Information Theoretic Approach To The Authentication Of Multimedia

Emin Martinian, Brian Chen, and Greg Wornell

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
{emin,bchen,gww}@allegro.mit.edu

ABSTRACT

In many multimedia applications, there is a need to authenticate a source that has been subjected to benign degradations in addition to potential tampering attacks. We develop one information-theoretic formulation of this problem, and identify and interpret the associated fundamental performance limits. A consequence of our results is that there is a tradeoff between embedding distortion and robustness to channel noise, but no such tradeoff between these parameters and security to forgery. To develop some intuition, we outline a sphere packing analogy and show that the results from sphere packing and information theory have the same form.

One important benefit of our framework is a coherent way to analyze and design authentication schemes for general source models, distortion metrics, and noisy channel models. We illustrate this by an example construction of a realizable authentication scheme. An important ingredient of our construction is the use of forward error correcting codes. We show that the application of fairly simple codes decreases the embedding distortion required by more than 5 dB without decreasing security or robustness. Our approach is general enough to be used in a wide variety of applications.

Keywords: multimedia, authentication, tamper proofing, digital watermarking, anti-spoofing, digital signatures

1. INTRODUCTION

In traditional authentication problems, one is interested in determining whether a received message is an exact replica of what was sent. Digital signature techniques have emerged as a natural tool for addressing such problems. However, in many emerging multimedia applications, the message may be an audio or video waveform, and even in the absence of a tampering attack, the waveform may experience routine degradation due to noise, compression, etc, before being received. Methods for reliably authenticating the received data in such cases are important as well.

As a motivating example, consider the authentication of drivers' licenses. Many jurisdictions print a hologram on the photograph portion of the license. The presence of the hologram indicates that the license is legitimate but does not add excessive distortion. Imprinting a hologram on a license is a particular implementation of a larger class of authentication schemes. More generally, special markings are embedded into the photograph. A decoder uses these markings to extract an authentic representation of the original. The special markings should be embedded so that the distortion between the original and embedded photographs is small; thus, someone without the appropriate decoder can still use the license to check the identity of the bearer. In addition, the special markings need to be robust to perturbations in the form of smudges or other degradation due to routine handling: the decoder should still declare the photo authentic if only these are present. Finally the special markings should be inserted so that no other agent can create a successful forgery.

Researchers have proposed approaches to this class of problems based on digital watermarking, cryptography, and content classification [1]–[8]. Ultimately, the methods developed to date attempt to balance the competing goals of robustness to benign degradation, security against tampering attacks, and embedding distortion. For example, previous work suggests that reducing the embedding distortion or increasing the security to forgery reduces the robustness to noise.

For a simple but reasonable model, we examine these tradeoffs from an information-theoretic perspective. We begin by proposing a simple, well-defined formulation of the authentication problem which illustrates the fundamental tradeoffs between robustness, security, and embedding distortion. This allows us to characterize asymptotically achievable embedding distortions which serve as performance bounds and provide insights into the design of practical schemes. We demonstrate the application of this framework via an example construction specifically optimized for a uniform source over an additive white Gaussian noise channel using quadratic distortion. To illustrate the benefits of coding, we show that relatively simple error correction codes can decrease the embedding distortion required by more than 5 dB without decreasing security or robustness to channel noise.

The rest of this paper is organized as follows. We briefly outline the problem and present the main result in Section 2. Then we precisely define the problem and conditions when this result applies in Section 3. In Section 4 we discuss the main result and provide a geometric interpretation in terms of sphere packing. We use these ideas in Section 5 to construct an example authentication scheme and evaluate its performance. We close with some conclusions.

2. PREVIEW OF THE PROBLEM AND MAIN RESULTS

While a variety of models will apply in practice, for the purpose of illustration we consider a particularly simple and tractable one in this section. Specifically, we model the original source as a stochastic process, $\{X_i\}_{i=1}^n$, where the X_i's are independent and identically distributed (i.i.d.) according to some known distribution $p(x)$. The encoder modifies the original source, producing Y_1^n, which then passes through a noisy channel with a known, memoryless probability distribution $p(z|y)$. The output of the noisy channel then passes through an insecure channel. The insecure channel does not have a probability model. Instead a malicious attacker may modify the input to the insecure channel to produce a potential forgery, W_1^n. It is important to emphasize that our results do not depend critically on the i.i.d. property. In fact, the i.i.d. model is pessimistic; better performance can usually be obtained when correlation is present. Our analysis can be extended to cover such cases using standard techniques such as water-pouring, transform coding, whitening, equalization, etc.

The decoder takes the potential forgery as input and attempts to extract an authentic representation of the original source. The diagram in Figure 1 illustrates this scenario. Performance is measured according to three criteria: security, robustness, and distortion. For a scheme to be secure, it should be impossible or infeasible for an attacker to fool the decoder. For a scheme to be robust, the decoder should almost never declare a signal to be a forgery unless the attacker has tampered with the signal. Finally, a good scheme should keep the distortion between the original source and the received signal as small as possible.

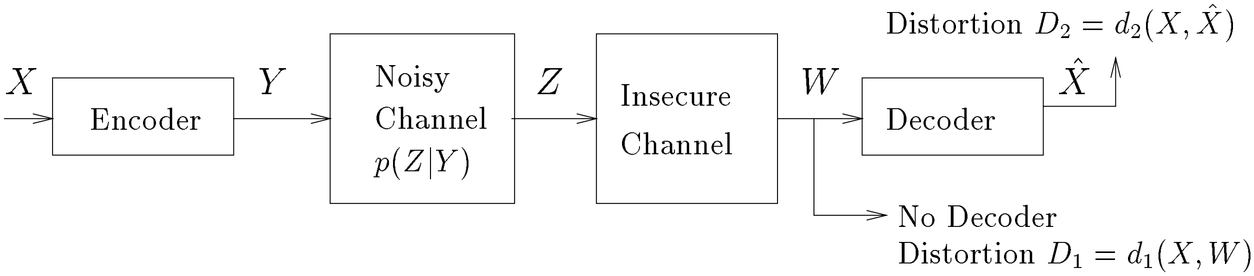

Figure 1. A diagram of the authentication problem.

We label the two relevant distortions D_1 and D_2. The distortion between the original source and the output of the noisy channel is D_1. This is the distortion a receiver sees when no decoder is available and no forgery is attempted. Hence, D_1 should be small so that a receiver without a decoder receives a good representation of the source. By contrast, D_2 is the distortion between the source and the representation extracted by the decoder. Distortion D_2 should be small so that a receiver with a decoder can extract a high quality, authenticated representation of the source.

In the license example, a small D_1 corresponds to an encoding procedure that does not distort the original image much. This allows an individual without a decoder to use the license to check the bearer's identity. A small D_2 allows an official with a decoder to extract a high quality authentic representation of the image from the license. Ideally, the pair (D_1, D_2) should be as small as possible.

Our analysis shows that reducing the embedding distortion for a scheme also reduces the robustness to noise, but increasing the security to forgery does not effect robustness or embedding distortion. Thus our work confirms that a tradeoff exists between distortion and robustness, but shows that no such tradeoff exists between these goals and the goal of security.

In [9], we prove that authentication schemes that are secure, robust and satisfy a distortion constraint pair (D_1, D_2) are asymptotically achievable if and only if there exists a conditional distribution $p(y|x)$ and a scalar function $f(\cdot)$ such that

$$I(X;Y) \leq I(Y;Z), \qquad E[d_1(X,Z)] \leq D_1, \qquad E[d_2(X, f(Y))] \leq D_2 \tag{1}$$

where $I(\ \cdot\ ;\ \cdot\ )$ denotes mutual information between a pair of random variables, and $d_1(\cdot,\cdot)$ and $d_2(\cdot,\cdot)$ are the distortion measures of interest. After precisely defining the problem for which these results apply, we provide a geometric interpretation in Section 4.

3. PRECISE PROBLEM STATEMENT

Next we precisely define an idealized version of the problem for which asymptotic results can be obtained. An instance of the general authentication problem consists of the 5-tuple $(\mathcal{X}^n, p(x), p(z|y), d_1(\cdot,\cdot), d_2(\cdot,\cdot))$. The source is a sequence of n values drawn from a finite set $\mathcal{X}^n$ according to the known i.i.d. distribution $p(x)$. Without loss of generality all random variables are assumed to take values in this set. The noisy channel follows a known probability law $p(z|y)$. When the channel is used repeatedly it is memoryless: $p(z^n|y^n) = \prod p(z_i|y_i)$. The attacker can look at the input and output of the noisy channel, Y_1^n, Z_1^n, and arbitrarily choose the final output W_1^n. Specifically, the attacker is not bound by any distortion constraint. Distortion for a received sequence is measured by summing the bounded single letter distortion functions. Therefore $D_1 = (1/n)\sum_{i=1}^n d_1(X_i, Z_i) \leq d_{\max,1}$, and $D_2 = (1/n)\sum_{i=1}^n d_2(X_i, \hat{X}_i) \leq d_{\max,2}$, where $\hat{X}_1^n$ is the estimate of the source produced by the decoder.

An encoding scheme, $\mathcal{G}_n$, consists of an algorithm which returns a triple $(Q_n(\cdot), \Phi_n(\cdot), p_s)$ consisting of an encoding function, decoding function, and secret key. The secret key consists of k bits known only to the encoder and decoder. All other information is known to all parties including the attacker.

The encoder is a mapping from source sequences and secret key to codewords: $Q_n(X_1^n, p_s) : \mathcal{X}^n \times \{0,1\}^k \to \mathcal{X}^n$. The decoder is a mapping from channel outputs and the secret key to estimated source sequences or a special symbol, $\varnothing$, which indicates decoding failure: $\Phi_n(W_1^n, p_s) : \mathcal{X}^n \times \{0,1\}^k \to \mathcal{X}^n \cup \varnothing$. Although we consider sequences of encoding schemes, encoder, decoders, etc., we omit the subscript n for clarity when it will not cause confusion. Also, we assume the secret key is provided where necessary and omit the term p_s to simplify the notation.

If $\mathcal{G}$ makes random choices, then all probabilities are taken over these random choices as well as other stochastic processes such as the random source, channel law, etc. Note that we do not assume any probability model for the behavior of the attacker.

3.1. Error Events And Achievable Distortions

In order to define a notion of achievable distortions, we define three error events. The first type of event, which we call an undetected error, corresponds to the attacker tricking the decoder into accepting a forgery. A secure encoding scheme should make it extremely unlikely that the attacker can create a successful forgery. The last two error events correspond to the encoding scheme introducing too much embedding distortion. Good encoding schemes should keep the distortion to within the specified tolerances. We require that achievable schemes have the overall probability of error decay to 0 as $n \to \infty$.

3.1.1. Definition Of Security And Undetected Error

In order to define an undetected error event we need to define what constitutes a forgery and what constitutes secure operation. Multiple notions of security are possible. In [9] we discuss several definitions of security and examine their implications. For convenience of analysis, in this paper we consider a simple, strong notion of security by restricting our attention to a 2-stage decoder: $\Phi(\cdot) = \Phi_B(\Phi_A(\cdot))$. The first stage, $\Phi_A(\cdot)$ produces an estimate of Y_1^n and the second stage, $\Phi_B(\cdot)$ converts the estimate of Y_1^n into a reconstruction of X_1^n. Secure operation corresponds to the event $\mathcal{S} = \{\Phi_A(W_1^n) = Y_1^n\} \cup \{\Phi_A(W_1^n) = \varnothing\}$. When the event $\mathcal{S}$ occurs, either the decoder bases its reconstruction solely on the encoder output or declares an encoding failure. Thus the attacker can only cause a decoding failure, but if no decoding failure occurs the output of the decoder is independent of the attacker's actions. Consequently, we define an undetected error as the event that the attacker tricks the decoder into accepting a forgery $\mathcal{E}_U = \mathcal{S}^c = \{\Phi_A(W_1^n) \neq Y_1^n\} \cap \{\Phi_A(W_1^n) \neq \varnothing\}$.

3.1.2. Excess Distortion Error:

In order to insure that the encoding satisfies the distortion constraint we define the excess distortion error events as $\mathcal{E}_{D_1} = \{(1/n)\sum_{i=1}^{n} d_1(X_i, Z_i) > D_1\}$ and $\mathcal{E}_{D_2} = \{(1/n)\sum_{i=1}^{n} d_2(X_i, \Phi_i(Z_1^n)) > D_2\}$. An excess distortion error corresponds to either the encoding process introducing too much distortion or the combined encoding and decoding process not producing an accurate reconstruction. Note that if the decoder erroneously declares a decoding failure on input Z_1^n, this will cause the event $\mathcal{E}_{D_2}$ to occur. This follows since no matter what fixed value is chosen for $\Phi_B(\varnothing)$, the distortion between X and $\Phi_B(\varnothing)$ will be large with high probability. Consequently, requiring $\Pr[\mathcal{E}_{D_2}]$ to be small implies that the probability of a decoding failure is also small.

3.1.3. Achievable Distortions

Define an overall error as any of the previous events $\mathcal{E} = \mathcal{E}_{D_1} \cup \mathcal{E}_{D_2} \cup \mathcal{E}_U$. We define a distortion pair, (D_1, D_2), for the noisy authentication problem $(\mathcal{X}^n, p(x), p(z|y), d_1(\cdot,\cdot), d_2(\cdot,\cdot))$ as achievable if there exists a sequence of encoding schemes, $\mathcal{G}_n$, such that $\lim_{n\to\infty} \Pr[\mathcal{E}_n] = 0$.

Our main result is summarized below in Theorem 3.1 and proved in [9].

Theorem 3.1. *A distortion pair, (D_1, D_2), is achievable if and only if there exists a probability distribution $p(y|x)$ and a scalar function $f(\cdot)$ such that the conditions in Equation (1) are satisfied.*

The scalar function, $f(\cdot)$, corresponds to the second stage of the decoder, $\Phi_B(\cdot)$, mentioned in Section 3.1.1. Generally, once the decoder has determined Y_1^n, instead of producing the estimate $\hat{X}_1^n = \hat{Y}_1^n$, the decoder can create a better reconstruction by estimating X_1^n given Y_1^n. For example, if the distortion metric was mean square error, then $f(\cdot)$ would correspond to the minimum mean square estimator of each component of X_1^n given each component of Y_1^n. Generally, $f(\cdot)$ will correspond to the scalar minimum distortion estimator of X_i given Y_i. Clearly $D_1 \geq D_2$. Additional structure between the distortions can be derived as shown in Section 4.1.

4. GEOMETRIC INTERPRETATION

In this section we develop a sphere packing view to provide an intuitive understanding of Theorem 3.1 and to present a general method of constructing authentication schemes. We consider a source, X_1^n, with mean 0 and variance σ_X^2 elements with distortion measured according to mean square error. We model the noisy channel as additive noise with mean 0 and variance σ_N^2 elements.

The law of large numbers implies that with high probability an outcome of the source, X_1^n, will approximately lie in an n-dimensional sphere of radius $\sqrt{n\sigma_X^2}$ centered on the origin. The codewords are points in this sphere[*]. Again, by the law of large numbers, the noise vector will lie near the shell of an n-sphere of radius $\sqrt{n\sigma_N^2}$. We will show that designing a good authentication scheme corresponds to packing spheres of radius $\sqrt{n\sigma_N^2}$ (i.e. codewords) into a sphere of radius $\sqrt{n\sigma_X^2}$ (i.e. the space of source outcomes).

To meet the goal of fidelity, there should be enough codewords so that most source outcomes are close to a codeword. To achieve the goal of security we partition the codewords into two types: the admissible codewords and the non-admissible codewords. The encoder and decoder know which codewords are admissible and the attacker does not. The encoder only transmits admissible codewords, so if the received signal is closer to a non-admissible codeword than to an admissible codeword, the decoder can deduce that the signal is a forgery. To achieve robustness to channel noise, the codewords must be far enough apart that they can be distinguished at the output of the noisy channel. Figure 2 illustrates this idea. We develop this analogy further in the remainder of this section.

4.1. Detailed Analysis Of Sphere Packing Analogy

In Section 3.1.1 we restricted the decoder structure to the form $\Phi(\cdot) = \Phi_B(\Phi_A(\cdot))$, where the first stage produces an estimate $\hat{Y}_1^n = \Phi_A(W_1^n)$ and the second stage converts this into an estimate of the source $\hat{X}_1^n = \Phi_B(\hat{Y}_1^n)$. To develop the analogy, we further restrict the decoder by choosing the second stage to be the identity function so $\hat{X}_1^n = \hat{Y}_1^n = \Phi_A(W_1^n)$. Furthermore we first consider a decoder that maps the received signal to the nearest codeword. This allows us to focus our attention on the embedding distortion. With this structure, the distortion at the output of the decoder will be $d_2(X_1^n, Y_1^n)$ provided successful decoding occurs. By expanding $D_1 = E[(X - Z)^2]$

[*] Since the source lies inside the n-sphere of radius $\sqrt{n\sigma_X^2}$ with high probability, codewords outside the sphere would never be used. Consequently to simplify the analysis we assume that all the codewords are inside the sphere.

using $Z = Y + N$ we obtain $D_1 = D_2 + \sigma_N^2$. Therefore, in the additive noise model where $\Phi_B(\cdot)$ is the identity function, finding the smallest pair of achievable distortions is equivalent to finding the smallest achievable distortion for $D_2 = d_2(X_1^n, Y_1^n)$.

4.1.1. Fidelity

The encoder, $Q(\cdot)$, maps a source outcome, X_1^n, to a codeword $Y_1^n = Q(X_1^n)$. To keep the total squared distortion between X_1^n and Y_1^n smaller than nD_2 requires that for each value of X_1^n, there exists a codeword within Euclidean distance $\sqrt{nD_2}$. We call the set of all codewords the codebook $\mathcal{C}$ and denote the number of codewords as $K = |\mathcal{C}|$. Since we are interested in the fidelity, security and robustness of the encoding, the size of the codebook is not of direct importance. However, because our authentication is implemented in terms of quantizers, the codebook size and the associated codebook rate, $R = (1/n) \log K$, turn out to be useful intermediate parameters in the analysis.

If we denote the volume of an n-sphere of radius r as $V_n(r)$, then the number of codewords per unit volume is $\alpha = K/V_n(\sqrt{n\sigma_X^2})$. Consider a sphere of volume $V_n(1/\alpha)$ centered on a source outcome, X_1^n. If the codewords are evenly distributed, the sphere will contain one codeword on average (if the codewords are not evenly distributed the sphere will contain less codewords on average). The Euclidean distance between X_1^n and the codeword must be less than the radius of the sphere. Consequently, keeping the distortion smaller than nD_2 requires packing enough codewords into the space of possible source outcomes so that $V_n^{-1}(1/\alpha) < nD_2$. Using the simple manipulations below we can transform this into a constraint on the codebook rate:

$$\frac{1}{\alpha} < V_n(nD_2) \Rightarrow \frac{V_n\left(\sqrt{n\sigma_X^2}\right)}{K} < V_n(nD_2) \Rightarrow K > \frac{V_n\left(\sqrt{n\sigma_X^2}\right)}{V_n(nD_2)} \Rightarrow K > \left(\frac{\sigma_X^2}{D_2}\right)^{n/2} \Rightarrow R > \frac{1}{2}\log\frac{\sigma_X^2}{D_2} \quad (2)$$

4.1.2. Security

To achieve security it should be infeasible for the attacker to find a signal which is accepted by the decoder. For example, if the attacker chooses a random signal, the decoder should reject it with high probability. This implies that the volume of the space accepted by the decoder should be small compared to the volume of the space of source outcomes. To achieve this, we modify the original codebook $\mathcal{C}$, encoder, $Q(\cdot)$, and decoder, $\Phi(\cdot)$ as follows. Choose a $\gamma > 0$ such that $\gamma \ll R$ and generate the modified codebook, $\mathcal{C}' \subset \mathcal{C}$, by randomly choosing $2^{n(R-\gamma)}$ of the 2^{nR}

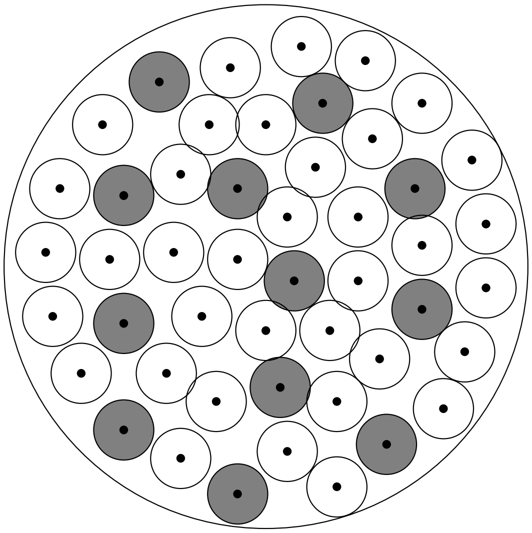

Figure 2. The large n-sphere of radius $\sqrt{n\sigma_X^2}$ represents the space of possible source outcomes. The small spheres of radius $\sqrt{n\sigma_N^2}$ are centered on codewords. Since the noise vector, N_1^n, will be smaller than $\sqrt{n\sigma_N^2}$ and the small spheres do not overlap, the codewords can be determined from the output of the noisy channel $Z_1^n = Y_1^n + N_1^n$. The shaded spheres represent the admissible codewords known only to the encoder and decoder. Since the attacker does not know which codewords are admissible, when he creates a forgery the decoder will check that the closest codeword is not admissible and detect the forgery.

codewords in $\mathcal{C}$. We call $\mathcal{C}'$ the admissible codebook. The knowledge of $\mathcal{C}'$ is the secret key which is known only to the encoder and decoder and concealed from the attacker. The modified encoder encodes X_1^n by mapping it to the nearest codeword in $\mathcal{C}'$. The modified decoder first maps a signal to the nearest codeword, $c \in \mathcal{C}$. Since the decoder knows that the encoder only produces codewords in $\mathcal{C}'$, if $c \notin \mathcal{C}'$, the decoder declares the signal to be a forgery. Otherwise if $c \in \mathcal{C}'$, the decoder produces c as the estimate of X_1^n.

If the attacker sees a valid codeword, c, and tries to create a forgery, W_1^n, there are three possibilities for the behavior of the decoder. In the first case, $\Phi\left(W_1^n\right) = c$: the attacker does not modify the forgery much so the decoder obtains the same result as if no tampering had occurred. The decoder is not fooled in this case. In the second case, $\Phi\left(W_1^n\right) = c_f \notin \mathcal{C}'$: the decoder declares the signal a forgery which is correct. In the third case, $\Phi\left(W_1^n\right) = c_f \neq c$ and $c_f \in \mathcal{C}'$: the decoder does not realize the signal is a forgery and produces c_f as an allegedly authentic reconstruction. A good scheme should make the third case unlikely. Recall that the attacker has no knowledge of $\mathcal{C}'$ and the set of codewords in $\mathcal{C}'$ were chosen randomly and independently of the attacker. So if $c_f \neq c$, then the probability that $c_f \in \mathcal{C}'$ is

$$\Pr[\{c_f \neq c\} \cap \{c_f \in \mathcal{C}'\}] = \frac{|\mathcal{C}'|}{|\mathcal{C}|} = \frac{2^{n(R-\gamma)}}{2^{nR}} = 2^{-n\gamma}.$$

Consequently the forgery will fail with high probability so the scheme is secure.

The distortion for the modified codebook, $\mathcal{C}'$, can be computed by evaluating the average number of codewords per unit volume as before. Since $|\mathcal{C}'| = |\mathcal{C}|2^{-n\gamma}$, this calculation shows that the distortion of the modified codebook is $D_2 2^{\gamma}$. Thus making the scheme secure increases the distortion slightly. The increase in distortion can be made negligible by choosing $\gamma > 0$ small enough. Alternatively we can start with $2^{n(R+\gamma)}$ codewords in $\mathcal{C}$ and create $\mathcal{C}'$ by choosing 2^{nR} codewords from $\mathcal{C}$. The result is that both security and fidelity can be achieved simultaneously.

4.1.3. Robustness

Next we address the goal of robustness to noise. According to the law of large numbers, the noise vector N_1^n will have length close to $\sqrt{n\sigma_N^2}$. Therefore if codeword Y_1^n is transmitted, the output of the noisy channel will lie on the surface of an n-sphere of radius $\sqrt{n\sigma_N^2}$ centered on the transmitted codeword Y_1^n. If the noise spheres centered on the codewords in $\mathcal{C}$ overlap, then decoding errors can occur if the noise perturbs $c = Y_1^n$ such that the received signal Z_1^n is closer to another codeword c_n. If $c_n \neq c$, then with high probability $c_n \notin \mathcal{C}'$. Consequently the decoder will declare the received signal to be a forgery, when in fact it is a valid signal perturbed by noise. To prevent this type of false alarm error, the codewords must be far enough apart that the noise spheres do not overlap. The maximum number of noise spheres of radius $\sqrt{n\sigma_N^2}$ which can be packed into the sphere of possible source outcomes without overlap is proportional to the ratio of their volumes. Therefore to insure correct decoding, the total number of codewords must satisfy

$$K < V_n\left(\sqrt{n\sigma_X^2}\right)/V_n\left(\sqrt{n\sigma_N^2}\right) \quad \Rightarrow \quad R < \frac{1}{2}\log\frac{\sigma_X^2}{\sigma_N^2}. \tag{3}$$

Combining this with Equation (2) implies $(1/2)\log(\sigma_X^2/D_2) + \gamma < R < (1/2)\log(\sigma_X^2/\sigma_N^2)$ must hold to achieve fidelity, security, and robustness. Since γ can be made negligible, we obtain the requirement $D_2 > \sigma_N^2$.

4.2. Uniform Source, AWGN Channel Model, Quadratic Distortion

For a more specific example, consider authenticating a uniform source over an additive white Gaussian noise channel with quadratic distortion. The sphere packing argument shows that an authentication scheme for this scenario is possible when $D_2 > \sigma_N^2$. Similar results can be obtained by applying Theorem 3.1. In addition, Theorem 3.1 can provide a lower bound for D_2. Specifically, in [10] we show that

$$\frac{1}{2}\log\frac{6\sigma_X^2}{\pi e D_2} \leq I(X;Y), \quad I(Y;Z) \leq \frac{1}{2}\log\frac{\sigma_X^2 + D_2 + \sigma_N^2}{\sigma_N^2} \tag{4}$$

Combining these with the requirement $I(X;Y) \leq I(Y;Z)$ from Theorem 3.1 yields

$$\frac{1}{2}\log\frac{6\sigma_X^2}{\pi e D_2} \leq \frac{1}{2}\log\frac{\sigma_X^2 + D_2 + \sigma_N^2}{\sigma_N^2} \quad \Rightarrow \quad D_2 \geq \frac{-(\sigma_X^2 + \sigma_N^2) + \sqrt{(\sigma_X^2 + \sigma_N^2)^2 + 24\frac{\sigma_X^2\sigma_N^2}{\pi e}}}{2}$$

Rewriting this in terms of the distortion-to-noise ratio $\mathrm{DNR} = D_2/\sigma_N^2$ and the signal-to-noise ratio $\mathrm{SNR} = \sigma_X^2/\sigma_N^2$ we obtain

$$\mathrm{DNR} \geq \left(\frac{1+\mathrm{SNR}}{2}\right)\left(-1+\sqrt{\frac{6}{\pi e}\frac{4\mathrm{SNR}}{(1+\mathrm{SNR})^2}}\right) = \mathrm{DNR}_{min} \tag{5}$$

Note that the left inequality in Equation (4) has the same form as the $R > (1/2)\log(\sigma_X^2/D_2)$ result in Equation (2) obtained from the sphere packing argument. The right inequality in Equation (4) has the same form as the $R < (1/2)\log(\sigma_X^2/\sigma_N^2)$ term in Equation (3). In general Theorem 3.1 can be viewed as a more precise version of the sphere packing argument which applies for arbitrary distortion measures, source distributions, and channels.

5. CONSTRUCTIONS

In this section we describe constructions of authentication schemes incorporating the ideas from Section 4. One of our goals is to illustrate how error correcting codes can decrease the embedding distortion and increase the reliability of authentication schemes. Consequently, we focus upon the simple model of an i.i.d. source $\{X_i\}_{i=1}^n$ uniformly distributed over $[-L, L]$ and an additive white Gaussian noise channel. Distortion is measured using mean square error. The components of the source could be pixel values, DCT coefficients, wavelet coefficients, etc. The noise could correspond to JPEG compression, smudges, half-toning, format changes, or other benign perturbations. To simplify the exposition and provide a point of comparison we first develop an uncoded scheme and evaluate its performance. Then we show how relatively simple error correcting codes can be incorporated to provide significant performance gains.

5.1. Designing An Authentication Scheme Without Error Correcting Codes

5.1.1. Fidelity

As discussed in Section 4.1.1, the encoder should be able to choose from a large number of possible codewords to keep the distortion small. To design the encoder, we choose a rate R uniform scalar quantizer to use in quantizing each sample of the source. The 2^R reconstruction points are at

$$c_i = \left(i - \frac{2^R-1}{2}\right)\frac{L}{2^{R-1}} \quad \forall i \in \{0, 1, 2, ..., 2^R - 1\}.$$

Note that the reconstruction points are a subset of a shifted and scaled version of the integer lattice. A simple calculation shows that the expected quantization distortion is

$$D_2 = \frac{4L^2}{12}2^{-2R} = \sigma_X^2 2^{-2R} \tag{6}$$

Therefore to obtain a target distortion, D_2, R should be chosen such that $R > (1/2)\log(\sigma_X^2/D_2)$. This achieves the goal of fidelity.

To encode a source, X_1^n, the encoder first quantizes X_1^n. This maps the sequence of n source symbols to a sequence of n blocks of 2^R bits each. We call the resulting bit sequence $F(X_1^n)$. The codebook, $\mathcal{C}$ corresponds to the set of reconstruction points of the quantizer.

5.1.2. Security

To achieve security, we need to create a modified codebook $\mathcal{C}' \subset \mathcal{C}$. Randomly choosing $2^{n(R-\gamma)}$ codewords from $\mathcal{C}$ as in Section 4.1.2 is generally impractical. If $\mathcal{C}'$ was chosen in this manner, then encoding would require searching through all $2^{n(R-\gamma)}$ codewords which is prohibitively complex. Furthermore, this would require a method for exchanging secret keys between the encoder and decoder. Consequently we describe a method based on the well-known tools of public-key digital signatures. The digital signature based method has the advantages of low complexity and a public key; however, other techniques to achieve security are also possible.

A digital signature algorithm consists of a key generation algorithm $(p_k, s_k) = \mathcal{K}_n$, a signing algorithm, $\tau = \mathcal{S}(m, s_k)$, and a signature verifying algorithm $\mathcal{V}(m, \tau, p_k)$. The signing algorithm is used to generate a tag, τ, by signing the message m with the secret key s_k. We denote the length of the tags produced as γ since the tag length plays a role analogous to the parameter γ used in Section 4.1.2. The verifying algorithm returns true only when called on a valid message–tag pair generated with the secret key matching the public key p_k. Furthermore, it is computationally infeasible for an attacker to generate a valid message–tag pair without knowing the secret key.

We define $\mathcal{C}'$ to be all the codewords $c \in \mathcal{C}$ such that two pieces of the codeword form a valid message–tag pair for a digital signature scheme. The details for this procedure are as follows. The encoder chooses γ distinct indices from 1 to n: $\{I_1, I_2, ..., I_\gamma\}$ to store digital signature information. One good method is to choose these randomly and uniformly over all the samples. Then the encoder chooses his public and private key pair, (p_k, s_k), for the digital signature algorithm. The type of digital signature chosen, the public key, the reconstruction points, rate of the quantizer, and the indices where the digital signature tag will be embedded are made publicly available. For a quantized block, $F(X_1^n)$, let $\tau(F(X_1^n))$ correspond to the least significant bits of sample I_1 through I_γ. Let $G(F(X_1^n))$ correspond to setting the least significant bit of sample I_1 of $F(X_1^n)$ to 0 and repeating for samples I_2 through I_γ. $\mathcal{C}'$ consists of all the codewords in $\mathcal{C}$ such that $\mathcal{V}(G(F_1^n), \tau(F_1^n), p_k)$ returns true.

The encoder can encode X_1^n to a valid codeword by first computing $F(X_1^n)$. This corresponds to scalar quantization and is therefore a simple operation. Next, the encoder computes the γ-bit tag as $\tau = \mathcal{S}(G(F(X_1^n)), s_k)$. The encoder sets the least significant bit in sample I_j to be the jth bit of the tag, τ. We call the resulting sequence of bits $Q(G(F(X_1^n)))$. The encoder then constructs $Y_1^n = P(Q(G(F(X_1^n))))$ by reconstructing the sequence of bits using the reconstruction points for the quantizer specified earlier. Figure 3 shows a diagram of the encoding process.

The process of embedding the digital signature tag in the least significant bits is based on a digital watermarking method called Low Bit Modulation (LBM) [11]. More efficient watermarking schemes such as Quantization Index Modulation (QIM) [12] could also be used. For most digital signature schemes, the tag length is small enough that the distortion difference between LBM and QIM is negligible compared to the overall processing distortion.

The decoder receives W_1^n and performs maximum likelihood decoding. The decoder first quantizes W_1^n with the same quantizer used by the encoder to get $F(W_1^n)$. Next the decoder chooses the first bit of the tag, τ, to be the least significant bit of block I_1 and repeats this process for $\{I_2, I_3, ..., I_t\}$. Then the decoder sets the least significant bit in block I_1 to 0 and repeats this process for $\{I_2, I_3, ..., I_t\}$ to get $G(F(W_1^n))$. Finally the decoder verifies the digital signature by checking if $\mathcal{V}(\tau, G(F(W_1^n)), p_k) = 1$. If this is the case, then the decoder accepts the decoded result as $\hat{Y_1^n} = P(Q(G(F(W_1^n))))$. Otherwise the receiver declares a decoding failure. Figure 4 shows a diagram of the decoding process.

To create a forgery, the attacker must find a new codeword which is a valid message–tag pair. This requires cracking the digital signature scheme. A discussion of the security of digital signature schemes is beyond the scope of this paper. Various signature schemes exist which provide strong levels of security such that the probability of the attacker cracking the scheme is $\mathcal{O}(2^{-\gamma})$. Consequently, by making γ large enough, security can be achieved.

The distortion for the unmodulated samples was computed in Equation (6). The distortion for the low bit modulated samples, $\{I_1, I_2, ..., I_t\}$, will be greater. If we model the tag bits as equally likely to be 0 or 1, the

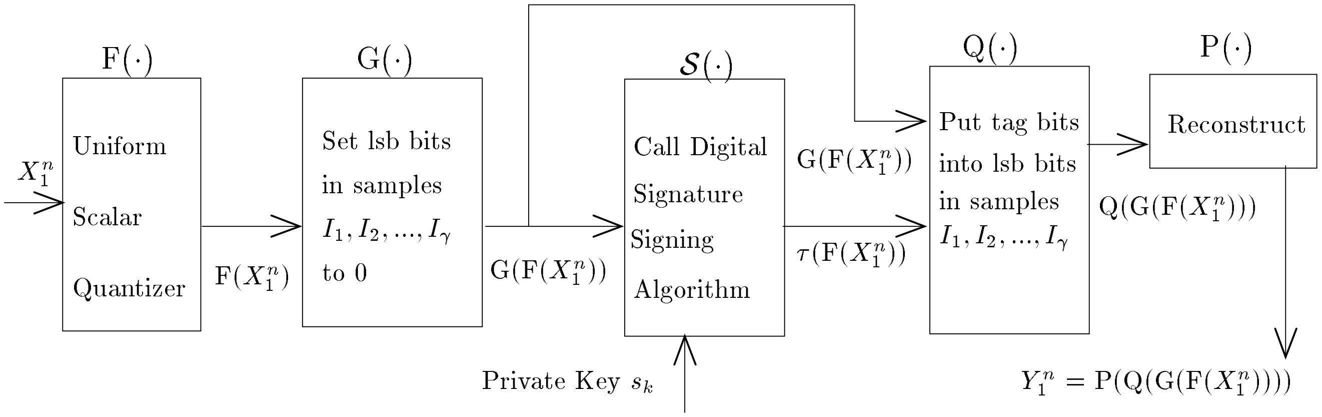

Figure 3. Diagram of the encoding process.

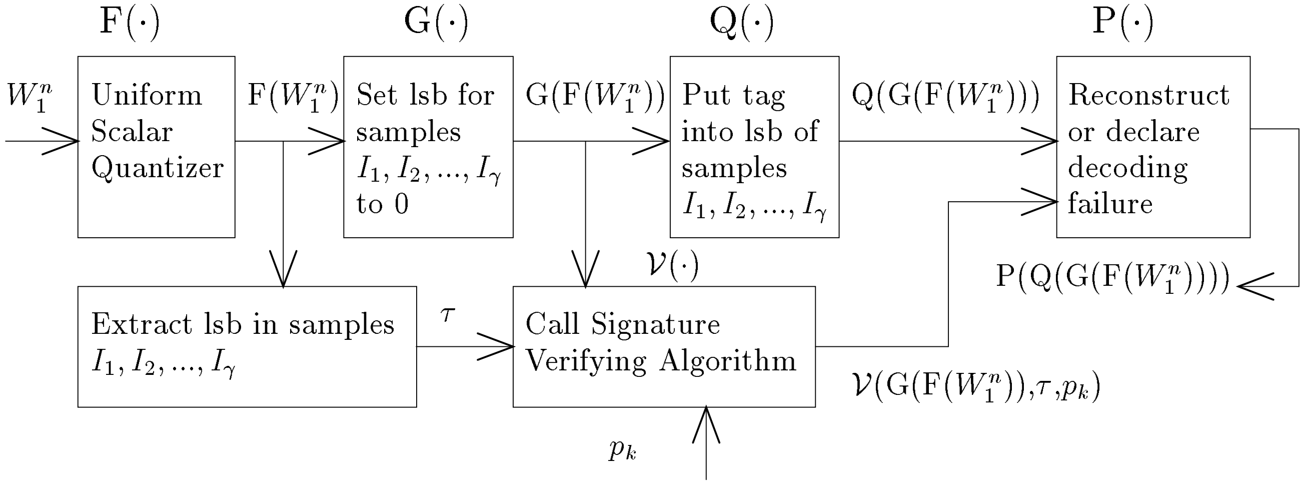
F(·)
G(·)
Q(·)
P(·)
W_1^n
Uniform Scalar Quantizer
$F(W_1^n)$
Set lsb for samples $I_1, I_2, ..., I_\gamma$ to 0
$G(F(W_1^n))$
Put tag into lsb of samples $I_1, I_2, ..., I_\gamma$
$Q(G(F(W_1^n)))$
Reconstruct or declare decoding failure
$P(Q(G(F(W_1^n))))$
$\mathcal{V}(\cdot)$
Extract lsb in samples $I_1, I_2, ..., I_\gamma$
τ
Call Signature Verifying Algorithm
$\mathcal{V}(G(F(W_1^n)),\tau,p_k)$
p_k

Figure 4. Diagram of the decoding process.

expected distortion for the modulated samples is

$$E[(X_{I_j} - Y_{I_j})^2] = \sigma_X^2 2^{-2R} + \sigma_X^2 \frac{15\gamma}{16n} 2^{-2R} \tag{7}$$

Thus the total encoding distortion is computed by averaging the distortion in equations (6) and (7) to get

$$D_2 = \frac{1}{n}\sum_{i=1}^{n} E[(X_i - Q(X_i))^2] = \sigma_X^2 2^{-2R}(1 + \frac{15\gamma}{16n}). \tag{8}$$

For n large enough to make γ/n small, the additional distortion due to the tag is negligible. Consequently both security and fidelity can be achieved simultaneously.

5.1.3. Robustness

Next we calculate the probability of decoding error, p_e. If each symbol of Y_1^n is perturbed by less than half the distance between quantization points, then no errors occur. This condition is equivalent to the event $\cap_{i=1}^{n}\{|N_i| < L2^{-R}\}$. Thus the probability that symbol i is in error will be $p_s = Pr[|N_i| > L2^{-R}]$. For $\gamma \ll n$ we can write p_s in terms of D_2 as $p_s = \Pr[|N_i| > \sqrt{3D_2}]$. Since N_i is Gaussian with mean 0 and variance σ_N^2 we get $p_s = 2Q(\sqrt{3D_2/\sigma_N^2})$. If any symbol is in error then the whole sequence will be in error. Since symbol errors are independent, the total probability of error is $p_e = 1-(1-p_s)^n$. For $p_s \ll 1$, the total probability of error is roughly $p_e \approx np_s$. We can relate the embedding distortion, probability of decoding error and probability of forgery with

$$p_e \approx 2nQ\left(\sqrt{\frac{3D_2}{\sigma_N^2}\frac{1}{1+\frac{15\gamma}{16n}}}\right) = 2nQ\left(\sqrt{\frac{3\text{DNR}}{1+\frac{15\gamma}{16n}}}\right) \approx 2nQ(\sqrt{3\text{DNR}}) \tag{9}$$

This equation shows a tradeoff between robustness to noise and fidelity. Specifically for large n the factor γ/n will be small so security can be achieved without affecting distortion or robustness. Conversely, to make the probability of decoding error negligible we must increase the embedding distortion. In the next section, we show how to decrease the probability of decoding error and at the same time decrease the embedding distortion.

5.2. Designing An Authentication Scheme Using Error Correcting Codes

The distortion to noise ratio required by an uncoded scheme is roughly 10 dB from the lower bound we derived at $p_s = 10^{-6}$. This distortion gap is present because the uncoded scheme does not pack the codewords as densely as suggested by the sphere packing analogy. Specifically, the codewords in the scheme described previously are elements of a shifted and scaled integer lattice. Since integer lattices do not have good packing properties, the uncoded scheme is suboptimal. By using better codewords, we can obtain a superior authentication scheme.

In general, the uncoded scheme can be modified to use a code, $\mathcal{C}$, by changing the uniform scalar quantizer, $F(X_1^n)$, into a quantizer which quantizes X_1^n to the nearest codeword in $\mathcal{C}$. To illustrate this procedure we describe a construction using lattices based on trellis codes. As shown in [13], [14], [15], these lattices are superior to the integer lattice both in terms of reducing the quantization distortion and increasing the distance between codewords.

We use the same encoding scheme as shown in Figure 3 except that the uniform scalar quantizer is replaced with a trellis quantizer designed for a uniform source as in [14]. The tag bits are then computed and embedded into the least significant bits of the codeword as before[†] and the resulting codeword is reconstructed to obtain the encoded signal Y_1^n. The decoding operation is the same as in the uncoded case except the scalar quantizer is replaced with the trellis quantizer.

By using trellis codes, we reduce both the quantization distortion and the probability of decoding error. We derived a lower bound on the minimum possible embedding distortion in Equation (5). To obtain a fair performance metric, we define LDNR_{norm} as how much more DNR a scheme requires than the lower bound at a given probability of symbol error: $\mathrm{LDNR}_{norm} = \mathrm{DNR}/\mathrm{DNR}_{min}$. Every achievable scheme must have $\mathrm{LDNR}_{norm} \geq 1$. Schemes with lower LDNR_{norm} are better in the sense that they come closer to achieving the lower bound on distortion.

Rather than develop a closed form expression, we evaluate p_s numerically. The results of simulations for some reasonable design parameters are plotted in Figure 5. This plot shows the probability of decoding error per symbol, p_s, as a function of the normalized distortion to noise ratio, LDNR_{norm}[‡]. We plot the probability of symbol error, p_s, instead of the total probability of decoding error, p_e, to separate the issue of sequence length from probability of symbol error.

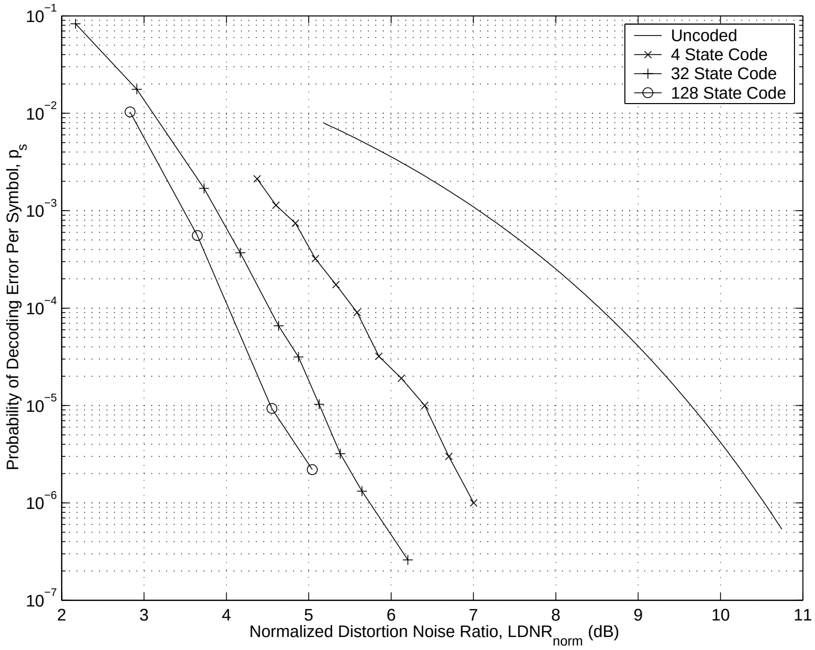

Figure 5. The vertical axis is proportional to the probability that a valid, authentic signal is erroneously declared a forgery by the decoder due to channel noise. By using coding, the distortion required to each a given error probability can be significantly decreased.

For these experiments, the specific design parameters were based on an i.i.d source uniform over the range $[-4, 4]$, quantized using rate 4 trellis coded quantization. The rate 1/2 convolutional codes used for the coset selector are the 1-D codes in Table I of [16]. The results for the uncoded case were computed analytically while the results for the

[†] Due to the structure of the quantizer, determining which bits are the least significant bits requires more care than with the uniform scalar quantizer. See [10] for details.

[‡] Our plot ignores the extra distortion due to embedding the tag bits since we have shown that this extra distortion is negligible for large n.

coded case were collected using $10^4 - 10^6$ sequences of length 1000 (more sequences were used to collect the points with low p_s). The number of trials were chosen so that all results are with ±10% with 95% confidence. Since the source size and the rate were fixed for all trials, the SNR and DNR were varied by changing the noise variance.

The plot shows that trellis coded schemes achieve the same level of robustness but require 3.5-5.3 dB less embedding distortion at $p_s \approx 10^{-6}$. As the DNR increases the distortion reduction of the coded systems increase. Consequently we can bound the probability of error of a coded system with the equation $p_s \leq K_{\text{eff}} \cdot Q(\sqrt{3\text{DNR} \cdot G_c})$ where G_c represents a factor between 3.5-5.3 dB due to the benefit of coding and K_{eff} represents the effective number of nearest neighbors. Following the communications literature we identify G_c as a coding gain since it amplifies the effect of the DNR in determining robustness.

Throughout this paper we have used quadratic distortion for analytical convenience. To show techniques such as coding provide significant gains for other measures such as human perceptual system and for broader classes of sources, we provide encoded images. Figure 6 shows the result of using both schemes to encode the 256 by 256 gray scale Lena image. Since trellis coding effectively includes a type of dithering, the uncoded image is dithered to provide a fair comparison. Without dithering the visual quality of the uncoded image is even worse. Dithering does not significantly help the coded image. The embedding distortions were chosen so that both images would have an overall probability of authentication error of 10^{-3} (corresponding to $p_s \approx 10^{-9}$) when subjected to additive noise with PSNR = 40 dB. Since the coding gain increases with increasing DNR, the coded image will have increasingly better noise robustness than the uncoded image as the PSNR is increased.

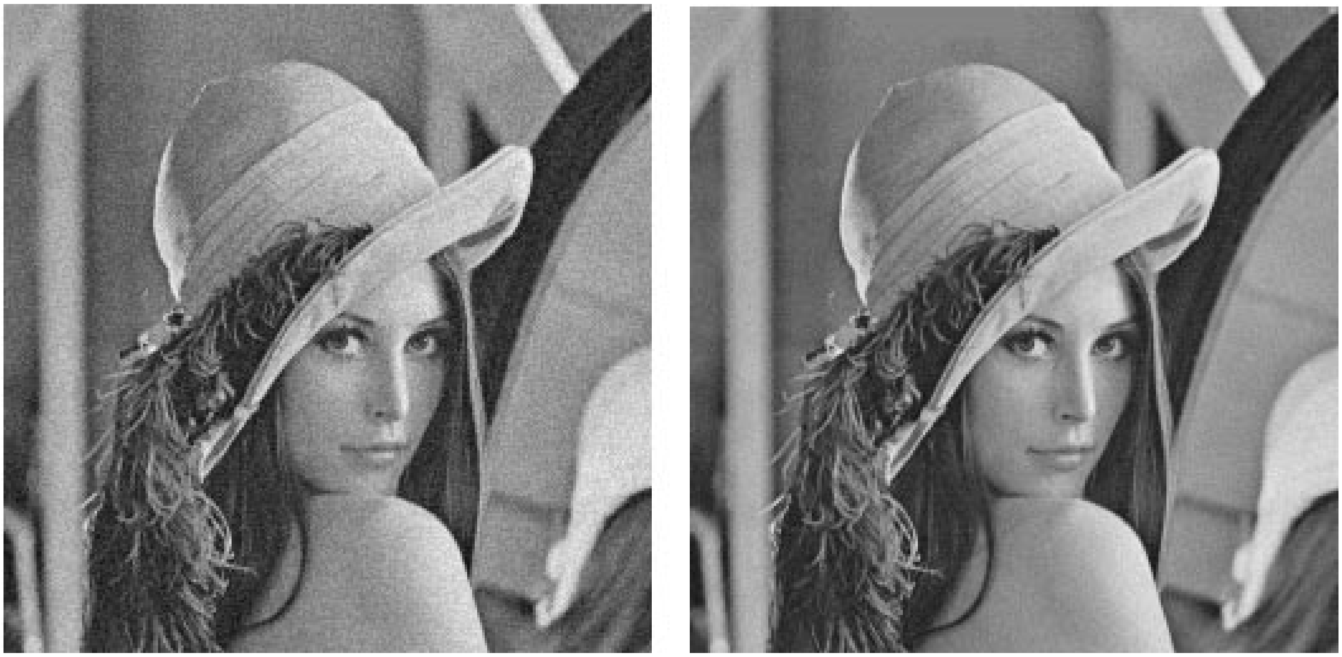

Figure 6. The image on the left was created using the uncoded authentication scheme described in Section 5.1 while the image on the right was created using the coded system described in Section 5.2 with a 128 state trellis code. The peak signal to embedding distortion ratio (PSDR) for the uncoded image is 30.040 dB and 36.190 dB for the coded image. The coded image has superior probability of decoding error provided the PSNR is at least 40 dB. Consequently coding provides a PSDR gain of 6.15 dB.

6. CONCLUSIONS

We presented one formulation of the multimedia authentication problem and provided a single letter expression for the set of achievable distortions in Theorem 3.1. This result can be used to measure the performance of practical schemes as well as to understand the fundamental limits of authentication. To develop intuition for this result, we outlined a sphere packing analogy for authenticating an i.i.d. source in additive noise with quadratic distortion. The results from the sphere packing argument have the same form as the information theoretic results.

We considered a relatively simple model for our constructions in Section 5.1 and Section 5.2 in order to illustrate the applications of the information theoretic framework. Our goal is not to suggest that these simple models are important in themselves, but to show that the tradeoff between fidelity and robustness to noise is a significant issue in multimedia authentication. Designing an efficient authentication scheme requires choosing the codewords to optimally balance these competing goals as suggested by the sphere packing analogy.

In Section 5.1, we explored an uncoded authentication scheme and analyzed its performance. We showed the distortion for the uncoded scheme is about 10 dB above the bounds obtained from Theorem 3.1. Next in Section 5.2, we showed how to modify the uncoded scheme using trellis codes. We presented simulation results which show that simple coding reduces the distortion by roughly 3.5-5.3 dB at $p_s \approx 10^{-6}$. The substantial coding gain obtained by using trellis codes suggests that coding can provide significant advantages in authentication. Further coding gains could be obtained using more powerful codes such as multistage trellis codes, turbo codes, or LDPC codes.

Our framework can be valuable in designing authentication schemes for more complicated models as well. In many areas of signal processing, transform techniques followed by scalar processing have proven valuable. This idea can be applied to authentication. For example, by using an appropriate transform, a correlated source model could be converted to a basis where the source is uncorrelated. Quantization and coding can then be performed in the transform domain. Similarly a correlated noise model could be addressed by using a whitening filter at the decoder to transform the noise vector into uncorrelated samples. Blockwise transforms could be used to address distortion metrics which weight distortions as a function of frequency.

In conclusion the information-theoretic framework provides a link between multimedia authentication and a wide array of powerful results from signal processing and information theory. One such link examined in this paper is the use of error correcting codes in authentication. Some other ideas from these fields which could be applied to authentication include transform techniques, techniques for unknown source models such as universal coding, techniques for unknown channel models such as blind or adaptive equalization, techniques applying channel side information at the transmitter or receiver, and techniques for burst noise channels.

REFERENCES

1. J. Fridrich, “Methods for tamper detection in digital images,” *Proceedings of Multimedia and Security Workshop at ACM Multimedia* , 1999.
2. D. Kundur and D. Hatzinakos, “Digital watermarking for telltale tamper proofing and authentication,” in *Proceedings of the IEEE*, vol. 87, pp. 1167–1180, IEEE, July 1999.
3. P. W. Wong, “A public key watermark for image verification and authentication,” in *ICIP 98*, vol. 1, pp. 445–459, International Conference On Image Processing, 1998.
4. M. Wu and B. Liu, “Watermarking for image authentication,” in *ICIP 98*, vol. 2, pp. 437–441, International Conference On Image Processing, 1998.
5. M. P. Queluz, “Towards robust, content based techniques for image authentication,” in *Multimedia Signal Processing*, pp. 297–302, IEEE Second Workshop on Multimedia Signal Processing, 1998.
6. S. Bhattacharjee and M. Kutter, “Compression tolerant image authentication,” in *ICIP 98*, vol. 1, pp. 435–439, International Conference On Image Processing, 1998.
7. G. L. Friedman, “The trustworthy digital camera: Restoring credibility to the photographic image,” *IEEE Transactions on Consumer Electronics* **39**, pp. 905–910, November 1993.
8. M. Schneider and S. Chang, “A robust content based digital signature for image authentication,” in *ICIP 96*, vol. 3, pp. 227–230, International Conference On Image Processing, 1996.
9. E. Martinian, B. Chen, and G. Wornell, “Authentication with distortion constraints.” In preparation.
10. E. Martinian, “Authenticating multimedia in the presence of noise,” Master’s thesis, Massachusetts Institute of Technology, 2000.
11. M. Swanson, M. Kobayashi, and A. Tewfik, “Multimedia data-embedding and watermarking technologies,” in *Proceedings of the IEEE*, vol. 86, pp. 1064–1087, June 1998.
12. B. Chen and G. W. Wornell, “Digital watermarking and information embedding using dither modulation,” in *Multimedia Signal Processing*, pp. 273–278, IEEE Second Workshop on Multimedia Signal Processing, 1998.
13. R. J. Barron, B. Chen, and G. Wornell, “The duality between information embedding and source coding with side information and its implications and applications.” submitted to IEEE Transactions on Information Theory, Jan. 200.
14. M. W. Marcellin and T. R. Fischer, “Trellis coded quantization of memoryless and Gauss-Markov sources,” *IEEE Transactions on Communications* **38**, pp. 82–93, January 1990.
15. G. D. Forney, Jr., “Coset codes – Part 1: Introduction and geometrical classification,” *IEEE Transactions on Information Theory* **34**, pp. 1123–1151, September 1988.
16. G. Ungerboeck, “Trellis-coded modulation with redundant signal sets Part 2: State of the art,” *IEEE Communications Magazine* **25**, pp. 12–21, February 1987.