channel is unknown to the encoder—whence the minimization over A in (7.5).

Example: Bernoulli-Hamming case: The capacity formula (7.5) can be evaluated in closed form for a few simple problems. One of these is the case of binary alphabets: $\mathcal{S} = \mathcal{X} = \mathcal{Y} = \{0, 1\}$ and Hamming distortion constraints D_1 and D_2 for the embedder and attacker, respectively. As expected, capacity is strictly higher for nonblind watermarking relative to blind watermarking. Capacity for nonblind watermarking is given by [34]

$$C^{\text{priv}} = h(D_1 \star D_2) - h(D_2) \tag{7.7}$$

where $D_1 \star D_2 = D_1(1 - D_2) + (1 - D_1)D_2$. Capacity for blind watermarking is given by [79]

$$C^{\text{pub}} = \begin{cases} \frac{D_1}{\delta_2}\left[h(\delta_2) - h(D_2)\right], & \text{if } 0 \leq D_1 < \delta_2; \\ h(\delta_2) - h(D_2), & \text{if } \delta_2 \leq D_1 \leq \frac{1}{2}; \\ 1 - h(D_2), & \text{if } D_1 > \frac{1}{2}, \end{cases} \tag{7.8}$$

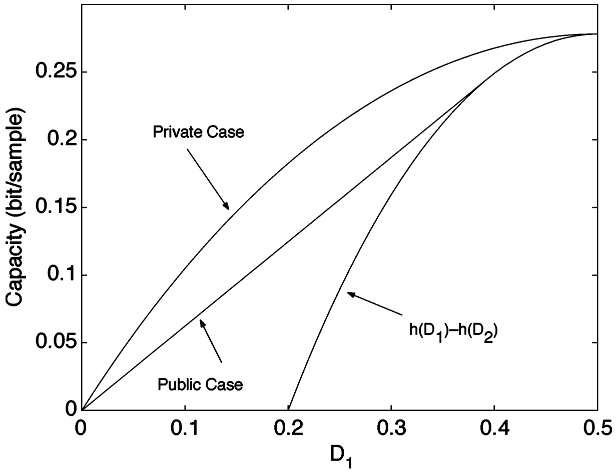


Fig. 18. Capacity functions for Bernoulli-Hamming problem when $D_2 = 0.2$.

where $\delta_2 = 1 - 2^{-h(D_2)}$ and $h(p) = -p\log_2 p - (1 - p)\log_2(1-p)$ is the binary entropy function. The straight-line portion of the capacity function is achieved by time-sharing. See Fig. 18. In both cases, the worst attack is a binary symmetric channel (BSC) with crossover probability D_2. The capacity formula (7.8) was derived in [80] and [81] under the assumption of a fixed attack channel.

B. Random Binning

In principle, the capacity bound can be approached using a *random binning* coding technique [28], [30], which exemplifies the role of the covert channel Q; see Fig. 19. A size--$2^{N(I(U;Y,K)-\epsilon)}$ codebook $\mathcal{C}$ is constructed for the variable $\mathbf{U}$ by randomly sampling the capacity-achieving distribution $p(\mathbf{u})$, and partitioning the samples into $|\mathcal{M}|$ equal-size subsets (lists). The actual embedding of a message $m \in \mathcal{M}$ proceeds as follows: first identify an element $\mathbf{u}(m)$ from the list of elements indexed by m in the codebook $\mathcal{C}$, in such a way that $\mathbf{u}(m)$ is statistically typical with the current $(\mathbf{s}, \mathbf{k})$, then generate watermarked data $\mathbf{x}$ according to the pmf $p(\mathbf{x}|\mathbf{u}(m), \mathbf{s}, \mathbf{k})$. The decoder finds $\hat{\mathbf{u}}$ that is statistically typical with $(\mathbf{y}, \mathbf{k})$, and obtains $\hat{m}$ as the index of the list to which $\hat{\mathbf{u}}$ belongs. However, memory and computational requirements grow exponentially with block length N, and so such approaches are known to be infeasible in practice. Developing structured binning schemes that approach the capacity bound is an active research area [33], [61], [64]–[66], [82]–[84]. This problem is closely related to the problem of developing good nested lattice codes in Euclidean spaces which was introduced in Section V-C and will be further developed in Section VIII. For each m, the mapping from $\mathcal{S}^N$ to the list of vectors $\mathbf{u}$ indexed by m may be thought of as a generalized VQ mapping.

C. Gaussian Channels

Theorem 7.1 can be generalized to the case of infinite alphabets $\mathcal{S}$, $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{U}$, $\mathcal{K}$. The case of Gaussian S and squared-error distortion measure is of considerable practical and theoretical interest, as it becomes possible to explicitly compute the distributions that achieve capacity, leading

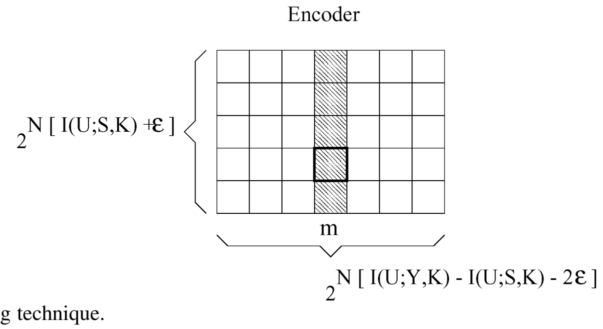

Encoder
Decoder
$2^{N[I(U;S,K)+\varepsilon]}$
m
$2^{N[I(U;Y,K)-I(U;S,K)-2\varepsilon]}$
m

Fig. 19. Random binning technique.

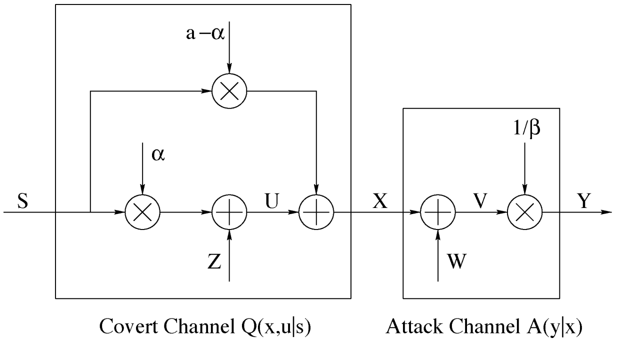


Fig. 20. Optimal data-hiding and attack strategies for Gaussian host data $S \sim \mathcal{N}(0, \sigma^2)$. Here $Z \sim \mathcal{N}(0, aD_1)$ and $W \sim \mathcal{N}(0, \beta(D_2 - D_1))$ are mutually independent random variables, where $a = 1 - D_1/\sigma^2$ and $\beta = \sigma^2/(\sigma^2 - D_2)$. The optimal channels $p(x|s)$ and $A(y|x)$ are Gaussian test channels with distortion levels D_1 and $D_2 - D_1$, respectively. For blind data hiding, $\alpha = aD_1/(aD_1 + D)$; for nonblind data hiding, one may choose $\alpha = a$.

to insightful results. We refer to this case as the Gaussian channel. Let $\mathcal{S} = \mathcal{X} = \mathcal{Y}$ be the set $\mathbb{R}$ of real numbers, and $d(x, y) = (x - y)^2$ be the squared-error metric. Also let $S \sim \mathcal{N}(0, \sigma^2)$, meaning that S follows a Gaussian distribution with mean zero and variance σ^2. Assume as in (7.2) that the attack channel is memoryless.

A remarkable result is that *the data-hiding capacity is the same for both blind and nonblind data-hiding problems.* Under the average distortion constraints (2.5) and (2.8), we obtain [88]

$$\begin{aligned} C &= C_G(\sigma^2, D_1, D_2) \\ &\triangleq \begin{cases} \frac{1}{2}\log\left(1 + \frac{D_1}{D}\right) : & \text{if } D_1 \le D_2 < \sigma^2, \\ 0 \quad : & \text{if } D_2 \ge \sigma^2 \end{cases} \end{aligned} \tag{7.9}$$

where $D \triangleq \sigma^2(D_2 - D_1)/(\sigma^2 - D_2)$. When $D_2 < \sigma^2$, the optimal distributions turn out to be Gaussian test channels [30], [46], [88]; see Fig. 20.

Closely related to this result is one derived by Costa [29] in 1983 for communications on an additive white Gaussian noise channel (with power D_2) in the presence of an i.i.d. Gaussian interference (with power σ^2) that is known at the encoder but not at the decoder. When the channel input power is constrained not to exceed D_1, Costa showed that the capacity of the channel is exactly the same as if the interference was also known to the decoder

$$C = \frac{1}{2}\log\left(1 + \frac{D_1}{D_2}\right).$$

The analogy to the data-hiding problem is remarkable: the host signal $\mathbf{S}$ plays the role of the known interference. Capacity in the data-hiding problem is slightly lower than in the Costa problem because the optimal Gaussian attack is not additive; however, the gap vanishes in the low-distortion limit ($D_1/\sigma^2 \to 0$ and $D_2/\sigma^2 \to 0$). In this case, we have

$$\alpha = \frac{\text{WNR}}{1 + \text{WNR}} \tag{7.10}$$

which admits an elegant MMSE (minimum mean squared error) interpretation [85]; also see (6.17).

Additional extensions of Costa's result have recently appeared [65], [86], [87]. In particular, the capacity formula $C = (1/2)\log(1 + (D_1/D_2))$ is still valid if the interference $\mathbf{S}$ is *any finite-power sequence*, for *any* values of D_1 and D_2. Also, the capacity for the following two data-hiding games are identical: (i) the game with average distortion constraint (2.7) and memoryless attack channel, known to the decoder, and (ii) the game subject to the maximum-distortion constraint (2.6) with a decoder uninformed about the attack channel [86].

The optimal decoding rule for Fig. 20 is a minimum-distance decoding rule

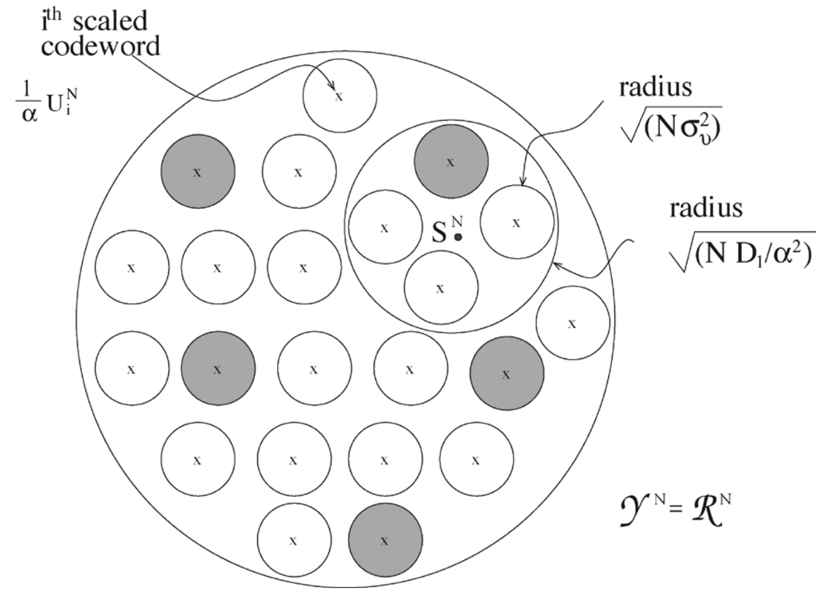
$$\hat{\mathbf{u}} = \operatorname*{argmax}_{\mathbf{u} \in \mathcal{C}} p(\mathbf{u}|\mathbf{y}) = \operatorname*{argmin}_{\mathbf{u} \in \mathcal{C}} \|\mathbf{u} - \gamma\mathbf{y}\|^2 \tag{7.11}$$

where $\gamma \sim \alpha$ as $D_1/\sigma^2 \to 0$ and $D_2/\sigma^2 \to 0$. For large N, we have $\|\mathbf{u}\|^2 \sim N\sigma_u^2$, and (7.11) is asymptotically equivalent to a correlation rule

$$\hat{\mathbf{u}} \sim \operatorname*{argmax}_{\mathbf{u} \in \mathcal{C}} \mathbf{u}^T\mathbf{y}. \tag{7.12}$$

This rule is remarkable in its simplicity and robustness. For instance (7.12) is also optimal if the attacker is allowed to scale the output of the Gaussian channel by an arbitrary factor, because all correlations are scaled by the same factor. Also (7.12) turns out to be the optimal universal decoding rule in Cohen and Lapidoth's setup [86]; see Section VII-E.

The property that capacity is the same whether or not $\mathbf{S}$ is known at the decoder is illustrated in Fig. 21 using sphere-packing arguments. Assume that $D_1, D_2 \ll \sigma^2$. With overwhelming probability, the scaled codewords $(1/\alpha)\mathbf{U}$ live in



i^{th} scaled codeword
$\frac{1}{\alpha}\mathrm{U}_{\mathrm{i}}^{\mathrm{N}}$
radius $\sqrt{(\mathrm{N}\sigma_{\upsilon}^2)}$
radius $\sqrt{(\mathrm{N}\,\mathrm{D}_1/\alpha^2)}$
S^{N}
x
$\mathcal{Y}^{\mathrm{N}} = \mathcal{R}^{\mathrm{N}}$

Fig. 21. Sphere-packing interpretation of blind Gaussian information hiding. Shaded spheres are indexed by the same message m.

a large sphere of radius $\sqrt{N\sigma^2(1+\epsilon)}$ centered at 0. The encoder in the random binning construction selects a scaled codeword $(1/\alpha)\mathbf{U}$ inside the medium-size sphere of radius $\sqrt{ND_1/\alpha^2}$ centered at $\mathbf{S}$.[19] There are approximately 2^{NC} codewords (one for each possible message m) within this medium-size sphere. The received data vector $\mathbf{Y}$ lies within a small sphere of radius $\sqrt{N\sigma_v^2}$ centered at $(1/\alpha)\mathbf{U}$. Decoding by joint typicality means decoding $\mathbf{Y}$ to the center of the closest small sphere. To yield a vanishing probability of error, the small spheres should have statistically negligible overlap. The number of distinguishable messages, 2^{NC}, is independent of the size of the large sphere $(N\sigma^2)$.

D. Parallel Gaussian Channels

Real-world signals such as images do not follow i.i.d. Gaussian models; however they can be decomposed into approximately independent Gaussian components [46]. Data-hiding capacity can be evaluated by solving a certain power-allocation problem, as described below.

Assume $\mathbf{S}$ is a collection of n_P independent sources S_i, $1 \le i \le n_P$, each producing N_i i.i.d. Gaussian random variables from the distribution $\mathcal{N}(0, \sigma_i^2)$, where $\sum_{i=1}^{n_P} N_i = N$. Thus, we have n_P parallel Gaussian channels, with samples $\{S_i(n))\}$, and rates $r_i = N_i/N$, $1 \le i \le n_P$. The distortion metric is squared error. Let

$$d_{1i} = \frac{1}{N_i}\sum_{n=1}^{N_i} \mathbb{E}\left[X_i(n) - S_i(n)\right]^2 \quad \text{and}$$
$$d_{2i} = \frac{1}{N_i}\sum_{n=1}^{N_i} \mathbb{E}\left[Y_i(n) - S_i(n)\right]^2 \qquad (7.13)$$

be the distortions introduced by the embedder and the attacker in channel i, respectively. We have distortion constraints

$$\sum_{i=1}^{n_P} r_i d_{1i} \le D_1 \quad \text{and} \quad \sum_{i=1}^{n_P} r_i d_{2i} \le D_2. \qquad (7.14)$$

As in the Gaussian case, capacity is the same for both blind and nonblind data hiding [46], [88]

$$C = \max_{\{d_{1i}\}} \min_{\{d_{2i}\}} \sum_{i=1}^{n_P} r_i C_G\left(\sigma_i^2, d_{1i}, d_{2i}\right) \qquad (7.15)$$

where the maximization and minimization over power allocations are subject to the distortion constraints (7.14). The capacity-achieving distributions are product distributions, i.e., the n_P channels are decoupled. The distributions in each channel take the form of Fig. 20, where the weights a, α and β depend on the channel index i. We may therefore think of the weights a_i and α_i as optimal host signal preprocessing filters for the embedder, and of β_i as an optimal attack filter.

In many signal processing problems, the appropriate distortion metric is not squared error but weighted squared error: $d(\mathbf{s}, \mathbf{x}) = \sum_{i=1}^{n_P} w_i \sum_{n=1}^{N_i}(x_k(n) - s_k(n))^2$, where w_i are nonnegative weights [41]. For instance, $w_i = 0$ if channel i is perceptually irrelevant. The ordinary squared error metric is obtained by choosing $w_i \equiv 1$. Under the weighted squared error metric, capacity is still given by (7.15), but with $w_i\sigma_i^2$ in place of σ_i^2 [148].

In some problems, the host signal components may be coarsely classified into two categories: significant ones ($\sigma_i^2 \gg D_1$, D_2) and insignificant ones ($\sigma_i^2 \ll D_1$, D_2). In

[19] Again, this selection step may be thought of as a VQ mapping.

this case the capacity expression (7.15) reduces to a much simpler formula

$$C = \frac{\rho}{2}\log\left(1 + \frac{D_1}{D_2}\right) \tag{7.16}$$

where $\rho \leq 1$ is the fraction of significant components in the host signal. This result is consistent with the intuition that for a data-hiding code to be robust, information should be embedded in perceptually significant components of the host signal [50].

E. *Attack Channels With Memory*

Recently, Somekh-Baruch and Merhav [89], [90] have shown that the capacity formula (7.5) holds under milder assumptions on the attacks and decoder. They assume the maximum-distortion constraints (2.4) and (2.6). The decoder does not know the attack channel $p_{\mathbf{Y}|\mathbf{X}}$, which is any channel that satisfies (2.6). Therefore $p_{\mathbf{Y}|\mathbf{X}}$ has arbitrary memory. The key alphabet $\mathcal{K}$ is allowed to be unbounded.

Capacity can again be achieved using a random binning scheme closely related to the one described above, and a particular universal decoder based on the method of types [30], [91]. This decoder evaluates the empirical first-order statistics of the pairs $(\mathbf{u}, \mathbf{y})$, for all possible codewords $\mathbf{u} \in \mathcal{C}$. The binning scheme is such that the probability distribution for $\mathbf{X}$ (*averaged over* $\mathbf{k}$) is memoryless: $p_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{N} p_X(x_i)$, where p_X is obtained from the optimal covert channel Q. (If the key space $\mathcal{K}$ were too constrained, this memoryless property could not be obtained, and there would be dependencies between $\{x_i\}$.) Loosely speaking, the randomization over $\mathbf{k}$ is such that the attacker derives no advantage from using arbitrary memory in his attack. The data-hiding code is thus secure.

VIII. Capacity of Constrained Systems

While the theory above provides fundamental limits for reliable data hiding, it does not indicate how to construct practical codes. The codes used to prove the capacity theorems are random codes which cannot be used in practice due to the exponential complexity of the storage and encoding and decoding procedures.

The lattice QIM codes mentioned in Section V-C are practical, but is their performance good enough to approach the unconstrained capacity (7.9)? Recently Erez and Zamir proved that the answer is "yes" [73]. Roughly speaking, this requires the use of lattices with nearly spherical Voronoi cells. The information-bearing sequence $\mathbf{U}$ selected by the lattice encoder (5.18) plays the same role as the sequence $\mathbf{U}$ in the random-binning technique of Section VII.

For any practical lattice code, one would like to quantify the performance gap relative to an unconstrained system. We first consider the case of scalar quantizers.

A. *Capacity of Scalar QIM Systems*

Equation (6.22) describes the transmission of two possible length-N codewords $\mathbf{d}_0$ and $\mathbf{d}_1$ over the MAN channel of Fig. 14. The channel adds independent samples $V_1, \cdots, V_N$ to the input codewords. The addition is modulo Δ, the step size of the scalar quantizer. Referring to (6.10), the noise V has two parts: self-noise due to quantization and aliased attacker's noise. The tradeoff parameter α controls the probability distribution of V. If we want to transmit many codewords (as described in Section VI-D), what is the maximum rate of reliable transmission?

The answer is given by analyzing the MAN channel of Fig. 14. The maximum rate of reliable transmission for scalar QIM using parameter α and input alphabet $\mathcal{D}$ is obtained by maximizing mutual information between input and output of the MAN channel

$$R_1(\alpha, \mathcal{D}, \text{WNR}) = \max_{p_D} I(D; \tilde{Y}) \tag{8.1}$$

where p_D is a probability distribution over $\mathcal{D}$. If the codeword letters are in the binary alphabet $\mathcal{D} = \{\pm(\Delta/4)\}$ (as was assumed in Section VI-C2), the maximizing distribution is symmetric: $p_D(-(\Delta/4)) = p_D(\Delta/4) = 1/2$. But a larger value of $I(D; \tilde{Y})$ may be obtained by enlarging $\mathcal{D}$. The best choice is $\mathcal{D} = [-(\Delta/2), (\Delta/2))$, and the resulting optimal p_D is again uniform over $\mathcal{D}$.

The maximum rate of reliable transmission for any scalar QIM system using alphabet $\mathcal{D}$ is obtained by optimizing α

$$C_1(\mathcal{D}, \text{WNR}) = \max_{0 \leq \alpha \leq 1} R_1(\alpha, \mathcal{D}, \text{WNR}). \tag{8.2}$$

Using the optimal (largest) choice of $\mathcal{D}$ given above, we obtain the constrained capacity

$$C_1(\text{WNR}) = \max_{0 \leq \alpha \leq 1} \max_{\mathcal{D}} R_1(\alpha, \mathcal{D}, \text{WNR}). \tag{8.3}$$

The value of the maximizing α is obtained numerically and is not the same as the MMSE choice (7.10). A good approximation proposed by Eggers *et al.* [62] is $\alpha_{1,\text{opt}} = \sqrt{\text{WNR}/(\text{WNR} + 2.71)}$. Both the exact value and its approximation are close to (7.10) for WNR ≥ 1. Fig. 22 shows capacity as a function of WNR for scalar QIM and compares it with the capacity expression (7.9) for unconstrained systems. The gap is approximately 2 dB at a rate of 0.5 bit/sample.

When the input to the MAN channel is binary-valued, $C_1(\mathcal{D}, \text{WNR})$ cannot exceed 1 bit/sample. The performance loss due to binary alphabets is however insignificant at rates below 0.7 bit/sample. At high WNRs, the gap between constrained capacity $C_1(\text{WNR})$ and unconstrained capacity $C(\text{WNR})$ is equal to the shaping gain of scalar quantizers, $(1/2)\log_2(2\pi e/12) \approx 0.254$ bit; see Section VIII-C for more details.

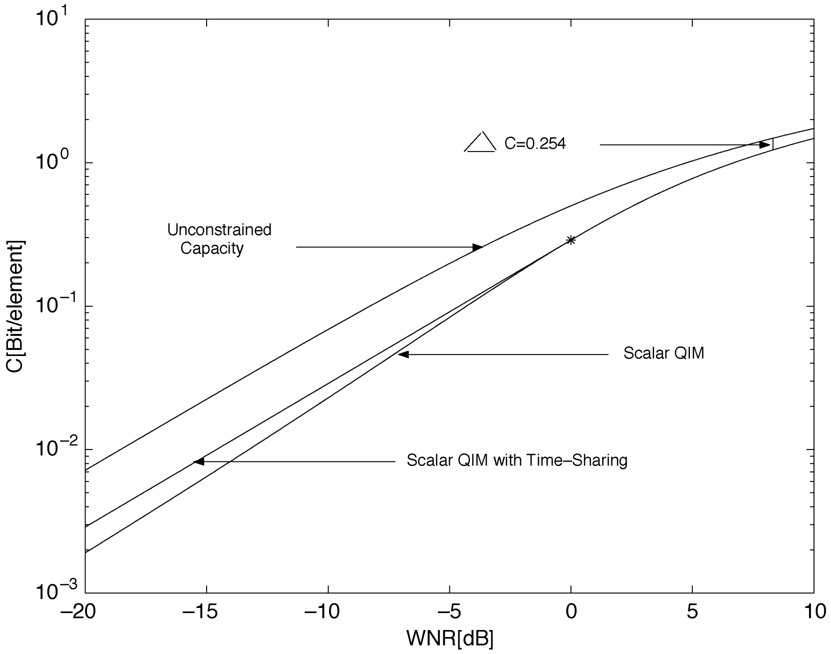


Fig. 22. Capacity versus WNR for scalar QIM.

B. Capacity of Sparse QIM Systems

It is easy to relate the capacities of QIM and the sparse QIM systems of Section V-B. The rate of reliable transmission for a sparse QIM code with sparsity factor τ is given by the time-sharing formula

$$R_1^{\text{sparse}}(\tau,\alpha,\mathcal{D},\text{WNR}) = \tau R_1\left(\alpha,\mathcal{D},\frac{\text{WNR}}{\tau}\right), \qquad 0<\alpha,\tau\le 1. \quad (8.4)$$

Optimizing over τ, α and $\mathcal{D}$, we obtain the constrained capacity

$$\begin{aligned} C_1^{\text{sparse}}(\text{WNR}) &\triangleq \max_{\tau,\alpha,\mathcal{D}} R_1^{\text{sparse}}(\tau,\alpha,\mathcal{D},\text{WNR}) \\ &= \max_{0\le\tau\le 1} \tau C_1\left(\frac{\text{WNR}}{\tau}\right). \end{aligned}$$

Based on numerical experiments with scalar quantizers, Eggers *et al.* [62] observed the following properties:

1) For WNR above a certain critical value WNR*, the optimal sparsity factor is $\tau = 1$, i.e., the system is the same as a standard nonsparse QIM system.
2) For WNR below WNR*, the optimal τ is less than one, i.e., sparse QIM systems outperform their nonsparse counterparts.

Interestingly, this property is related to information-theoretic time-sharing ideas:[20] the curve $C_1(\text{WNR})$ is nonconvex at low $\text{WNR}'s$, and the curve $C_1^{\text{sparse}}(\text{WNR})$ is the upper convex envelope of $C_1(\text{WNR})$. Thus

$$C_1^{\text{sparse}}(\text{WNR}) = \begin{cases} \frac{\text{WNR}}{\text{WNR}^*} C_1(\text{WNR}^*) : & \text{WNR} \le \text{WNR}^* \\ C_1(\text{WNR}) \qquad : & \text{else} \end{cases} \quad (8.5)$$

is a straight line for $0 \le \text{WNR} \le \text{WNR}^*$ and coincides with $C_1(\text{WNR})$ beyond WNR*. Here WNR* is the unique solution to the nonlinear equation

$$\left.\frac{d}{d\text{WNR}} \ln C_1(\text{WNR})\right|_{\text{WNR}=\text{WNR}^*} = \frac{1}{\text{WNR}^*}.$$

In conclusion, sparse QIM methods are advantageous at low WNR but not at high WNR.

C. Capacity of Lattice QIM Systems

Further improvements can be obtained by replacing scalar quantizers with L-dimensional lattice VQs (as described in Section V-C). We outline Erez and Zamir's analysis [73], which sheds insight into the coding problem.

Denote by $\mathcal{V}$ the Voronoi cell for the coarse lattice Λ_c, assumed to satisfy the embedding-distortion constraint (6.30). Analogously to (8.3), the resulting constrained capacity is

$$C_L(\text{WNR}) = \max_{0\le\alpha\le 1} \max_{\Lambda_c} \max_{p_{\mathbf{D}}} I(\mathbf{D};\tilde{\mathbf{Y}}), \qquad L \ge 1. \quad (8.6)$$

[20] See Erez and ten Brink [92] for an equivalent description of the time-sharing concept.

where $p_{\mathbf{D}}$ is a pdf over $\mathcal{D} = \mathcal{V}$. Clearly we must have

$$C_L(\text{WNR}) \leq \frac{1}{2}\log_2(1+\text{WNR}) \tag{8.7}$$

but can equality be achieved using suitable α and lattice code?

Due to (6.32), the noise vector $\mathbf{V}$ in the MAN channel has mean zero and mean-squared value per component $\sigma_v^2 = (1/L)\mathbb{E}\|\mathbf{V}\|^2$, where

$$\sigma_v^2 = (1-\alpha)^2 D_1 + \alpha^2 D_2 \geq \frac{D_1 D_2}{D_1 + D_2} \tag{8.8}$$

and equality is achieved above for the MMSE choice $\alpha = D_1/(D_1+D_2)$ of (7.10). For any α, Λ_c, $p_{\mathbf{D}}$, we have

$$\begin{aligned} C_L(\text{WNR}) &\geq \frac{1}{L}I(\mathbf{D};\tilde{\mathbf{Y}}) = \frac{1}{L}\left[h(\tilde{\mathbf{Y}}) - h(\tilde{\mathbf{Y}}|\mathbf{D})\right] \\ &= \frac{1}{L}\left[h(\tilde{\mathbf{Y}}) - h(\mathbf{V})\right] \end{aligned} \tag{8.9}$$

where $h(X) = -\int p_X(x)\log_2 p_X(x)dx$ denotes differential entropy of a random variable X. Since $\mathbf{V}$ is independent of the channel input $\mathbf{D}$, the capacity-achieving distribution $p_{\mathbf{D}}$ is uniform over $\mathcal{V}$. For any $p_{\mathbf{V}}$, we have the following properties:

- The pdf of $\tilde{\mathbf{Y}}$ is uniform over $\mathcal{V}$;
- $(1/L)h(\mathbf{V}) \leq (1/2)\log_2(2\pi e\sigma_v^2)$, where the right side is the entropy of a $\mathcal{N}(0,\sigma_v^2)$ random variable.

Using the first property and (6.36), we have

$$\frac{1}{L}h(\tilde{\mathbf{Y}}) = \frac{1}{L}\log_2 \text{Vol}(\mathcal{V}) = \frac{1}{2}\log_2\frac{D_1}{G(\Lambda_c)}.$$

Using the second property, we obtain

$$\frac{1}{L}I(\mathbf{D};\tilde{\mathbf{Y}}) \geq \frac{1}{2}\log_2\frac{D_1}{\sigma_v^2} - \frac{1}{L}\log_2\left(2\pi e G(\Lambda_c)\right).$$

To maximize the lower bound on $C_L(\text{WNR})$, we select α that minimizes σ_v^2 in (8.8) and Λ_c that minimizes $G(\Lambda_c)$ among all L-dimensional lattices

$$C_L(\text{WNR}) \geq \frac{1}{2}\log_2(1+\text{WNR}) - \min_{\Lambda_c}\frac{1}{L}\log_2\left(2\pi e G(\Lambda_c)\right). \tag{8.10}$$

Now there exist "good lattices" such that $G(\Lambda_c) \downarrow 1/2\pi e$ as $L \to \infty$. Combining the upper and lower bounds (8.7) and (8.10) on $C_L(\text{WNR})$, we conclude that $\lim_{L\to\infty} C_L(\text{WNR}) = (1/2)\log_2(1+\text{WNR})$, i.e., lattice VQ is asymptotically optimal. Furthermore, the capacity gap $(1/L)\log_2(2\pi e G(\Lambda_c))$ can be evaluated using known formulas [78] from lattice theory.

We conclude this section with a note about sparse lattice QIM systems: the gains due to time-sharing are negligible for large L because $C_L(\text{WNR})$ tends to $C(\text{WNR})$ which is convex and thus cannot be improved by convexification.

IX. Desynchronization Attacks

In addition to noise attacks, an attacker may introduce filtering, amplitude scaling, modulation, delays, warping, etc. in an attempt to desynchronize the decoder. The perceptual effects of such operations are normally quite weak, but the effects on decoding performance can be devastating. Below we use the terminology "basic decoder" to refer to the standard correlation decoder for SSM and the standard lattice decoder for QIM. Thus one can ask three basic questions:

1) How does the performance of the basic decoders degrade under such operations?
2) What is the capacity of the data-hiding systems under a distortion metric such as (2.3), which does not penalize delays and scaling factors?
3) How can one improve basic decoders to better cope with desynchronization attacks?

This line of research has recently gained some interest. To illustrate the concepts, we consider five simple desynchronization attacks. Each one takes the form

$$\mathbf{y} = T_\theta \mathbf{x} + \mathbf{w} \tag{9.1}$$

where the desynchronization operator T_θ is defined below, and $\mathbf{w}$ is signal-independent noise. Without loss of generality, we assume that $\mathbf{w}$ is zero-mean.

1) **Offset**. Let $T_\theta x(n) = \theta + x(n)$ for all $n \in \{1,2,\cdots,N\}$.
2) **Amplitude scaling**. Let $T_\theta \mathbf{x} = \theta\mathbf{x}$.
3) **Cyclic Delay**. Let T_θ be a cyclic shift by θ, i.e., $T_\theta x(n) = x(n - \theta \bmod N)$ if θ is an integer. For noninteger θ, we use the more general formula $T_\theta x(n) = \sum_{i=1}^{N} x(i)\varphi(n-i)$ where $\varphi(t) = \sin \pi t/(N \sin \pi t/N)$ is the periodic sinc interpolating function.
4) **Erasures**. Some samples $x(n)$ are erased, resulting in a shortened received sequence.
5) **Insertions**. New values are inserted in the sequence $\mathbf{x}$, resulting in a longer received sequence.

We focus on the more challenging problem of blind data hiding; otherwise the host signal can serve as a powerful synchronization signal.

A. *Performance of Basic Decoders*

SSM. From the basic SSM embedding formula (3.1) and the noise model (9.1), we obtain

$$\mathbf{y} = T_\theta\left(\gamma\mathbf{p}^{(m,k)} + \mathbf{s}\right) + \mathbf{w}.$$

The basic blind SSM decoder (3.4) computes correlation statistics $\mathbf{y}^T\mathbf{p}^{(m,k)}$ for all $m \in \mathcal{M}$. In general, T_θ can have

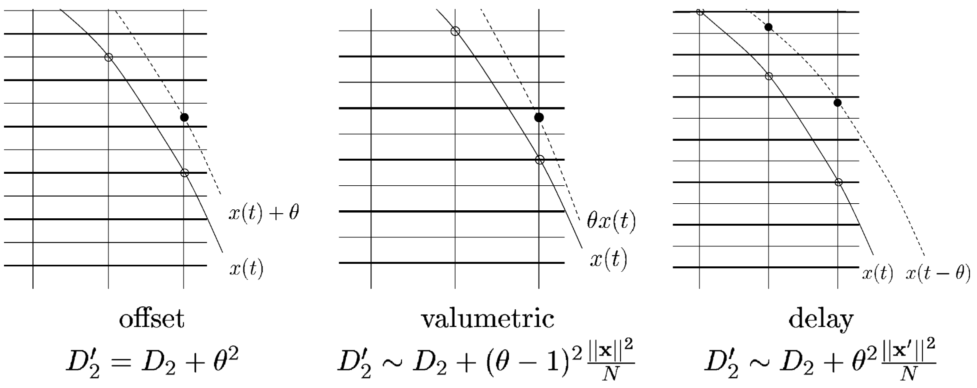

$x(t)+\theta$
$x(t)$
$\theta x(t)$
$x(t)$
$x(t)$ $x(t-\theta)$
offset
$D_2' = D_2 + \theta^2$
valumetric
$D_2' \sim D_2 + (\theta-1)^2 \frac{\|\mathbf{x}\|^2}{N}$
delay
$D_2' \sim D_2 + \theta^2 \frac{\|\mathbf{x}'\|^2}{N}$

Fig. 23. Offset, valumetric, and delay attacks.

a strong effect on the correlation statistics. For instance if $\mathbf{p}^{(m,k)}$ is a white-noise-like sequence, a slight delay would suffice to destroy the correlation between $\mathbf{y}$ and $\mathbf{p}^{(m,k)}$.

To see the problem from a slightly more general perspective, consider the following linear approximation, which is acceptable for desynchronization attacks such as amplitude modulation or time warping:

$$T_\theta\left(\gamma\mathbf{p}^{(m,k)}+\mathbf{s}\right)\approx\gamma T_\theta\mathbf{p}^{(m,k)}+T_\theta\mathbf{s}.$$

To mitigate the effects of T_θ on the correlation statistics, we would like to have $T_\theta\mathbf{p}^{(m,k)} \approx \mathbf{p}^{(m,k)}$. In other words, for the basic correlation decoder to perform as intended, the watermarks should be nearly invariant against desynchronization attacks. For instance, a slowly varying sequence $\mathbf{p}^{(m,k)}$ does not change much under moderate delays. See [93], [94] for an application to warping.

QIM. The noise V at the decoder is still a weighted average of quantization noise and aliased attacker's noise, however a new term $T_\theta\mathbf{x}-\mathbf{x}$ is added to the attacker's noise signal-independent noise $\mathbf{w}$ (see Fig. 23 for an illustration).

1) For an offset attack, the new term is the sequence whose components are all equal to θ. The mean-squared error (MSE) of the attack noise $\mathbf{y}-\mathbf{x}$ is increased from $D_2 = (1/N)\|\mathbf{w}\|^2$ to $D_2+\theta^2$, which is significant if $|\theta| > \sqrt{D_2}$.
2) For an amplitude scaling attack, the new term is equal to $(\theta-1)\mathbf{x}$. If $\mathbb{E}[\mathbf{w}^T\mathbf{x}] = 0$, the MSE of the attack noise becomes $D_2+(\theta-1)^2(1/N)\|\mathbf{x}\|^2$. This effect is significant if $(\theta-1)^2$ exceeds the noise-to-host power ratio, $\|\mathbf{w}\|^2/\|\mathbf{x}\|^2$.
3) For a cyclic delay, the MSE of the attack noise is asymptotic to $D_2+\theta^2(1/N)\|\mathbf{x}'\|^2$ as $\theta\to 0$, where $x'(n) = \sum_{i=0}^{N-1} x(i)\varphi'(n-i)$ denotes the sampled derivative of the signal x. This effect is significant if $|\theta-1| > \|\mathbf{w}\|/\|\mathbf{x}'\|$.
4) Erasures and insertions can have a similar catastrophic effect.

Therefore, the effect of even mild desynchronization attacks on unsuspecting QIM decoders can be catastrophic.

B. Capacity

First consider the case where the dimension of the parameter θ is fixed and independent of N.[21] As is the case with more traditional communication problems [91], such desynchronization attacks have no effect on capacity [34], [46]. The reason is that desynchronization does not introduce sufficient randomness. Capacity can be achieved, for instance, using random coding together with pilot sequences (entailing vanishing rate loss for large N). The poor performance of basic QIM decoders under desynchronization attacks should thus be attributed to the suboptimality of these decoders rather than a fundamental performance limit.

Desynchronization attacks which introduce substantial randomness (e.g., random jitter [95]) are more pernicious and generally cause a loss of capacity.

C. Improved Systems

Several ideas are being developed in the literature to better cope with desynchronization attacks. These includes:

- Two-step decoders. In the first step, the desynchronization parameter θ is estimated, possibly using a large search over the parameter space Θ. In the second step, the desynchronization attack is "inverted" using the estimated θ, and the resulting sequence is fed into the basic decoder. One problem with this method is the potential computational complexity of the search.
- Pilot sequences [62], [96]–[102] for estimating desynchronization parameters. The idea is to embed a known sequence in the host (in addition to the information-bearing sequence) and have the decoder estimate the desynchronization parameter θ from the received data. If the dimensionality of θ is small relative to N, this can be done reliably using the method of maximum likelihood or some other consistent estimator [38]. Moreover, the pilot can be designed to provide high estimation accuracy at the receiver, and to facilitate the search. The difficulty with these methods is twofold: 1) the sequence should be suitably randomized so that the attacker cannot remove it, and 2) the pilot does not convey

[21] This condition can be relaxed to account for more complex desynchronization problems.

information about the actual message, so less power is available for the information-bearing signals.

- Embedding in invariant domain [103]–[105]. The difficulty with these methods is to construct suitable invariants. This has been done for operations such as scaling, translation, and rotation, but it is difficult to extend this approach to more complex desynchronization attacks. A promising idea is to construct invariants based on perceptually important signal features [106]–[109].
- Embedding redundancy in the data-hiding code. The simplest example is perhaps repetition codes to combat cropping or facilitate resynchronization after a delay attack. Reed–Solomon codes have also been used for coping with insertions and deletions [68], and synchronization codes have been developed for coping with more general insertions, deletions, and substitutions [67], [110].

While promising results have been achieved in limited settings, the gap between theory and practice is still significant as of the time of this writing. Much research is going into the design of practical codes that can survive a broad range of desynchronization attacks. The state of the art for data hiding in images is overviewed in Section XII.

X. Secret Codes

So far our presentation has focused on the robustness properties of the data-hiding code—more specifically its ability to withstand the addition of memoryless noise (Sections VI–VIII) and desynchronization operations (Section IX). Decoding performance may however collapse if the attacker develops an appropriate strategy with memory.

To see why, suppose we make the code completely public—we do not use any secret key at all. It would be remarkable that such a code could resist the efforts of determined attackers, but let us see what can be done. The first observation is that an attacker is able to produce a reliable (maybe even perfect) estimate $\hat{m}$ of the embedded message m—just like any public decoder. Say the probability of correct decoding is $1-\epsilon$.

Assume for now an embedding function of the form $\mathbf{x} = f(\mathbf{s}, m)$, where f is deterministic and known to the adversary, and $m \in \mathcal{M}$ is the embedded message.

- If $f(\cdot, m)$ is invertible for all $m \in \mathcal{M}$, the adversary first computes $\hat{\mathbf{s}} = f^{-1}(\mathbf{x}, \hat{m})$, which coincides with the original host $\mathbf{s}$ with probability $1-\epsilon$. Next, the attacker may select $\hat{\mathbf{s}}$ as a forgery, thereby implementing a so-called *estimation attack* [25]. This strategy has been used by Mıhçak *et al.* to crack a popular audio watermarking scheme [111]. A related idea is to embed a fake message $m' \neq \hat{m}$ into $\hat{\mathbf{s}}$, implementing a *remodulation attack*: $\mathbf{y} = f(\hat{\mathbf{s}}, m')$. In either case, the correct m can no longer be reliably decoded.

  The spread-spectrum schemes of Section III-A are reversible and therefore inherently vulnerable to disclosure of f.
- If $f(\cdot, m)$ is noninvertible, i.e., a many-to-one map, the adversary cannot reliably reconstruct $\mathbf{s}$. The quantization-based schemes of Section V have this property (provided that $\alpha > 0$). Still a good strategy for the attacker is to apply the shortest perturbation vector leading the degraded signal to an incorrect decoding region. For scalar and hexagonal QIM systems, the distortion incurred by the adversary is of the same order as the embedding distortion [112]. It is likely that the same result holds for higher-dimensional QIM systems as well.

To defend against the estimation and modulation attacks above, one could work with an embedding function f that is mathematically invertible, but computationally hard to invert [112]–[116]. To our knowledge no practical scheme has yet been proposed and successfully tested based on this concept.

Another possible idea is to make f stochastic, meaning the receiver is unaware of the particular realization of f that generated the marked data. Such codes are called *stochastic codes* in the communication literature. They are viewed with some suspicion, because they generally have poor theoretical performance relative to *randomized codes*, in which the receiver knows f [91]. Several stochastic codes have been proposed in the watermarking literature [112], [115]–[117], but their own inventors and colleagues have found ways to defeat them. Stochastic codes have also been used for QIM steganography; see Section XI-B.

A. Randomized Codes

Several ideas can be used to randomize the codebook. Mathematically, the embedding function mapping $(\mathbf{s}, m)$ to the marked signal $\mathbf{x}$ should depend on a random variable k shared by the encoder and decoder, but unknown to the attacker. Therefore, the notation $\mathbf{x} = f(\mathbf{s}, m, k)$ adopted in (2.1) and throughout this paper already accounts for the use of randomized codes. The results by Cohen and Lapidoth [86] and Somekh-Baruch and Merhav [89], [90] demonstrate that suitably randomized codes can be made perfectly secure against adversaries with arbitrary memory and unlimited computational resources.

The common source of randomness k between encoder and decoder is usually independent of the host $\mathbf{S}$, in which case we think of it as a conventional cryptographic key. As discussed in Section II-A, occasionally we might want k to depend on $\mathbf{S}$ in order to provide side-information to the decoder; in this case, we talk about signal-dependent keys. In the remainder of this section, we restrict our attention to conventional cryptographic keys.

Following cryptographic terminology, two basic types of systems can be used: private-key systems, based on Shannon's theory of security [118]; and public-key systems as introduced by Diffie and Hellmann [119].[22]

The study of practical, secure QIM codes is still in its infancy. Ideas include randomized sparse QIM [33], randomized dithering [65] and look-up tables [121], and randomized lattice rotations [70]. Rotations may be implemented explicitly for low-dimensional lattices. For longer linear codes, one

[22] Note that the use of private or public key systems is possible whether or not the original host is available at the decoder, i.e., we can have nonblind watermarking using public-key cryptography or blind watermarking using private-key cryptography.

may use a randomized generator matrix, or randomized interleavers in the case of turbo codes.

B. Private-Key Systems

In some cases a secret key is shared between the encoder and decoder. This assumes both parties have been able to exchange this key prior to the watermark transmission, which may be difficult if not outright unrealistic in many applications. The advantage of private-key systems is that they can be made *provably* secure.

To study the secrecy of codes based on private-key systems, one can compute the mutual information $I(\mathbf{X};\mathcal{C})$ between a marked signal $\mathbf{X}$ and the codebook $\mathcal{C}$ used to generate it. Equivalently, if $\mathcal{C}$ is a one-to-one function of the secret key $\mathbf{K}$, we have $I(\mathbf{X};\mathcal{C}) = I(\mathbf{X};\mathbf{K})$. The code is perfectly secure if $I(\mathbf{X};\mathcal{C}) = 0$, i.e., observing the marked data does not convey *any* information about the code to the attacker. A related security requirement is $I(\mathbf{X};M) = 0$, i.e., observing the marked data does not convey information about the message to the attacker either. From the viewpoint of an authorized decoder however, we need $I(\mathbf{X};M|\mathbf{K}) > 0$ in order to reliably decode the message given the marked data and the key. Randomized nonlinear codes can be constructed that have the above properties [89], [90], but they are not practical.

As shown below, it is possible in some very special cases to construct simple randomized codes with the above properties.

Example 3, Revisited: Consider the data-hiding code of Table 3, which lists all 16 codewords $\mathbf{x} = \phi(\mathbf{s}, m)$ in a codebook $\mathcal{C}$. There the host $\mathbf{s} \in \{0,1\}^7$, and the message $m \in \{0,1\}$. Observe that the Voronoi cell $\mathcal{V}$ of the coarse lattice contains the all-zero sequence and the seven sequences with Hamming weight one. We can randomize $\mathcal{C}$ using a key space of cardinality 8, i.e., $k \in \{0,1,\cdots,7\}$. All we need to do is to assign a different sequence $\mathbf{d} \in \mathcal{V}$ to each value of k. Next, in place of the traditional quantizer of Example 3, we use a dithered quantizer with dither sequence $\mathbf{d}$. The resulting code is given by $\mathbf{d} + \mathcal{C}$, i.e., its codewords are obtained by adding $\mathbf{d}$ to the entries of Table 3. Observe that if $\mathbf{S}$ uniformly distributed over the space $\{0,1\}^7$ and k is uniformly distributed over $\{0,1,\cdots,7\}$, then $\mathbf{X}$ is uniformly distributed over $\{0,1\}^7$ as well. Furthermore, an attacker observing $\mathbf{X}$ gains information about the pair (m,k) but not about m or k individually: we therefore have $I(\mathbf{X};M) = 0$ and $I(\mathbf{X};K) = 0$.

This example can be straightforwardly generalized. If $\mathbf{S}$ is uniformly distributed over Hamming space $\{0,1\}^n$ and randomized nested lattice codes are used [65], [71], [80]; then $\mathbf{X}$ is also uniformly distributed over Hamming space, no matter which linear code was used! Here again $I(\mathbf{X};M) = 0$ and $I(\mathbf{X};K) = 0$. However, it appears doubtful that both properties would be achievable for linear codes in more general settings, when $\mathbf{S}$ is nonuniformly distributed. See [122] for recent, related work.

Furthermore, even in the special setting above, the perfect-secrecy property comes at a cost. To see why, assume our host sequence does not have length 7 as in Example 3, but say, length 7^B, where B is a large integer. We can then view the host sequence as the concatenation of B blocks, and apply the embedding above to each block. Note that we need to generate B independent keys in order to retain perfect secrecy. This means the size of the key space is now $8^B = 2^{3B}$, i.e., the length, $3B$, of the binary key string is linear in the length of the host sequence. In general, perfect secrecy would be practically infeasible for applications of data hiding to media signals.

To summarize, the main disadvantages of information-theoretically secure private-key systems are: (1) the key exchange protocol, which requires a secure back channel; and (2) the length of the key, which is prohibitive in media applications.

C. Public-Key Systems

In case private key exchange between the sender and receiver is neither possible nor desirable, public-key cryptographic algorithms such as RSA or elliptic-curve cryptography can be used. The idea of using public-key cryptography in watermarking can be traced back to Hartung and Girod [120]. More recent work includes [112]–[116].

Public-key systems have the following ingredients:

- a secret key k_s and a public key k_p for the receiver;
- an encryption rule $e_{k_p}(\cdot)$;
- a decryption rule $d_{k_p}(\cdot)$ with the following properties: (1) $d_{k_p}(e_{k_p}(x)) = x$, and (2) $e_{k_p}(\cdot)$ is a *trapdoor one-way function*, i.e., it is computationally infeasible to invert it (i.e., implement the decryption rule d_{k_p}) without knowing the secret key k_s.

A pratical application of this approach to data hiding could work as follows. A binary-string representation of the codebook (e.g., its generator matrix, the seed of a PRN sequence, and possibly other parameters needed for watermark decoding) is produced and encrypted using RSA. The encrypted string $e_{k_p}(\mathcal{C})$ is made publicly available. The receiver decrypts $\mathcal{C}$ and therefore obtains the codebook parameters needed for decoding.

D. Security Weaknesses

If some amount of information about the code leaks to the adversary, how can he exploit it and develop a powerful attack? This topic has seen quite a bit of research activity lately. A typical scenario is one where a key-dependent block code is used, but the same key is used over multiple blocks, or over multiple images, etc. An intelligent adversary could estimate the key (the reliability of this estimation increases with the number of copies available) and implement a remodulation attack as described in [112], [123]. All these attacks are part of the same framework of estimation attacks that was discussed earlier in this section, and can be devastating if reliable estimates of (m,k) can be formed.

If independent keys are used for different blocks, the adversary should be unable to form reliable estimates of (m,k). He may still be able to develop "surgical attacks" that exploit the structure of the code. In general, a code that is insufficiently randomized is vulnerable to surgical attacks. Some preliminary results in that direction have been reported in

[70], where a L-dimensional lattice code was used, and a key was independently generated for each length-L block. The worst L-dimensional attack pdf was derived by minimizing the Bhattacharyya performance metric for the detector. In that sense dithering (as described in Section V-C3) provides some security, but randomized lattice rotations provide a higher level of security.

XI. Related Topics

This section provides an overview of various modifications of the generic data-hiding problem studied so far. These modifications range from system attacks to problems such as steganography, authentication, fingerprinting, and media forensics, to information-theoretic duality issues.

A. *System Attacks*

In Section X we have alluded to the attacks that exploit the code structure. To guard against such attacks, the embedder and decoder need to use randomized codes, indexed by the secret key k. In this subsection we briefly discuss additional attacks in which the attacker exploits weaknesses in the communication protocol.

Sensitivity Attacks [124]–[126]. If the attacker has unlimited access to the decoder, he could iteratively modify the signal $\mathbf{x}$ and monitor the decoder's response until he is able to force an incorrect decision. The main application studied so far has been a copyright protection problem, in which the receiver makes a binary decision (watermark present or absent). The motivation for the attacker to cause an incorrect decision might be, for instance, illegally playing a watermarked CD or DVD. Can the attacker do better than using a brute-force approach (which would be infeasible if the key space is large)? The answer is "yes" for a basic spread-spectrum scheme [125] but unknown for more complex schemes.

Copy Attacks [127], [128]. Here the attacker illegally embeds a watermark derived from one document into a new document. For instance, if the auxiliary document is an image marked using an LSB embedding technique, the LSB plane is simply copied to the new image (replacing the original LSB plane). The attacker can then claim ownership of the new document. The copy attack is generally effective against nonrobust methods which embed information in perceptually insignificant components of a signal. For a more elaborate example, consider the following attack against a textured image: replace textured patches with similar patches taken from other images (e.g., replace a grassy patch with another grassy patch, etc.). It appears to be harder to develop an effective copy attack against robust watermarking methods, in which watermark and content cannot be easily separated.

Ambiguity Attacks. The main application is *proof of ownership* [129]. The attacker creates a forgery: a fake original host $\tilde{\mathbf{s}}$, together with a fake watermark (indexed by a fake message $\tilde{m}$ and a forged key $\tilde{k}$). He claims to have produced the disputed marked signal using $\mathbf{x} = f(\tilde{\mathbf{s}}, \tilde{m}, \tilde{k})$. This attack is successful if he can create such a forgery and the decoder returns $\tilde{m} = g(\mathbf{x}, \tilde{k})$. We have assumed here that f and g are fixed. Such attacks have been successful against nonblind spread-spectrum watermarking systems [129] and against some public spread-spectrum systems [4]. To guard against such attacks, one needs f, g to be one-way functions, in the cryptographic sense of the word: it should be computationally very hard to create a forgery that matches $\mathbf{x}$.

Protocol Attacks. The ambiguity attack is an example of a protocol attack, in which the attacker does not remove the watermark but makes it impossible for the document owner to prove ownership. Other protocol attacks are described in [130]–[132].

B. *Steganography*

Steganography is a data-hiding problem, with the distinguishing feature that the marked signal should "appear" like a normal unmarked signal. The problem of detecting the presence of hidden information is known as *steganalysis*.

If one needs to transmit only a few bits of information, a foolproof steganographic method can be devised. Say the transmitter (Alice) sends an image $\mathbf{x}$ containing a message $m \in \{0,1\}^b$ (i.e., b bits) to the receiver (Bob). Alice and Bob have agreed upon the following code: m will be decoded by reading the LSBs at b predetermined pixel locations in the transmitted image. If Alice has access to a database of photographic images, all she has to do is to find one that will be decoded as m. Roughly speaking, the probability that an arbitrary image satisfies this matching condition is 2^{-b}; therefore Alice has to search through an expected number 2^b of images to find a match. The image Alice selects is a perfectly natural one, and the steganalyzer (also called warden Willie by analogy to a prisoner's game [133]) is fooled.

The above method is computationally infeasible if the length of the message sequence is large. For such applications, other steganographic methods must be devised. The LSB embedding method of Section III was a simple and popular method during the 1990s, the premise being that changing the value of bits in the LSB plane does not cause any visual degradation of the image. Unfortunately LSB replacement produces *unnatural* statistical artifacts: the LSB plane of a photographic image exhibits some small but characteristic dependencies, and more significantly, dependencies with higher-order bit planes as well. These ideas are described in papers by Fridrich *et al.*, who developed a simple but surprisingly powerful algorithm (*RS steganalysis*) to detect the presence of hidden information in the LSB plane [134], [135].

Recently improved LSB steganographic methods have been developed that can resist RS steganalysis, but these new methods are themselves vulnerable to more advanced steganalysis methods. Where does the cat-and-mouse game stop?

Again, statistical detection theory provides a natural and fundamental framework to answer this question [34], [136]–[141]. The steganalyzer is essentially faced with a binary choice: decide whether data are hidden or not. Assume a statistical model (pdf p_0) for images or image features is available. If the steganographic algorithm is also known, the steganalyzer can infer the pdf p_1 for marked images (or image features). Then the steganalyzer's decision is whether the observed signal $\mathbf{x}$ was generated from p_0 or from p_1. Based on this model, one can use detection-theoretic measures of discrepancy between pdf's to bound the steganalyzer's ability to make the correct decision. If the two pdf's are identical, the steganalyzer has a 50\% probability of error. When the two pdf's differ, discrepancy measures such as Kullback-Leibler distance or Chernoff distances may be used to quantify the performance of optimal statistical tests.

At the time of this writing, the theory is sound but difficult to apply to practical problems because no universal statistical image model is known. Therefore modifications of the above techniques are required, e.g., developing universal detectors [138], [141]. While these practical difficulties might seem overwhelming for Willie, he still has the advantage that he can select arbitrary image features and test them for "naturalness." Examples of this approach may be found in [142]–[144].

C. Signature Verification

So far we have focused on coding problems, in which the decoder knows that one of $|\mathcal{M}|$ possible messages is embedded in the data, and attempts to reliably decode the message. As discussed in Section I-D, the problem is quite different when the receiver must perform the simpler binary decision: Is the received signal marked using a *given* signature $m \in \mathcal{M}$ or not? An application of this problem is signal authentication, where the signal is declared authentic if the mark is present [12]–[14], [145]. For convenience one can always assume that $\mathcal{M}$ contains a special symbol $\emptyset$ indicating the absence of any digital signature. In some applications, it is known that either the test signature $m = 1$ is embedded, or no signature at all is embedded; we may then simply write $\mathcal{M} = \{1, \emptyset\}$.

If the goal is to detect *any* tampering of the data, a *fragile watermarking* technique is often used. A rudimentary example of a fragile watermarking code would be a LSB method in which the LSB plane is a signature known to the detector, and the detector declares an error if even one bit in the LSB plane has been modified. (This method can be easily defeated by an attacker; see [4] for examples of more secure fragile watermarking schemes.)

To analyze the above problem as well as more general signature verification problems involving admissible attacks (e.g., transmission noise and/or desynchronization operations), we can define an appropriate class of channels $p_{\mathbf{Y}|\mathbf{X}}$ as in Section II. The block-diagram of the system is as in Fig. 1, with suitable modifications. The receiver has access not just to the degraded data $\mathbf{y}$ and the key k, but also to the signature $m \in \mathcal{M}$. The decoding function $\hat{m} = g(\mathbf{y}, k)$ is replaced with a binary decision rule $g(\mathbf{y}, k, m)$ taking values in {0,1} and indicating the absence or presence of the tested signature, respectively.

Given (k, m), the basic hypothesis testing setup is

$$\begin{cases} H_0 : \mathbf{Y} \sim p_{m'}(\cdot|k), & \text{for some } m' \neq m \\ H_1 : \mathbf{Y} \sim p_m(\cdot|k) \end{cases} \tag{11.1}$$

where H_0 and H_1 are respectively the "signature absent" and "signature present" hypotheses. The challenge is to design a good embedding code. The two possible error events at the detector are*false positives* (deciding H_1 when H_0 is true) and *false negatives* (deciding H_0 when H_1 is true). Unlike the coding problems studied so far, it is often useful to trade off one type of error against the other one.[23] For any detection test of the form[24]

$$\tilde{g}(\mathbf{y}|k, m) \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{11.2}$$

by varying T we obtain a curve giving the probability of true positives versus the probability of false positives. This curve is the receiver operating characteristic (ROC) [38] for the detection test. If the ROC is nonconvex, it can be improved (convexified) by randomizing T.

It turns out that QIM codes are good verification codes as well. The paper [145] contains the first application of QIM to signature verification, with encouraging results in image authentication applications. The special case $\mathcal{M} = \{1, \emptyset\}$ was analyzed in [13], [54].

The fundamental limits of signature verification schemes with distortion constraints have been studied by Steinberg and Merhav [18]. They proved that the detection problem (11.1) is dramatically easier than the full decoding problem (due to the small size of the decision space). They assumed a class of distortion-constrained memoryless channels, as in (7.4). For a normal decoding problem the receiver can reliably distinguish between 2^{NC} messages (where C is capacity); for the signature verification problem, the receiver can reliably identify as many as $2^{2^{NC}}$ signatures! A decision region $\mathcal{D}_m \subset \mathcal{Y}^N \times \mathcal{K}$ is associated with each signature m; the detector decides H_1 when $(\mathbf{y}, k) \in \mathcal{D}_m$. The number of all possible decision regions is doubly exponential in N, and so is the number of "good" decision regions.

D. Fingerprinting

In a typical fingerprinting problem, N_u users receive a marked copy of the same document. The mark is different for each user. A user may try to remove his watermark, exactly as in the basic watermarking problem. Some users could also *collude*, combining their copies to produce a better forgery (which will evade detection). For instance, they could "average" their copies in a variety of ways, they could add noise, or they could try to crack the fingerprinting code. Realistically it may be impossible for many users to collude: the maximum number N_c of colluders may be much smaller than N_u. This is a reasonable assumption when the users are only loosely acquainted.

The detection problem can be set up as ascertaining the presence of all residual marks in the forgery, i.e., catching all colluders. Unfortunately the number of combinations is

[23] The $|\mathcal{M}|$ hypotheses have equal probabilities in the coding problems.

[24] Note that H_0 is a composite hypothesis [38], and unlike in simple hypothesis testing, there is no guarantee in general that tests of the form (11.2) have optimality properties.

N_u choose N_c, which can be extremely large. The detection problem is often formulated as catching only one of the colluders: there are only $N_u + 1$ hypotheses, and the probability of getting caught is $1/N_c$, which can be large enough to deter would-be forgerers.

From a communication standpoint, the problem is essentially a multiuser version of the watermarking problems considered so far, which involved one transmitter and one receiver. The key paper by Boneh and Shaw [11] derives a lower bound on the maximum number of colluders the system can accommodate. The derivation is based on the assumption of binary sequences as well as a *marking assumption* under which the users do not flip bits at locations at which their sequences coincide. The marking assumption is not a natural one for media fingerprinting problem, because it precludes some useful strategies by the colluders (such as adding noise) and does not take distortion constraints into account. Performance analyses have recently been derived for media fingerprinting problems [146]–[150]. A typical strategy for the colluders involves linear averaging of their signals and addition of independent noise. The design of fingerprinting codes is also an active area of research [150]–[154].

E. *Media Forensics*

Data-hiding codes may also be constructed for the purpose of extracting information about the attack channel. The concept was studied by Kundur and Hatzinakos [14] under the name of *tell-tale watermarks*. Examples of tell-tale watermarks include the following.

- *Semifragile watermarks*. Here the receiver can make three possible decisions: H_1: no tampering took place; H_2: some acceptable degradation was introduced; H_0: anything else. The media is declared nonauthentic under H_0.
- Watermarks that convey information about which frequency bands of the signal might have been distorted.
- Watermarks that convey information about which areas of an image might have been distorted [60].

Security aspects of such codes have been studied in [4].

F. *Duality Issues*

We have seen that blind data hiding is a communication problem with side information at the encoder. The problem is the dual of a certain source coding problem with side information at the decoder; such problems have been studied by Wyner and Ziv [56]. The duality aspects of both problems have been studied in detail in [80]–[82].

XII. Data Hiding in Images

This section illustrates the application of the theory to images. The main challenges are to identify perceptually significant image components, resolve desynchronization issues between encoder and decoder, and develop codes that can not only cope with desynchronization but also with attacks such as addition of colored Gaussian noise and image compression. To this end, we first apply the parallel-Gaussian channel theory of Section VII-D to images [148]. Next we present a practical, recently developed QIM method [68] and outline its connection to the theory. This method represents the current state of the art of published research in data hiding for images—a line of research that started in 1999 and includes [35], [63], [67], [68], [155]–[157].

A. *Capacity Estimates*

Several transforms, including the two-dimensional (2-D) block DCT and the 2-D discrete wavelet transform [158] decompose images into approximately independent components that describe the local spatial-frequency contents of the image. To simplify the presentation, we focus on the 2-D block DCT using 8 × 8 blocks, which is the transform used in the JPEG image compression standard. Each DCT coefficient corresponds to one of 64 spatial frequencies. Let us make the approximation that these coefficients are Gaussian distributed (in fact, a Laplacian model would be more accurate but that would not add any further insight to the exposition here). We may then represent the image as a parallel Gaussian channel, with $n_P = 64$ equal-size channels, each corresponding to a different spatial frequency. The number of samples per channel is equal to n_B, the number of 8 × 8 blocks in the image ($n_B = 4096$ for a 512 × 512 image). We then compute empirical variances σ_i^2, $1 \leq i \leq 64$, for the DCT coefficients in that channel. A natural choice for the distortion metric is weighted squared error. The weighting factors w_i are chosen to be inversely proportional to the square of the default JPEG quantizer step sizes Δ_i. With this choice, noise with variance distribution $\{w_i\}$ is perceptually white. Overall mean-squared distortion levels $D_1 = 10$ and $D_2 = 50$ are chosen such that the embedding distortion is just noticeable, and the attack noise is noticeable.

The capacity limit C, evaluated from Section VII-D, is then equal to 0.01 bits per pixel. To correctly interpret this number, we need to recall that C is an asymptotic bound on the rate of reliable transmission, achievable as N tends to infinity. Due to the limited number of host samples available for embedding in each channel and to the limitations of the codes used, we may need to transmit at a rate well below C to obtain a sufficiently low probability of bit error.

B. *Practical Codes*

The paper by Solanki *et al.* [68] shows how information-theoretic concepts can be applied to practical applications of data hiding in images. The key ingredients of their framework are: (1) control of local embedding distortion based on a perceptual image model, and (2) use of erasures and errors correcting codes to handle attacks and desynchronization problems between encoder and decoder. They describe two schemes, respectively named entropy thresholding (ET) and selective embedding in coefficients (SEC). Either scheme can be used to embed thousands of information bits into an $N = 512 \times 512$ image and withstand various types of attacks, without incurring a single bit decoding error.

Here we describe their ET scheme. The image is partitioned into 8 × 8 blocks, to which the block DCT is applied. The energy (or l_2-norm entropy) of each block is computed (excluding the zero frequency component), and only those blocks whose energy exceeds a predefined threshold are selected for embedding. There are n_B such blocks. Next, n_P DCT coefficients are selected at predefined positions (spatial frequencies) within each block. Scalar QIM is then used to embed a bistream $\tilde{\mathbf{b}}$ into the sequence of selected coefficients. The quantizer step size Δ_i for each coefficient is determined from the standard JPEG quantization table, scaled according to a predefined quality factor. The quantizer step size represents a visually acceptable distortion level at that frequency. The modified $n_B n_P$ coefficients, together with the remaining unmodified ones, are transformed back to the image domain using the inverse 2-D block DCT.

This method implicitly defines n_P parallel channels with n_B samples per channel, perceptual weight $w_i \propto \Delta_i^{-2}$ for the squared-error distortion in channel i, and an overall distortion level D_1 that is controlled by the predefined energy threshold and quality factor.

The decoder computes the energy of each block to decide whether data are hidden there. Observe that two kinds of incorrect decisions can be made: incorrectly believing there are hidden data in the given block (which is equivalent to *inserting* n_P bits inside the sequence $\tilde{\mathbf{b}}$) or the converse (effectively *deleting* n_P bits from the sequence $\tilde{\mathbf{b}}$). To cope with these insertions and deletions, $\tilde{\mathbf{b}}$ should be the output of a code that can correct a number of insertions and deletions, and has the original information bit sequence $\mathbf{b}$ as input. The authors in [68] used a Reed–Solomon code, which is easily implementable. To better cope with deletions and erasures occurring in bursts (say due to cropping or tampering of parts of the image), interleaving (randomized permutation) of the information sequence is used. Interleaving distributes errors and erasures more evenly across codewords.

The setup described above fits in the general framework of Fig. 10, where $\tilde{\mathbf{b}}$ plays the role of $\tilde{m}$, and the information bit sequence $\mathbf{b}$ plays the role of the message m. The lattice is a cubic lattice, and the channel from $\mathbf{X}$ to $\mathbf{Y}$ introduces insertions and deletions.

An example presented in [68] is that of a Reed–Solomon code with 2^7 symbols (alphabet size 128), length 128, and dimension 32 (rate $R_{\rm RS} = 1/4$). There are 7 information bits per symbol of the Reed–Solomon code and 32 symbols per codeword. Using $n_P = 14$ DCT coefficients per block, they map these 14 coefficients into two code symbols. A 512 × 512 image contains 4096 8 × 8 blocks. This yields $(2 \times 4096)/128 = 64$ codewords for the whole image. The total number of embedded bits is therefore be 64 × 32 × 7 = 14 336, corresponding to a data-hiding rate $R = 0.0547$ bits/pixel. A fraction of the blocks fail the energy threshold test (say one half), causing erasures at the encoder. Nevertheless the information bits can be perfectly recovered provided that $e + 2r \leq 64(128 - 32)$, where e is the number of erasures, and r the number of errors.

A useful property of the decoding scheme is that it provides information about the location of insertions and deletions. This is particularly useful if the image has been tampered with; see Fig. 24 for an example taken from [68].

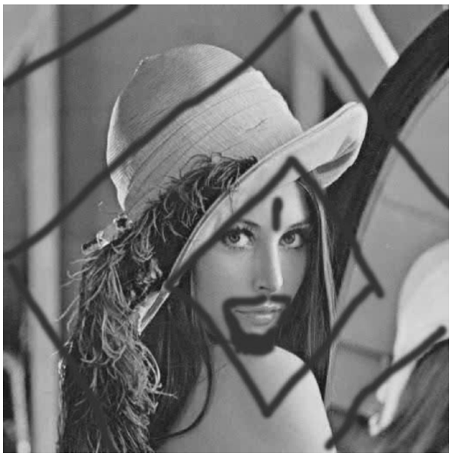


Fig. 24. Tampered *Lena* image. Reproduced with permission from [68].

The ET scheme is inherently robust against JPEG compression attacks. It should however be noted that Reed–Solomon codes are not effective against additive white Gaussian noise (AWGN). A classical coding approach to deal with that difficulty consists in using the Reed–Solomon code as an outer code, following an inner code matched to AWGN channels. The authors in [68] did not pursue this approach, but their SEC scheme copes with insertions, deletions, erasures, JPEG compression, and AWGN attacks.

Recently the same authors have also demonstrated a scheme that can resist print-scan attacks [157]. The attack consists of printing the marked image and then rescanning it. This process introduces nonlinearities, correlated high-frequency noise, and some geometric distortions. The scheme proposed in [157] applies QIM to the difference in phase of adjacent spatial-frequency components of the image.

XIII. Discussion

This paper has reviewed some basic theory for data hiding, focusing on the fundamental roles of information theory, coding theory, game theory, and signal processing. The tradeoffs between embedding distortion, attack distortion, embedding rate, and error probability can be derived quantitatively, by application of basic principles. From a qualitative standpoint, some of the most important conclusions are the following.

- When the host signal is unavailable to the receiver (blind data hiding), special embedding techniques must be devised to achieve high communication performance. The best methods known to date are based on the information-theoretic concept of binning.
- Practical binning schemes have already been developed based on this theory. They exhibit very good performance under memoryless noise attacks.
- Spread-spectrum techniques continue to be popular but have severe theoretical limitations for blind data hiding.

- Much research is still needed to design practical binning schemes that are reliable under complex desynchronization attacks. Likewise, research on secure data-hiding codes is still in its infancy. Sophisticated attacks should be expected in presence of an adversary, but need not be a concern in applications where no adversary is present.

The last ten years have seen rapid improvements in the understanding of this field and in the design of good codes. They have also seen the emergence of a plethora of new potential applications. Developing good, practical data-hiding codes that can resist sophisticated attacks appears to be a hard task. However, research is now at a point where state-of-the-art data-hiding codes have a valuable potential role to play in applications requiring a low-to-medium level of security as well as specialized applications involving private networks. While such applications differ in their specifics, solutions can be sought based on the general principles and methodology surveyed in this paper.

APPENDIX A. CODING THEORY BASICS

The theory and practice of watermarking is closely related to coding-theoretic notions. In this appendix we give a short introduction to some of the relevant basic concepts of coding theory.

The primary goal of coding is to represent signals as robustly as possible with respect to a given set of channel distortions. In the simplest case we might consider a binary communication scheme with the goal of transmitting binary digits over a channel. The channel may be modeled as a probabilistic device which reproduces the input symbol zero or one with probability $1-\epsilon$ at the receiver and changes a bit either from one into zero or one into zero with probability ϵ. This simple channel, usually referred to as the binary symmetric channel, poses the challenge that any sequence of transmitted bits may be altered into another sequence observed at the receiver. Thus, if we, for example, transmit a sequence $\mathbf{y} = (0\,000\,000)$ we may receive a sequence $\mathbf{y}' = (0\,000\,100)$ containing one error. Assume now the receiver knows that out of the possible 2^7 sequences of length seven, the transmitter only transmitted one of the 16 sequences

$$\begin{aligned}&(0\,000\,000), (1\,011\,000), (0\,101\,100), (0\,010\,110),\\&(0\,001\,011), (1\,001\,110), (1\,000\,101), (1\,100\,010),\\&(0\,110\,001), (1\,111\,111), (0\,100\,111), (1\,010\,011),\\&(1\,101\,001), (1\,110\,100), (0\,111\,010), (0\,011\,101).\end{aligned}$$

It is easy to verify that the received sequence $\mathbf{y}'$ differs from the all-zero sequence in only one position, while it differs from any other sequence in at least two positions. Provided we can assume that fewer errors are more likely than more errors (equivalent to the condition $\epsilon < 1/2$) we can conclude that the most likely transmitted sequence has been the all-zero sequence and that a single error occurred. Indeed, any other explanation for the observed sequence $\mathbf{y}'$ would imply at least two errors.

Formalizing the above setup we define a binary code $\mathcal{C}$ of length n simply as a collection of binary sequences of length n, i.e. $\mathcal{C} \subset \{0,1\}^n$. A code is coarsely characterized by its size $M = |\mathcal{C}|$, i.e. the number of codewords in the code and the so-called minimum Hamming distance of the code $d(\mathcal{C})$ defined as

$$d(\mathcal{C}) = \min_{\mathbf{c},\mathbf{c}' \in \mathcal{C}: \mathbf{c} \neq \mathbf{c}'} |\{i : c_i \neq c_i'\}| \,.$$

The size of a code relates to the data rate R, i.e. the number of bits that we can transmit in the n channel uses, as $R = (1/n)\log_2 M$. The significance of the parameter $d(\mathcal{C})$ is that a code with minimum Hamming distance at least $2t+1$ is guaranteed to correct t errors in a channel. It is easily verified that the above 16 sequences constitute a code of length seven, size 16, and minimum Hamming distance three. Indeed, we are guaranteed to be able to correct one error.

In practice it would be completely infeasible to keep track of codes by lists of codewords; additional structure on codes is required in order to keep their description small. In the example above it can be seen that the 16 codewords can be added as vectors over the binary field $\mathbb{F}_2 = \{0,1\}$ using the familiar XOR sum yielding another codeword in $\mathcal{C}$. Thus the code forms a vector space over $\mathbb{F}_2$ and it may be described by a generator matrix for this vector space. Any code with this property is called a *linear* code. It is easily verified that, indeed, the code may be described as the set of linear combinations of rows of the generator matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Given two words $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{F}_2^n$, we can define an inner product $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i$ where the sum corresponds to a sum in $\mathbb{F}_2$ (which means it is computed modulo 2). With this definition we can define a dual space to any vector space $\mathcal{C}$ as

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_2^n : \langle \mathbf{x}, \mathbf{c} \rangle = 0,\ \forall \mathbf{c} \in \mathcal{C}\}$$

The space $\mathcal{C}^\perp$, that is dual to $\mathcal{C}$, is itself a vector space generated by the rows of a so called parity-check matrix H for $\mathcal{C}$. This name reflects the fact that membership in $\mathcal{C}$ can be tested by verifying that all parity-check equations (i.e. inner products of a given vector formed with rows of the parity-check matrix) evaluate to zero. For the above code $\mathcal{C}$ a parity check matrix is given as

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Indeed, it is easily verified that $\langle \mathbf{h}_i, \mathbf{c}\rangle$ equals zero for all $\mathbf{c} \in \mathcal{C}$ and rows $\mathbf{h}_i$ of H. It is worthwhile pointing out that the space $\mathcal{C}^{\perp}$ itself constitutes a linear code with eight codewords and minimum Hamming distance four. (The reader is invited to check this.)

Linear codes are one of the cornerstones of coding theory and are being used throughout modern communications and a fair deal is known about the tradeoff between the three parameters n, M and d [36].

In Euclidean space $\mathbb{R}^n$ a binary, linear code gives rise to point sets $\mathcal{C} \subset \mathbb{R}^n$ via the simple embedding that associates real-valued vectors $\mathbf{x} \in \mathbb{R}^n$ with codewords $\mathbf{c} \in \mathbb{F}^n$ via the mapping $x_i = (-1)^{c_i}$. It can be verified that the point sets $\mathcal{C} \subset \mathbb{R}^n$ obtained in this way from a binary code $\mathcal{C}$, have minimum squared Euclidean distance $d_E^2 = 4d(\mathcal{C})$. For the connection between coding theory and codes and lattices in Euclidean space we refer to [78].

Appendix B. Vector Quantization Basics

The problem of VQ is closely related to the problem of compressing data with a maximal distortion guarantee. Assume we observe the output of a source that produces vectors of length n and let $\mathbf{x}$ be such a vector. Moreover, assume we are given a collection $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{M}$ of M vectors which are to be used for the VQ task. The goal of VQ is to find the word $\hat{\mathbf{c}}$ that is "closest" to the vector $\mathbf{x}$. Once this word $\hat{\mathbf{c}}$ is found it suffices to transmit the index of this work in the codebook $\mathcal{C}$. This can be accomplished at an expense of transmitting $\log_2 M$ bits, which is usually much less than the number of bits required for a precise reproduction of $\mathbf{x}$.

In order to give a concrete example consider again the code $\mathcal{C}$ consisting of the 16 sequences

$$\begin{aligned}&(0\,000\,000), (1\,011\,000), (0\,101\,100), (0\,010\,110),\\&(0\,001\,011), (1\,001\,110), (1\,000\,101), (1\,100\,010),\\&(0\,110\,001), (1\,111\,111), (0\,100\,111), (1\,010\,011),\\&(1\,101\,001), (1\,110\,100), (0\,111\,010), (0\,011\,101).\end{aligned}$$

Moreover assume a binary source produces a sequence $\mathbf{x} = (0\,100\,010)$. In order to perform the VQ task with respect to a Hamming distortion, i.e. we would like to reproduce $\mathbf{x}$ with a codeword at minimum Hamming distance, we choose $\hat{\mathbf{c}} = (1\,100\,010)$. Indeed, the Hamming distance between $\hat{\mathbf{c}}$ and $\mathbf{x}$ is only one. We then may use four bits to transmit which of the 16 codewords is the reproduction vector causing least distortion. In fact it is an easy exercise to check that *any* binary vector of length seven is at distance of *at most* one from one of the codewords in $\mathcal{C}$. Thus using the above code $\mathcal{C}$ we have achieved a compression ration of 7/4 at the expense of a reproduction vector of Hamming distance at most one from the source sequence.

While the above example is meant to exemplify the idea of VQ, we would like to emphasize that it is this simple principle underlying all of data compression. Indeed, given an image in raw data format a compression according to the JPEG standard follows the same ideas: A source output described by a number of bits in e.g. TIFF format is represented by a reproduction image that is as similar to the original image. The JPEG file can be interpreted as the index of the reproduction image in the codebook that consists of all possible JPEG encoded images. The involved techniques are of course far more sophisticated, but at the core all compression algorithms can be identified as VQs with specific codebooks and distortion constraints.[25]

In the main body of this paper we often resort to somewhat idealized problem settings. In particular, the VQ of Gaussian sources plays a prominent role. The natural codebooks for VQ are lattice quantizers, i.e. the set of reproduction vectors for the quantization task are given by (a subset) of lattice points. We would like to stress that the spirit of our results does not hinge around this idealized setting. In fact any VQ for realistic data may in principle replace the lattice quantizer in our setting. While analytic expressions are then hard to find, the basic concepts remain unchanged.

Appendix C. Detection Theory Basics

The most basic detection problem is deciding which of *two* hypotheses H_0 and H_1 is true. For instance, one may need to decide whether the observed data $\mathbf{y} \in \mathcal{Y}^N$ are noise only (H_0) or signal plus noise (H_1). There are two types of errors: deciding in favor of H_1 when H_0 is true (often called false alarm, or false positive), and conversely, deciding in favor of H_0 when H_1 is true (often called miss, or false negative). The statistical test takes the form

$$\begin{cases} H_0 : \mathbf{Y} \sim p_0 \\ H_1 : \mathbf{Y} \sim p_1 \end{cases} \tag{C.1}$$

where the notation $\mathbf{Y} \sim p$ indicates that $\mathbf{Y}$ is a random vector with probability distribution $p(\mathbf{y})$. The detector often forms a *test statistic* $t(\mathbf{y})$, a function of the data, and compares it with a threshold τ. If $t(\mathbf{y}) > \tau$, the decision is H_1; if $t(\mathbf{y}) < \tau$, the decision is H_0. If $t(\mathbf{y}) = \tau$, the decision may be randomized.

Optimal detection rules can often be derived by exploiting knowledge of the statistics of $\mathbf{Y}$. For instance, if both hypotheses are equally likely, the detector that minimizes probability of error is the maximum likelihood (ML) detector [38]

$$L(\mathbf{y}) = \frac{p_1(\mathbf{y})}{p_0(\mathbf{y})} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{C.2}$$

where $L(\mathbf{y})$ is the likelihood ratio.

The probability of error for the test (C.2) is[26]

$$P_e = \frac{1}{2} \int_{\mathcal{Y}^N} \min\left(p_0(\mathbf{y}), p_1(\mathbf{y})\right) d\mathbf{y}. \tag{C.3}$$

[25] In many practical compression schemes the distortion constraints are given implicitly by the source coding algorithm.

[26] The integral is a sum if $\mathcal{Y}$ is a discrete set.

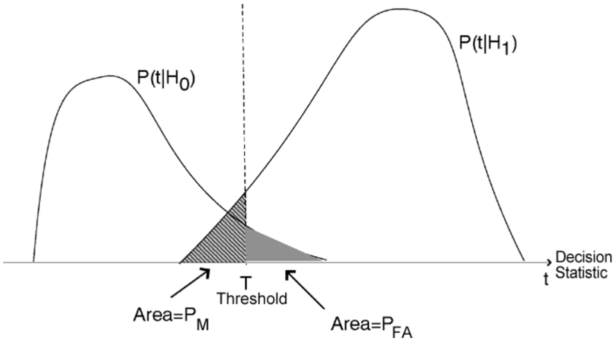


Fig. 25. Testing between two statistical hypotheses.

Fig. 25 depicts the distribution of the test statistic (here the likelihood ratio) under hypotheses H_0 and H_1. The two types of error are shown in the figure. P_e is the average of these two error probabilities, P_{FA} and P_M.

In some simple cases, P_e can be evaluated explicitly. For instance, if the rival pdf's are Gaussian, $p_0 = \mathcal{N}(0, \sigma^2)$ and $p_1 = \mathcal{N}(\mu, \sigma^2)$, then $P_e = Q(d/2)$, where $d = \mu/\sigma$ is the normalized distance between the two pdf's, and $Q(t) = \int_t^\infty (2\pi)^{-1/2} e^{-x^2/2} dx$ is the Q function. Observe that $P_e \to 1/2$ as $d \to 0$, i.e., detection becomes completely unreliable.

For most other problems, including those commonly encountered in practice, where N is large or even moderately large, exact calculation of the N-dimensional integral formula (C.3) for P_e is intractable. In many cases though, good approximations can be derived (and bad approximations as well!).

Consider the core problem encountered in this paper, where all N components of $\mathbf{y}$ are mutually independent under H_0 as well as H_1, with respective pdf's q_0 and q_1. Then we have $p_0(\mathbf{y}) = \prod_{i=1}^n q_0(y_i)$ and $p_1(\mathbf{y}) = \prod_{i=1}^n q_1(y_i)$. Taking the logarithm of both sides of (C.2) we obtain

$$t(\mathbf{y}) \triangleq \sum_{i=1}^{N} \ln \frac{q_1(y_i)}{q_0(y_i)} \underset{H_0}{\overset{H_1}{\gtrless}} 0. \tag{C.4}$$

The mean of the test statistic t is equal to $-ND(q_0 \| q_1)$ under H_0 and to $ND(q_1 \| q_0)$ under H_1, where $D(p\|q) = \int_{\mathcal{Y}} p(y) \ln(p(y)/q(y)) dy$ denotes Kullback-Leibler divergence. Denote by σ^2 the variance of $\ln(q_1(Y)/q_0(Y))$ under H_0 (in many problems, this is also the variance of $\ln(q_1(Y)/q_0(Y))$ under H_1). Then, analogously to the definition of the normalized distance d in the Gaussian case above, one can define $d^2 = N[D(q_0\|q_1) + D(q_1\|q_0)]^2/\sigma^2$, which is called *deflection coefficient*, or *generalized SNR*. While d^2 is sometimes useful as a rough measure of separation of the rival pdf's, it is not necessarily a meaningful predictor of detection performance. For instance, if the rival pdf's have disjoint supports, perfect discrimination is possible ($P_e = 0$) even though d^2 is finite. The often encountered approximation $P_e \approx Q(d/2)$ is meaningful only when the test statistic t has Gaussian tails.

For large values of N, excellent approximations to P_e can be obtained based on large-deviations theory. P_e vanishes exponentially fast with N. The errors are due to rare events whose probability is determined by the tails of the rival pdf's. The tails could be much heavier or much lighter than Gaussian tails. For large N, the crude approximation $P_e \approx Q(d/2)$ becomes overly optimistic or pessimistic, respectively, by many orders of magnitudes.

The following upper bound on P_e holds for any N:

$$P_e \le \frac{1}{2} e^{-NB(q_0,q_1)}$$

where

$$B(q_0, q_1) = -\ln \int_{\mathcal{Y}} \sqrt{q_0(y) q_1(y}\, dy \tag{C.5}$$

is the so-called Bhattacharyya coefficient, or Bhattacharyya distance between the pdf's q_0 and q_1. In the problems encountered in this paper, q_0 and q_1 satisfy a symmetry property, and the bound is tight in the exponent

$$\lim_{N\to\infty} \left[-\frac{1}{N} \ln P_e \right] = B(q_0, q_1).$$

Hence $B(q_0, q_1)$ is a more useful predictor of detection performance than is GSNR. It is easy to compute, and can be used to determine how large N should be to guarantee a prescribed probability of error.

Acknowledgment

The authors would like to thank their current and former students, A. Briassouli, A. K. Goteti, M. Kesal, T. Liu, M. K. Mıhçak, and Y. Wang, for their contributions to this paper. Special thanks are also due to the reviewers for their thorough work and for suggestions that have considerably improved this paper.

References

[1] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE,* vol. 86, no. 6, pp. 1064–1087, Jun. 1998.
[2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proc. IEEE,* vol. 87, no. 6, pp. 1062–1078, Jul. 1999.
[3] M. Barni and F. Bartolini, *Watermark Systems Engineering*. New York: Marcel Dekker, 2004.
[4] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco, CA: Morgan-Kaufmann, 2002.
[5] J. Eggers and B. Girod, *Informed Watermarking*. Boston, MA: Kluwer, 2002.

[6] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding. Steganography and Watermarking—Attacks and Countermeasures*. Boston, MA: Kluwer, 2001.

[7] , S. Katzenbeisser and F. A. Petitcolas, Eds., *Information Hiding Techniques for Steganography and Digital Watermarking*. Norwood, MA: Artech House, 2000.

[8] J. A. Bloom, I. J. Cox, T. Kalker, J.-P. M. G. Linnartz, M. L. Miller, and C. B. S. Traw, "Copy protection for digital video," *Proc. IEEE (Special Issue on Identification and Protection of Multimedia Information),* vol. 87, no. 7, pp. 1267–1276, Jul. 1999.

[9] C. Herley, "Why watermarking is nonsense," *IEEE Signal Process. Mag.,* vol. 19, no. 5, pp. 10–11, Sep. 2002.

[10] P. Moulin, "Comments on 'Why watermarking is nonsense'," *IEEE Signal Process. Mag.,* vol. 20, no. 6, pp. 57–59, Nov. 2003.

[11] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory,* vol. 44, no. 5, pp. 1897–1905, 1998.

[12] F. Bartolini, A. Tefas, M. Barni, and I. Pitas, "Image authentication techniques for surveillance applications," *Proc. IEEE,* vol. 89, no. 10, pp. 1403–1418, Oct. 2001.

[13] E. Martinian and G. W. Wornell, "Authentication with distortion constraints," in *Proc. IEEE Int. Conf. Image Processing* 2002, pp. II.17–II.20.

[14] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proc. IEEE,* vol. 87, no. 7, pp. 1167–1180, Jul. 1999.

[15] J. Kelley, "Terror groups hide behind web encryption," *USA Today* Feb. 5, 2001 [Online]. Available: http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm

[16] B. Chen and C.-E. W. Sundberg, "Digital audio broadcasting in the FM band by means of contiguous band insertion and precancelling techniques," *IEEE Trans. Commun.,* vol. 48, no. 10, pp. 1634–1637, Oct. 2000.

[17] A. Baros, F. Franco, D. Delannay, and B. Macq, "Rate-distortion analysis of steganography for conveying stereovision disparity maps," *Proc. SPIE,* vol. 5306, pp. 268–273, Jan. 2004.

[18] Y. Steinberg and N. Merhav, "Identification in the presence of side information with application to watermarking," *IEEE Trans. Inf. Theory,* vol. 47, no. 4, pp. 1410–1422, May 2001.

[19] M. Holliman, N. Memon, and M. Yeung, "On the need for image dependent keys in watermarking," presented at the 2nd Workshop Multimedia, Newark, NJ, 1999.

[20] G. Depovere and T. Kalker, "Secret key watermarking with changing keys," in *Proc. Int. Conf. Image Proc.* 2000, pp. I.10–I.13.

[21] P. Lee, Disney Corp., keynote speech at, SPIE Conf. Watermarking and Security of Multimedia San Jose, CA, 2004.

[22] A. Kerckoffs, "La cryptographie militaire," *Journal des Sciences Militaires,* vol. 9, pp. 5–38, 1883.

[23] S. A. Craver, M. Wu, and B. Liu, "Reading between the lines: Lessons from the SDMI challenge," in *10th USENIX Security Symp.* Washington, DC, 2001.

[24] J. Boeuf and J. P. Stern, "An analysis of one of the SDMI candidates," in *Proc. Int. Workshop on Information Hiding* 2001, pp. 395–410.

[25] S. Voloshynovskiy, S. Pereira, and T. Pun, "Attacks on digital watermarks: Classification, estimation-based attacks, and benchmarks," *IEEE Commun. Mag.,* vol. 39, no. 8, pp. 2–10, Aug. 2001.

[26] H. C. Kim, H. Ogunleye, O. Guitart, and E. J. Delp, "The watermark evaluation testbed (WET)," in *Proc. SPIE* Jan. 2004, vol. 5306, pp. 236–247.

[27] J. M. Ettinger, "Steganalysis and game equilibria," Proc. 1998 Workshop Information Hiding Lecture Notes in Computer Science, Springer-Verlag, vol. 1525, 1998.

[28] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems Control Inf. Theory,* vol. 9, no. 1, pp. 19–31, 1980.

[29] M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory,* vol. 29, no. 3, pp. 439–441, May 1983.

[30] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[31] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE (Special Issue on Identification and Protection of Multimedia Information),* vol. 87, no. 7, pp. 1127–1141, Jul. 1999.

[32] F. M. J. Willems, "An information theoretical approach to information embedding," in *Proc. 21st Symp. Information Theory in the Benelux* 2000, pp. 255–260.

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

[34] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inf. Theory,* vol. 49, no. 3, pp. 563–593, Mar. 2003.

[35] M. L. Miller, G. J. Doërr, and I. J. Cox, "Applying informed coding and embedding to design a robust high-capacity watermark," *IEEE Trans. Image Process.,* vol. 13, no. 6, pp. 792–807, Jun. 2004.

[36] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2004.

[37] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[38] H. V. Poor, *An Introduction to Detection and Estimation Theory*. New York: Springer-Verlag, 1994.

[39] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE J. Sel. Areas Commun.,* vol. 16, no. 4, pp. 540–550, May 1998.

[40] J. L. Cannons and P. Moulin, "Design and statistical analysis of a hash-aided image watermarking system," *IEEE Trans. Image Process.,* vol. 13, no. 10, pp. 1393–1408, Oct. 2004.

[41] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. IEEE,* vol. 81, no. 10, pp. 1385–1422, Oct. 1993.

[42] A. B. Watson, "DCT quantization matrices optimized for individual images," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display IV,* vol. 1913, pp. 202–216, 1993.

[43] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inf. Theory,* vol. 8, pp. 84–92, 1962.

[44] Y. N. Wu, S. C. Zhu, and X. W. Liu, "Equivalence of Julesz ensemble and FRAME model," *Int. J. Comput. Vis.,* vol. 38, no. 3, pp. 247–265, Jul. 2000.

[45] S. Lyu and H. Farid, "How realistic is photorealistic?," *IEEE Trans. Signal Process. (Supplement on Secure Media),* vol. 53, no. 2, pp. 845–850, Feb. 2005.

[46] P. Moulin and M. K. Mıhçak, "A framework for evaluating the data-hiding capacity of image sources," *IEEE Trans. Image Process.,* vol. 11, no. 9, pp. 1029–1042, Sep. 2002.

[47] A. C. Bovik, Ed., *Handbook of Image and Video Processing*, 2nd ed. New York: Academic, 2005.

[48] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press, 1998.

[49] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multi-level image," in *Proc. IEEE Milcom* 1990, pp. 216–220.

[50] I. J. Cox, J. Killian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.,* vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[51] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proc. IEEE (Special Issue on Identification and Protection of Multimedia Information),* vol. 87, no. 7, pp. 1108–1126, Jul. 1999.

[52] P. Moulin and A. Ivanović, "The zero-rate spread-spectrum watermarking game," *IEEE Trans. Signal Process.,* vol. 51, no. 4, pp. 1098–1117, Apr. 2003.

[53] H. Malvar and D. Florêncio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Trans. Signal Process.,* vol. 51, no. 4, pp. 898–905, April 2003.

[54] T. Liu and P. Moulin, "Error exponents for watermarking game with squared-error constraints," in *Proc. Int. Symp. Info Theory* 2003, p. 190.

[55] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory,* vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[56] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory,* vol. 22, no. 1, pp. 1–10, 1976.

[57] B. Chen and G. W. Wornell, "An information-theoretic approach to the design of robust digital watermarking systems," presented at the Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Phoenix, AZ, 1999.

[58] F. M. J. Willems, "On Gaussian channels with side information at the transmitter," in *Proc. 9th Symp. Information Theory in the Benelux* 1988, pp. 129–135.

[59] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data hiding for video-in-video," in *Proc. ICIP* 1996, vol. 2, pp. 676–679.

[60] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. 4th IEEE Int. Conf. Image Processing (ICIP'97)* 1997, vol. 2, pp. 680–683.

[61] J. J. Eggers, J. K. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks," presented at the IEE Secure Images and Image Authentication Conf., London, U.K., 2000.

[62] J. J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar Costa scheme for information embedding," *IEEE Trans. Signal Process.,* vol. 51, no. 4, pp. 1003–1019, Apr. 2003.

[63] G.-I. Lin, "Digital Watermarking of Still Images Using a Modified Dither Modulation Algorithm," M. S. thesis, Dept. of Electrical and Computer Engineering, Univ. Illinois, Urbana-Champaign, 2000.

[64] M. Kesal, M. K. Mıhçak, R. Kötter, and P. Moulin, "Iteratively decodable codes for watermarking applications," presented at the 2nd Symp. Turbo Codes and Related Topics, Brest, France, 2000.

[65] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory,* vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

[66] J. Chou, S. S. Pradhan, and Ramchandran, "Turbo coded trellis-based constructions for data embedding: channel coding with side information," in *Proc. 35th Asilomar Conf.* 2001, pp. 305–309.

[67] J. Chou and K. Ramchandran, "Robust turbo-based data hiding for image and video sources," presented at the IEEE Int. Conf. Image Processing, Rochester, NY, 2002.

[68] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," *IEEE Trans. Image Process.,* vol. 13, no. 12, pp. 1627–1639, Dec. 2004.

[69] A. K. Goteti and P. Moulin, "Two private, perceptual data-hiding games," in *Proc. ICASSP* 2004, pp. III.373–III.376.

[70] ——, "QIM watermarking games," in *Proc. ICIP* Singapore, 2004, pp. II.717–II.720.

[71] P. Moulin, A. K. Goteti, and R. Koetter, "Optimal sparse-QIM codes for zero-rate blind watermarking," in *Proc. ICASSP* 2004, pp. III.73–III.76.

[72] U. Erez and R. Zamir, "Error exponents of modulo-additive noise channels with side information at the transmitter," *IEEE Trans. Inf. Theory,* vol. 47, no. 1, pp. 210–218, Jan. 2001.

[73] ——, "Achieving $(1/2)\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory,* vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[74] T. Liu and P. Moulin, "Error exponents for one-bit watermarking," in *Proc. ICASSP* 2003, pp. III-65–III-68.

[75] R. Zamir, "On lattice quantization noise," *IEEE Trans. Inf. Theory,* vol. 42, no. 4, pp. 1152–1159, Jul. 1996.

[76] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Trans. Commun.,* vol. 12, no. 4, pp. 162–165, Dec. 1964.

[77] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Trans. Inf. Theory,* vol. 38, no. 2, pp. 428–436, Mar. 1992.

[78] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York: Springer-Verlag, 1999.

[79] P. Moulin and Y. Wang, "Error exponents for channel coding with side information," in *Proc. IEEE Information Theory Workshop* 2004, pp. 353–358.

[80] R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. Inf. Theory,* vol. 49, no. 5, pp. 1159–1180, May 2003.

[81] S. S. Pradhan, J. Chou, and Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inf. Theory,* vol. 49, no. 5, pp. 1181–1203, May 2003.

[82] J. Chou, S. S. Pradhan, and Ramchandran, "On the duality between distributed source coding and data hiding," in *Proc. 33rd Asilomar Conf.* 1999, pp. 1503–1507.

[83] J. Chou, S. Pradhan, L. El Ghaoui, and K. Ramchandran, "A robust optimization solution to the data hiding problem using distributed source coding principles," *Proc. SPIE* vol. 3971, pp. 301–310, Jan. 2000.

[84] D. van den Borne, T. Kalker, and F. M. J. Willems, "Codes for writing on dirty paper," presented at the 23rd Symp. Information Theory in the Benelux, Louvain, Belgium, May 2002.

[85] G. D. Forney, Jr., "On the role of MMSE estimation in approaching the information-theoretic limits of linear Gaussian channels: Shannon meets Wiener," presented at the Allerton Conf., Monticello, IL, 2003.

[86] A. S. Cohen and A. Lapidoth, "The Gaussian watermarking game," *IEEE Trans. Inf. Theory,* vol. 48, no. 6, pp. 1639–1667, Jun. 2002.

[87] W. Yu *et al.*, "Writing on colored paper," in *Proc. IEEE Int. Symp. Information Theory* 2001, p. 302.

[88] P. Moulin and M. K. Mıhçak, "The parallel-Gaussian watermarking game," *IEEE Trans. Inf. Theory,* vol. 50, no. 2, pp. 272–289, Feb. 2004.

[89] A. Somekh-Baruch and N. Merhav, "On the error exponent and capacity games of private watermarking systems," *IEEE Trans. Inf. Theory,* vol. 49, no. 3, pp. 537–562, Mar. 2003.

[90] ——, "On the capacity game of public watermarking systems," *IEEE Trans. Inf. Theory,* vol. 50, no. 3, pp. 511–524, Mar. 2004.

[91] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inf. Theory,* vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[92] U. Erez and S. ten Brink, "Approaching the dirty paper limit for canceling known interference," presented at the Allerton Conf., Monticello, IL, 2003.

[93] P. Moulin, A. Briassouli, and H. Malvar, "Detection-theoretic analysis of desynchronization attacks in watermarking," in *Proc. 14th Int. Conf. Digital Signal Proc.* 2002, pp. I.77–I.84.

[94] A. Briassouli and P. Moulin, "Detection-theoretic anaysis of warping attacks in spread-spectrum watermarking," in *IEEE Proc. ICASSP* 2003, pp. III.53–III.56.

[95] V. Licks, F. Ourique, F. Jordan, and F. Perez-Gonzalez, "The effect of the random jitter attack on the bit error rate performance of spatial domain watermarking," in *Proc. ICIP* 2003, pp. 455–458.

[96] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Process.,* vol. 9, no. 6, pp. 1123–1129, Jun. 2000.

[97] N. Johnson, Z. Duric, and S. Jajodia, "Recovery of watermarks from distorted images," presented at the Information Hiding Conf., Dresden, Germany, 2000.

[98] R. Caldelli, M. Barni, F. Bartolini, and A. Piva, "Geometric-invariant robust watermarking through constellation matching in the frequency domain," in *Proc. ICIP* Vancouver, B.C., Sep. 2000, pp. 65–68.

[99] P. Moulin and A. Ivanović, "The fisher information game for optimal design of synchronization patterns in blind watermarking," in *Proc. IEEE Int. Conf. Image Processing* 2001, pp. II.550–II.553.

[100] M. Álvarez-Rodríguez and F. Pérez-González, "Analysis of pilot-based synchronization algorithms for watermarking of still images," *Signal Process. Image Commun.,* vol. 17, pp. 611–633, Sep. 2002.

[101] P. Moulin, "Embedded-signal design for channel parameter estimation. Part I: linear embedding," in *Proc. IEEE Statistical Signal Processing Workshop* 2003, pp. 38–41.

[102] ——, "Embedded-signal design for channel parameter estimation. Part II: quantization embedding," in *Proc. IEEE Statistical Signal Processing Workshop* 2003, pp. 42–45.

[103] M. Kutter, "Watermarking resisting to translation, rotation and scaling," *Proc. SPIE* vol. 3528, pp. 423–431, 1998.

[104] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Process.,* vol. 66, no. 3, pp. 303–317, 1998.

[105] C.-Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Process.,* vol. 10, no. 5, pp. 767–782, May 2001.

[106] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Toward second generation watermarking schemes," in *Proc. ICIP* 1999, vol. I, pp. 320–323.

[107] M. Alghoniemy and A. H. Tewfik, "Geometric distortions correction in image watermarking," *Proc. SPIE* vol. 3971, pp. 82–89, 2000.

[108] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Process.,* vol. 11, no. 9, pp. 1014–1028, Sep. 2002.

[109] P. Bas, J.-M. Chassery, and B. Macq, "Image watermarking: an evolution to content based approaches," *Pattern Recognit.,* vol. 35, pp. 545–561, 2002.

[110] M. C. Davey and D. J. C. Mackay, "Reliable communication over channels with insertions, deletions and substitutions," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 687–698, Feb. 2001.

[111] M. K. Mıhçak, R. Venkatesan, and M. Kesal, "Cryptanalysis of discrete-sequence spread spectrum watermarks," presented at the 5th Information Hiding Workshop, Noordwijkerhout, The Netherlands, 2002.

[112] J. J. Eggers, J. K. Su, and B. Girod, "Asymmetric watermarking schemes," presented at the Sicherheit in Mediendaten, Berlin, Germany, 2000.

[113] P. Guillon, T. Furon, and P. Duhamel, "Applied public-key steganography," *Proc. SPIE* vol. 4675, pp. 38–49, Jan. 2002.

[114] G. Hachez and J.-J. Quisquater, "Which directions for asymmetric watermarking?," presented at the EUSIPCO, Toulouse, France, 2002.

[115] M. Barni, F. Bartolini, and T. Furon, "A general framework for robust watermarking security," *Signal Process.,* no. 83, pp. 2069–2084, 2003.

[116] T. Furon and P. Duhamel, "An asymmetric watermarking method," *IEEE Trans. Signal Process.,* vol. 51, no. 4, pp. 981–995, Apr. 2003.

[117] R. G. van Schyndel, A. Z. Tirkel, and I. D. Svalbe, "Key independent watermark detection," presented at the IEEE Int. Conf. Multimedia Computing and Systems, Florence, Italy, 1999.

[118] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.,* vol. 28, no. 4, pp. 656–715, 1949.

[119] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory,* vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[120] F. Hartung and B. Girod, "Fast public-key watermarking of compressed video," in *Proc. ICIP* 1997, pp. 528–531.

[121] M. Wu, "Joint security and robustness enhancement for quantization based data embedding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 13, no. 8, pp. 831–841, Aug. 2003.

[122] L. Pérez-Freire, P. Comesana, and F. Pérez-González, "Information-theoretic analysis of security in side-informed data hiding," in *Pre-Proc. 7th Information Hiding Workshop* 2005, pp. 107–121.

[123] M. F. Mansour and A. H. Tewfik, "Attacks on quantization-based watermarking schemes," in *Proc. 7th Int. Symp. Signal Processing and Its Applications* 2003, pp. 367–370.

[124] I. J. Cox and J.-P. Linnartz, "Public watermarks and resistance to tampering," in *Proc. Int. Conf. Image Proc.* 1997, pp. III.26–III.29.

[125] T. Kalker, J.-P. Linnartz, and M. van Dijk, "Watermark estimation through detector analysis," in *Proc. Int. Conf. Image Proc.* 1998, pp. I.425–I.429.

[126] J.-P. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," presented at the 2nd Information Hiding Workshop, Portland, OR, 1998.

[127] M. Kutter, S. Voloshinovskiy, and A. Herrigel, "The watermark copy attack," *Proc. SPIE* vol. 3971, pp. 371–380, 2000.

[128] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Process.,* vol. 9, no. 3, pp. 432–441, Mar. 2000.

[129] S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications," *IEEE J. Sel. Areas Commun.,* vol. 16, no. 4, pp. 573–586, May 1998.

[130] L. Qian and K. Nahrstedt, "Watermarking schemes and protocols for protecting rightful ownership and customer's rights," *J. Visual Commun. Image Represent.,* vol. 9, pp. 194–210, Sep. 1998.

[131] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol," *IEEE Trans. Image Process.,* vol. 10, no. 4, pp. 643–649, Apr. 2001.

[132] K. Gopalakrishnan, N. Memon, and P. L. Vora, "Protocols for watermark verification," *IEEE Multimedia (Special Issue on Security),* vol. 8, no. 4, pp. 66–70, Oct.–Dec. 2001.

[133] G. J. Simmons, "The prisoner's problem and the subliminal channel," in *Proc. Advances in Cryptology (CRYPTO 1983)* pp. 51–70.

[134] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color and gray-scale images," *IEEE Multimedia (Special Issue on Security),* vol. 8, no. 4, pp. 22–28, Oct.–Dec. 2001.

[135] J. Fridrich and M. Goljan, "Practical steganalysis of digital images—State of the art," *Proc. SPIE, Photonics West* vol. 4675, pp. 1–13, Jan. 2002.

[136] C. Cachin, "An information-theoretic model for steganography," in *Proc. 1998 Workshop on Information Hiding* 1998, vol. 1525, Lecture Notes in Computer Sciences, pp. 306–318.

[137] T. Mittelholzer, "An information-theoretic approach to steganography and watermarking," presented at the 3rd Workshop Information Hiding, Dresden, Germany, 1999.

[138] Y. Wang and P. Moulin, "Steganalysis of block-DCT steganography," in *Proc. IEEE Statistical Signal Processing Workshop* 2003, pp. 339–342.

[139] ——, "Steganalysis of block-structured stegotext," *Proc. SPIE* vol. 5306, pp. 477–488, Jan. 2004.

[140] P. Moulin and Y. Wang, "New results on steganographic capacity," presented at the Conf. Information Systems and Science '04, Princeton, NJ.

[141] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekharan, and B. S. Manjunath, "Detection of hiding in the least significant bit," *IEEE Trans. Signal Process. (Supplement on Secure Media),* vol. 52, no. 10, pp. 3046–3058, Oct. 2004.

[142] A. G. Flesia and D. L. Donoho, "Implications for image watermarking of recent work in image analysis and representation," in *Proc. Int. Workshop Digital Watermarking* 2002, pp. 139–155.

[143] H. Farid, "Detecting hidden messages using higher-order statistical models," in *Proc. ICIP* 2002, pp. II.905–II.908.

[144] S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machines," *Proc. SPIE* vol. 5306, pp. 35–45, Jan. 2004.

[145] J. J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *Proc. ICASSP* 2001, pp. III.1977–III.1980.

[146] J. Kilian, F. T. Leighton, L. R. Matheson, T. G. Shamoon, R. E. Tarjan, and F. Zane, "Resistance of digital watermarks to collusive attacks," in *Proc. IEEE Int. Symp. Information Theory* 1998, p. 271.

[147] F. Ergun, J. KIlian, and R. Kumar, "A note on the bounds of collusion resistant watermarks," in *Proc. EUROCRYPT* 1999, pp. 140–149.

[148] P. Moulin and A. Briassouli, "The Gaussian fingerprinting game," presented at the Conf. Information Systems and Science '02, Princeton, NJ, 2002.

[149] A. Somekh-Baruch and N. Merhav, "On the capacity game of private fingerprinting systems under collusion attacks," in *Proc. IEEE Int. Symp. Information Theory* 2003, p. 191.

[150] Z. J. Wang, M. Wu, H. Zhao, W. Trappe, and K. J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. Image Process.,* vol. 14, no. 6, pp. 804–821, Jun. 2005.

[151] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.,* vol. 51, no. 4, pp. 1069–1087, Apr. 2003.

[152] A. Barg, G. R. Blakley, and G. Kabatiansky, "Digital fingerprinting codes: problem statements, constructions, identification of traitors," *IEEE Trans. Inf. Theory,* vol. 49, no. 4, pp. 852–865, Apr. 2003.

[153] A. Silverberg, J. Staddon, and J. Walker, "Efficient traitor tracing algorithms using list decoding," *IEEE Trans. Inf. Theory,* vol. 49, no. 5, pp. 1312–1318, May 2003.

[154] M. Fernandez, "Identification of traitors in algebraic-geometry traceability codes," *IEEE Trans. Signal Process. (Supplement on Secure Media),* vol. 52, no. 10, pp. 3073–3077, Oct. 2004.

[155] M. K. Mıhçak and P. Moulin, "Information-embedding codes matched to local Gaussian image models," in *Proc. IEEE Int. Conf. Image Proc.* 2002, pp. II-137–II-140.

[156] K. Solanki, N. Jacobsen, S. Chandrasekaran, U. Madhow, and B. S. Manjunath, "High-volume data hiding in images: introducing perceptual criteria into quantization based embedding," in *Proc. ICASSP* 2002, pp. 3485–3488.

[157] K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Modeling the print-scan process for resilient data hiding," *Proc. SPIE* vol. 5681, pp. 418–429, Jan. 2005.

[158] S. G. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, CA: Academic, 1999.

[159] S. LoPresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conf. '97* pp. 221–230.

Pierre Moulin received the Ingénieur civil électricien degree from the Faculté Polytechnique de Mons, Belgium, in 1984 and the M.Sc. and D.Sc. degrees in electrical engineering from Washington University, St. Louis, MO, in 1986 and 1990, respectively.

He was a researcher at the Faculté Polytechnique de Mons in 1984–1985 and at the Ecole Royale Militaire, Brussels, Belgium, in 1986–1987. He was a Research Scientist at Bell Communications Research in Morristown, NJ, in 1990–1995. In 1996, he joined the University of Illinois, Urbana-Champaign (UIUC), where he is currently Professor in the Department of Electrical and Computer Engineering, Research Professor at the Beckman Institute and the Coordinated Science Laboratory, and Affiliate Professor in the Department of Statistics. His fields of professional interest are image and video processing, compression, statistical signal processing and modeling, decision theory, information theory, information hiding, and the application of multiresolution signal analysis, optimization theory, and fast algorithms to these areas.

Dr. Moulin received a 1997 Career award from the National Science Foundation (NSF) and a IEEE Signal Processing Society 1997 Senior Best Paper award. He is also coauthor (with J. Liu) of a paper that received an IEEE Signal Processing Society 2002 Young Author Best Paper award. He was 2003 Beckman Associate of UIUC's Center for Advanced Study. He is currently serving on the Board of Governors of the IEEE Signal Processing Society and as Editor-in-Chief of the upcoming TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He has served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE TRANSACTIONS ON IMAGE PROCESSING, Co-chair of the 1999 IEEE Information Theory Workshop on Detection, Estimation, Classification and Imaging, Chair of the 2002 NSF Workshop on Signal Authentication, and Guest Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY's 2000 special issue on information-theoretic imaging and of the IEEE TRANSACTIONS ON SIGNAL PROCESSING's 2003 special issue on data hiding. During 1998–2003 he was a member of the IEEE IMDSP Technical Committee. More recently he has been Area Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and Guest Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING's supplement series on Secure Media.

Ralf Koetter (Member, IEEE) received the Diploma degree in electrical engineering from the Technical University Darmstadt, Germany, in 1990 and the Ph.D. degree from the Department of Electrical Engineering, Linköping University, Sweden.

From 1996 to 1997, he was a Visiting Scientist at the IBM Almaden Research Lab, San Jose, CA. He was a Visiting Assistant Professor at the University of Illinois, Urbana-Champaign, and Visiting Scientist at CNRS, Sophia Antipolis, France, during 1997–1998. He joined the faculty of the University of Illinois, Urbana/Champaign, in 1999 and is currently an Associate Professor with the Coordinated Science Laboratory. His research interest include coding and information theory and their application to communication systems.

Dr. Koetter received an IBM Invention Achievement Award in 1997, a National Science Foundation CAREER Award in 2000, and an IBM Partnership Award in 2001. He served as Associate Editor for Coding Theory and Techniques for the IEEE TRANSACTIONS ON COMMUNICATIONS in 1999–2001. In 2000, he started a term as Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY. He received the 2004 paper award of the IEEE Information Theory Society.