Hiding Secret Points amidst Chaff

Ee-Chien Chang and Qiming Li[⋆]

Department of Computer Science
National University of Singapore
changec@comp.nus.edu.sg qiming.li@ieee.org

Abstract. Motivated by the representation of biometric and multimedia objects, we consider the problem of hiding noisy point-sets using a secure sketch. A point-set X consists of s points from a d-dimensional discrete domain $[0, N-1]^d$. Under permissible noises, for every point $\langle x_1, .., x_d \rangle \in X$, each x_i may be perturbed by a value of at most δ. In addition, at most t points in X may be replaced by other points in $[0, N-1]^d$. Given an original X, we want to compute a secure sketch P. A known method constructs the sketch by adding a set of random points R, and the description of $(X \cup R)$ serves as part of the sketch. However, the dependencies among the random points are difficult to analyze, and there is no known non-trivial bound on the entropy loss. In this paper, we first give a general method to generate R and show that the entropy loss of $(X \cup R)$ is at most $s(d \log \Delta + d + 0.443)$, where $\Delta = 2\delta + 1$. We next give improved schemes for $d = 1$, and special cases for $d = 2$. Such improvements are achieved by pre-rounding, and careful partition of the domains into cells. It is possible to make our sketch short, and avoid using randomness during construction. We also give a method in $d = 1$ to demonstrate that, using the size of R as the security measure would be misleading.

1 Introduction

Many biometric data are noisy in the sense that small noises are introduced during acquisition and processing. Hence, two biometric samples that are different but close to each other, are considered to belong to the same identity. This poses technical challenges in applying classical cryptographic operations on them. Recently, new generic techniques such as fuzzy commitment [10], helper data [15] and secure sketch [7] are introduced to handle noisy data. These techniques attempt to remove the noise with the aid of some additional public data P. Here we follow Dodis et al. [7] and call such P a *sketch*. During registration, given original data X, a sketch P is constructed and made public. During reconstruction, given some other data Y and the sketch P, the original X can be reconstructed if Y is *close*[1] to X. In other words, the sketch aids in removing noise from noisy

[⋆] The author is currently with Department of Computer and Information Science, Polytechnic University.

[1] The formal definition of "closeness" will be given in Section 3.

data Y. It is important that such sketch P should be *secure* in the sense that it reveals only limited information about the original X, so that the privacy of the original data can be sufficiently maintained. In other words, it is desirable to bound the *entropy loss* of X given P (Section 3 gives the definitions).

Not surprisingly, the design of a secure sketch is very much dependent on the definition of "closeness". Secure sketch for the following two main types of data have been proposed: (1) The data are from a vector space, and two sequences are close to each other if their distance (e.g., Hamming distance) is less than a threshold. (2) The data X and Y are subsets of a universe $\mathcal{U}$, where $|X| = |Y| = s$, and they are close with respect to a threshold t, if the set difference $s - |X \cap Y| \leq t$.

We observe that in many applications, a combination of the above is required. For example, a fingerprint template is typically represented as a set of minutiae points in a discrete 2-dimensional space, or even 3-dimensional if the less reliable orientation attribute is included [6]. Under noise, each points may be slightly perturbed, and a small number of points may be replaced.

We study secure sketch schemes for such *point-sets*. A point-set X is a set of s points from a discrete d-dimensional domain $[0, N-1]^d$. Under permissible *white noise*, for every point $\langle x_1, .., x_d \rangle \in X$, each x_i, $1 \leq i \leq d$, may be perturbed by at most δ. In addition, under *replacement noise*, at most t points in X may be replaced by randomly selected points. Hence, two point-sets X and Y are close to each other if we can find a subset $X' \subset X$, $|X'| \geq s - t$, such that for each $x \in X'$, there is a unique $y \in Y$ that satisfies $\|x - y\|_\infty \leq \delta$, where $\| \cdot \|_\infty$ is the infinity norm. We assume that a point-set X is always *well-separated*, that is, for any $x, x' \in X$, the distant $\|x - x'\|_\infty \geq 3\delta$. This assumption is reasonable in practice. For example, in a fingerprint template, two minutiae points cannot be too close to each other, otherwise they will be considered as false minutiae and should be corrected [11].

Clancy et al. [5] give the following construction of a two-part sketch for a point-set. The first part of the sketch is a *codebook* $\mathcal{C}$, which is a collection of points that are well-separated. We call each point in $\mathcal{C}$ a *codeword*, and we assume that all codewords are properly indexed in a pre-defined manner. The codebook $\mathcal{C}$ is the union of the original data X and a set of random *chaff* points R, i.e., $\mathcal{C} = (X \cup R)$. Consider another point-set Y that is a version of X corrupted only by white noise. For each point $y \in Y$, the codeword in $\mathcal{C}$ that is closest to y must be the corresponding $x \in X$. Thus, with $\mathcal{C}$, the white noise can be corrected. Hence we call $\mathcal{C}$ the *white noise sketch.* The second part of the sketch is constructed from the indices of the points in X, where the index of a point $x \in X$ is its location in the codebook $\mathcal{C} = (X \cup R)$. By using existing schemes for set difference, replacement of at most t points can be corrected. Hence we call it the *replacement sketch.* In this paper, we will focus on the construction of the white noise sketch. That is, we study how to hide the original points X amidst some chaff points R.

Clancy et al. propose the following method to generate R: The points in R are iteratively selected. During each iteration, a chaff point is chosen uniformly at random. If it is too close to any previously selected points or a point in X, then it is discarded. Otherwise it is selected. The iteration is repeated until sufficient points are selected or it is impossible to add more points. The above process of selecting a set of random points is essentially the online parking process which has intrinsic statistical properties [14, 13, 8].

Due to the dependencies among the selected points, the analysis of online parking process is difficult. This is especially so in higher dimensions. Many fundamental questions remain open, for example, the Palasti's Conjecture [13]. In our context of secure sketch, there is no known non-trivial bound of the entropy loss by revealing $(X \cup R)$. Furthermore, although the points generated seem to be "random", due to the dependencies, the original X may be statistically distinguishable from R. Indeed, an empirical study suggests a method to find X among $(X \cup R)$ [4].

Therefore, we propose another method of generating the points. First, many points are generated independently. Next, some points are removed so that among the remaining points, no two points are near to each other. In this way, we can eliminate the dependencies among the chaff points and give an upper bound $\mathcal{L}_H$ on the information revealed (i.e., the entropy loss) by the codebook $\mathcal{C} = (X \cup R)$. There are many ways to generate the points independently. The challenging issue now is to find a method whereby the randomness invested during generation is not much less than the number of bits required to represent the codebook.

For the second part of the sketch that corrects the replacement noise, we employ known techniques for set difference. Let $\mathcal{L}_{SD}(s, t, n)$ be the entropy loss of the sketch for set difference, where $n = |\mathcal{C}|$ is the size of codebook. There are sketch schemes such that $\mathcal{L}_{SD}(s, t, n)$ is in $O(t \log n)$ (e.g., those proposed by Juels and Wattenberg [9], Dodis et al. [7], and Chang et al. [3]).

In this paper, we propose a generic method to generate the white noise sketch and show that the upper bound of the entropy loss $\mathcal{L}_H < s(d \log \Delta + d + \log(e/2))$, where $\Delta = 2\delta + 1$, e is the base of natural logarithm and $\log(e/2) \approx 0.443$. The overall entropy loss is at most $\mathcal{L}_H + \mathcal{L}_{SD}(s, t, N^d/(4\delta+1)^d)$. The bound is quite tight in the sense that there is a distribution of X such that the entropy loss of $\mathcal{C}$ is at least $\mathcal{L}_H - \epsilon$ where ϵ is a positive constant that is at most 3. When $t = 0$ (i.e., no replacement noise), a lower bound of the entropy loss is $sd \log \Delta$. Hence, the gap between our construction and the optimal is at most $s(d + \log(e/2))$. By pre-rounding and carefully partitioning the domain $[0, N-1]$ into cells, we can improve the entropy loss in $d = 1$ to at most $s(1 + \log(\Delta - 1)) + \mathcal{L}_{SD}(s, t, N/(3\delta))$. We further apply the technique of partitioning to some special cases in two dimensions ($d = 2$) and obtain some improvements. Such technique probably can be extended to $d = 2$ in general, and to higher dimensions. In addition, we give two methods to reduce the size of the sketch. In one of them, we can avoid using randomness during sketch construction, thus some limited form of reusability can be achieved [2]. We also give another method in one dimension to demonstrate that, using the size of R as the security measure would be misleading.

2 Related Works

Recently, a few new cryptographic primitives for noisy data are proposed. Fuzzy commitment scheme [10] is one of the earliest formal approaches to error tolerance. The fuzzy commitment scheme uses an error correcting code to handle Hamming distance. The notions of *secure sketch* and *fuzzy extractor* are introduced by Dodis et al. [7], which gives constructions for Hamming distance, set difference, and edit distance. Under their framework, a reliable key is extracted from noisy data by reconstructing the original data with a given sketch, and then applying a normal extractor (such as pair-wise independent hash functions) on the data.

An important requirement of a secure sketch scheme is that the amount of information about X revealed by publishing the sketch P should be limited. Dodis et al. [7] propose a notion of entropy loss to measure the security of the sketch. They also provide a convenient way to bound the entropy loss for **any** distribution of X. Such worst case analysis is important in practice because typically, the actual distribution of the biometric data is not known.

The issue of *reusability* of sketches is addressed by Boyen [2]. It is shown that a sketch scheme that is provably secure may be insecure when multiple sketches of the same biometric data are obtained. It is also shown by Boyen that a sketch that can be constructed deterministically can achieve some limited form of reusability [2].

The set difference metric was first considered by Juels and Wattenberg [9], who gave a *fuzzy vault* scheme. Later, Dodis et al. [7] proposed three constructions. The entropy loss by all these schemes are roughly the same. They differ in the sizes of the sketches, decoding efficiency and also the degree of ease in practical implementation. The BCH-based scheme [7] has small sketches and achieves "sublinear" (with respect to the size of the universe) decoding by careful reworking of the standard BCH decoding algorithm. Chang et al. [3] gave a scheme for multi-sets, using the idea in set reconciliation [12].

A *fuzzy fingerprint vault* scheme is proposed by Clancy et al. [5], which is to be used in secure fingerprint verification using a smart card. The security of the scheme is analyzed by considering force attackers. Yang and Verbauwhede [16] employed similar approaches with different fingerprint representation.

3 Preliminaries

Entropy and entropy loss. We follow the definitions of entropy by Dodis et al. [7]. They propose to examine the *average min-entropy* of X given P, which gives the minimum length of an almost uniform secret key that can be extracted even if the sketch P is made public.

Let $\mathbf{H}_\infty(A)$ be the min-entropy of the random variable A, i.e., $\mathbf{H}_\infty(A) = -\log(\max_a \Pr[A = a])$. For two random variables A and B, the average min-entropy of A given B is defined as $\widetilde{\mathbf{H}}_\infty(A \mid B) = -\log(\mathbb{E}_{b\leftarrow B}[2^{-\mathbf{H}_\infty(A|B=b)}])$.

The entropy loss of X given sketch P is defined as $\mathcal{L} = \mathbf{H}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X|P)$. When it is clear in the context, we simply call $\mathcal{L}$ the entropy loss of sketch P. This definition is useful in the analysis of entropy loss, since for any ℓ-bit string B, we have $\widetilde{\mathbf{H}}_\infty(A \mid B) \geq \mathbf{H}_\infty(A) - \ell$. For any secure sketch scheme, let R be the randomness invested in constructing the sketch, it can be shown that when R can be recovered from X and P, then

$$\mathcal{L} = \mathbf{H}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X \mid P) \leq |P| - \mathbf{H}_\infty(R). \tag{1}$$

Inequality (1) implies that the entropy loss can be bounded from above by the difference between the size of the sketch and the randomness we invested during construction. This gives a general method to find an upper bound of $\mathcal{L}$ that is independent of X, and hence it applies to any distribution of X. Therefore, $\mathcal{L}$ is an upper bound of entropy loss in the "worst-case".

Secure sketch. Let $\mathcal{M}$ be a set with a *closeness* relation $\mathsf{C} \subseteq \mathcal{M} \times \mathcal{M}$. When $(X, Y) \in \mathsf{C}$, we say the Y is close to X, or (X, Y) is a close pair. Similar to Dodis et al. [7], define

DEFINITION 1 *A sketch scheme is a tuple* $(\mathcal{M}, \mathsf{C}, \mathsf{Enc}, \mathsf{Dec})$, *where* $\mathsf{Enc} : \mathcal{M} \to \{0,1\}^*$ *is an encoder and* $\mathsf{Dec} : \mathcal{M} \times \{0,1\}^* \to \mathcal{M}$ *is a decoder such that for all* $X, Y \in \mathcal{M}$, $\mathsf{Dec}(Y, \mathsf{Enc}(X)) = X$ *if* $(X, Y) \in \mathsf{C}$. *The string* $P = \mathsf{Enc}(X)$ *is to be made public and we call it the sketch. We say that the sketch scheme is* $\mathcal{L}$*-secure if for all random variable* X *over* $\mathcal{M}$, *the entropy loss of* P *is at most* $\mathcal{L}$. *That is,* $\mathbf{H}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X \mid \mathsf{Enc}(X)) \leq \mathcal{L}$.

Closeness relations. For any two points x and y from the d-dimensional space $[0, N-1]^d$, we define the closeness C_δ, where $(x, y) \in \mathsf{C}_\delta$ if $\|x - y\|_\infty \leq \delta$. We further define the closeness $\mathsf{PS}_{\delta,s,t}$ for two point-sets.

DEFINITION 2 *For any two sets of s points* $X = \{x_1, \ldots, x_s\}$ *and* $Y = \{y_1, \ldots, y_s\}$, *we say that* $(X, Y) \in \mathsf{PS}_{\delta,s,t}$ *if there exists a 1-1 correspondence* f *on* $\{1, \ldots, s\}$ *such that* $|\{i \mid (x_{f(i)}, y_i) \in \mathsf{C}_\delta\}| \geq s - t$.

A lower bound of the entropy loss. Here we give a lower bound $\mathcal{L}_0$ of the entropy loss. We say that $\mathcal{L}_0$ is a lower bound if, for any sketch scheme $(\mathcal{P}([0, N-1]^d), \mathsf{PS}_{\delta,s,t}, \mathsf{Enc}, \mathsf{Dec})$, there exists a distribution of X such that the entropy loss of $P = \mathsf{Enc}(X)$ is at least $\mathcal{L}_0$.

For any distribution of X, let $\mathcal{X}_b$ to be the set of all possible original point-sets given sketch $P = b$. We observe that

$$\max_a \Pr[X = a \mid P = b] \geq \frac{1}{|\mathcal{X}_b|}.$$

Substitute it into the definition, we have

$$\widetilde{\mathbf{H}}_\infty(X|P) \leq \max_{b, \Pr[P=b] \neq 0} \log |\mathcal{X}_b|. \tag{2}$$

Now, by considering X that is uniformly distributed over all well-separated sets of size s in $[0, N-1]^d$, using (2), we can show that (details omitted) when $s < (\frac{N}{2\Delta})^d$ and $t < (\frac{N}{2\Delta})^{\frac{d}{2}}$, $\mathcal{L}_0$ is in

$$sd \log \Delta + \Omega(td \log \frac{N}{2\Delta}). \tag{3}$$

Recall that $\Delta = 2\delta + 1$. An intuitive interpretation of the bound is that, it is the minimum number of bits needed to describe the noise. The first term in (3) is for the white noise, and the second term is for the replacement noise. When $t = 0$ (i.e., there is no replacement noise), the bound becomes $sd \log \Delta$.

4 The Basic Construction

Recall that our sketch consists of two parts $P_H P_S$, where P_H is the white noise sketch that removes the white noise. During encoding, a large number of points R is generated to form the codebook $\mathcal{C} = (X \cup R)$, and P_H is its description. During decoding, the points in Y are matched with the nearest codewords in $\mathcal{C}$, so that white noise can be removed. The sketch P_S for set difference is constructed using known schemes on $\mathcal{C}$ to correct the replacement noise. We also assume that X is well-separated.

Here we focus on the construction of P_H. We will first give our basic construction in one dimension ($d = 1$), and then show that it can be extended to higher dimensions.

The main idea of our construction is to first independently generate many points, but avoiding regions near the original X. We can also view the generation of these points as a two dimensional Poisson process. Next, remove some points so that among the remaining points, no two points are near to each other. The retained points form the codebook $\mathcal{C}$. Since the points are generated independently, it is easier to bound the entropy loss. To minimize the entropy loss, we need to find a way so that the size of the sketch is not much larger than the randomness we invested during the construction.

4.1 Construction of P_H in One Dimension ($d = 1$)

For any point $x \in [0, N-1]$, call the set $S_1(x) = \{x+1, x+2, \ldots, x+2\delta\}$ the *half-sphere* of x.

Given $X = \{x_1, \ldots, x_s\}$, the white noise P_H is constructed as below. We first construct a sequence $\langle h_0, h_1, \ldots, h_{N-1} \rangle$, where each $h_i \in [0, p_1 - 1]$, and p_1 is a parameter that is chosen to be $p_1 = |S_1(x)| + 1 = 2\delta + 1$ for optimal performance.

1. For each $x \in X$, set $h_x = 0$, and for each $a \in S_1(x)$, h_a is uniformly chosen at random from $\{1, \ldots, p_1 - 1\}$.
2. For each h_i that has not been set in step 1, uniformly choose its value from $\{0, \ldots, p_1 - 1\}$.

For each $w \in [0, N-1]$, we select it to be in the codebook if and only if $h_w = 0$ and $h_a \neq 0$ for all $a \in S_1(w)$. Hence, if w is a codeword, there would be no other codeword in the half-sphere $S_1(w)$. The sequence $\langle h_0, \ldots, h_{N-1} \rangle$ is published as the white noise sketch P_H. Note that in practice, we can simply publish a description of the codebook $\mathcal{C}$ as the sketch. However, we choose to publish the entire sequence $\langle h_0, \ldots, h_{N-1} \rangle$ for the ease of analysis.

From the codebook $\mathcal{C}$, we can construct P_S, the second part of the sketch, using known schemes for set difference.

During decoding, given Y, each point $y \in Y$ is matched with its nearest codeword in $\mathcal{C}$. Suppose y is a noisy version of an $x \in X$, i.e. $|y - x| \leq \delta$, it is easy to verify that x is its closest point in $\mathcal{C}$. Hence, P_H can correct the white noise. Lemma 3 gives the entropy loss, and Lemma 4 shows that the bound is quite tight. Note that Lemma 3 and 4 still hold if we choose to publish a shorter description of the codebook instead of the entire sequence. In other words, publishing the entire sequence might seem to reveal more information about X, the "worst-case" entropy loss would not be much different.

Lemma 3 *The entropy loss of X given P_H is at most*

$$s\left(\log \mathit{\Delta} + (\mathit{\Delta} - 1)\log(1 + \frac{1}{\mathit{\Delta} - 1})\right)$$

which is less than $s\left(\log \mathit{\Delta} + \log e\right)$*, where e is the base of natural logarithm.*

Proof: Since the randomness invested in constructing P_H can be recovered from X and P_H, we can apply (1) in Section 3. In particular, we look at the difference between the size of the sketch P_H, which is $N \log p_1$, and the randomness invested in constructing P_H. For any h_i in P_H, if it is not set in Step 1 of the above construction, then $|h_i| = \log p_1$, which equals to the invested randomness, and hence it does not contribute to the difference. For each h_x such that $x \in X$, it is set to 0, which contributes $\log p_1$ to the difference. For each h_a such that $a \in S_1(x)$ for some $x \in X$, we use $\log(p_1 - 1)$ bits of randomness, hence the difference introduced is $\log \frac{p_1}{p_1 - 1}$.

Therefore, the total difference (hence the entropy loss) is no greater than

$$s\left(\log p_1 + 2\delta \log \frac{p_1}{p_1 - 1}\right) .$$

When $p_1 = 2\delta + 1$, and substituting $\mathit{\Delta} = 2\delta + 1$, we have

$$\mathcal{L}_H \leq s\left(\log \mathit{\Delta} + (\mathit{\Delta} - 1)\log(1 + \frac{1}{\mathit{\Delta} - 1})\right) .$$

Since $(1 + \frac{1}{\mathit{\Delta} - 1})^{\mathit{\Delta} - 1}$ approaches e from below when $\mathit{\Delta}$ approaches infinity, we have the above claimed bound. □

Lemma 4 *There exists a distribution of X, where the entropy loss of X given P_H is at least $s(\log \mathit{\Delta} + (\mathit{\Delta} - 1)\log(1 + \frac{1}{\mathit{\Delta} - 1})) - \epsilon$ for some positive constant ϵ.*

Proof: Consider the distribution $X = \{x_1, x_1 + 2\Delta, \cdots, x_1 + 2(s-1)\Delta\}$, where x_1 is uniformly chosen from a set $A = \{a_1, \cdots, a_\lambda\}$ of λ points. Hence, $\mathbf{H}_\infty(X) = \log \lambda$. Recall that, given P_H, a point w is a codeword if and only if $h_w = 0$ and $h_b \neq 0$ for all $b \in S_1(w)$. Certainly, each point x_i in X itself *must* be a codeword. Hence, each point $a_i \in A$ is a possible candidate of the original point x_1 if and only if all the points in $\{a_i, a_i + 2\Delta, \ldots, a_i + 2(s-1)\Delta\}$ are codewords in $\mathcal{C}$.

For any $a_i \neq x_1$, the probability that a_i is a possible candidate of x_1 is at most $\frac{1}{\Delta^s}(1 - \frac{1}{\Delta})^{(\Delta-1)s}$. Let C be the number of candidates of x_1 for a given P_H, then we have

$$\mathbb{E}[C] \leq 1 + \frac{\lambda - 1}{\Delta^s}(1 - \frac{1}{\Delta})^{(\Delta-1)s} \leq 1 + \frac{\lambda}{\Delta^s}(1 - \frac{1}{\Delta})^{(\Delta-1)s}.$$

Now by choosing

$$\lambda = 2^{s(\log \Delta + (\Delta-1)\log(1+\frac{1}{\Delta-1}))}$$

we have $\mathbb{E}[C] \leq 2$. By Markov's Inequality, we have

$$\Pr[C \leq 4] \geq 1 - \mathbb{E}[C]/4 \geq 1/2.$$

We note that

$$\begin{aligned} &\mathbb{E}_{b \leftarrow P_H}\left[2^{-\mathbf{H}_\infty(X|P_H=b)}\right] \\ =&\mathbb{E}_{b \leftarrow P_H}\left[\max_a \Pr[X = a|P_H = b]\right] \\ \geq&\frac{1}{4}\Pr[C \leq 4] \geq \frac{1}{8}\,. \end{aligned}$$

Therefore, the left-over entropy $\widetilde{\mathbf{H}}_\infty(X|P) \leq -\log\frac{1}{8} = 3$. Considering that $\mathbf{H}_\infty(X) = \log\lambda = s\left(\log\Delta + (\Delta-1)\log(1+\frac{1}{\Delta-1})\right)$, and let $\epsilon = 3$, we have the claimed bound. □

4.2 Extension to Higher Dimensions

The construction in one dimension can be easily extended to higher dimensions by giving an appropriate notion of half-sphere. Let us first define a total order for the points in $[0, N-1]^d$. Define $\langle x_1, x_2, \ldots, x_d\rangle \succ \langle x'_1, x'_2, \ldots, x'_d\rangle$ if and only if there exists an i such that $x_i > x'_i$ and $x_j = x'_j$ for all $1 \leq j < i$. We define the half-sphere of x in d-dimensions $S_d(x) = \{y \mid 0 < \|y - x\|_\infty \leq 2\delta \text{ and } y \succ x\}$.

The sketch P_H is a set of N^d symbols. For each $h_y \in P_H$, we have $y \in [0, N-1]^d$ and $h_y \in \{0, \ldots, p_d - 1\}$ for some parameter p_d that is to be chosen later. We construct P_H as below.

1. For each $x \in X$, set $h_x = 0$. For every $a \in S_d(x)$, uniformly choose h_a at random from $\{1, \ldots, p_d - 1\}$.

2. For each h_y that is not set in step 1, choose its value uniformly at random from $\{0, \ldots, p_d - 1\}$.

From P_H we can determine the codebook $\mathcal{C}$ as follows. A point $x \in [0, N-1]^d$ is in $\mathcal{C}$ if and only if $h_x = 0$ and for every $a \in S_d(x)$, we have $h_a \neq 0$. We can then construct the second part P_S of the sketch for set difference. Suppose y is a noisy version of an $x \in X$, that is, $\|y - x\|_\infty \leq \delta$, it is not difficult to verify that its closest point in $\mathcal{C}$ is x.

In fact, this construction is essentially the same as the construction for $d = 1$, except that $S_d(x)$ is larger when $d > 1$. By simple counting we have

$$|S_d(x)| = \frac{(4\delta + 1)^d - 1}{2}.$$

Similar to the one-dimensional case, we choose $p_d = |S_d(x)| + 1$. By substituting $\varDelta = 2\delta + 1$, we have

Theorem 5 *The entropy loss of X given sketch P_H is at most*

$$s\left(\log p_d + (p_d - 1)\log(1 + \frac{1}{p_d - 1})\right) \leq s\left(d \log \varDelta + d + \log \frac{e}{2}\right)$$

in d-dimensions, where $p_d = \frac{(4\delta+1)^d+1}{2}$, and e is the base of natural logarithm.

Similarly to the one-dimensional case, the above bound is tight. That is, there is a distribution of X such that the entropy loss is at least

$$s\left(\log p_d + (p_d - 1)\log(1 + \frac{1}{p_d - 1})\right) - \epsilon$$

for some positive constant ϵ. Taking into consideration the entropy loss of sketch for set difference, we have

Corollary 6 *In d-dimensions, the entropy loss of X given sketch $P_H P_S$ is at most* $s\left(d \log \varDelta + d + \log \frac{e}{2}\right) + \mathcal{L}_{SD}\left(s, t, \frac{N^d}{(2\delta+1)^d}\right)$.

5 Improved Schemes

The generic construction in Section 4.2 can indeed be further improved in terms of entropy loss. We employ two techniques. The first is *pre-rounding*. That is, each point in X and Y is rounded prior to both encoding and decoding. We observe that, the effect of the white noise is reduced on the rounded points. The second technique is *partitioning*, where we carefully partition the domain into cells. Instead of selecting points independently from the space, in the improved scheme, at most one point is selected in each cell. Both techniques are useful in reducing the randomness required in constructing P_H.

5.1 Improvement in One Dimension ($d = 1$)

First, we give an improvement for $\delta = 1$ using partitioning, and we observe that this scheme can be extended to any $\delta > 1$ by pre-rounding.

We partition the domain $[0, N-1]$ into cells of size 3, such that the i-th cell contains the 3 consecutive points $\{3i, 3i+1, 3i+2\}$. There are $n' = \lceil N/3 \rceil$ cells in total. We want to assign one bit h_i to the i-th cell for all $0 \le i \le n'-1$, and construct P_H as the binary sequence $\langle h_0, h_1, \ldots h_{n'-1} \rangle$.

Our main idea is to use this binary sequence to describe the codewords in the cells. At the first glance, it seems impossible since each cell would have three different possible codewords, which cannot be described by one bit. However, since two codewords cannot be too close to each other, we can eliminate certain cases by considering each two consecutive cells together. In this way, we can use only two bits to describe the codewords in two consecutive cells.

Here is how the values in the binary sequence are determined: For each $x \in X$, it is in the $i = \lfloor x/3 \rfloor$-th cell, and $r = x \mod 3$ indicates the location of x in the i-th cell. We set two values h_i and h_{i+1} in P_H according to Table 1(a). Since there are s points in X, the above process sets the values for $2s$ bits in $\langle h_1, h_2, \ldots h_{n'-1} \rangle$. For each h_i that is not set, we randomly assign a value from $\{0, 1\}$ to it.

Now, from $\langle h_0, h_1, \ldots h_{n'-1} \rangle$, we determine a set of "potential codewords". For each i-th cell, the potential codeword in the cell is determined by h_i and h_{i+1} using Table 1(b). Next, for a potential codewords x, if there is another potential codeword x' such that $x' \in S_1(x)$, then x is removed. The retained points form the codebook $\mathcal{C}$. By the design of Table 1(a) & (b), each $x \in X$ will be a codeword.

	h_i	h_{i+1}
$r = 0$	0	0
$r = 1$	0	1
$r = 2$	1	1

(a)

	$h_{i+1} = 0$	$h_{i+1} = 1$
$h_i = 0$	$3i$	$3i+1$
$h_i = 1$	$3i+2$	$3i+2$

(b)

Table 1. Improved Scheme for $d = 1$.

Similar to the basic construction, in practice, we can publish a description of $\mathcal{C}$ as the sketch. However, for the ease of analysis, we choose to publish $\langle h_0, h_1, \ldots, h_{n'-1} \rangle$. During decoding, each y is simply matched to the nearest codeword in $\mathcal{C}$.

Since we invested $n' - 2s$ bits of randomness, and the size of sketch is $2s$, the entropy loss is at most $2s$.

Extension to any δ. To extend this scheme to any δ, we employ rounding. The rounding is essentially a many-to-one mapping. For each point $w \in [0, N-1]$, we map it to $\hat{w} = \lfloor w/\delta \rfloor$. Note that under white noise, the perturbed point w' can only be mapped to $\hat{w}-1, \hat{w}$ or $\hat{w}+1$. In other words, under the mapping, the white noise (that appears to be on $\hat{w}$) is reduced to -1, 0, or $+1$, which corresponds to white noise with unit strength. Since the mapping is many-to-one, for each $x \in X$, we keep the rounding error $x - \delta(\lfloor x/\delta \rfloor)$ and publish it as part of the sketch. Hence, the additional entropy loss due to the rounding is at most $\log \delta$ for each $x \in X$. In total, we have

THEOREM 7 *The entropy loss for the above scheme is at most* $(2 + \log \delta)s + \mathcal{L}_{SD}(s, t, N/(3\delta))$.

5.2 Improvement for $d = 2$ and $\delta = 1$

For $\delta = 1$ in two dimensions, with a parameter $\alpha \in [0, 4]$, we partition the space such that every 5 points of the form $\{(w, 5k+\alpha), (w, 5k+\alpha+1), (w, 5k+\alpha+2), (w, 5k+\alpha+3), (w, 5k+\alpha+4)\}$ for some non-negative integer k, are grouped into a cell (Fig. 1). Each cell will be assigned a number $q \in [0, p_2 - 1]$ where p_2 is a constant to be decided later. If the assigned value q is less than or equal to 4, then we select the point $(w, 5k+q)$ to be a codeword in the cell, otherwise no codeword is selected in this cell.

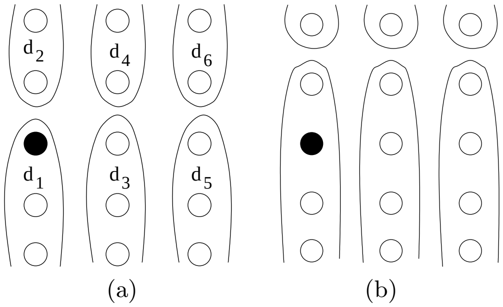

Fig. 1. Cells of size 5. For each scenario, the black point is a data point, the white points cannot be in the codebook.

There are five possible scenarios for a point $x \in X$, corresponding to the five different possible locations it occupies in a cell. Two of the five possible scenarios are illustrated in Fig. 1. Now we count the entropy loss for the scenario in Fig. 1(a). Same as in the basic construction, for any $x \in X$, all the points in the half-sphere $S_2(x)$ cannot be codewords. Therefore, all the white points in the figure cannot be codewords. Hence, for cell labeled d_1, there is only 1 choice for the value of the corresponding q, for d_3 and d_5, there are $p_2 - 3$ choices, and for d_2, d_4, and d_6, there are $p_2 - 2$ choices. Hence the entropy loss for this point is $\log p_2 + 2\log(p_2/(p_2-3)) + 3\log(p_2/(p_2-2))$.

Now we choose $p_2 = 14$, and the entropy loss for all five scenarios are as shown in Table 2.

(a)	$\log p_2 + 2\log(p_2/(p_2-3)) + 3\log(p_2/(p_2-2))$	< 5.1704
(b)	$\log p_2 + 2\log(p_2/(p_2-4)) + 3\log(p_2/(p_2-1))$	< 5.0990
(c)	$\log p_2 + 2\log(p_2/(p_2-5))$	< 5.0823
(d)	$\log p_2 + 2\log(p_2/(p_2-4)) + 2\log(p_2/(p_2-1))$	< 4.9921
(e)	$\log p_2 + 2\log(p_2/(p_2-3)) + 2\log(p_2/(p_2-2))$	< 4.9480

Table 2. Entropy loss of the five scenarios.

Next, we choose a value for α, such that scenario (e) happens most often. By this choice of α, we can show that $\mathcal{L}_H \leq 5.0750s$, whereas in the basic construction in Section 4.2, the bound is at least $5.0861s$ for $\delta = 1$.

Although the improvement is small, this construction suggests that the basic construction can be further improved by partitioning. There are many ways to partition the 2-d domain, and it is interesting to find the optimal partition in terms of entropy loss.

6 Short Description of P_H

In the basic constructions (Section 4.2), we can view the sketch P_H as a random sequence of length $N^d \log p_d$ with two types of constraints: Type 0 constraint is of the form $(k, 0)$, which requires that $h_k = 0$, and type 1 constraint is of the form $(k, 1)$ which requires that $h_k \neq 0$. The main idea is as follows: *Find the seed of some pseudo-random generator, such that the generated sequence satisfies all the type 0 and 1 constraints, and use the seed as the sketch.* In this section, we give two methods. The first method has efficient decoding and encoding algorithms, but still requires randomness. The second method eliminates all randomness but there is no known efficient encoder.

Using a high degree polynomial: Let $n = N^d$, and assign each $x \in [0, N-1]^d$ a unique index $\texttt{ind}(x)$ in $[0, n-1]$. Given a constraint set $S = \{(k_1, r_1), \ldots, (k_m, r_m)\}$, we construct a polynomial $f(x)$ of degree at most $m-1$ in $\mathbb{Z}_n$ as the following.

1. Uniformly choose $d_1, \ldots, d_m \in \mathbb{Z}_n$ at random such that for $1 \leq i \leq m$, if $r_i = 0$, then $d_i \equiv 0 \mod p_d$, otherwise $d_i \not\equiv 0 \mod p_d$.
2. Find the polynomial f of degree at most $m-1$ such that $f(\texttt{ind}(k_i)) \equiv d_i \mod n$ for $1 \leq i \leq m$.

The m coefficients of f is published as the sketch. During decoding, each h_k in P_H can be recovered by computing $h_k = (f(\texttt{ind}(k)) \mod n) \mod p_d$. Since for each point x we can have at most $|S_d(x)| + 1$ constraints, The polynomial f can be represented using $\frac{ds((4\delta+1)^d+1)}{2} \log N$ bits.

When p_d divides n, the entropy loss of this sketch is the same as the basic construction.

Using almost k-wise independence [1]. A sample space of n bits is almost k-wise independent if the probability distribution, induced on every k bit locations in a randomly chosen string from the sample space, is statistically close to uniform. The number of bits required to describe one sample is $(2+o(1))(\log\log n+3k/2+\log k)$. The sample space is pre-computed and made public.

We observe that this construction can be employed to make the sketch shorter. For instance, for $d = 1$ and $\delta = 1$ in our basic construction, we can construct such a sample space with $k = 3s$ and $n = N$. Given an original X, which in turn gives a set of constraints, we find the first sample that satisfies the constraints. The description of the sample is the sketch, whose size is in $o(s + \log\log N)$, which is also an upper bound for the entropy loss. In general, the size of the sketch would be in $o\left(s\Delta^d + \log\log(N^d)\right)$ in d-dimensional space. However, we are not aware of a better bound on the entropy loss other than the size of the sketch.

7 Entropy Loss of a Random Placement Method

Intuitively, it seems that it is better to have the codebook $\mathcal{C} = (X \cup R)$ as large as possible, since then a brute-force attacker will need to try more guesses to get X. In this section we give a seemingly natural random placement method to construct P_H with a large R in one dimension, and we show that the entropy loss is high for certain distributions of X.

The secure sketch P_H is a description of the sequence $\left\langle r_0, r_1, \ldots, r_{\lceil N/\Delta\rceil}\right\rangle$. Each r_i describes the gap between two consecutive codewords in $\mathcal{C}$ (except for r_0, which can be considered as the description of an "imaginary" starting codeword). Hence, instead of generating the codewords directly, we randomly choose the gaps between the codewords.

The sequence P_H is generated incrementally, starting from r_1. Most of the times the value of each gap can be chosen from Δ different values, but when a codeword w is close to a point $x \in X$, then the gap between w and the next codeword will be selected from a smaller interval (Steps 2 and 3).

1. Let $r_0 = -\delta$, $i = 1$.
2. If there is an $x \in X$ s.t. $x - r_{i-1} \in [2\delta, 4\delta]$ then let $r_i = x - r_{i-1}$.
3. If there is an $x \in X$ s.t. $x - r_{i-1} \in [4\delta + 1, 6\delta]$, uniformly choose r_i from $[\Delta - 1, x - r_{i-1} - \Delta]$. Otherwise, uniformly choose r_i from $[\Delta - 1, 2\Delta - 2]$.
4. Increase i by 1, and repeat from Step 2 until $i = \lceil N/\Delta\rceil + 1$.
5. Output $P_H = \left\langle r_1, \ldots, r_{\lceil N/\Delta\rceil}\right\rangle$.

The codewords can be recovered from P_H. In particular, the k-th codeword is $\sum_{i=0}^{k} r_i$, for $1 \le k \le \lceil N/\Delta\rceil$. If a codeword recovered in this process is greater than $N - 1$, it is removed. It is not necessary for P_H to have exactly $\lceil N/\Delta\rceil$ elements, and the extra padding is only for the ease of analysis.

Consider $X = \{x_1, x_1 + 2\Delta, \ldots, x_1 + 2(s-1)\Delta\}$, where x_1 is uniformly distributed. It can be shown that the entropy loss of X given P_H is at least $2s \log \Delta - \epsilon$ for some small positive constant ϵ. Comparing with other constructions in this paper, this method reveals the most information, even though it produces the largest number of codewords.

8 Conclusions and Discussions

In this paper, we investigate the technique of hiding a set of secret points by adding chaff points. Instead of considering brute force attackers as in known previous works, we give rigorous treatment under the secure sketch framework. We propose a construction of secure sketch for such point-sets, which can be extended to any dimension, and also some improvements for certain specific parameters. We give tight bounds of the entropy loss of our schemes.

Although we used infinity norm as the measure of closeness between any pair of points in the space, it is not difficult to extend our basic construction to any other closeness relations (e.g., using ℓ_2 norm). It seems that this is always possible as long as a total order can be defined on the points, so that the half-sphere of any given point is uniquely defined and is bounded.

On the other hand, the improvements in Section 5 are "ad-hoc" in the sense that they are specially designed for particular values of δ and d. We can also obtain improved schemes for another case where the white noise either leaves a coordinate unchanged or increased by one (we call this the `0-1` noise). An interesting question now is whether there is a generic method to find the "optimal" way of partitioning the space.

The proposed sketches are not suitable for large universe size N^d. The methods in Section 6 can reduce the sketch size, but the encoding and decoding algorithms can still be inefficient for large universe.

References

1. Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. In *Proc. of the 31st FOCS*, pages 544–553, 1990.
2. Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and Communications Security*, pages 82–91. ACM Press, 2004.
3. Ee-Chien Chang, Vadym Fedyukovych, and Qiming Li. Secure sketch for multi-set difference. Cryptology ePrint Archive, Report 2006/090, 2006. `http://eprint.iacr.org/`.
4. Ee-Chien Chang, Ren Shen, and Francis Weijian Teo. Finding the original point set hidden among chaff. In *ASIACCS*, 2006. To appear.
5. T.C. Clancy, N. Kiyavash, and D.J. Lin. Secure smartcard-based fingerprint authentication. In *ACM Workshop on Biometric Methods and Applications*, 2003.
6. Michael D.Garris and R.Michael McCabe. Fingerprint minutiae from latent and matching tenprint images. *NIST Special Database 27*, 2000.

7. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt'04*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.
8. E.G. Coffman Jr., L. Flatto, and P. Jelenković. Interval packing: The vacant-interval distribution. *The Annals of Applied Probability*, 10(1):240–257, 2000.
9. Ari Juels and Madhu Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002.
10. Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conf. on Computer and Communications Security*, pages 28–36, 1999.
11. D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, 2003.
12. Yaron Minsky, Ari Trachtenberg, and Richard Zippel. Set reconciliation with nearly optimal communications complexity. In *ISIT*, 2001.
13. I. Palasti. On some random space filling problems. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:353–359, 1960.
14. A. Rényi. On a one-dimensional problem concerning random space-filling. *Publ. Math. Inst. Hung. Acad. Sci.*, 3:109–127, 1958.
15. P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In *ECCV Workshop BioAW*, pages 158–170, 2004.
16. Shenglin Yang and Ingrid Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 609–612, 2005.