Small Secure Sketch for Point-Set Difference

Ee-Chien Chang Qiming Li

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

Abstract. A secure sketch is a set of published data that can help to recover the original biometric data after they are corrupted by permissible noises, and by itself does not reveal much information about the original. Several constructions have been proposed for different metrics, and in particular, set difference. We observe that in many promising applications, set difference alone is insufficient to model the noises. We propose to look into point-set difference, which measures noises that not only remove/introduce new feature points in the biometric objects, but may also perturb the points. In this paper, we first give an improvement for set difference construction that can be extended to multi-sets, where the sketch is small and there is an efficient decoding algorithm. We next give a sketch for point-set difference in both one and two-dimensional spaces. By using results in almost k-wise independence, the size of the sketch is reduced to near-optimal.
Keywords: biometrics, error-tolerant cryptography, secure sketch, point-set difference.

1 Introduction

Most biometric data are noisy in the sense that the capturing devices and extraction algorithms introduce inevitable noises. However, conventional cryptographic primitives do not tolerate even the slightest change in the data. For example, in an encryption scheme, decryption would fail if one bit of the decryption key is flipped. To use biometric data in cryptographic schemes (e.g., to use a fingerprint as the decryption key), new primitives are proposed (such as [9, 8, 6, 3]) to achieve robustness against noises. Secure sketch and fuzzy extractor were introduced in [6] as a generic way to reconstruct or extract a secret from noisy biometric data by publishing a "sketch". Given a set of original biometric data X captured during the registration process, the encoder computes a *sketch* P and publishes it. Later, when a set of corrupted biometric data Y is presented, the decoder can recover the original X from P and Y, as long as Y is close to X under certain distance measure. The security is measured by the amount of information about X revealed by the sketch P. Since the distribution of X may not be uniform, *fuzzy extractors* perform an additional step on the reconstructed X to obtain a uniformly random key.

Although information leakage is the main concern, in some applications, it is desirable to have small sketches. For example, *approximate message authentication codes* such as [11, 5] are developed to authenticate images under noises. Here, a short code is used to authenticate a long message received from a noisy channel. From another point of view, a small sketch simplifies the analysis of the information leakage, since the size of the sketch gives an upper bound of the information revealed.

Not surprisingly, the design of a secure sketch is very much dependent on the definition of "closeness" among biometric data, which in turn is determined by a distance function and

the type of noises to be tolerated. Secure sketch schemes for the following two main types of biometric data have been proposed: (1) The biometric data are from a vector space, and the distance is measured using a norm, e.g., Hamming distance. (2) The biometric data X is a subset of a universe $\mathcal{U}$, and the distance of two sets X and Y, where $|X| = |Y| = s$, is measured by the *set difference* $(s - |X \cap Y|)$.

We observe that for many biometric feature representations, especially those involve images, a combination of the above is required. For example, a fingerprint is typically represented as a set of minutia, which are points in a 2-dimensional space $[0, 1] \times [0, 1]$, or even 3-dimensional if the less reliable orientation attribute is included [4]. The noises introduced during scanning usually lead to small perturbation of the minutia, together with removal and addition of minutia. Hence, it is common to model the noises as a combination of two types of noises. The first type of noise perturbs each point in X by at most a small distance, say δ, and we call this the *white noise*. The *replacement noise* replaces some points in the perturbed X by randomly selected points in $\mathcal{U}$. The similarity between two sets X and Y can be measured by the maximum number of pairs of matched points, where two points x and y are considered a match if the distance between x and y is at most δ. Let us call this measure between two point-sets *point-set difference*. The following are a few observations that lead to our construction.

`0-1` *noise.* Let us illustrate an observation using point-set X from the one-dimensional interval $[0, n]$, and $\delta = 1/2$. To handle the white noise, prior to all operations, one may round each point to its nearest integer. Hence, if a point x is corrupted by a white noise, its rounded value could be unchanged, increased by 1, or decreased by 1. Therefore, to consider point-set difference in the continuous domain $[0, n]$, it suffices to consider points in $\mathbb{Z}_n$, whereby the white noise either leaves each point unchanged, or increases/decreases it by 1. By using two different rounding algorithms during the encoding and the decoding, we can further assume that the white noise is `0-1`, which either leaves each point unchanged, or increases it by 1 (Section 5). For a quick glance of why it is possible, refer to Fig. 1. Since it suffices to consider `0-1` noises, from now onward, we would only consider `0-1` noises in the discrete domain $\mathbb{Z}_n$. To avoid the special case at the boundary, we assume that the `0-1` noise has no effect on the point $x = n - 1$.

Tailor-made quantizer as sketch. For each discrete point x, we may further round it to an even number in $\{x - 1, x, x + 1\}$. If this extra rounding is predefined, e.g., it always rounds down, it will not be able to eliminate the `0-1` noise for certain point-set X. Nevertheless, for a given point x, we can always tailor-make a rounding function such that the `0-1` noise does not have any effect on the rounded value of x. This is illustrated in Fig. 2, where the points $x_1 = 0$ and $x_2 = 3$ will be rounded to the same values, 0 and 4 respectively, under the `0-1` noise. The description of such a rounding function has to be published so that the same rounding function can be used during both the encoding and the decoding. This leads to the main idea of our construction. Given X, we want to find a rounding function, such that the `0-1` noise does not have any effect on the rounded values of the points in X. The description of such a function will be part of the sketch. Since rounding is essentially a quantization process, we call a rounding function a *quantizer* and the rounded values the *quantized values*.

Well-separation and multi-sets. Let us consider the points x_3, x_4 and x_5 in Fig. 2. Under the quantization as illustrated in the figure, these points will be quantized to the same value 6. In this example, if the noise on x_5 happens to increase it by 1, then the quantized value would be different. To guarantee the consistency in the quantized values, we assume that the point-set X is *well-separated*. That is, for any two points $x_1, x_2 \in X$, $|x_1 - x_2| > 1$.

Such assumption is rather restrictive. In practice, it is difficult to ensure that all points are well-separated. If the input happens to contain 2 points that are the same, or close to each other, some mechanism is required. One method is to remove one of the points. However, this will lower the overall performance. We prefer another method which is to simply include both points. As we can see from Fig. 2, in certain cases, the quantized points will remain the same, hence the average performance potentially can be better than the first method. Section 6.3 discusses more on the practical issues and a way to reduce (but not eliminate) the `0-1` noise. More interestingly, in order to include repeated points, we need a sketch for set difference that can handle multi-sets (a multi-set is a set that may have repeated elements). Currently known constructions do not support multi-sets.

Another perspective of our construction. Here is another method that is unsatisfactory. The sketch includes a large point-set R such that $X \subset R$. During decoding, points in Y are matched with the points in R. Next, the techniques for set difference are used to recover the replaced points. However, this method reveals too much information about X, and its performance depends on the underlying distribution of X. For example, if the underlying distribution is likely to generate collinear points in X, then publishing R will reveal much about X. An improvement perturbs the points in X to obtain X', and the sketch includes a point-set R' such that $X' \subset R'$. The set X is randomly perturbed so as to reduce the influence of the underlying distribution. Although this approach seems feasible, there are many loose-ends. Our construction has many similarities with this approach, and indeed can be viewed as a method that realizes it.

In the rest of this paper, we first give a secure sketch for set difference that can handle multi-sets (Section 4). We then give a secure sketch for `0-1` noises for points in one-dimensional $\mathbb{Z}_n$ (Section 6), and extend it to 2-dimensional $\mathbb{Z}_n \times \mathbb{Z}_n$ (Section 7). Although similar ideas can be extended to higher dimensions, it might not be practical due to larger constant factors in the entropy loss.

Contributions and Results.

1. We give an approach to handle the combination of `0-1` noises and replacement noises, where the points are from a finite field $\mathbb{Z}_n$, and are well-separated. The total size of the sketch and the entropy loss depends on the choices of the sketch for set difference and the sketch that handles the `0-1` noise. If the encoder and the decoder have polynomial (with respect to $s, t, \log n$) computing time, the entropy loss is at most $1.5s + 1 + 2t(1 + \log n)$, which is in $O(s + t \log n)$, and the size of the sketch is in $O(s \log n)$. If the encoder is able to perform an exhaustive search in $2^{\Omega(s)}$ sequences, then both the size of the sketch and the entropy loss are in $O(s + t \log n)$.

2. We give an extension of the above to a 2-dimensional universe, $\mathbb{Z}_n \times \mathbb{Z}_n$. If the encoder and the decoder have polynomial computing time (with respect to $s, t, \log n$), the entropy loss is at most $4s + 4 + 2t(1 + 2\log n)$, while the size of the sketch is in $O(s \log n)$. When the encoder can do exhaustive search in $2^{\Omega(s)}$, then both the size of the sketch and the entropy loss are in $O(s + t \log n)$.
3. We give a scheme for set difference that handles multi-sets. The scheme has a simple and yet very efficient decoding algorithm, which amounts to solving linear systems with $2t$ equations, and root-finding for a polynomial of degree at most t. Hence, the number of arithmetic operations in $\mathbb{Z}_n$ is bounded by a polynomial of s and t. The size of the sketch and the entropy loss are at most $2t(1 + \log n)$. This construction is very similar to the set reconciliation technique in [10].

2 Related Works

Recently, a few new cryptographic primitives for noisy inputs are proposed. Fuzzy commitment scheme [9] 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 in [6], 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. The issue of *reusability* of sketches is addressed in [3]. It is shown that a sketch scheme that is provably secure may be insecure when multiple sketches of the same biometric data are obtained.

The set difference metric is first considered in [8], which gives a *fuzzy vault* scheme. Later, [6] 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 [6] has small sketches and achieves "sublinear" (with respect to n, the size of the universe) decoding by careful reworking of the standard BCH decoding algorithm. All these schemes can not handle multi-sets. The set reconciliation protocol presented in [10] is designed for two parties to jointly discover the union of their data, with as little communication cost as possible. Although the problem settings are different, the techniques in handling set difference is similar and can be employed to obtain a secure sketch.

Another line of research yields the constructions of *approximate message authentication codes* ([7, 2, 11, 5]), which can authenticate images that are corrupted by certain levels of noises, which are common to images (such as white noise and compression). There are also attempts in refining the extraction of biometric features so that the features are invariant to permissible noises [12]. Unfortunately, the reliability of such systems is not high.

3 Preliminaries

We follow the definitions of entropy and secure sketch in [6]. We also give the closeness and distance functions considered in this paper. A summary of notations is given in Appendix A.

Entropy. Let $\mathbf{H}_\infty(A)$ be the min-entropy of 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 $\widetilde{\mathbf{H}}_\infty(A|B) = -\log(\mathbb{E}_{b \leftarrow B}[2^{-\mathbf{H}_\infty(A|B=b)}])$. This definition is useful in the analysis, since for any ℓ-bit string B, we have $\widetilde{\mathbf{H}}_\infty(A|B) \geq \mathbf{H}_\infty(A) - \ell$.

Secure sketch. Let $\mathcal{M}$ be a set with a *closeness* relation $\mathtt{C} \subseteq \mathcal{M} \times \mathcal{M}$. When $(X, Y) \in \mathtt{C}$, we say the Y is close to X, or (X, Y) is a close pair. The closeness can be determined by a distance function $\mathsf{dist} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^{\geq 0}$ and a threshold Δ. That is, $(X, Y) \in \mathtt{C}$ iff $\mathsf{dist}(X, Y) \leq \Delta$.

Definition 1. *A sketch scheme is a tuple* $(\mathcal{M}, \mathtt{C}, \mathsf{Enc}, \mathsf{Dec})$*, 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* $X, Y \in \mathcal{M}$*,*

$$\mathsf{Dec}(Y, \mathsf{Enc}(X)) = X, \quad \text{if } (X, Y) \in \mathtt{C}.$$

The string $P = \mathsf{Enc}(X)$ *is to be made public and we call it the sketch. We say that the sketch scheme is* m*-secure if for all random variable* X *over* $\mathcal{M}$*, the entropy loss of* P *is at most* m*. That is,* $\mathbf{H}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X \mid \mathsf{Enc}(X)) \leq m$.

Closeness and Distance Functions. In this paper, $\mathcal{M}$ could be the collection of subsets or multi-sets of a universe $\mathcal{U}$. The universe could be $\mathbb{Z}_n$ or $\mathbb{Z}_n \times \mathbb{Z}_n$, where n is a prime. We consider 2 types of noises. The `0-1` noise, and the replacement noise that replaces certain elements in a set by randomly chosen elements. Here is a list of closeness and distance functions considered in this paper.

1. $\mathtt{C}_{s,t}$: The closeness determined by set difference, which is catered for the replacement noise. A pair $(X, Y) \in \mathtt{C}_{s,t}$ if $|X| = |Y| = s$ and $|X \cap Y| \geq s - t$.
2. `ZDist`: Distance function defined in $\mathbb{Z}_n$, which is catered for the `0-1` noise.

$$\mathtt{ZDist}(x, y) = \begin{cases} 0 & \text{if } y = x \vee y = x + 1 \\ \infty & \text{otherwise} \end{cases}. \tag{1}$$

   We define $\mathtt{ZDist}(n-1, 0) = \infty$. We can generalize this function to 2 or higher dimensions. For example, in $\mathbb{Z}_n \times \mathbb{Z}_n$, we have

$$\mathtt{ZDist}((x_1, x_2), (y_1, y_2)) = \begin{cases} 0 & \text{if } \mathtt{ZDist}(x_1, y_1) = 0 \wedge \mathtt{ZDist}(x_2, y_2) = 0 \\ \infty & \text{otherwise} \end{cases}. \tag{2}$$

   Note that `ZDist` is not symmetric, that is, it is not necessary that $\mathtt{ZDist}(x, y) = \mathtt{ZDist}(y, x)$.
3. $\mathtt{ZPS}_{s,t}$: Closeness for point-sets of size s. Suppose $X = \{x_1, \ldots, x_s\}$ and $Y = \{y_1, \ldots, y_s\}$, we have $(X, Y) \in \mathtt{ZPS}_{s,t}$ if there exists a 1-1 correspondence f on $\{1, \ldots, s\}$ such that

$$|\{i \mid \mathtt{ZDist}(x_{f(i)}, y_i) = 0\}| \geq s - t.$$

Well-separated point-sets. A point-set X is *well-separated* if for any $x, y \in X$, we have $\mathtt{ZDist}(x, y) > 0$. We will discuss about this in detail in Section 6.3.

Almost k-wise independent random variables. A sample space on n-bit strings is k-wise independent if, the probability distribution, induced on every k bit locations in a randomly chosen string from the sample space, is uniform. Alon et al [1] considered almost k-wise independence with small sample size, and gave several constructions.

Definition 2 (Almost k-wise independence [1]). *Let S_n be a sample space and $X = x_1 \cdots x_n$ be chosen uniformly from S_n. S_n is almost k-wise independent with ϵ statistical difference if, for any k positions $i_1 < i_2 < \ldots < i_k$, and any k-bit string α, we have*

$$\sum_{\alpha \in \{0,1\}^k} |\Pr[x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha] - 2^{-k}| \leq \epsilon. \tag{3}$$

If we choose $\epsilon = 2^{-k}$, the probability $\Pr[x_{i_1} x_{i_2} \ldots x_{i_k} = \alpha]$ in (3) is non-zero. To see this, let $X' = x_{i_1} x_{i_2} \ldots x_{i_k}$, then $\Pr[X' = \alpha] = 0$, and there exists some $\beta \neq \alpha$ such that $\Pr[X' = \beta] > 2^{-k}$, since otherwise $\sum_{\alpha \in \{0,1\}^k} \Pr[X' = \alpha] < 1$. Thus $|\Pr[X' = \alpha] - 2^{-k}| + |\Pr[X' = \beta] - 2^{-k}| > 2^{-k}$, which is a contradiction. Hence, we can always find such X given any $i_1, \ldots, i_k$ and any α. Furthermore, the number of bits required to describe the sample is $(2 + o(1))(\log\log n + 3k/2 + \log k)$ which is in $O(k + \log\log n)$.

4 Secure Sketch for Set Difference

In this section, we give a secure sketch for set difference, that is, with respect to the closeness $\mathsf{C}_{s,t}$. Our scheme can handle the case where X is a multi-set. The size of the sketch is at most $2t(1 + \log n)$. In addition, there exists a simple and yet efficient decoding algorithm – we just need to solve a linear system with $2t$ equations and unknowns and find the roots of two degree t polynomials.

To handle a special case, we assume that X does not contain any element in $\{0, 1, \ldots, 2t-1\}$, and will discuss how to remove this assumption later at the end of this section. Our construction is similar to the set reconciliation protocol in [10], but the problem settings are different.

The encoder Enc_s. Given $X = \{x_1, \ldots, x_s\}$, the encoder does the following.

1. Construct a monic polynomial $p(x) = \prod_{i=1}^{s}(x - x_i)$ of degree s.
2. Publish $P = \langle p(0), p(1), \ldots, p(2t-1) \rangle$.

The decoder Dec_s. Given $P = \langle p(0), p(1), \ldots, p(2t-1) \rangle$ and $Y = \{y_1, \ldots, y_s\}$, the decoder follows the steps below.

1. Construct a polynomial $q(x) = \prod_{i=1}^{s}(x - y_i)$ of degree s.
2. Compute $q(0), q(1), \ldots, q(2t-1)$.
3. Let $p'(x) = x^t + \sum_{j=0}^{t-1} a_j x^j$ and $q'(x) = x^t + \sum_{j=0}^{t-1} b_j x^j$ be monic polynomials of degree t. Construct the following system of linear equations with the a_j's and b_j's as unknowns.

$$q(i)p'(i) = p(i)q'(i), \quad \text{for } 0 \leq i \leq 2t-1 \tag{4}$$

4. Find one solution for the above linear system. Since there are $2t$ equations and $2t$ unknowns, such a solution always exists.
5. Solve for the roots of the polynomials $p'(x)$ and $q'(x)$. Let them be X' and Y' respectively.
6. Output $\widetilde{X} = (Y \cup X') \setminus Y'$.

The correctness of this scheme is straight forward. When there is exactly t replacement errors, we can view $p'(x)$ as the "missed" polynomial whose roots are in $X' = X \setminus Y$. Similarly, $q'(x)$ is the "wrong" polynomial, whose roots are in $Y' = Y \setminus X$. Since the roots of $p(x)$ and $q(x)$ are in X and Y respectively, we have $q(x)p'(x) = p(x)q'(x)$. This interpretation motivates the equation (4).

When there are less than t replacement errors, there will be many degree t monic polynomials $p'(x)$ and $q'(x)$ that satisfy $q(x)p'(x) = p(x)q'(x)$. For any such $p'(x)$ and $q'(x)$, they share some common roots, which could be some arbitrary multi-set Z. That is, $X' = (X \setminus Y) \cup Z$, and $Y' = (Y \setminus X) \cup Z$. In Step 6, this extra Z will be eliminated.

When $X \cap \{0, \ldots, 2t-1\} \neq \emptyset$, some equations in (4) would degenerate, which makes the rank of the linear system less than $2t$. In this case, it is not clear how to find the correct polynomial in the solution space. Hence we require that $X \cap \{0, \ldots, 2t-1\} = \emptyset$.

Note that in the above we do not require the elements of X and Y to be distinct, so this scheme can handle multi-sets. Furthermore, since the size of each $p(i)$ for $1 \leq i \leq 2t$ is $(\log n)$, the size of P is $2t(\log n)$. Therefore, we have the

Lemma 1. *When $X \cap \{0, \ldots, 2t-1\} = \emptyset$, the entropy loss due to $\mathsf{Enc}_s(X)$ is at most $2t \log n$.*

Removing the assumption on X and Y. The assumption that X cannot contain any element from $\{0, \ldots, 2t-1\}$ can be easily relaxed. We can find the smallest prime m such that $m - n \geq 2t$, and then apply the scheme on $\mathbb{Z}_m$. But instead of publishing $p(0), \ldots, p(2t-1)$, we publish $p(m-1), \ldots, p(m-2t)$. In this way, the size of the sketch is $2t \log m$. In practice, this is not a problem since the size of the universe may not be prime, and we will need to choose a larger finite field anyway. For t that is not too large (say, $t \leq n/4$), we can always find at least one prime in $[n + 2t, 2n]$. Hence, we have the

Lemma 2. *When $t \leq n/4$, the entropy loss due to $\mathsf{Enc}_s(X)$ is at most $2t(1 + \log n)$.*

5 Reduction from White Noise to `0-1` Noise

Now we describe how to reduce the problem of dealing with white noises in a continuous domain to `0-1` noises in a discrete domain.

First, let us consider points in $\mathbb{R}$, and the white noise that corrupts each point by at most δ. That is, for any x and its corrupted version y, we have $|y - x| \leq \delta$. We use two different quantizers, Q_e and Q_d, during encoding (on the original point-set X), and during decoding (on the corrupted set Y) respectively. During encoding, we define $Q_e(x) = k$ if and only if

$x \in [2k\delta, 2(k+1)\delta)$. During decoding, we define $Q_d(y) = k$ if and only if $y \in [(2k-1)\delta, (2k+1)\delta)$. Note that when $|x-y| \leq \delta$, we have $Q_d(y) \in \{Q_e(x), Q_e(x)+1\}$, hence the noise becomes `0-1`.

Therefore, instead of working with points from $\mathbb{R}$ and white noises in the range $[-\delta, \delta]$, it suffices to consider `0-1` noises in $\mathbb{Z}$. To avoid the special case at the boundary when working in the finite field $\mathbb{Z}_n$, we assume that the white noise has no effect on the last element $n-1$.

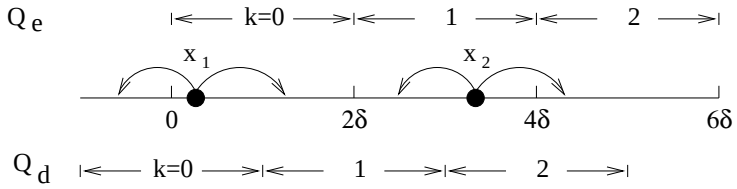

Fig. 1. Reduction from white noise to `0-1` noise.

6 `0-1` Noise with Replacement when $\mathcal{M} = \mathcal{P}(\mathbb{Z}_n)$

In this section, we consider `0-1` noise with replacement (the closeness $\mathtt{ZPS}_{s,t}$). The universe is the one-dimensional $\mathbb{Z}_n$. We assume that the original biometric data X is well-separated.

The final sketch published is a concatenation of two sketches, $P_H P_S$. The role of P_H is to correct the `0-1` noise (Section 6.1 and 6.2). P_H is the description of a quantizer H, whereby the quantized X remain the same under the `0-1` noise. Since the quantized points are consistent, we can apply the techniques on set difference on them to correct the replacement noise. The sketch for set difference is P_S. In this section, we focus on the construction of P_H.

6.1 Main Idea in Construction of P_H

We render the elements in $\mathbb{Z}_n$ using two colors, `black` and `white`. For any $x \in \mathbb{Z}_n$,

$$\mathsf{color}(x) = \begin{cases} \texttt{white} & \text{if } x \equiv 1 \mod 2 \\ \texttt{black} & \text{if } x \equiv 0 \mod 2 \end{cases} \tag{5}$$

We call a many-to-one function H a *quantizer* if for all x, $H(x) = x$ if x is black, and $H(x)$ is either $x-1$ or $x+1$ if x is white. Fig. 2 depicts such a function. For $X = \{x_1, \ldots, x_s\}$, we denote $H(X) \triangleq \{H(x_1), \ldots, H(x_s)\}$.

Given $X = \{x_1, x_2, \ldots, x_s\}$, our goal is to find a quantizer such that each point in X will be consistently quantized, even under the influence of `0-1` noises. In particular, we require that

$$\S 1 \quad \forall x \in X, H(x) = H(x+1).$$

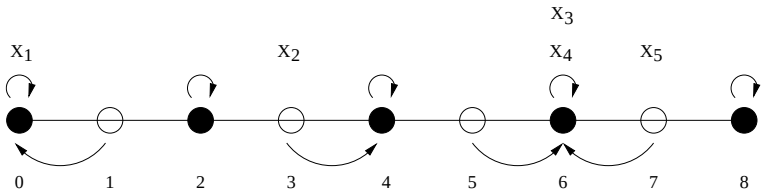

Fig. 2. A quantizer H. The arrows indicate how the points are quantized (or rounded) to even numbers.

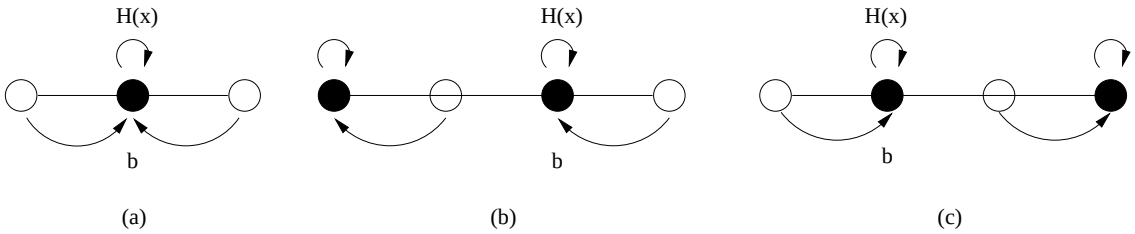

Fig. 3. Recovering x from $H(x)$. There are three scenarios for H when $H(x) = b$.

To reconstruct X given $H(X)$, we need to add more constraints on H. Suppose we know that $H(x) = b$ for some b, as illustrated in Fig. 3. In scenario (b) and (c), x must be b and $b-1$ respectively. However, when (a) happens, x can be either b or $b-1$.

To resolve the ambiguity, we add *one* but not both of the following two constraints:

$$\begin{array}{ll} \S 2a & \forall x \in X, H(x+2) = x+3 \qquad \text{if } x \text{ is white, or} \\ \S 2b & \forall x \in X, H(x-1) = x-2 \qquad \text{if } x \text{ is black.} \end{array}$$

Note that the constraint will not eliminate scenario (a). Nevertheless, even if scenario (a) happens, we can conclude that $x = b$ if §2a is imposed, and that $x = b-1$ if §2b is imposed.

6.2 Short Descriptions of H (Sketch P_H)

A quantizer H can be conveniently represented by a sequence $\langle h_1, h_3, h_5, \ldots, h_{n-2} \rangle$ where $h_i = 0$ if $H(i) = i-1$; otherwise, $h_i = 1$. Publishing such a sequence requires $\lfloor n/2 \rfloor$ bits, which is undesirable when n is large.

For any given X, there would be many quantizers that satisfy the constraints in Section 6.1. The constraints restrict the values of h_i for certain indices i's. Let W be the set of these indices. For any other $j \notin W$, h_j can be either 0 or 1.

The first constraint §1 restricts s of such h_i's. For constraints §2a and §2b, we choose §2a if white is the minority color in X, and §2b otherwise. In this way, the number of restricted h_i's is at most $0.5s$. Hence, we have $|W| \leq 1.5s$. Note that we need to publish an additional 1 bit to indicate which constraint, either §2a or §2b, we have used.

We give two constructions of short descriptions. The first one is simple and efficient but requires $|W| \log n$ bits. The second one requires space that is near-optimal, but it requires exhaustive search during encoding. In either case, the entropy loss is small, and there exists efficient decoding algorithms. Let $k = |W|$, and $W = \{w_1, \ldots, w_k\}$.

Construction I: Construct a polynomial $f(x)$ of degree $k-1$ as the following.

1. Randomly choose $d_1, \ldots, d_k \in \mathcal{U}$ such that for $1 \le i \le k$, $d_i \equiv h_{w_i} \mod 2$.
2. Find a degree $k-1$ polynomial f such that $f(w_i) = d_i$ for $1 \le i \le k$.

The sketch P_H is the k coefficients of f. Hence, the size of the sketch is $k \log n$. The size of this sketch could be reduced further to about $k \log(n/2)$, since we can work on a smaller finite field. This is possible because the number of h_i's is only $\lfloor n/2 \rfloor$, thus any finite field of size greater than $n/2$ is sufficient.

During decoding, the quantizer H to be used is the one represented by $h_i = f(i) \mod 2$ for all odd i.

Construction II: Firstly we construct an almost k-wise independent space on n bits [1], and $\epsilon = 2^{-k}$. Given W and $\langle h_{w_1}, \ldots, h_{w_k} \rangle$, we uniformly choose a sample $\beta = \beta_1 \cdots \beta_n$ in the sample space such that $\beta_{w_i} = h_{w_i}$ for all i. The representation of β is then the description of H, which is the sketch P_H. Note that the size of P_H is $\ell = (2 + o(1))(\log \log n + 3k/2 + \log k)$, which is in $O(s + \log \log n)$. However, we do not have an efficient way to find such sample, except by exhaustive search in the sample space of size 2^ℓ. Nevertheless, the decoding can be done efficiently.

Lemma 3. *The entropy loss due to P_H is at most $1.5s + 1$ for Construction I, and at most $(2 + o(1))(\log \log n + 9s/4 + \log(1.5s)) + 1$, which is in $O(s + \log \log n)$, for Construction II.*

Proof: For Construction I, we count the entropy. Let R be the randomness we invest in choosing the random numbers d_i's. The entropy of R is $k(\log n - 1)$ bits. The number of possible output of f is n^k. Together with the one additional bit to choose the constraint, the average min-entropy (X, R) given P_H is at least $\widetilde{\mathbf{H}}_\infty(X) - k - 1$. The randomness R can be recovered from X and P_H. Therefore, $\widetilde{\mathbf{H}}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X|P_H) \le k + 1$.

For Construction II, the entropy loss is simply bounded by the size of P_H. That is, $\widetilde{\mathbf{H}}_\infty(X) - \widetilde{\mathbf{H}}_\infty(X|P_H) \le |P_H| \le (2 + o(1))(\log \log n + 3k/2 + \log k) + 1$.

Since $k \le 1.5s$, we have the claimed bounds. □

Recall from Section 4 that P_S introduces entropy loss at most $2t(1+\log n)$, the total entropy loss of $P_H P_S$ is bounded by the following.

Theorem 1. *When $t \le n/4$, the entropy lost due to P_H and P_S is at most $h+1+2t(1+\log n)$, where $h = 1.5s$ if P_H is computed using Construction I, and $h = (2 + o(1))(\log \log n + 9s/4 + \log(1.5s))$, which is in $O(s + \log \log n)$, if P_H is computed as in Construction II.*

6.3 On the Assumption that Points are Well-Separated

If the points are not well-separated, the error tolerance of our scheme would be affected. For example, consider the points x_3, x_4 and x_5 in Fig. 2. If the `0-1` noise shifts x_5 from 7 to 8, it will be considered as a replacement error. If the noise happens to leave x_5 unchanged, then the scheme still works. In addition, if X contains duplicated elements, our scheme would work fine because the sketch in Section 4 can handle multi-sets. In other words, our scheme has a guaranteed error tolerance when there are no two points $x, x' \in X$ such that $x - x' = 1$.

During the reduction from white noise to `0-1` noise as in Section 5, we can choose a step size that is larger than 2δ, such that given the same white noise, the points are less likely to be shifted. In other words, the `0-1` white noise is reduced on average. Note that this introduces more entropy loss. Moreover, such trade-off depends on the distribution of the input biometric data. Therefore, we will not discuss it further in this paper.

In sum, with the requirement on well-separation, our scheme can tolerate the claimed noise in the worst case. Without the requirement, the scheme could still perform well in average. Hence, it is better to include those points that violate the well-separation requirement, instead of removing them. To include those points, we have to handle multi-sets.

7 `0-1` Noise with Replacement when $\mathcal{M} = \mathcal{P}(\mathbb{Z}_n \times \mathbb{Z}_n)$

We now extend our construction on `0-1` noise with replacement to $\mathcal{U} = (\mathbb{Z}_n \times \mathbb{Z}_n)$. Similar to one-dimension, the sketch is the concatenated $\widetilde{P_H}\widetilde{P_S}$. We will only discuss the sketch $\widetilde{P_H}$. We assume that X is always well-separated. That is, for any distinct $(u_1, v_1), (u_2, v_2) \in X$, $|u_1 - u_2| \geq 2$ and $|v_1 - v_2| \geq 2$.

7.1 Quantizer H

The elements in $\mathcal{U}$ are rendered with 4 different colors as below.

$$\mathsf{color}(u, v) = \begin{cases} \texttt{black} & \text{if } u \equiv v \equiv 0 \mod 2 \\ \texttt{white} & \text{if } u \equiv 0 \mod 2 \wedge v \equiv 1 \mod 2 \\ \texttt{red} & \text{if } u \equiv 1 \mod 2 \wedge v \equiv 0 \mod 2 \\ \texttt{green} & \text{if } u \equiv v \equiv 1 \mod 2 \end{cases} \tag{6}$$

The `0-1` noise either leaves a point x untouched, or shifts it to one of the 3 adjacent points. Let $\texttt{Neighbour}(u, v)$ be the set of 4 points that (u, v) may be shifted to by the `0-1` noise, namely $\{(u, v), (u + 1, v), (u, v + 1), (u + 1, v + 1)\}$.

In $\mathbb{Z}_n \times \mathbb{Z}_n$, a quantizer H maps each point (u, v) to a black point (u', v') where $|u - u'| \leq 1$ and $|v - v'| \leq 1$. Fig. 4 illustrates such a quantizer. We need an H that satisfies the following: For each $(u, v) \in X$, all the 4 points in $\texttt{Neighbour}(u, v)$ are mapped to the black point in $\texttt{Neighbour}(u, v)$. In other words,

$$\sharp 1 \quad \forall x \in X, \forall w \in \texttt{Neighbour}(x), H(x) = H(w).$$

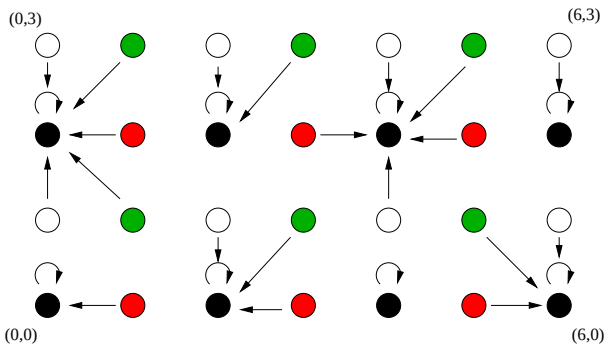

Fig. 4. Quantizer H in 2-D.

In this way, if $x \in X$ and y is a copy of x corrupted by `0-1` noise, then $H(x) = H(y)$.

For each white and red point, there are two possible black points that it can be mapped to. It seems at first that there are four choices for each green point, but after the quantization for red and white points are fixed, only two choices are left. For instance, in Fig. 4, from the surrounding white and red points, we can deduce that the green point $(3, 1)$ can only be mapped to either $(2, 0)$ or $(4, 2)$. Hence, a straight forward description of H requires $(1/4+1/4+1/4)n^2 = (3/4)n^2$ bits.

We could apply similar ideas as in Section 6.2 to these $(3/4)n^2$ bits to obtain $\widetilde{P_H}$. However, there are still ambiguities to resolve when we recover X from $H(X)$. For example, if $H(x) = (0, 2)$ as in Fig. 4, we would not be able to tell whether $x = (0, 1)$ or $x = (0, 2)$. Our basic idea is to apply the ambiguity resolving techniques in Section 6.1 by imposing more constraints on H. The details of the construction of $\widetilde{P_H}$ can be found in Appendix B.

Similar to 1-D, to obtain a short description H, we identify the number of bits that have to be fixed due to the imposed constraints. In total, $3s$ bits are imposed by $\sharp 1$, and an additional s bits are imposed by the constraints that resolve ambiguities. Then we can apply the techniques in Section 6.2. For instance, using Construction I as in the case of 1-D, we need to find a degree $4s-1$ polynomial, which gives entropy loss at most $4s$. We also need 4 additional bits to specify which color are the majority and minority in X. Thus, we have the following results.

Lemma 4. *The entropy loss due to $\widetilde{P_H}$ is at most $4s + 4$ with Construction I, and $(2 + o(1))(\log(2 \log n) + 6s + \log(4s)) + 4$, which is in $O(s + \log \log n)$, with Construction II.*

Theorem 2. *When $t \leq n^2/4$, the entropy loss due to $\widetilde{P_H}$ and $\widetilde{P_S}$ is at most $h + 4 + 2t(1 + 2 \log n)$, where $h = 4s$ if $\widetilde{P_H}$ is computed using Construction I, and $h = (2+o(1))(\log(2 \log n)+ 6s + \log(4s))$ with Construction II.*

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. G.R. Arce, L. Xie, and R.F. Graveman. Approximate image authentication codes. In *Proc. 4th Annual Fedlab Symp. on Advanced Telecommunications/Information Distribution*, 2000.
3. Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and Communications Security*, pages 82–91. ACM Press, 2004.
4. Michael D.Garris and R.Michael McCabe. Fingerprint minutiae from latent and matching tenprint images. *NIST Special Database 27*, 2000.
5. G. DiCrescenzo, R. Graveman, G. Arce, and R. Ge. A formal security analysis of approximate message authentication codes. In *Proc. CTA Comm. and Networks*, 2003.
6. 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.
7. R. Graveman and K. Fu. Approximate message authentication codes. In *Proc. 3rd Annual Fedlab Symp. on Advanced Telecommunications/Information Distribution*, 1999.
8. Ari Juels and Madhu Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002.
9. Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conf. on Computer and Communications Security*, pages 28–36, 1999.
10. Yaron Minsky, Ari Trachtenberg, and Richard Zippel. Set reconciliation with nearly optimal communications complexity. In *ISIT*, 2001.
11. L. Xie, G.R. Arce, and R.F. Graveman. Approximate image message authentication codes. In *IEEE Trans. on Multimedia*, pages 242–252, 2001.
12. W. Zhang, Y.-J. Chang, and T. Chen. Optimal thresholding for key generation based on biometrics. *Int. Conf. on Image Processing*, 2004.

A Summary of Notations

n An odd prime.

$\mathcal{U}$ The universe, which could be $\mathbb{Z}_n$, $\mathbb{Z}_n \times \mathbb{Z}_n$, or an Euclidean space.

$\mathcal{M}$ The set of biometric data. It is associated with a closeness relation.

X The original biometric data. $X = \{x_1, \ldots, x_s\} \in \mathcal{M}$.

Y A copy of X corrupted by noise. $Y = \{y_1, \ldots, y_s\} \in \mathcal{M}$.

s The size of the biometric data X.

t The number of errors (w.r.t. to replacement noise) the scheme is designed to tolerate.

δ The amount of error (w.r.t. to white noise) the scheme is to tolerate.

P A sketch.

H A quantizer that handles `0-1` noise.

$\mathbf{H}_\infty$ Entropy. $\mathbf{H}_\infty(A)$ is the min-entropy of random variable A: $\mathbf{H}_\infty(A) = -\log(\max_a \Pr[A = a])$. $\widetilde{\mathbf{H}}_\infty(A|B)$ is the average min-entropy of A given B: $\widetilde{\mathbf{H}}_\infty(A|B) = -\log(\mathbb{E}_{b \leftarrow B}[2^{-\mathbf{H}_\infty(A|B=b)}])$.

$\mathtt{C}_{s,t}$ The closeness relation defined by set difference. A pair $(X, Y) \in \mathtt{C}_{s,t}$ if $|X| = |Y| = s$ and $|X \cap Y| \geq s - t$.

`ZDist` Distance function defined in $\mathbb{Z}_n$, which caters for the `0-1` noise. For $x, y \in \mathbb{Z}_n$, $\mathtt{ZDist}(x, y)$ is defined to be 0 if $0 \leq y - x \leq 1$, infinity otherwise.

$\mathtt{ZPS}_{s,t}$ Closeness for point-sets of size s. For two point-sets X and Y, Y is close to X if at least $s - t$ elements in Y are close to a matching point in X under the `0-1` noise.

B Detailed Construction of $\widetilde{P_H}$

The quantizer H can be defined in terms of 3 functions $H_{\mathtt{w}}, H_{\mathtt{r}}$, and $H_{\mathtt{g}}$,

$$H(x) = \begin{cases} H_{\mathtt{w}}(x), & \text{if } \mathsf{color}(x) = \mathtt{white} \\ H_{\mathtt{r}}(x), & \text{if } \mathsf{color}(x) = \mathtt{red} \\ H_{\mathtt{g}}(x), & \text{if } \mathsf{color}(x) = \mathtt{green} \end{cases}$$

Each function maps its input to one of its neighbouring black points. For convenience, for any $x \in X$, let $x_{\mathtt{b}}, x_{\mathtt{w}}, x_{\mathtt{r}}$, and $x_{\mathtt{g}}$ be the black, white, red and green points in $\mathtt{Neighbour}(x)$ respectively. Hence, the first constraint $\sharp 1$ is equivalent to the following. For all $x \in X$,

$$\begin{array}{llll} \sharp 1a & H_{\mathtt{b}}(x) = x_{\mathtt{b}} & \sharp 1b & H_{\mathtt{w}}(x) = x_{\mathtt{b}} \\ \sharp 1c & H_{\mathtt{r}}(x) = x_{\mathtt{b}} & \sharp 1d & H_{\mathtt{g}}(x) = x_{\mathtt{b}}. \end{array}$$

As mentioned in Section 7, there are two possibilities for $H_{\mathtt{r}}(x)$ and $H_{\mathtt{w}}(x)$, and once they are fixed, there are also two possibilities for $H_{\mathtt{g}}(x)$. Hence, to describe H in a straight forward manner using a binary sequence, we only need $m = (1/4 + 1/4 + 1/4)n^2$ bits. Let $h = \langle h_1, \ldots, h_m \rangle$ be such a sequence.

For any $x \in X$, to satisfy constraint $\sharp 1b$, we need to restrict the value of at most one bit in h. Same for $\sharp 1c$ and $\sharp 1d$. We do not need to consider $\sharp 1a$ since it is implicit. Therefore, for s points, we need to restrict $3s$ bits in h.

Similar to the 1-dimensional case, there will be ambiguities when we try to recover X from an H that satisfies only $\sharp 1$. One of the worst (most ambiguous) scenarios of H is as illustrated by the solid arrows in Fig. 5. In this case, if $H(x) = x_1$ for some $x \in X$, any of the four points

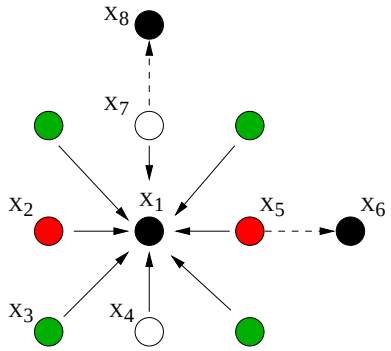

Fig. 5. Ambiguity resolving in 2-D. The solid arrows show how H quantizes each point, and the dashed arrows show how the ambiguities can be resolved.

x_1, x_2, x_3 and x_4 could be x.

To resolve the ambiguity while keeping the size of the sketch small, we impose more constraints on H. First, we find the most and the least frequently occurred colors in X. Without

loss of generality, we assume that they are black and green respectively. Then we require that the following constraints are satisfied by H. For all $(u, v) \in X$,

$\sharp 2a$	$H_{\mathtt{r}}(u+2, v) = (u+3, v)$	if (u, v) is red;
$\sharp 2b$	$H_{\mathtt{w}}(u, v+2) = (u, v+3)$	if (u, v) is white;
$\sharp 2c$	Both $\sharp 2a$ and $\sharp 2b$	if (u, v) is green.

With the above additional constraint, we can resolve the ambiguity by the following rules. If H is as shown by the solid arrows in the figure, we declare that $x = x_1$. If $H_{\mathtt{r}}(x_5) = x_6$ (the horizontal dashed arrow), and the rest follow solid arrows, we declare that $x = x_2$. If $H_{\mathtt{w}}(x_7) = x_8$ (the vertical dashed arrow), and the rest follow solid arrows, we declare that $x = x_4$. If both $H_{\mathtt{r}}(x_5) = x_6$ and $H_{\mathtt{w}}(x_7) = x_8$, we declare that $x = x_3$.

In this case, each red and white point would restrict one more bit by constraints $\sharp 2a$ and $\sharp 2b$ respectively, and each green point would restrict two more bits by constraint $\sharp 2c$. Since green is the least frequently occurred color, the number of green points is at most $s/4$. Similarly, the number of black points is at least $s/4$. In the worst case (in terms of the number of restricted bits), there are $s/4$ green points and $s/2$ red and white points. Therefore, the total number of bits restricted by these constraints is at most $(1/2 + 2/4)s = s$.

Therefore, $3s + s = 4s$ bits are restricted in the $(3/4)n^2$ bits of h. By applying the constructions in Section 6.2, we obtain the results presented in Section 7.