Robust, Short and Sensitive Authentication Tags Using Secure Sketch

Qiming Li*
Dept. of Computer and Information Science
Polytechnic University
qiming.li@ieee.org

Ee-Chien Chang
Dept. of Computer Science
National University of Singapore
changec@comp.nus.edu.sg

ABSTRACT

In order to apply cryptographic operations on noisy data, a recent approach employs some additional public data, known as *secure sketch*, to correct the noise so that consistent outcome can be obtained. This approach can be employed to extract authentication tags from noisy multimedia or biometric objects, by including the sketch in the tags. However, there are a few issues that need to be addressed. Firstly, those objects are typically represented in a continuous domain, and hence further quantization is required in order to obtain a short authentication tag. Secondly, for the purpose of authentication, forgery and preimage attacks are major concerns. However, such attacks are not considered in the notion of secure sketch. To handle the first issue, we give a construction using two levels of quantization. The second issue leads to the proposed additional requirement on *sensitivity*. We study how to choose the optimal parameters under the trade-off of robustness, size and sensitivity, and show that in many practical settings, the two-level quantization can be significantly more effective than a seemingly natural method of assigning one bit to each coefficient.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Authentication

General Terms

Security

Keywords

Authentication, secure sketch, robustness, sensitivity, two-level quantization

*Part of the work was done when the author was in the Department of Computer Science, National University of Singapore.

MM&Sec'06, September 26–27, 2006, Geneva, Switzerland.

1. INTRODUCTION

Recently, there are growing interests in applying classical cryptographic operations on noisy data, for example, to extract a consistent key from fingerprint [3, 1, 17], face images [14], or iris codes [9]. To handle the noise, a general approach [11, 5, 12, 15] uses an additional piece of public information, known as *secure sketch* [5] to obtain consistent results. During registration, the original object $\mathbf{x}$ is acquired and a sketch $P_{\mathbf{x}}$ is extracted from $\mathbf{x}$. The sketch is published in clear. Subsequently, during the verification process, another object $\mathbf{y}$ is acquired. By the properties of the sketch, the original $\mathbf{x}$ can be reconstructed exactly from $\mathbf{y}$ and $P_{\mathbf{x}}$, if $\mathbf{y}$ is similar to $\mathbf{x}$. Since the sketch $P_{\mathbf{x}}$ is publicly accessible, including the adversaries, it is important that it does not reveal too much information about the original $\mathbf{x}$.

More specifically, here is a typical sketch construction for objects represented as elements in a normed vector space. First, we choose a quantizer[1] $\mathcal{Q}_1$. During registration, given $\mathbf{x}$, we find its nearest codeword $\mathcal{Q}_1(\mathbf{x})$ and let the sketch be $P_{\mathbf{x}} = \mathbf{x} - \mathcal{Q}_1(\mathbf{x})$. During verification, given $\mathbf{y}$ and $P_{\mathbf{x}}$, we compute $(\mathbf{y} - P_{\mathbf{x}})$ and find its nearest codeword $c = \mathcal{Q}_1(\mathbf{y} - P_{\mathbf{x}})$. Finally, $c + P_{\mathbf{x}}$ is output as the reconstructed data. Note that $\mathbf{y} - P_{\mathbf{x}} = \mathcal{Q}_1(\mathbf{x}) + (\mathbf{y} - \mathbf{x})$. Hence, if $\mathbf{y}$ is close to $\mathbf{x}$, then $c = \mathcal{Q}_1(\mathbf{x})$ and $\mathbf{x}$ can be reconstructed.

Intuitively, such generic error-tolerant techniques can be employed to extract authentication tags from biometric and multimedia data, such that the authentication process is robust against noise. A straightforward method consists of the following steps during registration: Given a multimedia object $\mathbf{x}$, first extract a sketch $P_{\mathbf{x}}$. Next, known message authentication codes (MACs) are applied on $\mathbf{x}$ to obtain $\mathtt{mac}(\mathbf{x})$. An *authentication tag* is then constructed as the concatenation of $P_{\mathbf{x}}$ and $\mathtt{mac}(\mathbf{x})$. During verification, given another $\mathbf{y}$ and the authentication tag, we can retrieve the sketch $P_{\mathbf{x}}$. Together with $\mathbf{y}$, we can reconstruct another object $\widetilde{\mathbf{x}}$. By the properties of secure sketch, if $\widetilde{\mathbf{x}}$ and $\mathbf{x}$ are similar, then $\widetilde{\mathbf{x}} = \mathbf{x}$. Finally, the MAC of $\widetilde{\mathbf{x}}$ is computed and compared with $\mathtt{mac}(\mathbf{x})$, and $\mathbf{y}$ is deemed as authentic if and only if they are the same.

However, there are a few problems with the above method, which we discuss below.

1.1 Two-Level Quantization

Firstly, multimedia objects are typically represented in a continuous domain and their entropy is very high. For ex-

[1]Typically an error-correcting code is employed. However it is more natural to use source coding in our application.

ample, an image can be represented in its wavelet transform, and a face template can also be represented by a real vector resulted from the singular value decomposition of face images [14]. Thus, applying secure sketch directly will give a sketch $P_{\mathbf{x}} = \mathbf{x} - \mathcal{Q}_1(\mathbf{x})$ that is either not in a discrete form, or its entropy is high. Since the final tag contains the sketch, the tag will not be short.

Hence, we propose to further quantize the sketch. Instead of putting both $P_{\mathbf{x}}$ and $\mathtt{mac}(\mathbf{x})$ in the tag, we use a single $\mathbf{t} = \mathcal{Q}_2(P_{\mathbf{x}}) = \mathcal{Q}_2(\mathbf{x} - \mathcal{Q}_1(\mathbf{x}))$ as the tag. During verification, given the tag $\mathbf{t}$ and another object $\mathbf{y}$, we compute $\mathbf{y} - \mathbf{t}$ and determine its distance from the nearest codeword of $\mathcal{Q}_2$. If the distance is within a threshold, then $\mathbf{y}$ is declared to be authentic, otherwise, it is declared as unauthentic. More details of the proposed $\mathcal{Q}_1$ and $\mathcal{Q}_2$ will be given in Section 4. Note that the use of two-level quantization is similar to QIM watermarking [2].

Since information is discarded in getting $\mathbf{t}$ from $P_{\mathbf{x}}$, it is now impossible to reconstruct $\mathbf{x}$ from the tag. Nevertheless, it is sufficient to perform authentications.

1.2 Sensitivity

An important security requirement for multimedia and biometric authentication tags is not explicitly formulated in the notion of secure sketch. In particular, these tags have to be resistant to forgeries. Informally speaking, it should be difficult for an adversary, without the knowledge of a secret key shared by authorized users, to find an object and a tag that pass the verification process. Therefore, it is crucial that the scheme must be resistant to *preimage* attacks. In the scenario of classical MAC, given a message m and its tag $\mathbf{t} = \mathtt{mac}(m)$, a preimage attacker attempts to find another $m' \neq m$ such that m' is also considered by the verification algorithm as authentic when t is given as the tag. For noisy data, given $\mathbf{x}$ and its tag $\mathbf{t}$, a preimage attacker attempts to find another $\mathbf{y}$ that is very different from $\mathbf{x}$, and yet can be authenticated using the same tag $\mathbf{t}$. Such formulation is also discussed by Xie et. al [16].

For example, consider the application where an authentication tag $\mathbf{t}$ is constructed from an image $\mathbf{x}$. A forger wants to modify the image $\mathbf{x}$ substantially and yet the tag $\mathbf{t}$ wrongly indicates that the modified image is authentic. For instance, the forger may make many different random and significant modifications to $\mathbf{x}$, hoping that one of the tampered version of $\mathbf{x}$ can still be authenticated with the same tag. Hence, we require a quantitative assurance that, if the modification to $\mathbf{x}$ is above a certain threshold, the chances that the modified image is wrongly verified as authentic is very low. This leads to the formulation of *sensitivity*. Essentially, we say that a scheme achieves $(\sigma_{\mathrm{S}}, \epsilon_{\mathrm{S}})$-sensitivity when the following holds: Given the original $\mathbf{x}$ and its tag $\mathbf{t}$. Suppose $\widetilde{\mathbf{x}}$ is a corrupted version of the original $\mathbf{x}$ by noise exceeding a required level σ_{S}, then $\mathbf{t}$ is not the authentic tag with probability more than $(1 - \epsilon_{\mathrm{S}})$. Note that sensitivity is a stronger requirement compared to false alarm, which is the probability that a randomly chosen pair of object and tag is authentic.

In the above, we treat the modifications as random noise. However, a smart attacker who knows the algorithm and the quantizers, may exploit these public knowledge to find a preimage more efficiently. To prevent that, we apply an additional secret transformation to "whiten" the modification. This will be discussed in the next section.

1.3 Shared Secret Key

Similar to classical MACs, a security issue in the authentication for noisy objects is how to incorporate a shared secret key in the tag generation such that only authorized parties with the knowledge of the key can generate the tags.

A naive method is to computed the MAC on the "message" $(\mathcal{Q}_1(\mathbf{x}) + \mathbf{t})$. Given $\mathbf{y}$, we can compute the "reconstructed message" as $\mathcal{Q}_1(\mathbf{y} - \mathbf{t}) + \mathbf{t}$ and verify the MAC on it. Another naive method would further encrypt the tag $\mathbf{t}$ with a secure symmetric encryption function E_k where k is a shared secret key, and take the cipher text $E_k\{\mathbf{t}\}$ as the final tag. However, these methods are not sufficiently secure, since it is still possible for an adversary to perform preimage attacks before the encryption/MAC is applied.

A remedy is to apply a secret transformation on $\mathbf{x}$ prior to all operations. The transformed vector is $M_s\mathbf{x}$ where M_s is a secret orthogonal matrix derived from the secret key s. Overall, given $\mathbf{x}$ and a pair of secret keys k and s, the tag is computed as $E_k\left\{\mathcal{Q}_2(\mathcal{Q}_1(M_s\mathbf{x}) - M_s\mathbf{x})\right\}$. During verification, we have an object $\mathbf{y}$ and the encrypted tag $E_k\{t\}$. With the shared key k and s, we can obtain t, compute $M_s\mathbf{y} - t$ and determine its distance from the nearest codeword in $\mathcal{Q}_1$. In this paper, we focus on the sensitivity and the method of extracting the tag. We will not focus on the roles of the shared secret keys. Furthermore, M_s is orthogonal and thus preserves energy and distance. Hence, for simplicity, we will omit M_s and E_k in our analysis.

1.4 Remarks

Note a subtle difference on how the verification is done compared to classical MAC schemes and many other content-based image authentication schemes. Typically, given a tag t, a message or object $\mathbf{y}$ is declared authentic iff t is exactly same as the tag extracted from $\mathbf{y}$. In the proposed method, given the tag t and the object y, we do not extract another tag from y. Instead, decision is made directly based on t and y.

In this paper, we consider objects represented by vectors in $\mathbb{R}^n$. This is reasonable since many known representations for images and other media objects are in similar forms. For other forms of representation, for example sets of points or sets of lines, it could be possible to map it to a vector in $\mathbb{R}^n$. However, such mapping should be key dependent in order to prevent preimage attacks.

The transformation M_s also serves to whiten the objects and noise. It is important that the tag does not reveal any useful information of M_s. The encryption E_k is an additional step to ensure that. Since the adversaries do not know the secret transformation M_s, we assume that any modification of $\mathbf{x}$ amounts to a random white noise.

2. RELATED WORKS

Recently, there has been cryptographic primitives proposed to yield consistent cryptographic keys from noisy data, such as multimedia and biometric objects. The basic idea is that, given an original x, we can derive and publish a "description" P_x of x, so that when we receive a y that is similar to x, we can recover x exactly from y and P_x. After that, some key bounded with the original data can be retrieved and used. Fuzzy commitment scheme due to Juels and Wattenberg [11] is perhaps the earliest formal approach to achieve noise resistance. Their scheme uses an

error-correcting code to correct bit errors, i.e., the similarity is measured by Hamming distance. The scenario where the similarity is measured by set difference is first considered by Juels and Sudan [10], who proposed a fuzzy vault scheme. The notions of *secure sketch* and *fuzzy extractor* are introduced by Dodis et al. [5], who give constructions for three different similarity measures, namely, Hamming distance, set difference, and edit distance. Under their framework, a reliable and almost uniformly distributed key can extracted from noisy data by reconstructing the original data with a given sketch, and then applying a normal "strong-extractor" (such as pair-wise independent hash functions) on the original data. The major security concern for such secure sketch schemes is that the published P_x should not reveal too much information about x. The size of the sketch derived from those schemes is usually proportional to the size of the original data, which can be large.

Traditionally, a binary message can be authenticated by computing a message authentication code (MAC) using a hash function and an encryption algorithm, which is then attached to the message as an authentication tag. The receiver can verify the MAC by repeating the tag generation process on the received message, and compare the result with the MAC. The use of the hash function on the message makes the size of the tag small and independent from the size of the original data.

Another line of research includes the development of *approximate message authentication codes* (AMACs) [7, 16, 4]. The key idea involves the design of a distance preserving hash function, such that a small change in the original data is reflected by a small change in the hash values. Hence, messages can be considered as authenticated if the hash value of the corrupted data is within a certain distance from the hash value that can be decrypted from the authentication tags. Although the problem is similar, the constructions of such codes are in discrete domains, and they focus mainly on the distance preserving property of the hash functions.

There are also a number of empirical studies on how to extract consistent keys from biometric samples for the purpose of authentication. The noisy biometric samples considered include but not limited to handwritten signatures [8], voice [13], and face images [6]. All these techniques extract a single bit from a vector of coefficients, and each of them is represented either as a real number, or an integer in a large domain. Their methods usually involve setting a global threshold t, and each coefficient is mapped to 0 if it is less than t, or 1 otherwise.

3. FORMULATION

An object $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ consists of n coefficients, which do not necessarily correspond to the raw multimedia or biometric data. For example, in the case of digital images, they could be the coefficients after an image transformation, a feature extraction, or the application of a locality preserving function. We assume that the coefficients are independently and identically distributed (i.i.d.), and each of them follows the zero-mean, unit variance normal distribution. We also assume that the noise $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ is the additive white Gaussian noise. Hence, the z_i's are i.i.d. and each of them is normally distributed with zero-mean and variance σ^2.

There are two main components in the proposed scheme, namely, an *encoder* and a *detector*. The encoder `Enc`, given an object $\mathbf{x}$, outputs a binary string $\mathbf{v} = \texttt{Enc}(\mathbf{x})$ of m bits. The detector `Detect`, given an object $\widetilde{\mathbf{x}}$ and a binary string $\mathbf{v}$, outputs a decision of `yes` or `no`. The detector and the encoder have to satisfy the robustness requirement $(\sigma_{\mathrm{r}}, \epsilon_{\mathrm{r}})$ and the sensitivity requirement $(\sigma_{\mathrm{s}}, \epsilon_{\mathrm{s}})$.

1. **Robustness:** If $\mathbf{x}$ is a randomly chosen object and $\mathbf{z}$ is a random white noise with variance at most σ_{r}^2, then $\texttt{Detect}(\mathbf{x}+\mathbf{z}, \texttt{Enc}(\mathbf{x}))$ gives `yes` with probability more than $1 - \epsilon_{\mathrm{r}}$.

2. **Sensitivity:** If $\mathbf{x}$ is a randomly chosen object and $\mathbf{z}$ is a random white noise with variance more than σ_{s}^2, then $\texttt{Detect}(\mathbf{x}+\mathbf{z}, \texttt{Enc}(\mathbf{x}))$ gives `no` with probability more than $1 - \epsilon_{\mathrm{s}}$.

In other words, if an object is corrupted by some noise with level less than σ_{r}, then with high probability, the detector should declare that the object is authentic if it is robust. On the other hand, if the object is corrupted by some noise with level more than σ_{s}, then the detector should be sensitive enough to declare that it is not authentic, with very high probability. Note that it is necessary that $\sigma_{\mathrm{r}} \leq \sigma_{\mathrm{s}}$.

Since the application is in the authentication of multimedia and biometric objects, it is unreasonable to accept an object as authentic if it is corrupted by a noise with an energy higher than that of the original object. Thus, we assume that $\sigma_{\mathrm{s}} < 1$. The probability ϵ_{s} can be viewed as the probability that an illegal tempered object is declared as authentic. It is also an upper bound on the false alarm (i.e. the probability that a randomly chosen object is declared as authentic). Hence, it is necessary that $m > -\log_2(\epsilon_{\mathrm{s}})$. Typically, the false alarm has to be extremely small, partly to guard against attackers who make large number of random attempts. Thus, ϵ_{s} should be small, for example, not more than 2^{-50}. The false alarm is a weaker requirement compare to sensitivity.

In sum, the performance measure of an authentication scheme for n-coefficient object includes m the size of the authentication code, the robustness $(\sigma_{\mathrm{r}}, \epsilon_{\mathrm{r}})$ and the sensitivity $(\sigma_{\mathrm{s}}, \epsilon_{\mathrm{s}})$.

4. THE PROPOSED ENCODER AND DETECTOR

Our proposed encoder and decoder require 2 parameters, the step size Δ and the number of bins B. Each coefficient will be quantized to B bins. In other words, $log_2(B)$ bits will be allocated to each coefficient. Since we have the constraint of using m bits, only $m_0 = (m/log_2(B))$ coefficients can be considered. In this paper, we assume that the total number of available coefficients n is always more than m_0, and we simply select the first m_0 coefficients. Coefficients that are not selected will not be considered in both the encoding and the detection. In section 4.3, we discuss how to use a binary error-correcting code to enhance the performance by using m_0 coefficients during encoding, while utilizing all coefficients in detection.

4.1 The Encoder `Enc`

Given an object $\mathbf{x} = (x_1, \ldots, x_{m_0})$, the number of bins B, and the step size Δ, the encoder follows the procedure below.

1. For each coefficient x_i, compute the nearest codeword u_i and the difference d_i where

$$u_i = \Delta \cdot \lfloor x_i/\Delta \rfloor + \Delta/2 \quad (1)$$
$$d_i = x_i - u_i. \quad (2)$$

Let $U = (u_1, \ldots, u_{m_0})$.

2. Compute and output $\mathbf{v} = (v_1, \ldots, v_{m_0})$, where

$$v_i = \lfloor Bd_i/\Delta + 1/2 \rfloor \mod B. \quad (3)$$

An example of the encoding of one coefficient $x = 0.9\Delta$ is illustrated in Fig. 1. Note that $d_i \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$ and $v_i \in \{0, 1, 2, \ldots, B-1\}$. Hence, $\mathbf{v}$ can be represented in $m \approx m_0 \log_2 B$ bits.

To see that this encoder follows the 2-level quantization framework proposed in the introduction, observe that the collection of all possible U forms the codewords in the first quantizer $\mathcal{Q}_1$. The mapping that maps d_i's to the symbols $\{0, 1, \ldots, B-1\}$ is the second quantizer $\mathcal{Q}_2$.

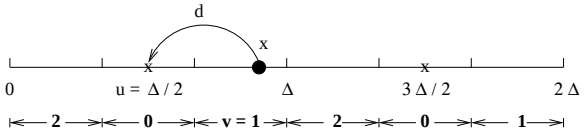

Figure 1: Encoder ($B = 3$). In this example $x = (0.9\Delta)$. Since $x \in [3\Delta/2, \Delta)$, we have the corresponding $v = 1$ and $d = x - \Delta/2$.

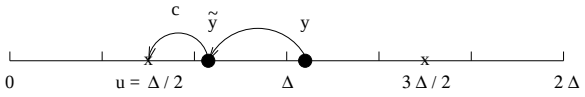

Figure 2: Detector ($B = 3$). Let y be the corrupted version of x in Fig. 1. The decoder computes $\widetilde{y} = y - \frac{v}{B}\Delta$ and finds the nearest codeword u. The correlation $c = (\widetilde{y} - u)^2$.

4.2 The Detector `Detect`

Given an object $\mathbf{y} = (y_1, \ldots, y_{m_0})$ and a binary string $\mathbf{v} = (v_1, \ldots, v_{m_0})$, the detector decides whether $\mathbf{y}$ is authentic. Besides the parameter Δ and B in the encoder, the detector requires a threshold T. We will discuss how to determine such T from Δ, B and the robustness and sensitivity requirements in Section 5. The detector follows the procedure below.

1. For each coefficient, compute:

$$\widetilde{y}_i = y_i - \Delta v_i/B \quad (4)$$
$$u_i = \Delta \cdot \lfloor \widetilde{y}_i/\Delta \rfloor + (\Delta/2) \quad (5)$$
$$c_i = \widetilde{y}_i - u_i. \quad (6)$$

2. Compute the correlation value $C = \frac{1}{m_0}\sum_{i=1}^{m_0} c_i^2$.

3. Output `yes` if $C < T$, output `no` otherwise.

An example on one coefficient is illustrated in Fig. 2.

4.3 Cases where $n > m_0$

When $n > m_0$, we need to select m_0 coefficients to extract the v_i during encoding, and compute the c_i during detection. We simply select the first m_0 coefficients for both encoding and detection. The disadvantage is that, during detection, since lesser coefficients are considered, the standard deviation of C would be larger.

Alternatively, one may use a QIM that incorporates a binary error-correcting code as the first quantizer $\mathcal{Q}_1$. This encoder first chooses a binary error-correcting code. During encoding, it finds the nearest codeword $u'_1 \ldots u'_n$ of the binary sequence $u_1 \ldots u_n$ in (1). Next, it computes d_i and v_i as in (2) and (3) respectively. Note that $\mathbf{v}$ can be compressed to less than $n \log_2(B)$ bits. This is because the Hamming distance between $u'_1 \ldots u'_n$ and $u_1 \ldots u_n$ is less than $n/2$ and thus there are more zeros in $\mathbf{v}$. The rate of the error-correcting code is chosen such that the size of the compressed $\mathbf{v}$ is expected to be m. During the detection, the nearest codeword (with respect to $\mathcal{Q}_1$) is determined and the correlation is computed accordingly. Although all coefficients are utilized during detection, it is not clear whether this method will always out-perform the simple selection algorithm. Further investigation is required.

5. ANALYSIS AND EXPERIMENTS

The performance of the proposed scheme can be analyzed by investigating the distributions of the correlation value C under different levels of noises. Since the coefficients x_i's and noise z_i's are independently distributed, we only need to investigate the distributions of one coefficient. In particular, let $x \in \{x_1, \ldots, x_n\}$ be one coefficient, z be the corresponding noise, and let $v = Q_2(x - \mathcal{Q}_1(x))$ be the authentication code for x as stated in (3).

Consider $y = x + z$ to be the corrupted version of x. Let c be the difference as given in (6). We want to determine the distribution of c^2. Fig. 3 shows this distribution when $\Delta = 1$ and the noise standard deviation $\sigma = 0.1$, for $B = 2, 3$. This distribution is obtained by a combination of simulation and analytical derivations (details omitted). One assumption we make in obtaining the distribution is that the d_i's, which are the differences between the coefficients and their quantized values, are uniformly distributed within the interval $[-\Delta/2, \Delta/2]$. This approximation is reasonable when Δ is small, say $\Delta < 3$.

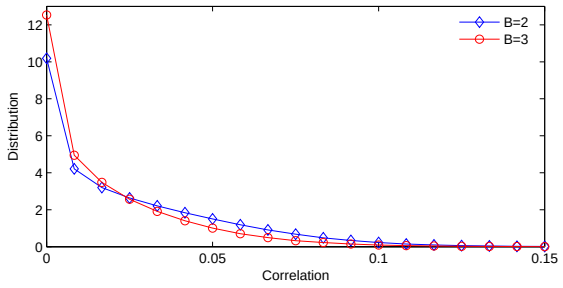

Figure 3: Distribution of correlation.

Suppose u and s are the mean and standard derivation of the distribution of c respectively. Since m_0 is large, we can assume that C is normally distributed with mean u, and

standard deviation $s/\sqrt{m_0}$.

Setting the threshold T: Now, given σ_{S} and ϵ_{S} , we have to set the threshold T small enough such that the sensitivity requirement is met. In particular, we can choose a T smaller than

$$u - \frac{sK_{\mathrm{S}}}{\sqrt{m_0}} \tag{7}$$

where K_{S} is a constant determined solely by ϵ_{S}. For example, if it is required that $\epsilon_{\mathrm{S}} \leq 2^{-163}$, then we can choose $K_{\mathrm{S}} = 15$. Fig. 4 shows the graph of (7) for different noise levels, where $K_{\mathrm{S}} = 15$. Similarly, to meet the requirement of robustness, we can choose the suitable threshold from the graph $u + sK_{\mathrm{R}}/\sqrt{m_0}$ where K_{R} is determined from ϵ_{R}, as illustrated in Fig. 5.

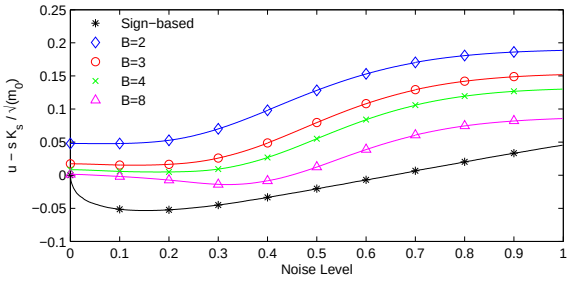

Figure 4: Sensitivity ($K_{\mathbf{S}} = 15$, $m = 1000$, $\Delta = 1$).

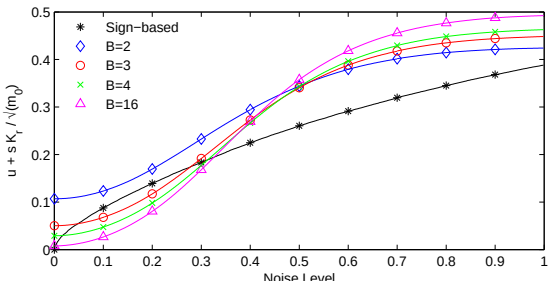

Figure 5: Robustness ($K_{\mathbf{R}} = 10$, $m = 1000$, $\Delta = 1$).

From the above two figures, we can determine the achievable robustness and sensitivity. To illustrate, for a given sensitivity, we can determined the maximum threshold to be set from Fig. 4. Using this threshold, from Fig. 5, we can determined the maximum robustness (for a fixed σ_{R}) that is achievable. This gives Fig. 6. The regions under the curves are the achievable sensitivity and robustness.

Choosing the optimal step size Δ: Observe that if the step size increases by a factor of 2, then the effect is essentially the same as decreasing the level of noise by a factor of 2. However, such a relationship does not hold when the step size is too large. This is because for large step size, the quantization difference d_i's may not be uniformly distributed.

Given the robustness and sensitivity requirements, we want to find an optimal choice of B and Δ. First, let us fixed B,

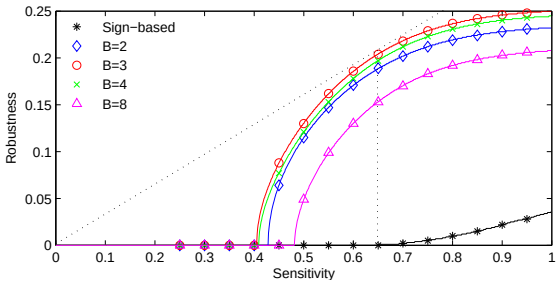

Figure 6: Lookup table: Obtaining robustness from sensitivity ($K_{\mathbf{S}} = 15$, $K_{\mathbf{R}} = 10$, $m = 1000$, $\Delta = 1$).

and then determine the optimal step size. Consider Fig. 6. When $B = 3$ and $\sigma_{\mathrm{S}} = 0.65$, the scheme is most effective in the sense that the ratio of $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}}$ is the largest. Hence, even if the required sensitivity is not 0.65, we can adjust Δ such that it becomes 0.65, thus achieving the maximum achievable robustness. For example, if the required σ_{S} is 0.5, we can choose $\Delta = 1.3$.

Therefore, for given B, ϵ_{R} and ϵ_{S}, the best performance in terms of the ratio $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}}$ can be determined from Fig. 6 by finding the tangent lines that pass the origin, for example, the dotted line in the figure for $B = 3$. From here we can see that $B = 3$ yields the best performance for the proposed method. We can also see that when B increases further (e.g., $B = 4$ and $B = 8$), the performance gradually decreases. This further gives Fig. 7 that shows the maximum $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}}$ for given values of m.

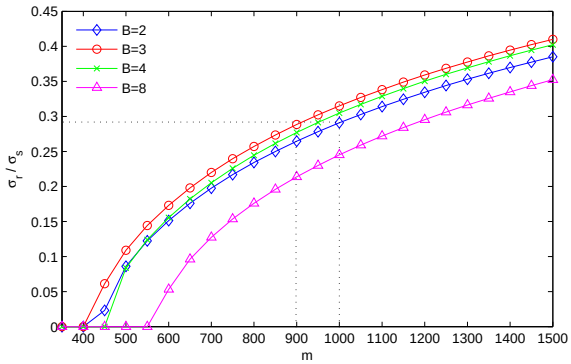

Figure 7: Maximum $\sigma_{\mathbf{R}}/\sigma_{\mathbf{S}}$ given m ($K_{\mathbf{R}} = 10$, $K_{\mathbf{S}} = 15$).

Choosing m and B: Suppose we have m bits to invest to the authentication code, we wish to use the optimal number of bins B such that the ratio $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}}$ is the maximum. On the other hand, suppose we have a fixed requirement on $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}}$, we wish to find a B such that m is minimized. These can be done by examining Fig. 7, where $K_{\mathrm{R}} = 10$, which is equivalent to $\epsilon_r \approx 2^{-73}$, and $K_{\mathrm{S}} = 15$, which is equivalent to $\epsilon_s \approx 2^{-163}$.

For example, as indicated by the dotted lines in Fig. 7, if we can invest $m = 1000$ bits to the authentication code, then using 3 bins will yield the best $\sigma_{\mathrm{R}}/\sigma_{\mathrm{S}} \approx 0.31$. If we fix

$\sigma_r/\sigma_s = 0.29$, then using 3 bins requires 900 bits, but using 2 bins costs 1000 bits.

This figure shows that, typically, it is more effective to allocate more than 1 bit to each coefficients by sacrificing the total number of coefficients utilized during detection.

Comparing with the "sign-based" method: Besides analyzing the performance for various parameters, we also compare our scheme with a simple method that assigns one bit per coefficient, which we call "sign-based" method in our experiments: During the encoding, each coefficient x_i is mapped to 0 if $x_i > 0$, and mapped to 1 otherwise. This gives a binary sequence which serves as the authentication code. During detection, the same algorithm as in the encoding is first applied to obtained a binary sequence. The Hamming distance between the two sequences is the correlation value. Under the settings studied, this method is always less effective than the proposed method. For instance, from Fig. 6, when $m = 1000$ and the required sensitivity is 1, then the maximum robustness is about 0.04. From Fig. 7 we can see that by choosing $B = 4$, similar robustness and sensitivity requirements can be achieved using less than 450 bits.

6. CONCLUSIONS

In this paper we study the problem of designing authentication tags for multimedia or biometric data. Due to possible permissible noise in the data, the authentication tags have to be robust against such noise, but still have to be sensitive to illegal tampering. Besides, it is desirable that these tags are short.

We observe that recent error-tolerant cryptographic techniques such as secure sketch can be employed to derive such tags. However, we need to take preimage attacks into consideration, which are not a concern in the notion of secure sketch. That is the why sensitivity is important in our setting.

We propose the use of two-level quantization, where we compute the difference between the noisy data and a codeword of the first quantizer, and further quantize the difference with the second quantizer. The result forms part of the tag. We show that we can trade-off among robustness, sensitivity and the size of the tags. We further study how to find the optimal parameter by simulations.

7. REFERENCES

[1] Ee-Chien Chang and Qiming Li. Hiding secret points amidst chaff. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 59–72, St. Petersburg, Russia, May 2006.

[2] B. Chen and G.W. Wornell. Quantization Index Modulation methods: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 49(4):1423–1443, 2001.

[3] T.C. Clancy, N. Kiyavash, and D.J. Lin. Secure smartcard-based fingerprint authentication. In *ACM Workshop on Biometric Methods and Applications*, pages 45–52, Berkley, California, 2003.

[4] G. Di Crescenzo, R. Graveman, G. Arce, and R. Ge. A formal security analysis of approximate message authentication codes. In *CTA Communications and Networks*, pages 217–221, College Park, MD, April 2003.

[5] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocypt*, volume 3027 of *LNCS*, pages 125–132, Interlaken, Switzerland, May 2004.

[6] A. Goh and D.C.L. Ngo. Computation of cryptographic keys from face biometrics. In *IFIP Communications and Multimedia Security*, volume 2828 of *LNCS*, pages 1–13, Torino, Italy, October 2003.

[7] R. Graveman and K. Fu. Approximate message authentication codes. In *3rd Annual Fedlab Symposium of Advanced Telecommunication/Inforamtion Distribution*, volume 1, College Park, MD, February 1999.

[8] F. Hao and C.W. Chan. Private key generation from on-line handwritten signatures. *Information Management and Computer Security*, 10(4):159–164, 2002.

[9] Feng Hao, Ross Anderson, and John Daugman. Combining cryptography with biometrics effectively. Technical Report UCAM-CL-TR-640, University of Cambridge, 2005.

[10] A. Juels and M. Sudan. A fuzzy vault scheme. *IEEE International Symposium on Information Theory*, page 408, June 2002.

[11] A. Juels and M. Wattenberg. A fuzzy commitment scheme. *6th ACM Conference on Computer and Communication Security*, pages 23–36, November 1999.

[12] J. P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *4th International Conference on Audio and Video Based Biometric Person Authentication*, volume 2688 of *LNCS*, pages 125–132, June 2003.

[13] F. Monrose, M.K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *IEEE Symposium on Security and Privacy*, pages 202–213, Oakland, CA, May 2001.

[14] Y. Sutcu, T. Sencar, and N. Memon. A secure biometric authentication scheme based on robust hashing. In *ACM Multimedia Security Workshop*, pages 111–116, New York City, NY, August 2005.

[15] P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In *Biometric Authentication Workshop*, pages 158–170, Prague, May 2004.

[16] L. Xie, G.R. Arce, and R.F. Graveman. Approximate image message authentication codes. *IEEE Trans. on Multimedia*, 3(2):242–252, 2001.

[17] Shenglin Yang and Ingrid Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 609–612, March 2005.