Secure Sketch for All Noisy Sources

Yen-Lung Lai, Zhe Jin

Monash University Malaysia,
Jalan Lagoon Selatan, Bandar Sunway, 47500 Subang Jaya, Selangor
yenlung.lai@monash.edu, jin.zhe@monash.edu

Abstract. Secure sketch produces public information of its input w without revealing it, yet, allows the exact recovery of w given another value w' that is close to w. Therefore, it can be used to reliably reproduce any error-prone a secret sources (i.e., biometrics) stored in secret storage. However, some sources have lower entropy compared to the error itself, formally called "more error than entropy", a standard secure sketch cannot show its security promise perfectly to these kinds of sources. This paper focused on secure sketch. We propose a concrete construction for secure sketch. We show security to all noisy sources, including the trivial source with zero min-entropy. In addition, our construction comes with efficient recovery algorithm operates in polynomial time in the sketch size, which can tolerate high number of error rate arbitrary close to 1/2.

Keywords: Secure Sketch · Error Correction · Coding Theory · Fuzzy Extractor

1 Introduction

Traditional cryptography systems rely on uniformly distributed and recoverable random strings for secret. For example, random passwords, tokens, and keys. These secrets must present exactly on every query for a user to be authenticated and get accessed into the system. Besides, it must also consist of high enough entropy, thus making it very long and complicated, further resulted in the difficulty in memorizing it. On the other hand, there existed plentiful non-uniform strings to be utilized for secrets in practice. For instance, biometrics (i.e., human iris, fingerprint) which can be used for human recognition/identification purpose. Similarly, long passphrase (S. N. Porter, 1982 [1]), answering several questions for secure access (Niklas Frykholm *et al.*, 2001 [2]) or personal entropy system (Ellison *et al.*, 2000 [3]), and list of favorite movies (Juels and Sudan, 2006 [4]), all are non-uniformly distributed random strings that can be utilized for secrets.

The availability of non-uniform information prompted the generation of uniform random string from non-uniform materials. Started by Bennette *et al.*, (1988) [5], identified two major approaches to derive a uniform string from noisy non-uniform sources. The first approach is *information-reconciliation*, by tolerating the errors in the sources without leaking any information. The second approach refers to the *privacy amplification*, which converts high entropy input into a uniformly random input. The information-reconciliation process can be classified into interactive (includes multi messages) and non-interactive (only includes single message) versions. For non-interactive line of work, it has been first defined by Dodis *et al.*, (2004) [6] called the fuzzy extractor. Likewise, the fuzzy extractor used two approaches to accomplish the task, which is the secure sketch - for error tolerance, and randomness extractor - for uniform string generation. Secure sketch is demanding because it enables information-reconciliation, e.g., exact recovery of a noisy secret while offering security assurance to it. Moreover, a secure sketch can be easily extended to fuzzy extractor for uniform string generation by using a randomness extractor. The generated random string can be used in independent security system for access control, identification, etc.

This work focus on secure sketch. We review the limitations of current secure sketch construction in Section 1.1. To overcome such limitations, we introduced the usage of resilience vector (RV) in Section 5 to support better understanding of the structure of the noisy sources. We proposed a concrete construction with included RV for sketching and recovery (secure sketch) in Section 6 and 7 respectively. Our proposed recovery mechanism has shown to be efficient in polynomial in the sketch size and allows error tolerance of error rate arbitrary close to 1/2 (Section 9). In the end, in Section 10 we formalize the security of our construction and show security to all noisy sources with computationally unbounded attacker. We also compared our proposal with existing secure sketch construction, showing our construction enjoys the better upper bound of min-entropy requirement for a standard secure sketch (Section 11).

1.1 Issues in Existing Secure Sketch Construction

There existing various secure sketch constructions in the literature. Some notable constructions involved the code-offset construction proposed by Juels and Wattenberg (1999) [7] that operates perfectly over hamming matric space. Besides, Juels and Sudan (2006) [4] have also proposed another construction for metric other than hamming called the fuzzy vault. An improved version of the fuzzy vault is proposed by Dodis *et al.*, (2004) [6], and also the Pin-sketch that relies on syndrome encoding/decoding with t-error correcting BCH code $\mathcal{C}$, which works well for non-fixed length input over a universe $\mathcal{U}$.

However, the above mentioned secure sketch construction only works for limited noisy sources. Briefly, given a point (some value) w, the sketch would allow the acceptance of its nearby point w' within distance t for exact recovery of w. Therefore, if an adversary can predict an accepting w' with noticeable probability, the sketch must reveal w to the adversary with noticeable probability as well. The tension between security and error tolerance capability is very strong. Precisely, the security is measured in term of the residual (min-) entropy, which is the starting entropy of w minus the entropy loss. Given some non-uniform sources with low min-entropy, especially, when the sources consist of *more error than entropy* itself, deducting the entropy loss from the sources' min-entropy always output a negative value, hence, show no security. Because of this, correcting t errors regardless of the structure of the input distribution would have to assume sufficient high min-entropy to the input sources. To show meaningful security for standard secure sketch, the min-entropy must at least half of the input length itself [8], hence, limiting the availability of secure sketch construction for low entropy sources.

Through exploitation of the struction of the input distributions, Fuller *et al.*, (2013) [9] have show that the crude entropy loss over 'more error than entropy' sources can be avoided by the measurement of fuzzy min-entropy, which defined as the min-entropy with maximized chances for a variable of W within distance t of w':

$$\mathrm{H}^{\text{fuzz}}_{t,\infty}(W) \overset{\text{def}}{=} -\log\left(\max_{w'} \Pr[W \in B_t(w')]\right),$$

where $B_t(w')$ denoted a hamming ball of radius t around w'. Conceivably, the fuzzy min-entropy is equivalent to the residual entropy, which is at least the min-entropy $\mathrm{H}_\infty(W)$ minus the loss signified by the hamming ball $B_t(w')$ of radius t, s.t.

$$\mathrm{H}^{\text{fuzz}}_{t,\infty}(W) \geq \mathrm{H}_\infty(W) - \log(B_t(w')).$$

$\mathrm{H}^{\text{fuzz}}_{t,\infty}(W)$ is useful for security measurement instead of $\mathrm{H}_\infty(W)$ especially when the residual entropy shows negative value (i.e. more error than entropy). However, due to the fact that $\mathrm{H}^{\text{fuzz}}_{t,\infty}(W)$ depends on the error tolerance distance t, and it is not necessary referring to the worst case distribution for W, therefore, traditional way of showing security with $\mathrm{H}^{\text{fuzz}}_{t,\infty}(W)$ measurement have to deal with such *distribution uncertainty* by considering a family of distributions $\mathcal{W}$ for different variables i.e., $\{W_1, W_2, \ldots\} \in \mathcal{W}$ rather than single distribution. Viewed this way, $\mathrm{H}^{\text{fuzz}}_{t,\infty}(W)$ measurement is only sufficient for computational secure sketch construction [9], [10], which means that the security property of such construction only hold for computationally bounded attacker (i.e., polynomial time bounded) accompanies with strong assumption on the user has a precise knowledge over $\mathcal{W}$. However, it is unrealistic to assume every sources distribution can be modelled precisely, especially for high entropy sources like human biometric.

2 Overview Results

We highlighted our main four results as follow.

To construct a secure sketch for all noisy sources, it was believed that the exploitation of the input structure is necessary [11]. Follow in this way, our works adopted the principle of *Locality Sensitive Hashing (LSH)* to generate a resilient vectors pair (trivially, a pair of longer strings with resilience property) for sketching and recovery. Details discussion on the resilient vector (RV) is covered in Section 5. The involvement of RV has greatly benefited our algorithm designation. In particular, because the RV pair possessing resilience property, i.e., distance preserving, we can make use of such information to derive their input structure. This attributed to our first result, which is the metric of correlation measure between the RV pair and their input pair (Eq. 1 and Eq. 2).

Since the RV is used for sketching, such correlation measurement implies the entropy loss from the input. Therefore, we have the minimum entropy loss from the sketch reduced to the maximum correlation measurement in between the RV pair, conditioned to their inputs. We formalize such minimum entropy loss based on the worst-case distribution for the RV pair and the given inputs. The way we identify such worst-case distribution is by random error parsing, that is, we parse random error to the input when generating RV in sketching. Doing so allow us to define a maximum tolerance distance corresponds to the worst-case input distribution (Corollary 1). This gives us our second results for the derived minimum entropy loss from the sketch (Eq. 4).

Our first result and second result give rise to the third result, where we show that the minimum entropy loss could be at least three bits with BCH error correction codes. This pushed the upper bound of minimum entropy requirement for our secure sketch construction to accept any sources of entropy at least three bits, which is much lower compared to existing construction.

At the end, we deemed that the three bits upper bound can be further pushed down to zero by considering any attacker could have unlimited computation power in modelling the input distribution with other random distribution (viewed as site information), rather than merely to decode the codeword from the sketch (Eq. 15). Nevertheless, the attacker has forced to brute-force the input where its security is defined by the Shannon entropy of the introduced error' distribution (for random error parsing). This refers to our fourth result, which is information-theoretic in claiming security for all noisy sources with any positive value of min-entropy, included the trivial source of min-entropy zero (Proposition 3).

3 Preliminaries

There are some preliminaries to introduce the background of a standard secure sketch, entropy, and error correction code.

Secure sketch: [6] An $(\mathcal{M}, m, \tilde{m}, t)$-secure sketch is a pair of randomized procedures "sketch" (SS) and "Recover" (Rec), with the following properties:

> SS: takes input $w \in \mathcal{M}$ returns a secure sketch (e.g., helper string) $ss \in \{0,1\}^*$.
> Rec: takes an element $w' \in \mathcal{M}$ and ss. If $\mathsf{dis}(w, w') \leq t$ for some tolerance threshold t, then $\mathsf{Rec}(w', ss) = w$ with probability $1 - \beta$, where β is some negligible quantity. If $\mathsf{dis}(w, w') > t$, then no guarantee is provided about the output of Rec.

The security property of secure sketch guarantees that for any distribution W over $\mathcal{M}$ with min-entropy m, the values of W can be recovered by the adversary who observes ss with probability no greater than $2^{-\tilde{m}}$. That is the residual entropy $\tilde{\mathrm{H}}_\infty(W|W') \geq \tilde{m}$.

Min-Entropy: For security, one is always interested in the probability for an adversary to predict a random value, i.e., guessing a secret. For a random variable W, $\max_w \Pr[W = w]$ is the adversary's best strategy to guess the most likely value, also known as the predictability of W. The min-entropy thus defined as

$$\mathrm{H}_\infty(W) = -\log(\max_w \Pr[W = w])$$

min-entropy also viewed as worst case entropy.

Conditioned min-entropy: Given pair of random variable W, and W' (possible correlated), given an adversary find out the value w' of W', the predictability of W is now become $\max_w \Pr[W = w \mid W' = w']$. The conditioned min-entropy of W given W' is defined as

$$\tilde{\mathrm{H}}_\infty(W|W') = -\log\left(\mathbb{E}_{w' \leftarrow W'}\left[\max_w \Pr[W = w \mid W' = w']\right]\right)$$

Error correction code: [12] Let $q \geq 2$ be an integer, let $[q] = \{1, \ldots, q\}$, we called an $[n, k, d]_q$-ary code $\mathcal{C}$ consist of following properties:

- $\mathcal{C}$ is a subset of $[q]^n$, where n is an integer referring to the *blocklength* of $\mathcal{C}$.
- The *dimension* of code $\mathcal{C}$ can be represented as $|\mathcal{C}| = [q]^k = V$
- The *rate* of code $\mathcal{C}$ to be the normalized quantity $\frac{k}{n}$
- The *min-distance* between different codewords defined as $\min_{c,c^* \in \mathcal{C}} \mathsf{dis}(c, c^*)$

It is convenient to view code $\mathcal{C}$ as a function $\mathcal{C} : [q]^k \to [q]^n$. Under this view, the elements of V can be considered as a message $v \in V$ and the process to generate its associated codeword $\mathcal{C}(v) = c$ is called *encoding*. Viewed this way, encoding a message v of size k, always adding redundancy to produce codeword $c \in [q]^n$ of longer size n. Nevertheless, for any codeword c with at most $t = \lfloor \frac{d-1}{2} \rfloor$ symbols are being modified to form c', it is possible to uniquely recover c from c' by using certain function f s.t. $\mathsf{f}(c') = c$. The procedure to find the unique $c \in \mathcal{C}$ that satisfied $\mathsf{dis}(c, c') \leq t$ by using f is called as *decoding*. A code $\mathcal{C}$ is said to be efficient if there exists a polynomial time algorithm for encoding and decoding. Sometime, we refer $[n, k, d]$ code $\mathcal{C}$ as $[n, k, t]$ code $\mathcal{C}$ if the error tolerance distance t is of interested rather than its minimum distance d.

4 Main Idea

We here highlight some common notation to be used in this work, and a brief overview of our construction, focus on binary metric space.

Notations: Let $\mathcal{M}_1 = \{0, 1\}^{k^*}$, and $\mathcal{M}_2 = \{0, 1\}^n$ denote two different sizes of metric spaces where $n > k^*$. The distance between different binary string w

and w' denoted as $\mathsf{dis}(w, w')$ is the binary hamming distance (e.g., the number of disagree elements), i.e., $\mathsf{dis}(w, w') = \|w \oplus w'\|$ where $\|.\|$ is the hamming weight that count the number of non-zero elements, and $\oplus$ is the addition modulo two operation (XOR). Besides, the error rate in between the input w $w' \in \mathcal{M}_1$ is denoted as $\|w \oplus w'\| \, (k^*)^{-1}$ which is simply their normalized hamming distance.

For error correction code notation, despite there existing a lot of error correction codes available in practice, due to the efficiency consideration, we used the commonly studied binary error correction code named as BCH code [13] with minimum distance $d \geq 2t + 1$ and efficient decoding algorithm f via algebric method, i.e., syndrome decoding [13]. Our construction used two BCH codes. We called one of these as 'inner' code $\mathcal{C}_{in}$, and another one called the 'outer' code $\mathcal{C}_{out}$. Both of them are chosen to be BCH codes with parameter $[n^*, k^*, t^*]_2$ for $\mathcal{C}_{in}$ and $[n, k, t]_2$ for $\mathcal{C}_{out}$, where $k^* < n^* < k < n$ holds. We denote the tolerance rate of code $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ as $\xi^* = t^*/n^*$ and $\xi = t/n$ respectively.

Overview Construction: Suppose Alice wishes to conceal a noisy non-uniform string $w \in \{0, 1\}^{k^*}$ while allows exact recovery of w by using another noisy string $w' \in \{0, 1\}^{k^*}$ that is close to w.

Firstly, Alice encodes w using the 'inner' code $\mathcal{C}_{in}$ to output a codeword c^*. Then, c^* is used to generate a noisy string $v^* \in \{0, 1\}^k$ with w. Eventually, v^* is being encoded by the 'outer' code $\mathcal{C}_{out}$ to output the final codeword $c \in \mathcal{C}_{out}$. Alice then conceals c by generating a sketch $ss = c \oplus \delta$ which is then made public and leaving the offset δ in the clear. The offset δ is characterized by a pair of resilient vectors $\phi, \phi' \in \{0, 1\}^n$, which is generated from a pair of random noisy strings $w'_e, w_e \in \{0, 1\}^{k^*}$, i.e., $w_e = w \oplus e$ (with additional error vector e) through Ω with public shared random string N. The resilient vectors offer resilience for the recovery of w from w' if $\|\delta\| \leq t$ and $\|w \oplus w'\| \leq t^*$.

5 Resilient Vector: Properties and Generation

Since RV is a core element of our construction, we here provide details discussion on its properties and how it can be generated. The concept of RV is derived from Locality Sensitive Hashing (LSH) defined as below.

Locality Sensitive Hashing [14] Given that $P_2 > P_1$, while $w, w' \in \mathcal{M}$, and $\mathcal{H} = h_i : \mathcal{M} \rightarrow U$, where U refers to the output metric space (after hashing), which comes along with a similarity function S, where i is the number of hash functions h_i. A locality sensitive hashing can be viewed as a probability distribution over a family $\mathcal{H}$ of hash functions follows $P_{h \in \mathcal{H}}[h(w) = h(w')] = S(w, w')$. In particular, the similarity function S described the hashed collision probability in between w and w'.

$$P_{h \in \mathcal{H}}(h_i(w) = h_i(w')) \leq P_1, \quad \text{if } S(w, w') < R_1$$
$$P_{h \in \mathcal{H}}(h_i(w) = h_i(w')) \geq P_2, \quad \text{if } S(w, w') > R_2$$

LSH transforms input w and w' to its output metric space U with property that ensuring similarity inputs render higher probability of collision over U, and vice versa.

For RV generation, we only focus on a particular LSH family called hamming-hash [15]. The hamming hash is considered as one of the easiest ways to construct an LSH family by bit sampling technique.

Hamming hash strategy: *Let $[k^*] = \{1, \ldots, k^*\}$. For Alice with $w \in \{0,1\}^{k^*}$ and Bob with $w' \in \{0,1\}^{k^*}$. Alice and Bob agreed on this strategy as follow:*

1. *They are told to each other a common random integer $N \in [k^*]$.*
2. *They separately output '0' or '1' depend upon their private string w and w', i.e., Alice output '1' if the N-th bit of w is '1', else output '0'.*
3. *They win if they got the same output, i.e., $w(N) = w'(N)$.*

Based on above strategy, we are interested in the probability for Alice and Bob outputting the same value. This probability can be described by a similarity function $S(w, w') = P$ where $P \in [0, 1]$.

Theorem 1. *Hamming hash strategy is a LSH with similarity function $S(w, w') = 1 - \|w \oplus w'\|(k^*)^{-1}$*

The hamming hash strategy can also be operated in between Alice and Bob in an non-interactive way. To do so, Alice and Bob simply repeat Step 1 and Step 2 for n number of times with a set of pre-shared random integers $N = [N(1), N(2), \ldots, N(n)] \in [k^*]^n$. In the end, they can output a n bits string ϕ, and ϕ' respectively over $\{0,1\}^n$, which we have earlier named as *resilient vectors*. We denote such non-interactive hamming-hash algorithm as $\Omega : \mathcal{M}_1 \times [k^*]^n \to \mathcal{M}_2$, which serves to sample the input binary string of size k^* into a longer binary string a.k.a resilient vector of size $n > k^*$.

Given input $w \in \{0,1\}^{k^*}$, and $N \in [k^*]^n$, the algorithm for $\Omega : \mathcal{M}_1 \times [k^*]^n \to \mathcal{M}_2$ can be described as follow:

```
$\Omega(w, N)$
$\phi \leftarrow \emptyset$
for $i = 1, \ldots, n$ do
    parse $x = w(N(i))$// $x$ is the $N(i)$-th bits of $w$
    $\phi = \phi \| x$
endfor
return $\phi$
```

Theorem 2. *Suppose two resilient vectors $\phi, \phi' \in \{0,1\}^n$ are generated from $w, w' \in \{0,1\}^{k^*}$ respectively using hamming hash algorithm Ω with a random integer string $N \in [k^*]^n$, then $\mathbb{E}[\|\phi \oplus \phi'\|] = n \|w \oplus w'\| (k^*)^{-1}$.*

Correlation Measure in RVs: Let Φ and Φ' be two random variables over $\{0,1\}^n$, and W and W' be two random variables over $\{0,1\}^{k^*}$. Given a resilience vector $\phi \in \Phi$ generated from $w \in W$ with random string N, it follows Φ must correlate with W where the probability to look for any random variable $\Phi \in B_t(\phi')$ (also means similar resilience vector s.t. $\|\phi \oplus \phi'\| \leq t$) varies conditioned on either $W \notin B_{t'}(w')$ or $W \in B_{t'}(w')$. Note that $W \in B_{t'}(w')$ implies the inputs $w \in W$ and $w' \in W'$ must similar within distance t' (e.g. $\|w \oplus w'\| \leq t'$), while $W \in B_{t'}(w')$ means $\|w \oplus w'\| > t'$. Such correlation can be measured by using the conditional probability described as $\Pr[\Phi \in B_t(\phi') \mid W \notin B_{t'}(w')]$ or $\Pr[\Phi \notin B_t(\phi') \mid W \in B_{t'}(w')]$ respectively. In particular, we are more interested on the maximum correlation, which can be conveniently expressed by the conditioned *maximum* probability in looking for $\Phi \in B_t(\phi')$ given $W \notin B_{t'}(w')$ defined as:

$$\mathbb{E}_{w' \leftarrow W'}\left[\max_{\phi'} \Pr[\Phi \in B_t(\phi') \mid W \notin B_{t'}(w')]\right] \tag{1}$$

On the other hand, the conditioned *maximum* probability in looking for $\Phi \notin B_t(\phi')$ given $W \in B_{t'}(w')$ is defined as:

$$\mathbb{E}_{w' \leftarrow W'}\left[\min_{\phi'} \Pr[\Phi \notin B_t(\phi') \mid W \in B_{t'}(w')]\right] \tag{2}$$

6 Sketching

We denote the sketching algorithm that employs the hamming-hash algorithm, Ω, an $[n^*, k^*, t^*]_2$ 'inner' code $\mathcal{C}_{in}$ and an $[n, k, t]_2$ 'outer' code $\mathcal{C}_{out}$ as $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$. The sketching algorithm $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$ with inputs w, N, and ϵ_{ss} is described as follow:

```
$\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss})$
$\mathcal{E}_{ss} \leftarrow\!\$\ \{0,1\}^{k^*}$ // initiate $\mathcal{E}_{ss}$ according to the error parameter $\epsilon$
$e \leftarrow\!\$\ \mathcal{E}_{ss}$ // sample $e$ from $\mathcal{E}_{ss}$ uniformly at random, where $\|e\| = \lceil k^* \epsilon_{ss} \rceil$
$c^* = \mathcal{C}_{in}(w)$; // encode $w$
$v_{syn} = c^* \oplus (0^{n^*-k^*} \| w)$
$v^* = 0^{k-n^*} \| v_{syn}$;
$c = \mathcal{C}_{out}(v^*)$; // encode $v^*$
$w_e = w \oplus e$;
$\phi \leftarrow \Omega(w_e, N)$
$ss = c \oplus \phi$;
return $ss$
```

Our sketching procedure consists of mainly two step encoding. Given an input $w \in \{0,1\}^{k^*}$, the first encoding stage used $\mathcal{C}_{in}$ to encode w to generate a codeword $c^* \in \{0,1\}^{n^*}$. In principle, c^* can be any random codeword over $\mathcal{C}_{in}$ to be concealed, including the trivial codeword of all zeros i.e. $c^* = 0^{n^*}$. Then, we pad w with zeros in front to generate a longer bit string, which can be viewed as the syndrome vector denoted as $v_{syn} = c^* \oplus (0^{n^*-k^*} \| w)$. Clearly, v_{syn} conceals c^* by using w. The syndrome vector itself is also a codeword $v_{syn} \in \mathcal{C}_{in}$. Then, the second encoding stage used $\mathcal{C}_{out}$ to encode $v^* = 0^{k-n^*} \| v_{syn}$ to generate the final code word c. The 0^{k-n^*} zeros in front is used to notify the recovery algorithm if the decoding is success. The final sketch is formed by hiding c with RV generated from the noisy string w_e.

For the realisation of the noisy string w_e, we parse additional error to the original input w using a random error vector $e \in \mathcal{E}_{ss}$ sampled from some random distribution $\mathcal{E}_{ss}$. Such error distribution is parametrized by an error parameter $\epsilon_{ss} > 0$. To be specific, we have all error vector $e \in \mathcal{E}_{ss}$ is of weight $\|e\| = \lceil k^* \epsilon_{ss} \rceil$. The error vector e is leaving in clear after it has being parsed into the input w to form w_e.

All steps on $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss})$ can be done in $O(n^2)$, and the size of ss is now depend upon the blocklength n of the chosen 'outer' code $\mathcal{C}_{out}$.

7 Recovery

We denote the recover algorithm that employed the hamming-hash algorithm, Ω, an $[n^*, k^*, t^*]_2$ 'inner' code $\mathcal{C}_{in}$ and an $[n, k, t]_2$ 'outer' code $\mathcal{C}_{out}$ as $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. The recover algorithm $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ with inputs ss, w', N, ϵ_{rec} to recover w is described as follow:

```
$\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}(ss, w', N, \epsilon_{rec})$
$\mathcal{E}_{rec} \leftarrow_\$ \{0,1\}^{k^*}$ // initiate $\mathcal{E}_{rec}$ with error parameter $\epsilon_{rec}$
    for $i = 1, \ldots, |\mathsf{supp}(\mathcal{E}_{rec})|$
    $e'_i \leftarrow_\$ \mathcal{E}_{rec}$ // sample $e'_i$ differently at random at random, where $\|e'_i\| = \lceil k^* \epsilon_{rec} \rceil$
    $w'_{e'_i} = w' \oplus e'_i$
    $\phi'_i \leftarrow \Omega(w'_{e'_i}, N)$
    $c'_i = ss \oplus \phi'_i$ // also $ss \oplus \phi'_i = c \oplus (\phi \oplus \phi'_i)$
    $c \leftarrow \mathsf{f}(c'_i)$ // first decoding
    return $v^* = \mathcal{C}_{out}^{-1}(c)$
        if $v^*[1], \ldots, v^*[k-n^*] = 0^{k-n^*}$ // first $k-n^*$ bits of $v^*$ are zeros
            set $v'_{syn}$ as the last $n^*$ elements of $v'^*$
            $c'^* = v'_{syn} \oplus (0^{n^*-k^*} \| w')$
            $c^* \leftarrow \mathsf{f}(c'^*)$ // second decoding
            return $w = {\mathcal{C}_{in}}^{-1}(c^*)$
            break
        endif
    endfor
```

Our proposal for the recovery algorithm consists of mainly two decoding process. for decoding, we refer f be the syndrome decoding algorithm which operate in $\mathcal{O}\left((n^*)^t\right)$ and $\mathcal{O}(n^t)$ for $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ respectively. The first decoding process is designed to be iterative decoding uses $\mathcal{C}_{out}$ and f to output the codeword c from the sketch. It can be conveniently viewed as a brute-force decoding procedure of $|\mathsf{supp}(\mathcal{E}_{rec})|$ trials with some distribution $\mathcal{E}_{rec}$, parametrized by another error parameter $\epsilon_{rec} > 0$. Noting that the error distribution introduced during the recovery phase is different compared to the sketching phase ($\mathcal{E}_{ss}$). This is because we are here to consider a more general case where the person in recovering w may or may not know $\mathcal{E}_{ss}$, therefore the brute-force complexity for the first decoding is highly depends on the given error distribution $\mathcal{E}_{rec}$, where all error vector $e' \in \mathcal{E}_{rec}$ is of weight $\|e'\| = \lceil k^* \epsilon_{rec} \rceil$. The main goal of the first decoding is to output the syndrome vector v_{syn}. This can be done by examining the recovered vector $v^* = 0^{k-n^*} \| v_{syn}$. If the first $k - n^*$ bits of v^* are all zeros, the decoding is viewed as success and thus the recovery algorithm could proceed to the second decoding stage to recover c^* and so w from v_{syn} using $\mathcal{C}_{in}$.

The second decoding process is basically a traditional BCH code decoding with f. It decodes the corrupted syndrome vector (viewed as the corrupted codeword c'^*) to output c^*. The decoding itself must success if $\|w \oplus w'\| \leq t^*$, thus w can be recovered from c^*.

8 Distribution Hiding with Random Error Parsing

Recall that the recover algorithm $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ can be viewed as a brute-force decoding procedure of $|\mathsf{supp}(\mathcal{E}_{rec})|$ trials. We stress that such brute-force trial is necessary to show optimal security while allowing more errors to be tolerated. This is mainly because the input $w \in W$ could be under some random distribution W over $\mathcal{M}_1$, yielding a random RV $\phi \in \varPhi$ of random distribution $\varPhi$ over $\mathcal{M}_2$. However, tolerating t errors using an $[n, k, t]$ error correction code eventually reveal W. This is because the encoding process must ensure all random variable $W \in B_t(w')$ can be tolerated by decoding function f, therefore, the encoding process must reveal W. Adding redundancy to W inevitably introduce t information loss, hence cannot show security to more error than entropy sources. Moreover, it is understood that potential attacker may have better computational power in modelling W, leading to better knowledge over W and so higher entropy loss from W is possible.

To resolve the above issue, a straightforward way is to hide W before adding redundancy to it. This can be done by parsing an error randomly and uniformly chosen from a list $\{e_1, e_2, \ldots, e_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \mathcal{E}_{ss}$ into the input string $w \in W$ during the sketching phase. Doing so will generate a list of noisy strings over different distributions $\{W_1, W_2, \ldots, W_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \mathcal{W}$ respectively. This implies we would have a list of possible RVs in different distributions $\{\varPhi_1, \varPhi_2, \ldots, \varPhi_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \varPsi$ as well. Remark here because the syndrome vector $v_{syn} \in \mathcal{C}_{in}$ is concealed by w over W before the second encoding stage take place. Therefore it is naturally to have $W \in \mathcal{W}$, referring to the trivial case when the error vector $e \in \mathcal{E}_{ss}$ is all zeros. Based on above argument, we should have the original distribution W is now concealed over $\mathcal{W}$. In such a case, we have to consider family of distributions in $\mathcal{W}$ and Ψ rather than single distribution W and Φ over $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively in deriving the security of the sketch.

Because the sketch is generated by concealing the final codeword c with an RV ϕ. It follows that the best (worst-case) security of the sketch is manifested by the worst case distribution over Ψ, where all points (or RVs) in this distribution are very close to each other. Suppose the worst-case distribution is $\Phi \in \Psi$ over Ψ, we shall use the min-entropy $\mathrm{H}_\infty(\Phi)$ measurement to measure the entropy of such worst-case distribution. Arguing that the points in the worst case distribution are most close to each other, it must offer highest probability of success in getting a similar RV within distance t. Compared to blindly modelling Ψ to determine Φ, it is relatively easier to measure its maximum probability of success in getting a similar RV within distance t among all possible distributions $\{\Phi_1, \Phi_2, \ldots, \Phi_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \Psi$. Such maximum probability must correspond to the maximum tolerance distance $t = t_{\max}$ in Ψ. To do so, since Φ is conditioned on some random distribution $W \in \mathcal{W}$, we have to first define a maximum tolerance distance $t_{(+)}$ over $W \in \mathcal{W}$ for sketching. We define such maximum tolerance distance as $t_{(+)} = \lfloor (\xi + \epsilon_{ss})k^* \rfloor$. where $\xi = t/n$ is parametrized by the chosen $[n, k, t]$ error correction code $\mathcal{C}_{out}$. Since the number of errors can be tolerated is bounded by the tolerance distance of the selected codes pair $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$. This also means that the value of ϵ_{ss} should be bounded by the maximum achievable error tolerance rate among $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$, i.e, $\epsilon_{ss} \leq \max\{\xi^*, \xi\}$. Because $t^* < t$, certainly $\epsilon_{ss} \leq \xi$. Above reasoning has answered why we shall refer to ξ but not ξ^* for $t_{(+)}$. With $\epsilon_{ss} > 0$ and $\max\{\xi^*, \xi\} = \xi$, we know that $t_{(+)}$ should be maximum in $W \in \mathcal{W}$. It follows the maximum probability to look for similar RV over Ψ can be reduced to measuring the maximum correlation among all variables $\{\Phi_1, \Phi_2, \ldots, \Phi_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \Psi$ given $W \in \mathcal{W}$ (see Eq. 1). The Corollary below characterized the worst case security of RVs. Such security is measured in term of conditioned maximum probability in getting similar RV within maximum tolerance distance $t = t_{\max}$ over Ψ given W.

Corollary 1. *Given $W \in \mathcal{W}$, for all RV over a family of distributions Ψ, the conditioned maximum probability to look for any similar RV ϕ in random distribution $\Phi \in B_t(\phi')$ over Ψ (i.e. $\|\phi \oplus \phi'\| = \|\delta\| \leq t$ for all $\phi \in \Phi$) when $W \notin B_{t_{(+)}}(w')$ is measured to be*

$$\begin{aligned}
&= \mathbb{E}_{w' \leftarrow W'}\left[\max_{\phi'} \Pr\left[\Phi \in B_t(\phi') \,\middle|\, W \notin B_{t_{(+)}}(w')\,\right]\right] \\
&= \min_{t = t_{\max}} \Pr\left[\|\delta\| \leq t \,\middle|\, \|w \oplus w'\| \geq t_{(+)}\,\right] \leq \exp\left(-2n\epsilon_{ss}^2\right) \qquad (3)
\end{aligned}$$

Proof. For $W \notin B_{t_{(+)}}(w')$, it means that for all $w \in W$, $\|w \oplus w'\| \geq t_{(+)}$, $\|w \oplus w'\|(k^*)^{-1} \geq \xi + \epsilon_{ss}$ is always true. Multiplying both sides of the inequality with n, then $n\|w \oplus w'\|(k^*)^{-1} \geq n\xi + n\epsilon_{ss}$ and yield $t \leq \mathbb{E}[\|\delta\|] - n\epsilon_{ss}$ (by Theorem 2). Let $\mathbb{E}[\|\delta\|] = t$, we shall have an $t = t_{\max}$ for LHS of the inequality $t \leq \mathbb{E}[\|\delta\|] - n\epsilon_{ss}$ s.t. $t_{\max} = t - n\epsilon_{ss}$, then the probability for $\|\delta\| \leq t_{\max}$ given

$\|w \oplus w'\|(k^*)^{-1} \geq \xi + \epsilon_{ss}$ can be computed by *Hoeffding's inequality* follows the last line of Eq. 3.

By Corollary 1, we shall have the conditioned min-entropy of RV over Ψ measured to be

$$\begin{aligned}
\tilde{\mathrm{H}}_\infty(\Psi|\mathcal{W}) &= -\log\left(\mathbb{E}_{W\leftarrow\mathcal{W}}[\Pr[\Psi = \Phi \mid \mathcal{W} = W]]\right) \\
&= -\log\left(\mathbb{E}_{w'\leftarrow W'}\left[\max_{\phi'}\Pr\left[\Phi \in B_t(\phi') \;\middle|\; W \notin B_{t_{(+)}}(w')\right]\right]\right) \\
&\geq \log(1/\exp(-2n\epsilon_{ss}^2)) \qquad (4)
\end{aligned}$$

9 Correctness with Regardless Computational Power

We are here to discuss the correctness of $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ in recovering w from a sketch. specifically, such correctness characterizes the *success rate* of the recovery of v_{syn} and so w from a sketch generated by $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$. Noting that the number of iterations (or computational power) requirement for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ is proportional to the value of $|\mathsf{supp}(\mathcal{E}_{rec})|$, which is parametrized by ϵ_{rec}. In such a case, higher value of ϵ_{rec} would result to higher number of $|\mathsf{supp}(\mathcal{E}_{rec})|$, implying higher computational power requirement for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. Besides, the efficiency of the decoding algorithm f used for the given error correction code, i.e. $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ also plays an important rules in computing the final computational power requirement for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. Nonetheless, we here consider the correctness with negligible error without the consideration of the computational power in running $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. Formally, it can be expressed in Eq. 5 below with some negligible probability $\beta > 0$ for all $\epsilon_{rec} > 0$, regardless the efficiency of the decoding algorithm f itself.

$$\Pr[\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}(\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss}), w', N, \epsilon_{rec}) = w] = 1 - \beta. \qquad (5)$$

We wish to show that the recovery of w can *at least* be done in probability $1 - \beta$. Therefore, we have to derive the maximum value for β. This can be measure with refer to the maximum error in recovering w from a sketch with $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$

Suppose a sketch ss generated through $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss}) = ss$, where $ss \in SS$ under some random distribution SS over $\mathcal{M}_2$. Since the total number of possible RVs generated from the input w (with random string N) implies the total number of possible sketches, therefore, all RVs over the family of distributions $\{\Phi_1, \Phi_2, \ldots, \Phi_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \Psi$ will yield different sketches in different distributions $\{SS_1, SS_2, \ldots, SS_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \mathcal{S}$ as well. In such a case, the error (in term of probability) to recover w from a sketch $ss \in SS$ reduced to the error to look for a random variable SS_i over family of distributions $\mathcal{S}$, where the RV pair (ϕ, ϕ') used for sketching and recovery is in distinct, (i.e., $\|\phi \oplus \phi'\| = \|\delta\| \geq t$). Therefore, to maximize such error for maximum value of β, we need to determine the optimal distribution from Ψ, where all points (or RVs) in such distribution are farthest away to each other. Suppose the optimal distribution is $\Phi \in \Psi$, since the points in Φ is farthest to each other, it should correspond to the highest probability in looking for a distinct RV. Likewise the way we used to determine the worst-case distribution from Ψ (refer to last section), instead of modelling Ψ for such optimal Φ we can derive the maximum value of β with refer to the minimum tolerance distance $t = t_{\min}$ over Ψ. This can be argued with the probability to look for two points which are distinct at least some distance t should be maximum given the distance is minimum $t = t_{\min}$. Before this, since $\{\Phi_1, \Phi_2, \ldots, \Phi_{|\mathsf{supp}(\mathcal{E}_{ss})|}\} \in \Psi$ are conditioned on $W \in \mathcal{W}$, we first have to define another minimum tolerance distance $t_{(-)} = \lceil (\xi - \epsilon_{ss})k^* \rceil$ over $W \in \mathcal{W}$ for sketching. As an important remark, for $t_{(-)}$ to be minimum given $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$, we need $t_{(-)} \leq t^* < t$ to hold as well. Eventually, we have the following Corollary revealing $\beta \leq \exp(-2n\epsilon_{ss}^2)$.

Corollary 2. *Given $W \in \mathcal{W}$, for all RV over a family of distributions Ψ, the conditioned maximum probability to look for any distinct RV ϕ in random distribution $\Phi \notin B_t(\phi')$ over Ψ (i.e. $\|\phi \oplus \phi'\| = \|\delta\| \geq t$ for all $\phi \in \Phi$) when $W \in B_{t_{(-)}}(w')$ is measured to be*

$$\begin{aligned} &\mathbb{E}_{w' \leftarrow W'} \left[\max_{\phi'} \Pr \left[\Phi \notin B_t(\phi') \;\middle|\; W \in B_{t_{(-)}}(w') \right] \right] \\ &= \max_{t = t_{\min}} \Pr \left[\|\delta\| \geq t \;\middle|\; \|w \oplus w'\| \leq t_{(-)} \right] \leq \exp\left(-2n\epsilon_{ss}^2\right). \end{aligned} \tag{6}$$

Proof. For $W \in B_{t_{(-)}}(w')$, it means that for all $w \in W$, $\|w \oplus w'\| \leq t_{(-)}$, $\|w \oplus w'\|(k^*)^{-1} \leq \xi - \epsilon_{ss}$ is always true. Multiplying both sides of the inequality with n, then $n\|w \oplus w'\|(k^*)^{-1} \leq n\xi - n\epsilon_{ss}$ and yield $t \geq \mathbb{E}[\|\delta\|] + n\epsilon_{ss}$ (by Theorem 2). Let $\mathbb{E}[\|\delta\|] = t$, we shall have an $t = t_{\min}$ for LHS of the inequality $t \geq \mathbb{E}[\|\delta\|] + n\epsilon_{ss}$ s.t. $t_{\min} = t + n\epsilon_{ss}$, then the probability for $\|\delta\| \geq t_{\min}$ given $\|w \oplus w'\|(k^*)^{-1} \leq \xi - \epsilon_{ss}$ can be computed by *Hoeffding's inequality* follows the last line of Eq. 6.

Therefore, we obtain the following Proposition by comparing Eq. 5 and Eq. 6.

Proposition 1. *For all $\epsilon_{rec} > 0$, $\epsilon_{ss} > 0$, and $\|w \oplus w'\| \leq t_{(-)} \leq t^*$,*

$$\begin{aligned} &\Pr\left[\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}(\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss}), w', N, \epsilon_{rec}) = w\right] \\ &\geq 1 - \exp\left(-2n\epsilon_{ss}^2\right) \end{aligned}$$

Proposition 1 concluded given an error parameter $\epsilon_{ss} > 0$. For all $\epsilon_{rec} > 0$, the success rate for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ in recovering w is at lest $1 - \exp\left(-2n\epsilon_{ss}^2\right)$ if $\|w \oplus w'\| \leq t_{(-)} \leq t^*$. This resultant probability is overwhelming when n is sufficiently large without considering the computational power in running $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$.

10 Correctness with Regard to Computational Power

Noting that our claim of correctness in Proposition 1 demonstrating at most $t_{(-)} \leq t^*$ of errors (or error rate of $\xi - \epsilon_{ss}$) can be tolerated over the family of input distributions $\mathcal{W}$ regardless the computational power in running $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. In this section, we will show that, if $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ has run in $|\mathsf{supp}(\mathcal{E}_{rec})|$ number of iterations, we can actually tolerate more error. In particular, when one set ϵ_{rec} of higher than ϵ_{ss}, i.e., $\epsilon_{rec} \geq 2\epsilon_{ss}$, he/she is possible of achieving higher error tolerance rate, arbitrary close to 1/2 over the input distribution $W \in \mathcal{W}$.

Our main intuition is to tolerate $t_{(+)} > t_{(-)}$ number of errors. Recall $t_{(+)} = \lfloor (\xi + \epsilon_{ss})k^* \rfloor$ and $\epsilon_{ss} \leq \xi$. This implies at most $2\xi < 1/2$ of the error rate can be tolerated by $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$. To do so, for any error described as the distance $\|w \oplus w'\| \leq t_{(+)}$, it means the error rate is $\|w \oplus w'\|(k^*)^{-1} \leq \xi + \epsilon_{ss}$. By introducing an error parameter ϵ_{rec} of higher value compared to ϵ_{ss}, e.g., $\epsilon_{rec} \geq 2\epsilon_{ss}$, the final worst case error rate can be described as $\|w \oplus w'\|(k^*)^{-1} \leq \xi + \epsilon_{ss} \pm \epsilon_{rec}$. Viewed this way, any error rate of $\|w \oplus w'\|(k^*)^{-1} \leq \xi + \epsilon_{ss}$ is possible to be reduced down to $\|w \oplus w'\|(k^*)^{-1} \leq \xi - \epsilon_{ss}$ given high enough ϵ_{rec} during recovery. Eventually, the remaining errors $\|w \oplus w'\| \leq t_{(-)}$ can be tolerated with overwhelming probability by Proposition 1.

To make the idea more explicit, suppose one knows the value of ϵ_{ss}, he/she can simply choose a minimum $\epsilon_{rec} = 2\epsilon_{ss}$ during recovery, this would allows him/her to generate another list of possible error vectors $\{e'_1, \ldots, e'_{|\mathsf{supp}(\mathcal{E}_{rec})|}\} \in \mathcal{E}_{rec}$, which would result to a list of noisy string $w_{e_i} \in W_i$ under a family of distributions $\mathcal{W}$, s.t. $\{W_1, \ldots, W_{|\mathsf{supp}(\mathcal{E}_{rec})|}\} \in \mathcal{W}$. Remark here we have the original input distribution $W \in \mathcal{W}$ holds by trivial case when $\epsilon_{ss} = 0$. Because the error e of weight $\lceil k^*\epsilon_{ss} \rceil$ sampled during the sketching phase is random. We shall have $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ running in $|\mathsf{supp}(\mathcal{E}_{rec})|$ iterations to try all possible $\{e'_1, \ldots, e'_{|\mathsf{supp}(\mathcal{E}_{rec})|}\} \in \mathcal{E}_{rec}$ for $W \in \mathcal{W}$. Let $d' = \lceil \xi k^* \rceil$. Suppose the original distance $\|w \oplus w'\| \leq d'$ holds (without considering ϵ_{ss}). For all error e' of weight $\lceil 2k^*\epsilon_{ss} \rceil$, there must be at least one $e' \in \mathcal{E}_{rec}$ for us to get

$$\begin{aligned} \|w_e \oplus w'_{e'}\| &= \|(w \oplus e) \oplus (w' \oplus e')\| = \|(w \oplus w') \oplus (e \oplus e')\| \\ &\leq d' + (\lceil k^*\epsilon_{ss} \rceil \pm \lceil 2k^*\epsilon_{ss} \rceil) \leq d' - \lceil k^*\epsilon_{ss} \rceil = t_{(-)} \end{aligned} \tag{7}$$

Doing so means we have to find the nontrivial error vector $e' \in \mathcal{E}_{rec}$ using error parameter ϵ_{rec} s.t. the distance $\|w \oplus w'\| \leq t_{(-)}$ holds. To look for such nontrivial error vector, the value of ϵ_{ss} used in the sketching phase have to be known by the recovery algorithm. Even so, only with the knowledge of ϵ_{ss}, there is no short-cut or direct way rather than brute-forcing for such e'. Therefore, we shall see that to tolerate more error (more than $t_{(-)}$), we now inevitably have to deal with the issue of computational power in running $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$, where the knowledge on the chosen error parameter is necessary.

Given $\epsilon_{ss} \in [(2k^*)^{-1}, 1/4]$, with minimum $\epsilon_{rec} = 2\epsilon_{ss}$, we can use *Stirling approximation* to obtain the minimum value for $|\mathsf{supp}(\mathcal{E}_{rec})|$:

$$|\mathsf{supp}(\mathcal{E}_{rec})| = \binom{k^*}{\lceil 2k^*\epsilon_{ss}\rceil} \leq 2^{\lceil k^* h_2(2\epsilon_{ss})\rceil} \tag{8}$$

where $h_2(x) = -x\log(x) - (1-x)\log(1-x)$ is the binary entropy function with input error rate of x. Result on Eq. 8 showing that after $2^{\lceil k^* h_2(2\epsilon_{ss})\rceil}$ number of iterations, we shall have $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to recover w successfully with overwhelming probability.

Follows Eq.7, we have $\|w \oplus w'\| \leq t_{(-)} = d' - \lceil k^*\epsilon_{ss}\rceil$ is the requirement for correctness. We only need to show meaningful correctness for any error in term of distance $\|w \oplus w'\| \geq 0$ which is positive. It means $d' \geq \lceil k^*\epsilon_{ss}\rceil$ must be the permissible minimum distance in our construction. This also implies $\lceil k^*\epsilon_{ss}\rceil \leq \|w \oplus w'\| \leq t_{(+)}$ would be the bounded error range tolerable by $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$.

10.1 Relaxation to Polynomial Time Recovery with Trapdoor Information

Nonetheless, Eq. 8 suggesting exponential computation time in the input size k^* for $\epsilon_{rec} = 2\epsilon_{ss}$, which is highly inefficient for large k^*. We want $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to run in an more efficient manner. Since $k^* < n$, we therefore use n as upper bound and define $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to be *efficient* if it can run in polynomial time $\mathsf{poly}(n)$ in the input sketch size n to show correctness.

Intuitively, to show efficient recovery, we first have to ensure the two decoding stages in $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ can done in an efficient manner. Since both decoding stages use syndrome decoding algorithm which operates in $\mathcal{O}\left((n^*)^{t^*}\right)$ and $\mathcal{O}(n^t)$ for $\mathcal{C}_{in} \in \{0,1\}^{n^*}$ and $\mathcal{C}_{out} \in \{0,1\}^n$ respectively. It is clearly shown that they are efficient in polynomial time $\mathsf{poly}(n)$. Nevertheless, we will show that for such efficiency claim, we would require $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to attain certain minimum correctness requirement. In other words, we would like to show that $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ can only be efficient given its correctness is claimed with some probability $1-\beta$, which can only be achieved with proper choice of the error correction codes $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$.

Recall that we have the derived $\beta \leq \exp(-2n\epsilon_{ss}^2)$. It is not difficult to see that the value of β is in fact closely tied to the numbers of zeros padding to the syndrome vector v_{syn} to form $v^* = 0^{k-n^*}\|v_{syn}$ (see Step 4 of $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$). This is because $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ will only proceed to the second decoding if the first decoding return v^* with the first $k-n^*$ bits are all zeros. Eventually, the second decoding must success in recovering w with probability at least $1-\beta$ given $\|w \oplus w'\| \leq t_{(-)}$. Due to the selection of $e' \in \mathcal{E}_{rec}$ is random, every iteration of the first decoding must return a random codeword $c \in \mathcal{C}_{out}$, hence its first $k-n^*$ bits should be random over $\{0,1\}^{k-n^*}$. In viewed of this, we could have $\beta = 2^{-(k-n^*)}$ of probability for the first $k-n^*$ bits are all zeros, which implies the second decoding shall success with probability $1-2^{-(k-n^*)}$ revealed by the number of zeros padding to the syndrome vector. However, for such argument to hold, we need to ensure

$$\beta \leq \exp(-2n\epsilon_{ss}^2) \leq 2^{-(k-n^*)}, \tag{9}$$

which shall give us the tighter upper bound of error rate measured as $2^{-(k-n^*)}$ compared to Eq. 6. As we shall see, Eq. 9 can be achieved easily with sufficient large value of n for a given ϵ_{ss}, n^* and k.

In such a case, the correctness derived previously in Section 9 reduced the number of zeros padding in the sketching phase, measured as $k - n^*$. Noting that such reduction is computational. It depends on the construction itself, where the selection of the error correction codes $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ (given ϵ_{ss}) with parameters $[n^*, k^*, t^*]$ and $[n, k, t]$ respectively is viewed as an important factor for efficiency claim. In particular, under the designation of an BCH code [13] used in our construction for $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$, its correctness is defined using some positive integer $m' \geq 3$. Given the value of tolerance distance $t < 2^{m'-1}$, we can construct an $[n, k, t]$ BCH code $\mathcal{C}_{out}$ with parameters $n = 2^{m'} - 1$, $n - k \leq m't$ and minimum distance $d \geq 2t - 1$ (something applied to $\mathcal{C}_{in}$). With such reasoning, follows Eq. 9, for sufficiently large n, we shall have the derived value for m':

$$m' = \left\lfloor \log(1/\exp(-2n\epsilon_{ss}^2)) \right\rfloor \geq 3 = k - n^* \tag{10}$$

Above statement also prove the existence of BCH codes for $\mathcal{C}_{out}$ and $\mathcal{C}_{in}$ in our construction with efficient syndrome decoding algorithm f only if Eq. 10 holds (i.e., at least three zeros padding to v_{syn}).

With efficient decoding stages, the efficiency of $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ thus reduced to the brute-force complexity itself, which is proportional to the value of $|\mathsf{supp}(\mathcal{E}_{rec})|$. Follows Eq. 8 and Eq. 10, if we properly choose $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ s.t. $\lceil k^* h_2(2\epsilon_{ss}) \rceil \leq k - n^*$, such complexity can be bounded in term of the sketch size n described as:

$$|\mathsf{supp}(\mathcal{E}_{rec})| \leq 2^{\lceil k^* h_2(2\epsilon_{ss}) \rceil} \leq 2^{(k-n^*)} = 2^{m'} = n + 1 \tag{11}$$

Expressing so would allow us to relax the number of iteration for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ from exponential time in k^* to linear time in n. In particular, since the syndrome decoding for both decoding stages operate in $\mathcal{O}(n^t)$, the remaining steps on $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ are operate in $\mathcal{O}\left(n^2\right)$. Therefore we shall have our overall brute-force complexity is in $\mathsf{poly}(n)$.

Remark that above complexity is derived given one has the knowledge of ϵ_{ss}. In such a case, ϵ_{ss} can be viewed as an trapdoor information for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to run in $\mathsf{poly}(n)$. By Eq. 10 and Eq. 11, we can formalise the following Proposition to characterize the correctness of $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ for higher error rate $\xi + \epsilon_{ss}$ in efficient manner.

Proposition 2. *For both $\mathcal{C}_{in} \in \{0,1\}^{k^*}$ and $\mathcal{C}_{out} \in \{0,1\}^n$ are BCH codes with syndrome decoding algorithm f, Given $\epsilon_{ss} \in [(2k^*)^{-1}, \xi]$, and $\epsilon_{rec} = 2\epsilon_{ss}$, where* *the following hold (for* $t_{(+)} = \lfloor (\xi + \epsilon_{ss})k^* \rfloor$*,* $\xi = t/n$*):*

1. $\lceil k^* \epsilon_{ss} \rceil \leq \|w \oplus w'\| \leq t_{(+)}$
2. $\lceil k^* h_2(2\epsilon_{ss}) \rceil \leq k - n^*$
3. $\lfloor \log(1/\exp(-2n\epsilon_{ss}^2)) \rfloor \geq 3$

Then, $\Pr[\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}(\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}(w, N, \epsilon_{ss}), w', N, \epsilon_{rec}) = w] \geq 0.875$ *can be achieved with* $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ *efficiently operating in time* $\mathsf{poly}(n)$*.*

11 Security

Based on the previous discussion, we learned that for some random variable $W \in \mathcal{M}_1$, rather than deal with the entire metric space $\mathcal{M}_1$ of size 2^{k^*}, sketching with $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$ is able to reduce our focus to a family of distribution $\mathcal{W}$ (by random errors parsing), where $W \in \mathcal{W}$ is concealed under such family of distributions over $\mathcal{M}_1$. Here, we want to show security for $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$. We know that $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$ hides the syndrome vector $v_{syn} \in W$, which concealing a random codeword c^* using $w \in W$ over W. Rather than define our security in term of the hardness in looking for a string $w \in W$, where $\|w \oplus w'\| \leq t_{(+)}$. It is more appropriately to define it as the hardness in looking for a variable $W \in \mathcal{W}$ satisfy $W \in B_{t_{(+)}}(w')$.

Recall that given the knowledge of ϵ_{ss}, we could set $\epsilon_{rec} = 2\epsilon_{ss}$, and tolerate at most $\|w \oplus w'\| \leq t_{(+)}$ number of error efficiently using $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ (see Proposition 2). Therefore, for any adversary knowing ϵ_{ss}, he/she should be able to look for $W \in B_{t_{(+)}}(w')$ efficiently by the correctness of the recovery algorithm itself. In such a case, we have to formalize our security for any random variable where $W \notin B_{t_{(+)}}(w')$ holds, which also means $\|w \oplus w'\| > t_{(+)}$ for all $w \in W$.

Based on the Eq. 10, we know that the the minimum information to show correctness using $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ (efficiently) could be expressed as the number of zeros padding to v_{syn}, which is $k - n^*$. Therefore, above statement argued that any adversary should be able to gain certain minimum information due to the zeros padding. This minimum information leakage is independent of how liberal or conservative is the selection of the input string $w \in W$ in concealing the codeword c^* over $v_{syn} \in W$. Rather, it is a limit (upper bound), which required to be set in before constructing any error tolerance system, to satisfy certain minimum requirement of correctness. Formally, follows Eq. 4 and Eq. 10, such minimum information leakage can be used to express the conditioned min-entropy of RV:

$$\tilde{\mathrm{H}}_\infty(\Psi|\mathcal{W}) \geq \lfloor \log(1/\exp(-2n\epsilon_{ss}^2)) \rfloor = k - n^* = m' \geq 3. \qquad (12)$$

However, merely consider the minimum information leakage is not sufficient to attain strong security claim. This is because the input $w \in W$ could be in some random distribution W, where $W \notin B_{t_{(+)}}(w')$ not necessary holds. This also can be argued with the adversary may have better computational power in modelling W using another family of distributions $\mathcal{I}$, leading to higher entropy loss from W compared to the minimum entropy loss derived in Eq. 12. In such a case, it is good for us to go into the analysis on how well is W being concealed over $\mathcal{W}$, given the additional information derived from $\mathcal{I}$. This question can also be interpreted as how well is the syndrome vector $v_{syn} \in W$ is hidden over $\mathcal{W}$ given some random distribution $I \in \mathcal{I}$ while sketching with $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$.

To analyse this, we have to go back to the sketching algorithm $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$ itself. Since we our construction accepts any random codeword c^*, included the trivial case when $c^* = 0^{n^*}$ is all zeros. Our analysis should cover the minimum entropy of W s.t. $\mathrm{H}_\infty(W) \geq m$ for some integer $m \geq 0$. Because the error parsing process is random, hiding W over $\mathcal{W}$ of size $|\mathcal{W}| = |\mathsf{supp}(\mathcal{E}_{ss})|$ would add entropy to it. The resultant entropy can be expressed as the min-entropy of $\mathcal{W}$:

$$\begin{aligned}&\mathrm{H}_\infty(\mathcal{W}) = \mathrm{H}_\infty(W, \mathcal{E}_{ss}) \geq \mathrm{H}_\infty(W) + |\log(\mathsf{supp}(\mathcal{E}_{ss}))| \\ &= m + \lceil k^* h_2(\epsilon_{ss})\rceil \end{aligned} \tag{13}$$

We now consider the entropy loss due to the zero bits padding to the syndrome vector v_{syn} to form $v^* = 0^{k-n^*} \| v_{syn}$. We claim it must leak information with any string in some distribution $I \in \mathcal{I}$ over $\{0,1\}^{k-n^*}$ by simply XOR operation with the first $k - n^*$ bits of v^*. Recall such zeros padding stage is necessary for $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ to be notified when the first decoding is success, which means $v_{syn} \in W$ is revealed from $\mathcal{W}$. Since $v^* \in W$ simply follows the distribution of the syndrome vector, therefore it can also be interpreted in such a way that certain understanding over W is necessary for our derived correctness (in Proposition 1) to hold. We call it as the *distribution precision*. Padding more zeros means higher distribution precision over W, thus $\mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}}$ can be well-notified that the first decoding stage is succeed. Our construction has perfectly captured such distribution precision notion, measured in term of the entropy loss proportional to the number of zeros padding to the syndrome vector v_{syn}. To be specific, it can be expressed as $m' = k - n^* \geq 3$ (see Eq. 10). In such a case, We therefore can describe the conditioned min-entropy of $\mathcal{W}$ given $\mathcal{I}$ using Eq. 13 as:

$$\tilde{\mathrm{H}}_\infty(\mathcal{W}|\mathcal{I}) = \tilde{\mathrm{H}}_\infty(W, \mathcal{E}_{ss}|\mathcal{I}) \geq m + \lceil k^* h_2(\epsilon_{ss})\rceil - (k - n^*) \tag{14}$$

Since we have define the min-entropy of W is at least m, straightforwardly, the worst case entropy loss due to the zeros bit padding is at most m. This means if $k - n^* \geq m$, we loss all the entropy can be supported by the sources. This also implies when worst comes to the worst, we could have such a powerful attacker with unlimited computational power that can reveal W precisely by using some random string in the worst-case distribution $I \in \mathcal{I}$ over $\{0,1\}^{k-n^*}$. In such a case, our security merely depends on the brute-force complexity to look for W concealed under $\mathcal{W}$. This security is measured in term of the conditioned min-entropy with the worst-case distribution $I \in \mathcal{I}$ described as below (say $H(X)$ is the Shannon entropy of distribution X):

$$\tilde{\mathrm{H}}_\infty(\mathcal{W}|\mathcal{I}) \geq \tilde{\mathrm{H}}_\infty(W, \mathcal{E}_{ss}|\mathcal{I} = I) \geq \lceil k^* h_2(\epsilon_{ss})\rceil \geq H(\mathcal{E}_{ss}) \tag{15}$$

Refer to Eq.13, under the worst-case when all input entropy is loss, we have the conditioned min-entropy of $\mathcal{W}$, which is at least the Shannon entropy over binary channel of rate ϵ_{ss}, i.e., $H(\mathcal{E}_{ss}) = k^* h_2(\epsilon_{ss})$. Such minimum entropy characterized the worst-case security of $\mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}$ even for computationally unbounded attacker in modelling the input distribution W with the worst-case distribution $I \in \mathcal{I}$ over $\{0,1\}^{k-n^*}$.

Based on the correctness discussed in Proposition 2, and the security reasoning follows Eq. 14, we formalise the below Proposition to characterize our construction as a standard secure sketch.

Proposition 3. *Given some integer $m_e > 0$. For any error distribution $\mathcal{E}_{ss}$ with entropy $H(\mathcal{E}_{ss}) \geq m_e$, then algorithm pair $\langle \mathsf{SS}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out}}, \mathsf{Rec}_{\Omega,\mathcal{C}_{in},\mathcal{C}_{out},\mathsf{f}} \rangle$ is an efficient $(\mathcal{M}_2, m, m_e, t_{(+)})$-secure sketch.*

12 Comparison

Table 1 below depicted the security bound (upper bound min-entropy requirement) for existing secure sketch construction.

Security Bound for Secure Sketch		
Computational	Best possible security	$\mathrm{H}_\infty(W) \geq \mathrm{H}^{\mathrm{fuzz}}_{t,\infty}(W) - \log(1-\beta)$
Computational	FRS sketch(universal hash functions) [11]	$\mathrm{H}_\infty(W) \geq \mathrm{H}^{\mathrm{fuzz}}_{t,\infty}(W) - \log(1/\beta) - \log\log(\mathsf{supp}(W)) - 1$
Computational	Layer hiding hash (strong universal hash function)[10]	$\mathrm{H}_\infty(W) \geq \mathrm{H}^{\mathrm{fuzz}}_{t,\infty}(W) - \log(1/\beta) - 1$
Info. theoretic	Fuzzy commitment with generic syndrome decoding [7]	$\mathrm{H}_\infty(W) \geq t\log(n)$ (when $t \ll n$)
Info. theoretic	Fuzzy vault [4]	$\mathrm{H}_\infty(W) \geq t\log(n)$
Info. theoretic	Improved Fuzzy vault [6]	$\mathrm{H}_\infty(W) \geq t\log(n) + 2$
Info. theoretic	Pinsketch [6]	$\mathrm{H}_\infty(W) \geq t\log(n+1)$
Info. theoretic	**Proposed**	$\mathrm{H}_\infty(W) \geq 0$ where security depends on $H(\mathcal{E}_{ss}) > 0$

Table 1: Summary of security bound of existing secure sketch in term of fuzzy-min entropy and min-entropy.

Obviously, compared to the computational secure sketch construction where its security property only holds for computationally bounded attacker, [9], [10], our construction offers *stronger security claim over computationally unbounded attacker.*

On the other hand compared to traditional single error correction code construction, i.e., [4], [6], [7], our construction is capable of *claiming security for all noisy sources with min-entropy* $m \geq 0$.

These results are clearly showed by Eq. 14 and 15, where $\tilde{\mathrm{H}}_\infty(W, \mathcal{E}_{ss} | \mathcal{I} = I) \geq H(\mathcal{E}_{ss})$ must hold regardless the value of m. Such property is crucial as merely claiming security according to the minimum residual entropy requirement, derived from the error correction construction itself cannot assure strong security. This is because any attacker could have higher computational power in modelling the input distribution W, results to significant low brute-force complexity in revealing w from W. This scenario is even worst for computationally bounded attacker assumption over computational secure sketch construction. The attacker can be running in exponential time still eventually reveal W. Therefore, hiding W over $\mathcal{W}$ is our main contribution to claim meaningful security for computationally unbounded attacker (information-theoretically secure).

Formally, our result from Eq. 15 suggested that Shannon entropy is necessary and sufficient condition to show meaningful security for a standard secure.

13 Concluding Remark

Existing secure secure constructions have shown limitation in providing security for noisy sources with low entropy, i.e., lower than half of its input size. To overcome such limitation, recent approaches [9], [10] suggested constructing secure sketch where its security property only holds for computationally bounded attacker. Such computational construction accompanies with stringent requirement, s.t. the user must have precise knowledge over the sources distribution. However, under practical scenario, a lot of noisy sources, for instance biometric (human face, iris, fingerprint, etc) are difficult to model, hence assuming precise knowledge over such noisy sources is unrealistic.

In this work, we proposed a concrete construction for secure sketch. We introduce the usage of RV for sketching to facilitate the understanding of the input distribution. Besides, we suggested parsing random error to the input, which we showed later it acts as trapdoor information to support efficient recovery, polynomial time in the sketch size. For security, under the worst-case where any attacker (computationally unbounded) could model and mount brute-force guesses over the input. The source entropy shall vanish, therefore, in principle, have no security to show. However, we showed that with the random error parsing to the input during sketching, we can still show meaningful security (information-theoretically) in term of the brute-force complexity to look for such nontrivial error vector. These results are significant, where it implies our construction could accept any sources, included the trivial sources with zero entropy, which have no prior construction have considered.

14 Appendix

Proof of Theorem 2

Proof. Let $\|\delta\| = \|\phi \oplus \phi'\|$, base on Theorem 1, we know that, for each time in comparing the hamming hash output (for $i = 1, \ldots, n$), the probability of disagree is describe as:

$$\Pr[\phi(i) \neq \phi'(i)] = \|w \oplus w'\| \, (k^*)^{-1} = 1 - P$$

Therefore, one has i.i.d variable (or Bernoulli variable) for each offset element, $\delta(i) = 1$ if $\phi(i) \neq \phi'(i)$ and $\delta(i) = 0$ if $\phi(i) = \phi'(i)$. Precisely, $\|\delta\| = \|\phi \oplus \phi'\| = \sum_{i=1}^{n} \delta(i)$, thus, $\|\delta\| \sim \mathrm{Bin}(n, 1-P)$ follows binomial distribution of expected distance $\mathbb{E}[\|\delta\|] = n(1-P)$ and s.d. $\sigma = \sqrt{nP(1-P)}$. Hence, $\mathbb{E}[\|\delta\|] = n(1-P) = n \, \|w \oplus w'\| \, (k^*)^{-1}$ and prove the theorem.

References

1. S. N. Porter, "A password extension for improved human factors," *Computers & Security*, vol. 1, no. 1, pp. 54–56, 1982.
2. N. Frykholm and A. Juels, "Error-tolerant password recovery," in *Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM, 2001, pp. 1–9.
3. C. Ellison, C. Hall, R. Milbert, and B. Schneier, "Protecting secret keys with personal entropy," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 311–318, 2000.
4. A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.
5. C. H. Bennett, G. Brassard, and J.-M. Robert, "Privacy amplification by public discussion," *SIAM journal on Computing*, vol. 17, no. 2, pp. 210–229, 1988.
6. Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 523–540.
7. A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*. ACM, 1999, pp. 28–36.
8. Y. Dodis and D. Wichs, "Non-malleable extractors and symmetric key cryptography from weak secrets," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 601–610.
9. B. Fuller, X. Meng, and L. Reyzin, "Computational fuzzy extractors," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2013, pp. 174–193.
10. J. Woodage, R. Chatterjee, Y. Dodis, A. Juels, and T. Ristenpart, "A new distribution-sensitive secure sketch and popularity-proportional hashing," in *Annual International Cryptology Conference*. Springer, 2017, pp. 682–710.
11. B. Fuller, L. Reyzin, and A. Smith, "When are fuzzy extractors possible?" in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*. Springer, 2016, pp. 277–306.
12. V. Guruswami, *List decoding of error-correcting codes: winning thesis of the 2002 ACM doctoral dissertation competition*. Springer Science & Business Media, 2004, vol. 3282.
13. W. W. Peterson and E. J. Weldon, *Error-correcting codes*. MIT press, 1972.
14. M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 2002, pp. 380–388.

15. A. Gionis, P. Indyk, R. Motwani *et al.*, “Similarity search in high dimensions via hashing,” in *Vldb*, vol. 99, no. 6, 1999, pp. 518–529.