# Correcting Errors Without Leaking Partial Information*

Yevgeniy Dodis† Adam Smith‡

November 5, 2004

**Abstract**

This paper explores what kinds of information two parties must communicate in order to correct errors which occur in a shared secret string $W$. Any bits they communicate must leak a significant amount of information about $W$ — that is, from the adversary's point of view, the entropy of $W$ will drop significantly. Nevertheless, we construct schemes with which Alice and Bob can prevent an adversary from learning any *useful* information about $W$. Specifically, if the entropy of $W$ is sufficiently high, then there is no function $f(W)$ which the adversary can learn from the error-correction information with significant probability. This leads to several new results:

- Code obfuscation: An obfuscator for a functionality $g$ generates a scrambled circuit $\tilde{C}$ which allows one to evaluate $g$ on any input, but leaks no additional information. Obfuscation of general functionalities is impossible (Barak et al. [2]).

  We show how to obfuscate *proximity queries*: we design a randomized function $Obf(w)$ such that for any $w$, given $Obf(w)$ one can verify if a candidate string $y$ is close to $w$, yet if an adversary's a priori probability of guessing $w$ was low, $Obf(W)$ reveals no function of $w$. This is the same as constructing noise-tolerant "perfectly one-way" hash functions in the sense of Canetti et al [10].

  The result does not contradict the impossibility results of Barak et al since the obfuscation guarantee requires $w$ to have high entropy.
- Private "Fuzzy Extractors": A fuzzy extractor (Dodis et al, [14]) takes a nonuniformly random, error-prone input $W$ (e.g. a fingerprint or iris scan) and produces two outputs, a public string $P$ and a key $R$, with two guarantees: $R$ is uniformly random given $P$, and yet given both $P$ and any string $Y$ close to $W$, one can recover $R$ exactly. Our constructions yield fuzzy extractors with an added privacy guarantee: $P$ reveals no function of the original input $W$. This means, for example, that a sensitive sub-string of $W$ will not accidentally be revealed.
- Noise Tolerance and Key Re-Use in the Bounded Storage Model: We give a scheme for key extraction in the bounded storage model with noise (Ding, [13]) which allows one to re-use the same initial key to derive many different session keys based on long public random strings. This answers the main open question from [13].

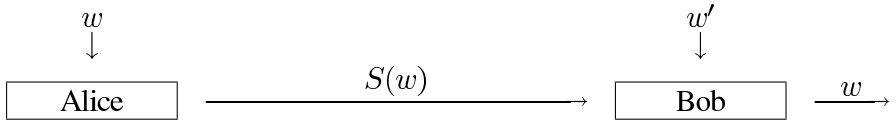
*Some of the results of this paper appears in the second author's Ph.D. thesis [41].

†New York University. Email: dodis@cs.nyu.edu

‡Weizmann Institute of Science. Email: adam.smith@weizmann.ac.il.

# 1 Introduction

This paper investigates what kind of information must be leaked to an eavesdropper when two cooperating parties communicate in order to correct errors in a shared secret string.




Suppose that Alice and Bob share an $n$-bit secret string. Alice's copy $w$ of the shared string is slightly different from Bob's copy $w'$. Alice would like to send a short message $S(w)$ to Bob which allows him to correct the errors in $w'$ (and thus recover $w$) whenever $w$ and $w'$ differ in at most $\tau$ bits. The randomized map $S()$ that Alice applies to $w$ to get the message she sends to Bob is called a *non-interactive information reconciliation scheme*, or simply a *sketch*, correcting $\tau$ errors. A typical example of a sketch is

$$S(w) = \mathsf{syn}_C(w),$$

where $\mathsf{syn}_C$ is the syndrome of a linear error-correcting code $C$ with block length $n$ (see below for definitions) [3]. If $C$ has dimension $k$, then $\mathsf{syn}_C(w)$ is only $n-k$ bits long. If the minimum distance of $C$ is at least $2\tau+1$, then $\mathsf{syn}_C(w)$ allows Bob to correct any $\tau$ errors in $w'$. Moreover, the process is efficient if the code can correct $\tau$ errors in polynomial time.

Enter Eve, who is tapping the line and trying to learn as much as possible. From her point of view, Alice and Bob hold a pair of random variables $W, W'$. Suppose that Alice and Bob do not share any secrets except this pair.[1] What kind of guarantees can Alice and Bob obtain on what Eve learns from seeing $S(W)$? This abstract game — and partial answers to this question — have been applied in several contexts, most notably in generating keys from long, noisy public strings and biometric authentication (see references below).

Standard notions of security do not fit here. The statement "$S(W)$ leaks no information about $W$" is normally formalized by requiring that $W$ and $S(W)$ be almost statistically independent or, equivalently, that the Shannon mutual information $\mathbf{I}(W; S(W))$ be very small. Such a strong requirement is impossible to achieve in our setting: a coding argument shows that the mutual information must be large (much larger than $\tau$) in general [7]. Even the analogue requirement for computationally bounded adversaries, *semantic security* [17], is impossible here: if Eve knows that $W$ is one of two strings $w_1, w_2$ which differ in only a few bits, then she can use whatever algorithm Bob would have run to compute $w_i$ from $S(w_i)$ and $w_1$.

The difficulty, then, is that the standard definitions of security require secrecy even when Eve knows a lot about $W$. We show that when this requirement is relaxed (that is, when Eve is sufficiently uncertain about $W$), a strong secrecy guarantee can be be provided.

A more suitable definition for our setting is *entropic security* [10, 37]. If $W, Y$ are (correlated) random variables, $Y$ *hides all functions of* $W$ if for every function $f$, it is nearly as hard to predict $f(W)$ given $Y$ as it is without $Y$, regardless of the adversary's computing power. A randomized map $S()$ is called *entropically secure* if $S()$ hides all functions of $W$ whenever the min-entropy[2] of $W$ is above a certain threshold. This definition of security has already produced surprising results in two contexts. Canetti, Micciancio and Reingold [9, 10] constructed hash functions whose outputs leak no partial information about the input. Russell and Wang [37] gave entropically-secure symmetric encryption schemes with keys much shorter than the length of the input, thus circumventing Shannon's famous lower bound on key length.

This paper introduces a third, very different application of entropic security: we construct secure sketches that are (a) efficiently decodable (that is, Bob's recovery algorithm is polynomial-time) and (b) entropically secure. In particular, for any entropy bound $t$ which is linear in $n$, we obtain sketches which can efficiently decode a constant fraction of errors and have leakage exponentially small in $n$. The core of our construction is a family of strong *randomness extractors* with an additional property: given the output of the extractor

[1] This rules out trivial solutions, such as Alice sending the encryption of $W$ with Bob's public key.

[2] Min-entropy measures the difficulty of guessing $W$ a priori: $\mathbf{H}_\infty(W) = -\log(\max_w \Pr[W = w])$.

and a string which is close to the source, one can efficiently recover the source exactly. We construct these extractors based on small random families of algebraic-geometric codes.

We apply our constructions to private storage of keys derived from biometric measurements, obfuscation of proximity queries, and key re-use in the bounded storage model. Perhaps the most surprising of these applications is to obfuscation: previous positive results for obfuscation were for "point functions," which are easy to obfuscate in the random oracle model [10, 27]. In contrast, it is not known how random oracles can help obfuscation of proximity queries.

**The Relation to Entropy Loss** The task of correcting errors in a joint string is usually called *information reconciliation* [3, 7, 8, 25, 13], *fuzzy cryptography* ([21], see [41] for a survey), or *document exchange* (in communication complexity, e.g. [11]). In contrast to this paper, previous work focused only on maximizing the length of a cryptographic key which can be derived from $W$ once the errors in $W'$ have been corrected. Because of that, they are only interested in bounding the drop in the entropy of $W$ from Eve's point of view when she sees the communication between Alice and Bob.

The security guarantee we provide is strictly stronger than in previous work. Entropic security implies a lower bound on the min-entropy of $W$ given the sketch $S(W)$. Min-entropy is a lower bound on all the measures of entropy used in the literature, and so entropic security implies an upper bound on entropy loss. The converse implication is not true: simply bounding the entropy loss does not prevent Eve from learning some particular function of $W$ with probability 1 (for example, the syndrome construction above always reveals a particular, fixed set of linear combinations of the bits of $W$). This can be a problem for several reasons. First, $W$ itself may be sensitive (say, if it is a biometric used for authentication [21, 22, 14]), in which case $S(W)$ might reveal sensitive information, such as a person's age. Second, when we use the error-correction protocol as a piece of a larger framework, entropy loss may not be a sufficient guarantee of secrecy; we will see an example of this in key agreement protocols which are secure against memory-bounded adversaries [13].

For completeness, we state the min-entropy loss of our constructions explicitly, since it is typically much lower than the bound implied by entropic security.

**Notation and Definitions** We denote the output of a randomized algorithm on input $x$ and random coins $r$ by $Y(x;r)$. We use the shorthand $Y(x)$ for (random) output when the string $r$ is chosen uniformly at random.

The *statistical difference* between two probability distributions $A$ and $B$ on the same space is $\mathbf{SD}(A,B) \stackrel{\text{def}}{=} \frac{1}{2}\sum_v \left|\Pr[A=v] - \Pr[B=v]\right|$ (that is, half the $L_1$ distance between the probability mass functions).

The main measure of entropy we use is *min-entropy*, which measures the difficulty of guessing a random variable $A$ a-priori: the best predictor succeeds with probability $p^* = \max_a \Pr[A=a]$, and the min-entropy is $\mathbf{H}_\infty(A) = -\log(p^*)$ (all logarithms are base 2 by default). $A$ is called a $t$-source if $\mathbf{H}_\infty(A) \geq t$. The conditional min-entropy of $A$ given $B$ is $\tilde{\mathbf{H}}_\infty(A \mid B) \stackrel{\text{def}}{=} -\log(\mathbb{E}_{b \leftarrow B}\left[2^{-\mathbf{H}_\infty(A|B=b)}\right])$. (This definition is not standard but very convenient.) We will use two properties: (1) if $B \in \{0,1\}^\ell$ then $\tilde{\mathbf{H}}_\infty(A \mid B) \leq \mathbf{H}_\infty(A) - \ell$ and (2) for any $A, B$, the event $\mathbf{H}_\infty(A \mid B = b) \geq \tilde{\mathbf{H}}_\infty(A \mid B) - \log\left(\frac{1}{\epsilon}\right)$ occurs with probability at least $1-\epsilon$ over $b \leftarrow B$.

We now turn to defining secure sketches and entropic security. Following [14], we incorporate entropy loss into the definition of a secure sketch; we state the definition of entropic security separately.

**Definition 1 ([14]).** *A $(t, t', \tau)$-*secure sketch *is a pair of (possibly) randomized maps $S : \{0,1\}^n \to \{0,1\}^*$ and $\mathsf{Rec} : \{0,1\}^* \to \{0,1\}^n$ such that:*

- *For all pairs of strings $w, w'$ of distance at most $\tau$, we have $\mathsf{Rec}(w', S(w)) = w$ with prob. 1.*[3]

[3] We consider this "worst-case" error model for simplicity. The discussion extends naturally to random errors, although some care must be taken: the results change depending on whether the adversary chooses $w'$ before or after seeing $S(w)$. The issue is discussed partially in [41].

– *For all $t$-sources $W$, we have $\tilde{\mathbf{H}}_\infty(W \mid S(W)) \geq t'$.*

*The* entropy loss *of a sketch is the difference $t - t'$.*
*The sketch is* efficient *if $S$ and* Rec *run in time poly($n$).*

**Definition 2 ([10, 37, 15]).** *The probabilistic map $Y()$ hides all functions of $W$ with leakage $\epsilon$ if for every adversary $\mathcal{A}$, there exists an adversary $\mathcal{A}'$ such that for all functions $f : \{0,1\}^* \to \{0,1\}^*$,*

$$\left| \Pr[\mathcal{A}(Y(W)) = f(W)] - \Pr[\mathcal{A}'() = f(W)] \right| \leq \epsilon.$$

*The map $Y()$ is called $(t, \epsilon)$-entropically secure if $Y()$ hides all functions of $W$, for all $t$-sources $W$.*

## 1.1 Our Contributions

As stated earlier, our main result is the construction of entropically secure sketches.

**Theorem 1.** *There exist efficient $(t, t', \tau)$-secure sketches for inputs in $\{0,1\}^n$ which are also $(t, \epsilon)$-entropically secure, such that (for infinitely many $n$)*

– *the tolerated error $\tau$ and the residual entropy $t'$ are linear in $n$, and*
– *the information leakage $\epsilon$ is exponentially small in $n$*

*whenever the original min-entropy $t$ is linear in $n$. (That is, whenever $t = \Omega(n)$ then we can find schemes where $\tau$, $t'$ and $\log\left(\frac{1}{\epsilon}\right)$ are $\Omega(n)$).*

Before proceeding, a word about parameters: the original entropy $t$ of the input $W$ is given by the context in which $W$ arises. The error tolerance $\tau$ will also typically be specified externally—it is the amount of noise to which $W$ will likely be subject. Thus, the goal is to get both the (entropic) security $\log\left(\frac{1}{\epsilon}\right)$ and the residual min-entropy $t'$ as high as possible. The quantity $\log\left(\frac{1}{\epsilon}\right)$ measures the difficulty of learning some function of $W$, while $t'$ measures the difficulty of guessing $W$ exactly. In particular, $t'$ is bounded below by $\log\left(\frac{1}{\epsilon}\right)$ (roughly), since by the definition of entropic security the adversary's probability of predicting the identity function $f(W) = W$ is at most $\epsilon + 2^{-t} \approx \epsilon$. Thus, it is sufficient to look for sketches will tolerate $\tau$ errors and are $(t, \epsilon)$-entropically secure for $\tau, \log\left(\frac{1}{\epsilon}\right) = \Omega(n)$. Theorem 1 states that such secure sketches do indeed exist.

**The Relation to Randomness Extraction** The starting point of the constructions is a result from earlier work stating that *randomness extractors* [33] are entropically secure, that is the output hides all functions of the source. We say a (randomized) map $Y()$ is $(t, \epsilon)$-indistinguishable if for all pairs of $t$-sources $W_1$, $W_2$, the distributions $Y(W_1)$ and $Y(W_2)$ are $\epsilon$-close. ($Y()$ is a randomness extractor in the special case where the output distribution is always close to uniform.) We will use the following result several times:

**Fact 2 ([15], Thm 2.1).** *If $Y()$ is $(t, \epsilon)$-entropically secure, then it is $(t - 1, 4\epsilon)$-indistinguishable. Conversely, if $Y()$ is $(t, \epsilon)$-indistinguishable, then it is $(t + 2, 8\epsilon)$-entropically secure.*

The second implication is the more interesting of the two. In particular, our main result is really a construction of randomness extractors whose output can be used to correct errors in the input. They are *strong* randomness extractors in the sense of Nisan and Zuckerman [33]: all the random coins used by by the extractor (the "seed") appear explicitly in the output. We will use the strong extractor property in the bounded storage model application. The construction is based on a random family of binary images of an algebraic-geometric code. It is explained in Section 2. We rephrase Theorem 1 in terms of extractors here:

**Theorem 3.** *For any constant entropy rate $t/n$, there is an explicitly constructible of ensemble of strong $(t, \epsilon)$-extractors $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell \times \{0,1\}^d$ with seed length $d = n$ such that (1) $\mathsf{Ext}$ extracts a linear amount of entropy from the input with exponentially small error and (2) $\mathsf{Ext}()$ corrects a linear number $\tau$ of errors in the source. That is, there is a polynomial time algorithm $\mathsf{Rec}$ such that for any strings $w, w'$ at distance at most $\tau$, $\mathsf{Rec}(w', \mathsf{Ext}(w; R)) = w$ with probability 1.*

**Applications** We present three new applications of this result:

- *Key Re-Use in the Bounded Storage Model* This is perhaps the least expected application of our technique, resolving the main open question left by Ding [13]. Namely, Ding considered the question of error correcting in the bounded storage model [29] which received a lot of attention recently (see [13, 44, 26] and references therein). The attractive feature of this model comes from the fact that is provides so called "everlasting" security of the derived keys (assuming the adversary has bounded storage capabilities at the time of transmission of a huge random string). Another nice feature of the recent constructions is the fact that the same long-term key can be used many times for subsequent session key derivations. This feature is called key reuse. On the other hand, one of the aspects limiting the usability of the current solutions comes from the fact that Alice and Bob must be error-free when receiving the satellite data. Ding [13] elegantly extended the bounded storage model to achieve error correction, but at the expense of considerably weakening the key reuse property: the parties must synchronously and periodically update their long-term secret keys. We resolve this open problem by showing that nearly optimal error-correction *can be achieved without sacrificing the key reuse property*.

- *Obfuscation and Perfectly One-Way Functions*. While general program obfuscation is impossible [2], obfuscation might be possible for specific functionalities. Indeed, Lynn, Prabhakaran and Sahai [27] formally showed that one can obfuscate equality queries in the random oracle model relative to some secret $w$, while the results of Canetti et al. [9, 10] on perfect one-way hash functions could be interpreted as obfuscating equality queries in the standard model, provided $w$ has high min-entropy. While equality queries are very natural for password authentication applications, for more general biometric applications it is more natural to consider more general proximity queries, where inputs $w'$ sufficiently close to $w$ should also be accepted. This was explicitly mentioned as an open problem in [27], who noticed that random oracles do not appear to be of much help for correcting unknown errors. We settle this problem in the affirmative in the standard model, but assuming that $w$ has high entropy (which is the model of [10]). Alternatively, this gives the error-tolerant construction of perfectly one-way hash functions. Along the way, we also improve the noise-free construction of [10], roughly halving the min-entropy requirement of their construction.

- *Privacy for Biometric Applications* Recently, Dodis, Reyzin and Smith [14] introduced a general framework for dealing with noisy and non-uniform biometric data, by defining two primitives termed secure sketches and fuzzy extractors aimed to provide noise-tolerant password recovery and randomness extraction, respectively. In both cases the goal was achieved by publishing some public function $P = P(W)$ which eliminated errors in subsequent imperfect readings of password $W$. However, in all the constructions in [14] the public information $P$ actually leaked some (potentially sensitive) information about the biometric input $W$. Our results here are two-fold. On the one hand, we show that $P$ *must* indeed leak some non-trivial amount of Shannon information about $W$. This conclusion is somewhat non-trivial for the case of fuzzy extractors, and critically uses the isoperimetric inequality. On the other hand and somewhat surprisingly, we construct secure sketches and fuzzy extractors which leak no deterministic function (such as a sensitive substring) of the biometric input $W$. This once again shows that Shannon security is stronger than semantic security for high-entropy distributions — a conclusion recently derived in a very different context of symmetric encryption [37, 15].

**This Abstract** The bulk of this abstract describes the construction of secure sketches which leak no partial information. Section 3 describes the application to the bounded storage model. The applications to fuzzy extractors and perfectly one-way hash functions are described in Appendix E and Appendix F, respectively.

## 2 Sketches That Hide All Partial Information

This section describes the main technical construction of the paper (Theorem 1). Our discussion refers often to the "code-offset" construction [3, 21]: if we view an error-correcting code as a function $C : \{0,1\}^k \to \{0,1\}^n$ with minimum distance $d$, the randomized map

$$S(w; R) = w \oplus C(R) \tag{1}$$

has entropy loss $t - t' = n - k$ (for any value of $t$) [14]. It can correct $\tau = \lfloor (d-1)/2 \rfloor$ errors, and is efficient if and only if $C$ has efficient encoding and error-correction algorithms. In the case of linear codes, this construction reduces to the syndrome construction in the introduction, since $w \oplus C(R)$ is a random element of the coset $\{x \in \{0,1\}^n \ : \ \mathsf{syn}_C(x) = \mathsf{syn}_C(w)\}$.

### 2.1 General Approach: Codes with Small Bias

We now turn to our constructions. Our starting point is the following fact about "small-bias" subsets of $\{0,1\}^n$ (defined below). If $A$ is randomly drawn from a subset of sufficiently small "bias," and $B$ is any random variable with sufficient min-entropy, then $A \oplus B$ is close to uniform on $\{0,1\}^n$. This fact was used to construct a nearly optimal entropically secure encryption scheme [37]. The intuition behind our approach, then, is simple:

> If $C$ itself is a small-bias set, then the code-offset construction $S(W) = W \oplus C(R)$ always yields distributions close to uniform, and hence $S()$ is entropically secure.

The problem with this intuition is that explicit constructions of codes with small bias are not known (in particular, such codes cannot be linear, and most explicitly constructible codes are linear).

We circumvent this difficulty and construct explicit, *efficient* entropically secure sketches. We show that the code-offset construction can be made indistinguishable (even with linear codes) when the choice of error-correcting code is randomized as opposed to always using the same fixed code.

Suppose that we have a family of $k$-dimensional linear error-correcting codes $\{C_i\}_{i \in I}$ indexed by some set $I$. Consider sketches of the form

$$\begin{aligned} & S(w; i) = (i, \mathsf{syn}_{C_i}(w)) \ , \text{ for } i \leftarrow I \\ \text{or, equivalently,} \quad & S(w; i, x) = (i, w \oplus C_i(x)) \ , \text{ for } i \leftarrow I, x \leftarrow \{0,1\}^k \end{aligned} \tag{2}$$

Below, we establish a necessary condition on the code family for the construction to leak no partial information about the input $w$.

1. We define a notion of "bias" for *families* of codes, and show that a small-bias family of codes also leads to an entropically-secure sketch. This allows us to work with linear codes.
2. To illustrate the framework, we show that random linear codes are optimal in terms of both error-correction and entropic security (this corresponds to reproving the "left-over hash" lemma [20]).
3. We construct explicit, efficiently decodable, small-bias families of codes by considering a subset of binary images of a fixed code over a large (but constant-size) alphabet $GF(2^e)$.

A number of interesting observations come out of our analysis. First of all, we derive a general sufficient condition for a set of *linear* functions to form a good randomness extractor; this may be of independent interest. We also obtain new bounds on the average weight enumerators of "generalized" algebraic-geometric codes.

**Bias and Secrecy** The *bias* of a random variable $A$ over $\{0,1\}^n$ is a (weak) measure of "pseudo-randomness": it measures how close $A$ is to fooling all statistical tests that look only at the parity of a subset of bits. Formally, the bias of $A$ with respect to a non-zero vector $\alpha$ is the distance between the dot product of $\alpha$ and $A$ from a fair coin flip, that is

$$\mathsf{bias}_\alpha(A) \stackrel{\text{def}}{=} \mathbb{E}\left[(-1)^{\alpha \odot A}\right] = 2\Pr[\alpha \odot A = 1] - 1$$

The random variable $A$ has bias $\delta$ if $|\mathsf{bias}_\alpha(A)| < \delta$ for all non-zero vectors $\alpha \in \{0,1\}^n$. The bias of a set $C$ is the bias of the uniform distribution over that set. It is known that the map $Y(W;A) = W \oplus A$ is a $(t,\epsilon)$-extractor whenever the bias of $C$ is sufficiently small ($\delta \leq \epsilon 2^{-(n-t-1)/2}$), e.g. [5].

We generalize this to a family of sets by requiring that on average, the square of the bias with respect to every $\alpha$ is low (at most $\delta^2$):

**Definition 3.** *A* family *of random variables (or sets) $\{A_i\}_{i\in I}$ is $\delta$-biased if, for all $\alpha \neq 0^n$,*

$$\sqrt{\mathbb{E}_{i\leftarrow I}\left[\mathsf{bias}_\alpha(A_i)^2\right]} \leq \delta.$$

Note that this is *not* equivalent, in general, to requiring that the expected bias be less than $\delta$. There are two important special cases:

1. If $C$ is a $\delta$-biased set, then $\{C\}$ is a $\delta$-biased set family with a single member.

   Constructing codes with good minimum distance and negligible bias seems difficult. Such codes do exist: a completely random set $C$ of $2^k$ elements will have both (1) minimum distance $d$, where $k/n \approx (1 - h_2(d/n))/2$ [28] and (2) bias approximately $2^{-(k-\log n)/2}$ [31]. However, these codes are neither explicitly constructed nor efficiently decodable. This raises a natural question:

   > Does there exist an explicitly-constructible ensemble of good codes with small bias and poly-time encoding and decoding algorithms (ideally, codes with linear rate and minimum distance, and negligible bias)?

   To the best of our knowledge, the problem remains open.

2. A family of linear codes $\{C_i\}_{i\in I}$ is $\delta$-*biased* if there is no word which is often in the dual $C_i^\perp$ of a random code $C_i$ from the family. Specifically, the bias of a linear space with respect to a vector $\alpha$ is always either 0 or 1:

   $$\mathsf{bias}_\alpha(C_i) = \begin{cases} 0 & \text{if} \quad \alpha \notin C_i^\perp \\ 1 & \text{if} \quad \alpha \in C_i^\perp \end{cases}$$

   Hence a family of codes is $\delta$-biased if and only if $\Pr_{i\leftarrow I}[\alpha \in C_i^\perp] \leq \delta^2$, for every $\alpha \neq 0^n$.

   Note that for a family of linear codes to satisfy Definition 3 the expected bias must be at most $\delta^2$, while for a single set the bias need only be $\delta$.

The general lemma below will allow us to prove that the randomized code-offset construction is indistinguishable (and hence entropically-secure).

**Lemma 4 (Small Bias Families Yield Extractors).** *Let $\{A_i\}_{i\in I}$ be a $\delta$-biased family of random variables over $\{0,1\}^n$, with $\delta \leq \epsilon \cdot 2^{-\frac{n-t-1}{2}}$. For any $t$-source $B$ (independent of $A_i$) the pair $(I, A_I \oplus B)$ is $\epsilon$-close to uniform.*

The proof of Lemma 4 is in Appendix D.1. It is a generalization of the proof that random walks on the hypercube converge quickly when the edge set is given by a small bias set. The basic idea is to bound the Fourier coefficients (over $\mathbb{Z}_2^n$) of the output distribution in order to show that it is close to uniform in the $\ell_2$ norm.

In order to apply Lemma 4 we will need a family of error-correcting codes with small bias. Our construction is described in the next section, and summarized here:

**Lemma 5 (Good Code Families Construction).** *For any constant $0 < \lambda < 1$, there exists an explicitly constructible ensemble of code families which* efficiently *correct $\tau = \Omega(n)$ errors and have square bias $\delta^2 < 2^{-\lambda n}$.*

**Proof of Theorem 1** We can combine this lemma and Lemma 4 to prove our main result, i.e. that that there are efficient, entropically secure sketches (Theorem 1). If $t/n$ is constant, we can set $\lambda = 1 - \frac{t}{2n}$. Picking a sequence of code families as in Lemma 5, we obtain a secure sketch scheme which corrects $\tau = \Omega(n)$ errors efficiently and is $(t, \epsilon)$-entropically secure, where $\epsilon = \delta \cdot 2^{(n-t)/2+O(1)}$. Since $\delta^2 \leq 2^{-\lambda n}$, the leakage $\epsilon$ is exponentially small. □

## 2.2 Small-Bias Families of Linear Codes: Constructions and Lower Bounds

**Inefficient Construction: Random Linear Codes** An easy observation is that the family of *all* linear codes of a particular dimension $k$ is has squared bias $\delta^2 < 2^{-k}$, although the codes are not known to be efficiently decodable. This bias is optimal. (The extractor one gets by plugging random linear codes into Lemma 4 is in fact the usual pairwise independent hashing construction [19]. See Appendix C.2 for a discussion). Random linear codes also exhibit the best known tradeoff between rate and distance for binary codes, as they lie near the Gilbert-Varshamov bound with high probability [28]. This gives us a point of reference with which to measure other constructions.

**Efficient Constructions via Random Binary Images** The basic idea behind our construction is to start from a single, fixed code $C'$ over a large (but constant) alphabet, and consider a family of binary codes obtained by converting field elements to bit strings in different ways.

Let $\mathcal{F} = GF(q)$, where $q = 2^e$. Starting from a $[n', k', d]_q$ code $C'$ over $\mathcal{F}$, we can construct a binary code by taking the *binary image* of $C'$, that is by writing down the codewords of $C'$ using some particular $e$-bit binary representation for elements of $\mathcal{F}$. More formally, fix a basis of the field $\mathcal{F}$ over $\mathbb{Z}_2$, and let $\mathsf{bin}(a) \in \{0,1\}^e$ be the binary representation of a field element $a$ in the basis (the exact choice of basis does not matter). For a vector $\alpha = (a_1, ..., a_{n'}) \in \mathcal{F}^{n'}$, let $\mathsf{bin}(\alpha)$ be the concatenation $(\mathsf{bin}(a_1), ..., \mathsf{bin}(a_{n'}))$. Finally, let $\mathsf{bin}(C')$ denote the set of binary images of the codewords, $\mathsf{bin}(C') \stackrel{\text{def}}{=} \{\mathsf{bin}(c) : c \in C'\}$.

We can randomize the code $C'$ by

1. Permuting the $n'$ coordinates of $\mathcal{F}^{n'}$,
2. Multiplying each coordinate of the code by some random non-zero scalar in $\mathcal{F}$, and
3. Taking the binary image of the result.[4]

These operations affect neither the dimension nor the decodability of $C'$: they are invertible and preserve Hamming distances in $\mathcal{F}^{n'}$. Describing the particular operations that were applied to the code requires $O(n' \log n' + n' \log(q-1))$ bits (we must describe a permutation of $n'$ positions and $n'$ non-zero scalars).

When the initial code $C'$ is a Reed-Solomon code or an algebraic-geometric (AG) code, the family of codes obtained as above is called a "generalized" Reed-Solomon (resp. AG) code. The bias of such a code family can be computed from the (average) weight distribution[5] of the codes in the family. These weight distributions have been studied before [35, 36, 45, 43], but the existing bounds do not apply to the range of parameters relevant here. We prove a new bound based on the minimum distance of the dual code of $C'$.

[4] For the bounds stated in this abstract to hold, it is not necessary to permute the coordinates of the code—multiplying the components by scalars provides enough randomness. Thus, only $O(n)$ random bits are needed to select a code from the family. As noted at the end of the proof of Lemma 6, permuting the coordinates does allow the potential of a much better bound on the bias of the code.

[5] The weight distribution of a code $C$ is a vector of $n$ integers $A_0, A_1, ..., A_n$, where $A_w$ is the number of codewords in $C$ with weight exactly $w$.

**Lemma 6 (Random binary images).** *Let $C'$ be a linear $[n', k', d]_q$ code over $\mathcal{F} = GF(q)$, with $q = 2^e$. Let $\{C'_i\}$ be the set of $[n', k', d]_q$ codes over $\mathcal{F}$ obtained by permuting the coordinates and multiplying each coordinate by a non-zero scalar in $\mathcal{F}$. Let $C_i = \mathsf{bin}(C'_i)$. Then*

1. *The $C_i$ are $[n, k, d]_2$ codes with $n = n'e$ and $k = k'e$. (Note that the rates $k/n$ and $k'/n'$ are equal).*
2. *If $C'$ can correct $\tau$ errors in $\mathcal{F}^{n'}$ efficiently, then each $C_i$ can efficiently correct $\tau$ errors in $\{0,1\}^n$.*
3. *If $(C')^\perp$ has minimum distance $d_\perp$, then the average square bias of $\{C_i\}$ is*

$$\delta^2 = \max_{\alpha \in \{0,1\}^n, \alpha \neq 0^n} \{ \Pr_i[\alpha \in C_i^\perp] \} \leq 1/(q-1)^{d_\perp - 1}.$$

*Note that in the last statement, the dual code $(C')^\perp$ is taken with respect to the dot product in $\mathcal{F}^{n'}$, while the dual code $C_i^\perp$ is taken with respect to the dot product in $\{0,1\}^n$.*

Finally, applying this lemma to algebraic-geometric codes yields the following lemma, which implies Lemma 5. The proofs of both Lemma 6 and Lemma 7 may be found in Appendix D.

**Lemma 7 (Good Code Families Construction).** *For any constant $0 < R < 1$, and any $q = 2^{2k}$ where $k$ is an integer, $k \geq 2$, there exists an explicitly constructible ensemble of code families which* efficiently *correct $\tau$ errors and have square bias $\delta^2$ where:*

$$\tau \geq \frac{n}{\log q}\left(1 - R - \frac{1}{\sqrt{q}-1}\right) \qquad \textit{and} \qquad \log\left(\tfrac{1}{\delta}\right) \geq \frac{nR}{2}\left(1 - \frac{1}{R(\sqrt{q}-1)}\right)\left(1 - \frac{\log q}{q-1}\right)$$

# 3 Application: Noise Tolerance and "Everlasting Security"

In this section we resolve the main open question of [13]: we show that there is a noise-tolerant "locally computable extractor" which allows its key to be reused many times.

**Bounded Storage Model (BSM).** We first briefly recall the basics of the bounded storage model [29]. Alice and Bob share a short, "long-term" secret key $K$. A sequence of huge random strings $X_1, X_2, \ldots$[6] is broadcast to both of them. Alice and Bob then apply a deterministic function $f_K$ to derive relatively short one-time pads $R_i = f_K(X_i)$. Traditionally, there are two main considerations in the bounded storage model: efficiency and *everlasting security*. Efficiency means that that $f_K$ depends on a few bits of the source $X_i$, and these bits can be easily determined from the long-term key $K$ alone. Concretely, for typical setting of parameters we usually want this number of bits to be linear in the length of the extracted one-time pad $R_i$, and perhaps polylogarithmic in the length $N$ of the source $X_i$. Security means that as long as the adversary does not know the secret key $K$ and cannot store each source $X_i$ in "its entirety", the one-time pads $R_i$ are statistically close to uniform, even if the adversary later gets the long-term secret key $K$. A bit more formally (see [44] for a complete definition), if the adversary is allowed to adaptively choose a storage function $g_i : \{0,1\}^N \to \{0,1\}^{\gamma N}$, where $\gamma < 1$ is a fixed constant, if $I = \langle g_1(X_1), \ldots, g_t(X_t), K \rangle$ denotes all the data available to the adversary, the joint distribution of $\langle I, R_1 \ldots R_t \rangle$ is $t2^{-\Omega(N)}$-close to the distribution $\langle I, U_1 \ldots U_t \rangle$, where $U_i$ are independent, truly uniform keys of the same length as $R_i$.

The BSM has received a lot of attention recently (see [13, 44, 26] and references therein). The current technique for achieving everlasting security [26, 44] in this model is the "sample-then-extract" approach. The high-level idea sufficient for our purposes is to have $K$ consist of two keys $K_s$ and $K_e$, where $K_s$ is used to obliviously sample a small potion $X_s^i$ of the bits of $X_i$, and then $K_e$ is used as a key for a strong randomness extractor [33]. Using optimal parameter settings, one can achieve a total key of size $O(\log N + \log\left(\frac{1}{\epsilon}\right))$.

[6] More generally, it is sufficient that each $X_i$ has high min-entropy conditioned on the other $X_j$ for $j \neq i$.

**Error-Correction in BSM.** Recently Ding [13] considered the problem of the error-correction in the bounded storage model, where it is assumed that Bob will not necessarily receive the same string $X_i$ as Alice, but instead will receive some $\tilde{X}_i$ which is guaranteed (or expected) to be close to $X_i$ in the Hamming distance. Ding proposed the following simple idea to overcome such errors, which we first describe for a single sample (i.e., $t = 1$). After receiving the source $X$ and sampling the substring $X^s$ (using $K_s$), Alice will simply send to Bob — over a public channel — the string $P = \mathsf{syn}_C(X^s)$, where $C$ is a good error-correcting code.[7] Bob will sample the string $\tilde{X}^s$ which is going to be close to $X^s$ (due to the properties of the sampler), which means that he can recover $X^s$ from $P$ and $\tilde{X}^s$, after which he can use $K_e$ to extract the final randomness $R$.

It is easy to see that this idea works for $t = 1$ and might initially appear to work for arbitrary number of repetitions $t$. However, Ding pointed out the following subtle problem. The value $\mathsf{syn}_C(X^s)$ leaks some information about $X^s$, which in turn could conceivably leak information about the long-term key $K_s$, since $X_s$ depends on $K_s$. But now the security in the BSM model crucically assumes that the key $K_s$ is *independent* from the source. Now, leaking $P_1$ conceivably leaks information about $K_s$, which in principle means that the attacker can choose the storage function $g_2$ as if it depends on $K_s$. But this means that the conditional distribution of $X_2$ given $g_2(X_2)$ can *no longer be argued independent from the sampling key* $K_s$. And this means that the analysis does not go through.

Ding addressed this problem by making Alice and Bob synchronized and stateful. Specifically, after each communication they not only extract a fresh one-time pad $R_i$, but also refresh the *long-term key* $K$ (specifically, $K_s$ must be replaced). While Ding showed that this solution achieves very good parameters, it obviously creates a lot of inconvenience for the sender and the receiver.

**Our Contribution.** Using our technique, we resolve the main open problem of [13]. Specifically, our construction gives a family of codes $\{C_i\}$ with the property that a syndrome of a randomly selected code is a strong randomness extractor, provided that the input distribution has enough min-entropy. Specifically, we have that the following distributions are statistically close ($U$ is the uniform distribution):

$$\langle i, \mathsf{syn}_{C_i}(W)\rangle \approx \langle i, U\rangle \tag{3}$$

where in case $|W| = n$ and $\mathbf{H}_\infty(W) = t = \Omega(n)$, we can have codes correcting $\Omega(n)$ errors and the residual min-entropy of $W$ given $\langle i, \mathsf{syn}_{C_i}(W)\rangle$ is $t' = \Omega(n)$. Moreover, the length of the code index $i$ is $\Theta(n)$.

Now, instead of sending Bob a fixed syndrome of the sampled source $X^s$, Alice will additionally share with Bob the random index $i$ of the code $C_i$, and will send Bob the value $\mathsf{syn}_{C_i}(X^s)$. It is easy to see that our modification resolves the reusability problem of [13]; We give brief reasoning here (below "high" denotes $\Omega(n)$, where $n$ is the length of $X^s$). Indeed, by the property of averaging samplers, proved in [44, 33], the joint distribution of $\langle g(X), K_s, X^s\rangle$ is statistically close to $\langle g(X), K_s, Y\rangle$, where $Y\,|_{K_s=a, g(X)=b}$ has high min-entropy, for every setting of $a, b$. By Equation 3 and the fact that the syndrome length is shorter than the residual min-entropy of $X^s$ given $g(X)$ and $K_s$, this means that the distribution $\langle g(X), K_s, i, \mathsf{syn}_{C_i}(X^s), X^s\rangle$ is statistically close to $\langle g(X), K_s, i, \mathsf{syn}_{C_i}(X^s), Z\rangle$, where $Z$ has high min-entropy given any setting for the other variables. Finally, the properties of the strong extractor $\mathbf{Ext}$ mean that $\langle g(X), K_s, i, \mathsf{syn}_{C_i}(X^s), K_e, R = \mathbf{Ext}_{K_e}(X^s)\rangle$ is statistically close to $\langle g(X), K_s, i, \mathsf{syn}_{C_i}(X^s), K_e, U$, where $U$ is a truly uniform string of length $\Omega(n)$. This shows "one-time" security. Now, given that the above "one-time" indistinguishability holds even conditioned on the sampling key $K_s$, the multiple time security of this locally computable extractor holds using the standard hybrid argument, much like for the error-free case of [26, 44]. In contrast, the argument of Ding could not condition of the sampling key $K_s$, since the syndrome could in fact reveal some information about $K_s$, which forced him to update the "compromised" value of $K_s$ with part of the freshly extracted key $R$, leading to the stateful construction.

[7] More precisely, if the adversary is allowed to corrupt $\delta$-fraction of the bit in $X$, $C$ should be able to correct slightly more than $\delta$-fraction of errors.

This construction achieves similar parameters to those of Ding — arbitrary high storage threshold $\gamma < 1$, linear fraction of corrected errors, up to linear number of extracted bits $\ell$ (in previous notation, $\ell = \Omega(n)$, which in principle could be as high as $\Omega(N)$), small number of sampled bits $n = O(\ell)$. There are two significant differences. First, the long-term key now has to include the index $i$ of the code, which in our construction is $\Theta(n) = \Theta(\ell)$. This could be considerably larger than the $O(\log N + \log\left(\frac{1}{\epsilon}\right))$-key size achieved in [13], but it is still sublinear in N, and moderate enough to be stored. For example, in applications when the extracted string $R$ is a 128-bit key to a computationally secure cipher, this value of $\ell = 128$ is of the same order as $\log N + \log\left(\frac{1}{\epsilon}\right)$. A second drawback is that the sketches constructed in this paper do not tolerate as many errors for a given level of entropy loss as do sketches without the requirement of entropic security. Finally, we remark that the linear key length in the number of extracted bits of our construction is asymptotically similar to the well known leftover hash lemma, which gives a more than sufficient extractor for most cryptographic applications, while our construction additionally supports error correction.

# References


[1] E. Agrell, A. Vardy, and K. Zeger. Upper bounds for constant-weight codes. *IEEE Transactions on Information Theory*, **46**(7), pp. 2373–2395, 2000.

[2] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, K. Yang. On the (Im)possibility of Obfuscating Programs. In *Advances in Cryptology — CRYPTO 2001*, pp. 1–18.

[3] C. Bennett, G. Brassard, and J. Robert. Privacy Amplification by Public Discussion. *SIAM J. on Computing*, **17**(2), pp. 210–229, 1988.

[4] C. Bennett, G. Brassard, C. Crépeau, and U. Maurer. Generalized Privacy Amplification. *IEEE Transactions on Information Theory*, **41**(6), pp. 1915-1923, 1995.

[5] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, Avi Wigderson: Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. STOC 2003: 612-621

[6] B. Bollobás. *Combinatorics*. Cambridge University Press, 1986.

[7] Gilles Brassard, Louis Salvail. Secret-Key Reconciliation by Public Discussion. In *Advances in Cryptology — EUROCRYPT 1993*, p. 410–423.

[8] Christian Cachin, Ueli M. Maurer. Linking Information Reconciliation and Privacy Amplification. In *J. Cryptology*, **10**(2), 97-110, 1997.

[9] R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology — CRYPTO 1997*.

[10] R. Canetti, D. Micciancio, O. Reingold. Perfectly One-Way Probabilistic Hash Functions. In *Proc. 30th ACM Symp. on Theory of Computing*, 1998, pp. 131–140.

[11] Graham Cormode, Mike Paterson, S?leyman Cenk Sahinalp, Uzi Vishkin: Communication complexity of document exchange. *Proc. ACM Symp. on Discrete Algorithms, 2000*, p. 197–206.

[12] T. Cover, J. Thomas. *Elements of Information Theory*. Wiley series in telecommunication, 1991, 542 pp.

[13] Y.Z. Ding. Error Correction in the Bounded Storage Model. In *Theory of Cryptography 2005*.

[14] Y. Dodis, L. Reyzin and A. Smith. Fuzzy Extractors and Cryptography, or How to Use Your Fingerprints. In *Advances in Cryptology — EUROCRYPT 2004*. Originally appeared as IACR Eprint Report 2003/235, November 2003.

[15] Y. Dodis and A. Smith. Entropic Security and the Encryption of High-Entropy Messages. In *Theory of Cryptography 2005*. Originally appeared as IACR Eprint Report 2004/219, September 2004.

[16] D. Forney. *Concatenated Codes*. MIT Press, 1966.

[17] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, **28**(2), pp. 270–299, April 1984.

[18] J. Håstad, R. Impagliazzo, L. Levin, M. Luby. A Pseudorandom generator from any one-way function. In *Proc. 21st ACM Symp. on Theory of Computing*, 1989.

[19] R. Impagliazzo, L. Levin, M. Luby. Pseudo-random Generation from one-way functions. In *Proc. ACM Symp. on Theory of Computing*, 1989.

[20] R. Impagliazzo and D. Zuckerman. How to Recycle Random Bits. In *Proc. 30th IEEE Symp. on Foundations of Computer Science*, 1989.

[21] A. Juels, M. Wattenberg. A Fuzzy Commitment Scheme. In *Proc. ACM Conf. Computer and Communications Security, 1999*, pp. 28–36.

[22] A. Juels and M. Sudan. A Fuzzy Vault Scheme. In *IEEE International Symposium on Information Theory*, 2002.

[23] J.-P. M. G. Linnartz, P. Tuyls. New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates. In *AVBPA 2003*, p. 393–402.

[24] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, 1992, 183 pp.

[25] Shengli Liu and Henk C. A. Van Tilborg and Marten Van Dijk. Practical Protocol for Advantage Distillation and Information Reconciliation. In *Des. Codes Cryptography*, **30**(1), 39–62, 2003.

[26] Chi-Jen Lu. Encryption against Storage-Bounded Adversaries from On-Line Strong Extractors. *J. Cryptology*, 17(1): 27-42 (2004).

[27] Ben Lynn, Manoj Prabhakaran, Amit Sahai. Positive Results and Techniques for Obfuscation. *Advances in Cryptology — EUROCRYPT 2004*, p. 20–39.

[28] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, New York, Oxford, 1978.

[29] U. Maurer. Secret Key Agreement by Public Discussion. *IEEE Trans. on Info. Theory*, 39(3):733–742, 1993.

[30] Silvio Micali, Chris Peikert, Madhu Sudan, and David Wilson. Cryptographic Sieving: Optimal Error Correction Against Computationally Bounded Noise. In *Theory of Cryptography 2005*.

[31] J. Naor, M. Naor. Small-Bias Probability Spaces: Efficient Constructions and Applications. In *SIAM J. Comput.* 22(4): 838-856 (1993).

[32] New York Times. "Arrest in Bombing Inquiry Was Rushed, Officials Say", May 8, 2004.

[33] N. Nisan, D. Zuckerman. Randomness is Linear in Space. In *JCSS*, **52**(1), pp. 43–52, 1996.

[34] J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. In *Proc. 38th IEEE Symp. on Foundations of Computer Science*, 1997, pp. 585–594.

[35] Charles T. Retter. The Average Binary Weight Enumerator for a Class of Generalized Rees-Solomon Codes. In *IEEE Trans. Info. Theory*, 37(2), 1991.

[36] V. Roychowdhuri and F. Vatan. Bounds on Weight Distributions of Weakly Self-Dual Codes. *IEEE Trans. Info. Theory*, 47(1), 2001.

[37] A. Russell and Wang. How to Fool an Unbounded Adversary with a Short Key. In *Advances in Cryptology — EUROCRYPT 2002*.

[38] R. Shaltiel. Recent developments in Explicit Constructions of Extractors. *Bulletin of the EATCS*, **77**, pp. 67–95, 2002.

[39] C. Shannon. Communication Theory of Secrecy systems. In *Bell Systems Technical J.*, 28:656–715, 1949. Note: The material in this paper appeared originally in a confidential report 'A Mathematical Theory of Cryptography', dated Sept. 1, 1945, which has now been declassified.

[40] C. Shannon. A Mathematical Theory of Communication. *Bell System Technical J.*, 27 (July and October 1948), pp. 379-423 and 623-656. Reprinted in D. Slepian, editor, *Key Papers in the Development of Information Theory*, IEEE Press, NY, 1974.

[41] A. Smith. Maintaining Secrecy When Information Leakage is Unavoidable. Ph.D. Thesis, Massachusetts Institute of Technology, 2004.

[42] H. Stichtenoth. Algebraic Function Fields and Codes. *Springer-Verlag*, Berlin, 1993.

[43] T. Umeda, K. Sakakibara, M. Kasahara. Notes on the Average Binary Weight Enumerator of Generalized Algebraic-Geometric Codes. In *IEICE Trans. Fundamentals*, E79-A(9), September 1996.

[44] Salil P. Vadhan. Constructing Locally Computable Extractors and Cryptosystems in the Bounded-Storage Model. *J. Cryptology* 17(1): 43-77 (2004).

[45] S. Vledutz and A. N. Skorobogatov. Weight Distributions of Subfield Subcodes of Algebraic-Geometric Codes. *Problems in Information Transmission*, 27(1), 1991.

# A Background on Coding and Information Theory

This appendix provides the notation and basic concepts from information theory that we use in the text. We assume that most readers are familiar with these concepts, but we have tried to state all the facts we will need explicitly.

We use capital letters (e.g. $A$) to refer to both random variables and the distributions from which they are drawn, and lower case letters to denote particular values which the variables may take on. The expression $a \leftarrow A$ denotes that $a$ is sampled according to the distribution (r.v.) $W$. If $S$ is a set, $a \leftarrow S$ denotes drawing $x$ from the uniform distribution on $S$. We sometimes also use $U_n$ to denote the uniform distribution on $\{0,1\}^n$. If $\mathcal{A}(x;r)$ is a randomized algorithm with random input $R$, we will sometimes use $\mathcal{A}(x)$ to denote the distribution on outputs when $r$ is drawn uniformly at random. $\mathbb{E}[A]$ denotes the expectation of a real-valued random variable and $\mathsf{Var}[A]$, its variance.

The *statistical difference* between two probability distributions $A$ and $B$ on the same space is $\mathbf{SD}(A,B) \stackrel{\text{def}}{=} \frac{1}{2}\sum_v \left|\Pr[A=v] - \Pr[B=v]\right|$ (that is, half the $L_1$ distance between the probability mass functions). The *collision probability* of $X$ is $\mathsf{Col}(X) \stackrel{\text{def}}{=} \sum_x \Pr[X=x]^2$. If $X \in S$, and $\mathsf{Col}(X) \leq (1+2\epsilon^2)/|S|$, then $\mathbf{SD}(X,U) \leq \epsilon$, where $U$ is uniform over $S$.

The main measure of entropy we use is *min-entropy*, defined as the negative log of the probability of predicting a random variable $X$ *a priori*, that is $\mathbf{H}_\infty(A) \stackrel{\text{def}}{=} -\log \max_a \Pr[A = a]$ (all logarithms are base 2 by default). $A$ is called a $t$-source if $\mathbf{H}_\infty(A) \geq t$. The conditional min-entropy of $A$ given $B$ is $\tilde{\mathbf{H}}_\infty(A \mid B) \stackrel{\text{def}}{=} -\log(\mathbb{E}_{b \leftarrow B}\left[2^{-\mathbf{H}_\infty(A|B=b)}\right])$. (NB: This definition is not standard but very convenient.) We will use two properties: (1) if $B \in \{0,1\}^\ell$ then $\tilde{\mathbf{H}}_\infty(A \mid B) \leq \mathbf{H}_\infty(A) - \ell$ and (2) for any $A, B$, the event $\mathbf{H}_\infty(A \mid B = b) \geq \tilde{\mathbf{H}}_\infty(A \mid B) - \log\left(\frac{1}{\epsilon}\right)$ occurs with probability at least $1 - \epsilon$ over $b \leftarrow B$.

We will also use the *Shannon entropy* of a distribution, defined as $\mathbf{H}_{sh}(A) \stackrel{\text{def}}{=} \sum_a \Pr[A = a] \log \frac{1}{\Pr[A=a]}$. The conditional entropy of $A$ given $B$ is $\mathbf{H}_{sh}(A \mid B) \stackrel{\text{def}}{=} \mathbb{E}_b\left[\mathbf{H}_{sh}(A \mid B = b)\right]$. The mutual information between $A$ and $B$ is the entropy loss in $A$ when given $B$: $\mathbf{I}(A;B) \stackrel{\text{def}}{=} \mathbf{H}_{sh}(A) - \mathbf{H}_{sh}(A|B)$.

**Randomness Extractors** An extractor is a function which takes as input some imperfect source of randomness (the various bits of which may be biased or correlated) and a short truly random "seed", and produces as output something close to uniformly random string.

**Definition 4.** $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^{k+\ell}$ *is a strong* $(t', \epsilon)$*-extractor if for all min-entropy* $t'$ *distributions* $X$*, the output* $\mathsf{Ext}(X; U_k)$ *is* $\epsilon$*-far from* $U_{k+\ell}$*. A* strong *extractor has output of the form* $\mathsf{Ext}(x; r) = r, \mathsf{Ext}'(x; r)$ *where* $\mathsf{Ext}'$ *outputs* $\ell$ *bits.*

The difference $t' - (\ell + k)$ is the *entropy loss* of the extractor. The number of truly random bits $k$ is the seed length of the extractor. Much research has been devoted to improving these parameters in extractors. The easiest construction of strong extractors is given by the "left-over hash" lemma, also called the "privacy amplification" lemma [18, 20, 4]. A family of functions $\{h_i\}_{i \in \mathcal{I}}$ from $n$ bits to $\ell$ bits is *(pairwise) XOR-independent* if the event $h_i(x) + h_i(y) = z$ occurs with probability $2^{-\ell}$ when $i$ is chosen uniformly from $\mathcal{I}$, for any choice of $x, y \in \{0,1\}^n$ and $z \in \{0,1\}^\ell$.

**Lemma 8 (Left-over hash/ priv. amp.).** *If* $\{h_i\}_{i \in \mathcal{I}}$ *is a family of XOR-independent functions from* $n$ *bits to* $\ell$ *bits, then* $\mathsf{Ext}(x; i) = h_i(x)$ *is a strong* $(t, \epsilon)$*-extractor whenever* $t \geq \ell + 2\log\left(\frac{1}{\epsilon}\right) + 1$.

**Distance and Error-Correcting Codes** Many of the ideas discussed in this paper extend to correcting errors in almost any metric space. For simplicity, we will work mostly over the Hamming cube $\{0,1\}^n$, where $\mathsf{dist}(x, y)$ is the number of bits in which strings $x$ and $y$ differ. We also use the Hamming metric over larger alphabets such as $[q]^n$ where $[q] = 1, \ldots, q$, $q$ is a power of 2, and Hamming distance is the number of symbols in $[q]$ in which two strings differ. We will associate $[q]$ with the field $GF(q)$. The weight $wt(w)$ of a word $w \in GF(q)^n$ is the number of positions in which it is non-zero, that is $dist xy = wt(x - y)$.

A $[n, k, d]_q$-code is a linear subspace of $C \subseteq \mathcal{F} = GF(q)^n$ of dimension $k$, such that every pair of strings $x, y \in C$ is at distance at least $d$. Such a code can correct any $\tau = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors in a codeword unambiguously. Let $H \in \mathcal{F}^{(n-k)\times n}$ be a matrix whose kernel (null space) is exactly $C$. The syndrome with respect to $C$ is $\mathsf{syn}_C(x) = Hx$. The *syndrome* of a codeword is $0^{n-k}$, and for any word $x$, the syndrome of $x$ depends only on the subset of bits in which $x$ differs from the nearest codeword $c$: if $x = c \oplus e$, then $\mathsf{syn}_C(x) = \mathsf{syn}_C(e)$. A linear code can correct any $\tau$ errors efficiently if and only if there is an algorithm which efficiently computes $e$ from $\mathsf{syn}_C(e)$ whenever $wt(e) \leq \tau$.

# B Alternative Error Models

The error model in the definition of secure sketches above is very restrictive: we require that the sketch correct *any* $\tau$ errors with probability 1. We made this choice for simplicity, since such a strong requirement will be sufficient in any application. However, for some applications one can get substantially better performance by considering less stringent error models.

The main relaxation is to require that error-correction occur only with high probability. There are several variants on the problem at that point. Perhaps the most subtle issue is that in some situations (such as biometric authentication), the errors introduced in $w'$ by the adversary may somehow depend on the sketch $S(w)$. In contrast, the set-up of the introduction implicitly suggests that the errors in $w'$ are decided on ahead of time, non-adaptively. This ambiguity is not a problem in our model since we require correction with probability 1. However, it can be a problem in general. Another issue is whether or not the errors are introduced by a computationally bounded process (see, e.g., [30] for techniques to exploit such bounds).

The ideas and techniques of this paper can be extended to yield better performance in these relaxed error models; we do not discuss those extensions in the paper. A very partial discussion may be found in Smith's thesis [41].

# C Context: Lower Bounds on Secure Sketches and Small-Bias Families

This section describes lower bounds (some old, some new) on the parameters achievable by secure sketches: this provides some useful context for understanding the parameters achieved by our constructions. The bounds relating to distance are specific to the stringent error model we consider here (see Appendix B). Bounds for high-probability error correction will look more like the Shannon bounds on channel capacity; we do not discuss the various distinctions here since they do not really shed light on our focus, which is entropic security.

1. *Min-Entropy Loss [14]:* Let $d^*(n,k)$ be the minimum distance of the best binary $n$-bit block code with $2^k$ codewords. If $t = n$ (i.e. $W$ is uniform), then any secure sketch correcting $\tau = \lfloor (d^* - 1)/2 \rfloor$ errors has residual entropy at most $t' \geq k$ (i.e. entropy loss at least $n - k$).
2. *Shannon Entropy Loss [7]:* When $\tau = \Omega(n)$, the drop in Shannon entropy, $\mathbf{I}(W; S(W))$, is at least $nh_2^{-1}(\tau/n)$, where $h_2()$ is the binary entropy function.

The first bound above gives a (nearly) complete picture of the performance of sketches for the Hamming metric with respect to entropy loss (the bound is matched by the code-offset construction above); we will use the bound as a benchmark for comparisons.

## C.1 Bounds on Entropic Security

Next, we can relate entropic security to the residual entropy $t'$ of a sketch (this is the minimum of $\tilde{\mathbf{H}}_\infty(W \mid S(W))$ over all $t$-sources $W$). First, it is easy to prove that the leakage $\epsilon$ of a $(t, \epsilon)$ secure sketch will always be at least $2^{-t'} - 2^{-t}$ (Proposition 10).

We can in fact get a better bound for a large class of canonical schemes. Recall that $S()$ is $(t, \epsilon)$ entropically secure only if $S(W)$ is always $4\epsilon$-close to some particular "target" distribution when $W$ is a $(t-1)$-source (Fact 2). In the special case where the "target" distribution is uniform, $S()$ is a $(t-1, 4\epsilon)$-*extractor* in the sense [33]. We can then apply:

**Fact 9 (Radakrishanan and Ta-Shma, [34]).** *Suppose $S(W)$ is a $(t, \epsilon)$-extractor. If $S()$ uses $r$ random coins as extra input, and always outputs $\ell$ bits, then $2\log\left(\frac{1}{\epsilon}\right) < t - \ell + r$.*

In all the schemes we discuss, the sketch will in fact be an extractor and the residual entropy will satisfy $t' = t - \ell + r$. For these schemes, the bound above implies that $\epsilon > 2^{-t'/2}$ (i.e. $t' > 2\log\left(\frac{1}{\epsilon}\right)$). We conjecture that this bound actually holds for all entropically secure sketches. For now, we only have the weaker bound:

**Proposition 10.** *If a $(t, t', \tau)$-secure sketch is $(t, \epsilon)$-entropically secure, then $\epsilon > 2^{-t'} - 2^{-t} \approx 2^{-t'}$.*

*Proof.* Consider the function $f(w) = w$, and any source $W$ with min-entropy exactly $t$. The best adversary's expected probability of guessing $f$ after seeing $S(W)$ is exactly $2^{-\tilde{\mathbf{H}}_\infty(W|S(W))}$. Without $S(W)$, it is $2^{-\mathbf{H}_\infty(W)} = 2^{-t}$. By entropic security, $\epsilon > 2^{-\tilde{\mathbf{H}}_\infty(W|S(W))} - 2^{-\mathbf{H}_\infty(W)} > 2^{-t'} - 2^{-t}$. □

### C.2 Bounds on Small-Bias Code Families

In this section, we explore some consequences of Lemma 4. If applied to a family of linear codes with sufficiently small bias, Lemma 4 shows that the sketch $S(w;i) = i, \mathsf{syn}_{C_i}(w)$ is a strong extractor.

How close to optimality is this extractor? Entropy loss is defined slightly differently for extractors and secure sketches. For the code offset construction, the entropy-loss-as-extractor is $t + k - n$ where $k$ is the dimension of the code. By a lower bound of Radakrishnan and Ta-Shma [34] (described in Appendix C), the entropy loss of an extractor is at least $2\log\left(\frac{1}{\epsilon}\right) - O(1)$. In our context, this yields the bound $\delta^2 \geq 2^{n-t-k-O(1)}.2^{-n+t} = 2^{-k-O(1)}$. We can conclude both that the average square bias is bounded by $2^{-k-O(1)}$ and that codes which match this bound yield nearly optimal extractors.

Code families with optimal bias do exist (the set of all linear codes is an example, and in that case Lemma 4 reduces to the left-over hash lemma [19]). However, these codes are not efficiently decodable. We do not know constructions of efficiently decodable families of codes with minimal bias, although the constructions in terms of algebraic geometric codes can get the ratio $2\log\left(\frac{1}{\delta}\right)/k$ arbitrarily close to 1.

## D Proofs from the Main Construction

### D.1 Proof of Secrecy for Small-Bias Families

*Proof of Lemma 4.* The proof uses elementary Fourier analysis over the hypercube $\mathbb{Z}_2^n$. The intuition comes from the proof that Cayley graphs based on $\epsilon$-biased spaces are good expanders: adding a $\delta$-biased family of random variables to $B$ will cause all the Fourier coefficients of $B$ to be reduced by a factor of $\delta$, which implies that *the collision probability* of $B$ (see below) gets multiplied by $\delta$ also.

Let $D_i$ be the distribution $A_i \oplus B$. Recall that for any probability distribution $D$ on a set of size $K$, if $\mathsf{Col}(D) \leq (1+\epsilon^2)/K$, then $D$ is within statistical distance $\epsilon$ of the uniform distribution (see, e.g., [20]). Hence to prove the theorem it is sufficient to show that the collision probability of the *pair* $D = (i, D_i) = (i, A_i + B)$ is bounded above by $\frac{(1+2\epsilon^2)}{|\mathcal{I}|2^n}$.

*Claim:* $\mathsf{Col}(D) = \frac{1}{|\mathcal{I}|}\mathbb{E}_{i \leftarrow \mathcal{I}}\left[\mathsf{Col}(D_i)\right]$.

*Proof.* We can write out the probability of a collision (here prime $'$ denotes an independent copy):

$$\Pr[(I, D_I) = (I', D'_{I'}] = \sum_i \Pr[I = I' = i] \Pr[D_i = D'_i]$$

Factoring out $\frac{1}{|\mathcal{I}|}$, we get $\mathsf{Col}(D) = \frac{1}{|\mathcal{I}|}\sum_i \frac{1}{|\mathcal{I}|}\mathsf{Col}(D_i)$, as desired. □

To bound $\mathsf{Col}(D)$, we need only bound the average collision probability of $D_i$. To do so, we use a standard fact from Fourier analysis over the hypercube:

**Fact 11.** *For any distribution $D_i$ on $\{0,1\}^n$, the collision probability $\mathsf{Col}(D_i)$ is given by the sum of the squared biases of $D_i$ with respect to all possible vectors:*

$$\mathsf{Col}(D_i) = \frac{1}{2^n}\sum_{\alpha \in \{0,1\}^n} \mathsf{bias}_\alpha(D_i)^2 = \frac{1}{2^n} + \frac{1}{2^n}\sum_{\alpha \neq 0} \mathsf{bias}_\alpha(D_i)^2.$$

Since $D_i = A_i \oplus B$ (that is, the distribution of $D_i$ is the convolution of $A_i$ and $B$), we can compute the bias of $D_i$ as a product of the biases of $A_i$ and $B$:

$$\begin{aligned} \mathsf{bias}_\alpha(D_i) &= \mathbb{E}\left[(-1)^{\alpha \odot (A_i \oplus B)}\right] \\ &= \mathbb{E}\left[(-1)^{\alpha \odot (A_i)}\right]\mathbb{E}\left[(-1)^{\alpha \odot B}\right] = \mathsf{bias}_\alpha(A_i)\mathsf{bias}_\alpha(B). \end{aligned}$$

We now want to bound the bias of $D_i$. We don't know how this bias will behave for particular values of $i$, but we can use the fact that $\{A_i\}$ is $\delta$-biased family to bound the *average* squared bias:

$$\mathbb{E}_i\left[\mathsf{bias}_\alpha(D_i)^2\right] \leq \mathbb{E}_i\left[\mathsf{bias}_\alpha(A_i)^2\right] \mathsf{bias}_\alpha(B)^2 \leq \delta^2 \mathsf{bias}_\alpha(B)^2.$$

Finally, we can combine these bounds:

$$\mathsf{Col}(D) = \frac{1}{|\mathcal{I}|}\mathbb{E}_i\underbrace{\left[\frac{1}{2^n} + \frac{1}{2^n}\sum_{\alpha\neq 0}\mathsf{bias}_\alpha(D_i)^2\right]}_{\mathsf{Col}(D_i)} = \frac{1}{|\mathcal{I}|2^n}(1+\delta^2\sum_{\alpha\neq 0}\mathsf{bias}_\alpha(B)^2)$$

By the fact above, the sum of squared biases of $B$ is at most $2^n\mathsf{Col}(B)$. Since the min-entropy of $B$ is at least $t$, its collision probability is at most $2^{-t}$, and we get the bound $\mathsf{Col}(D) \leq \frac{1}{|\mathcal{I}|2^n}(1+\delta^2 2^{-t+n})$. By hypothesis, $\delta \leq \epsilon 2^{-(n+t)/2}$, which implies the desired bound $\mathsf{Col}(D) \leq \frac{1}{|\mathcal{I}|2^n}(1+\epsilon^2)$. □

## D.2 Analysis of Random Binary Images

*Proof of Lemma 6.* (1),(2): The first two statements are straightforward since the multiplication by non-zero scalars in one component and permutations of positions are easily invertible isometries of $\mathcal{F}^{n'}$.

(3): There are really two separate stages to proving this statement. In the first stage, we have to relate the dual of a $q$-ary code to the dual of a binary code. Second, we will bound the bias of the $q$-ary codes $\{C'_i\}$.

To clarify the notion of "dual" code, let $\odot_2$ denote binary inner product on $\{0,1\}^n$, and let $\odot_\mathcal{F}$ denote the standard inner product in $\mathcal{F}^{n'}$. The duals of the codes $C_i \subseteq \{0,1\}^n$ are defined with respect to the binary inner product, while the duals of the $C'_i \in \mathcal{F}^{n'}$ are defined w.r.t. the dot product over $\mathcal{F}^{n'}$:

$$\begin{aligned} C_i^\perp &= \{y \in \{0,1\}^n \,:\, y \odot_2 x = 0 \;(\forall x \in C_i)\} \\ (C'_i)^\perp &= \{y' \in \mathcal{F}^{n'} \,:\, y' \odot_\mathcal{F} x' = 0_\mathcal{F} \;(\forall x' y \in C'_i)\} \end{aligned}$$

For the rest of the proof, fix some $\alpha \in \{0,1\}^n$, and let $\alpha'$ be the corresponding vector in $\mathcal{F}^{n'}$, that is $\alpha = \mathsf{bin}(\alpha')$. The statement to be proved follows from two claims:

*Claim 1:* For all $\alpha \in \{0,1\}^n$, there exists $\alpha' \in \mathcal{F}^{n'}$ s.t. $\Pr_i[\alpha \in C_i^\perp] = \Pr_i[\alpha' \in (C'_i)^\perp]$.

*Claim 2:* For all $\alpha' \in \mathcal{F}^{n'}$, we have: $\Pr_i[\alpha' \in (C'_i)^\perp] \leq 1/(q-1)^{d_\perp - 1}$.

*Proof of Claim 1.* The first claim is mostly a careful unwinding of the definitions. We will use the trace function $\mathrm{Tr} : \mathcal{F} \to \{0,1\}$. The exact definition of the trace is not important here (see, e.g. [28]). All we require is that the trace is linear, i.e. $\mathrm{Tr}(a+b) = \mathrm{Tr}(a)+\mathrm{Tr}(b)$, and not identitically zero. $\mathrm{Tr}(ab)$ is a bilinear map from $\mathcal{F}\times\mathcal{F}$ to $\{0,1\}$, and so there exists an invertible linear transformation $B : \{0,1\}^e \to \{0,1\}^e$ such that for all scalars $a, b \in \mathcal{F}$, we have $B(\mathsf{bin}(a)) \odot_2 \mathsf{bin}(b) = \mathrm{Tr}(ab)$.

Fix $\alpha \in \{0,1\}^n$. We can choose the unique vector $\alpha'$ in $\mathcal{F}^{n'}$ such that $\alpha$ is the concatenation of the $e$-bit vectors $B(\mathsf{bin}(\alpha'_i))$. Then for any vector $x' \in \mathcal{F}^{n'}$, we have:

$$\alpha \odot_2 \mathsf{bin}(x') = \mathrm{Tr}(\alpha' \odot_\mathcal{F} x')$$

*Sub-Claim:* $\alpha$ is in $C_i^\perp$ if and only if $\alpha'$ is in $(C'_i)^\perp$.

One direction of the sub-claim is easy: suppose $\alpha' \in (C'_i)^\perp$. Then for any vector $x \in C_i$, we have $\alpha \odot_2 x = \mathrm{Tr}(\alpha' \odot_\mathcal{F} \mathsf{bin}^{-1}(x))$. Now the image of $x$ in $\mathcal{F}^{n'}$ is in $C'_i$, and so $\mathrm{Tr}(\alpha' \odot_\mathcal{F} \mathsf{bin}^{-1}(x)) = \mathrm{Tr}(0_\mathcal{F}) = 0$. In the other direction (of the sub-claim), suppose that $\alpha \in C_i^\perp$. Suppose, to get a contradiction, that there is some $x' \in C'_i$ such that $\alpha' \odot_\mathcal{F} x' \neq 0_\mathcal{F}$. Then there exists some non-zero scalar $b \in \mathcal{F}$, such that $0 \neq \mathrm{Tr}(b(\alpha' \odot_\mathcal{F} x')) = \mathrm{Tr}(\alpha' \odot_\mathcal{F} (bx')) = \alpha \odot_2 \mathsf{bin}(bx')$. But the vector $bx'$ is in $C'_i$ since $C'_i$ is a linear

code, and so the inner product of its binary image with $\alpha$ should be 0. Thus, we get a contradiction and conclude that $\alpha' \in (C'_i)^\perp$, completing the proof of the sub-claim.

Based on the sub-claim, we can conclude that $\Pr_i[\alpha \in C_i^\perp] = \Pr_i[\alpha' \in (C'_i)^\perp]$. □

*Proof of Claim 2.* The main observation behind this proof is that the randomization operations we use behave nicely in the dual space. Permuting the coordinates of the code $C'$ induces the same permutation on the coordinates of $C'$. Similarly, if we multiply the $n'$ coordinates by non-zero scalars $b_1, ..., b_{n'} \in \mathcal{F}$, then we multiply the dual code by the inverses $b_1^{-1}, ..., b_n^{-1}$. Thus we get the same family of $q$-ary codes $C'_i$ by applying the randomization procedure to the dual instead of the primal code.

Now fix some vector $\alpha' \in \mathcal{F}^{n'}$. By symmetry, we can imagine that the randomizing operation is applied to the target word $\alpha'$ instead of to the code itself. *This maps $\alpha'$ to a random word in $\mathcal{F}^{n'}$ of the same weight as $\alpha'$.* The probability that this hits a codeword is exactly the fraction of words of a given weight $w$ which are in the code. We call the set of words in $\mathcal{F}^{n'}$ with weight exactly $w$ the $w$-slice. To complete the proof, we need only prove the following:

*Sub-Claim (Singleton bound for constant weight codes):* For any code over $\mathcal{F} = GF(q)$ of minimum distance $d_\perp$, the fraction of codewords in any slice of $\mathcal{F}^{n'}$ is bounded above by $(q-1)/(q-1)^{d_\perp}$ (except for the trivial slice $\{0^{n'}\}$).

To prove the sub-claim, fix some weight $0 < w \leq n'$. We can partition the slice of weight $w$ according to which $w$ positions in a word are non-zero. Each of these partitions can further be subidivided into pieces where all but $d_\perp$ of the non-zero values are fixed, i.e. sets of the form

$$(\underbrace{0, ..., 0}_{n-w \text{ times}}, \underbrace{b_1, ..., b_{w-d_\perp}}_{\text{non-zero scalars}}, \underbrace{*, ..., *}_{d_\perp \text{ times}}),$$

up to permutation of coordinates, where $*$ may take any non-zero value.

Now within any such piece, there can be at most $q-1$ codewords (since the codewords must differ in $d_\perp$ positions). There are $(q-1)^{d_\perp}$ words in the piece, and so overall the fraction of codewords in any constant-weight slice is at most $(q-1)^{d_\perp - 1}$. □

This completes the proof of Lemma 6. □

**Remark D.1.** The key piece of the proof above is a bound on the number of codewords of a given weight, based only on the minimum distance of the code. This corresponds to bounding the size of a "constant weight" code. The bound we give is the analogue of the Singleton bound. It is tight in some cases, such as for Reed-Solomon codes. However, it is quite loose in cases where the alphabet size is small (in that case, there are other much better bounds on constant weight codes [1]. It is sufficient for our purpose: we are mainly interested in proving that reasonable families of codes exist (rather than trying to optimize the parameters).

### D.3 Constructions of Small-Bias Families from Specific Codes

We can now use Lemma 6 to construct small-bias families from known code families.

**Warm-up: Reed-Solomon-Based Constructions** Reed-Solomon (RS) codes are a class of efficiently-decodable $[n', k', d]_q$ linear codes over a large alphabet: $q = 2^e$ must be at least $n$. They have distance $d = n' - k' + 1$ and, because the dual of a Reed-Solomon code is another Reed-Solomon code, they have dual distance $d_\perp = k' + 1$ (see, e,g., [24]).

Consider the family $\{C_i\}$ of binary images of a fixed RS code $C'$. By Lemma 6, the probability that a non-zero word $a$ lies in the dual is at most $\delta^2 = (q-1)^{-d_\perp + 1} = (q-1)^{-k'}$. Since $k < q$ and $(1 - 1/q)^q > 1/3$, we can in fact write $\delta^2 \leq 3q^{-k'} = 3 \cdot 2^{-k}$. Thus, binary images of RS codes (often called "generalized

Reed-Solomon codes") have optimal bias: $\log\left(\frac{1}{\delta}\right) = k/2 - O(1)$, as with random linear codes, matching the lower bound (see Appendix C.2).

Unfortunately, the conversion to a binary alphabet increases the code length and dimension without increasing the distance. Thus, these codes are only guaranteed to correct about $\frac{n-k}{2\log n}$ errors. Nevertheless, for *large alphabets*, these codes do very well. That is, if the metric in which we care about error-correction for the sketch is Hamming distance in $GF(q)^{n'}$, then we get as good a secure sketch as possible, with as small a bias as possible.

**Proposition 12 (RS-based Families for Large Alphabets).** *For all $k < n \leq q = 2^e$, there exists a family $\{C_i\}$ of $[n, k, d]_q$ linear codes for $q \geq n$ with bias $\delta \leq 2^{-k/2+1}$, correcting $\tau \geq \frac{n-k}{2}$ errors efficiently.*

**Algebraic-Geometric Constructions** We now turn to our main construction. Our starting point is a construction of "algebraic-geometric" (AG). We get binary codes with exponentially small bias and linear minimum distance. We will need the following fact:

> [Algebraic-geometric codes, see [42]] Let $q \geq 4$ be an even power of a prime, $q \geq 16$. There exists an infinite ensemble of $[n', k', d]_q$ linear codes $C'$ (over $GF(16)$) with minimum distance at least $d = n' - k' - \frac{n'}{\sqrt{q}-1}$ and dual minimum distance $d' \geq k' - \frac{n'}{\sqrt{q}-1}$. Moreover, these codes have efficient algorithms for decoding up to $\lfloor (d-1)/2 \rfloor$ errors.

This follows from well-known bounds on algebraic-geometric codes (see, e.g., [42], section II.2). The main fact we need is that the dual of an AG code is an AG code for the same curve, and the distance of an AG code is bounded below by $n - k + 1 - g$, where $g$ is the genus of the underlying curve. For infinitely many $n$, there exist curves over $GF(16)$ with $n'$ points and genus at most $n'/3$.

We can now prove Lemmas 7 and 5, which we restate here in a single statement.

**Lemma 13 (Good Code Families Construction).** *For any constant $0 < \lambda < 1$, there exists an explicitly constructible ensemble of code families which* efficiently *correct $\tau = \Omega(n)$ errors and have square bias $\delta^2 < 2^{-\lambda n}$. More specifically, for any constant $R \in [0, 1]$, any even power of two, for inifinitely many $n$ there is a family of binary codes which can efficiently correct $\tau$ errors and have bias $\delta$, where:*

$$\begin{aligned} \tau &\geq \frac{n}{\log q}\left(1 - R - \frac{1}{\sqrt{q}-1}\right) \\ &\textit{and} \\ \log\left(\tfrac{1}{\delta}\right) &\geq \frac{nR}{2}\left(1 - \frac{1}{R(\sqrt{q}-1)}\right)\left(1 - \frac{\log q}{q-1}\right) \end{aligned}$$

In fact, the codes can be made arbitrarily close to optimal, at some cost in error-correction. That is, for any $\gamma > 0$, we can have $\log\left(\frac{1}{\delta}\right) > k/2(1-\gamma)$ and still correct a linear number of errors.

*Proof.* Suppose that $R > 1/2$ (this is the interesting case, since it corresponds to small entropy loss; the case $R < 1/2$ is similar). Let $q$ be any (constant) even power of two. By the facts above on AG codes, there exist $[n', k', d]_q$ codes with rate $k'/n' = R$, minimum distance at least $d = n'(1 - R - \frac{1}{\sqrt{q}-1}) \geq n'(1-R)/2$, and dual distance $d_\perp \geq n'(R - \frac{1}{\sqrt{q}-1}) \geq n'(3R-1)/2$.

We can now apply Lemma 6 to get a family of codes which correct $\tau$ (binary) errors and have bias $\delta$,

where:

$$\begin{aligned}
\tau &\geq \frac{n}{\log q}\left(1 - R - \frac{1}{\sqrt{q}-1}\right) \\
&\text{and} \\
\log\left(\tfrac{1}{\delta}\right) &\geq \frac{1}{2}(d_\perp - 1)(\log(q-1)) \\
&= \frac{n}{2}\left(R - \frac{1}{\sqrt{q}-1}\right)\left(1 - \frac{\log q}{q-1}\right) \\
&= \frac{k}{2}\left(1 - \frac{1}{R(\sqrt{q}-1)}\right)\left(1 - \frac{\log q}{q-1}\right)
\end{aligned}$$

By choosing $q$ to be large enough (but constant), we get codes with constant error-correction rate and exponentially small bias, as desired. In fact, we can get $\log\left(\frac{1}{\delta}\right) > \frac{1}{2}n(R-\gamma)$ for any $\gamma > 0$, and still correct a linear number of errors. Let $\lambda$ be any positive constant less than 1. Setting $R = 1 - (1-\lambda)/2$, and $\gamma = (1-\lambda)/2$, we get $\log\left(\frac{1}{\delta}\right) > \frac{1}{2}\lambda n$, as desired. □

# E Application: Secrecy for Fuzzy Extractors

Fuzzy extractors were introduced in [14] to cope with keys derived from biometrics and other noisy measurements. In this section we show that for fuzzy extractors, as for secure sketches, leaking Shannon information is unavoidable. We also show that the straightforward construction of fuzzy extractors from secure sketches, which extracts a key from $W$ using a pairwise independent hash function, preserves entropic security.

**Definition 5 ([14]).** *An* $(t, \ell, \tau, \epsilon)$ fuzzy extractor *is a given by two procedures* $(\mathsf{Gen}, \mathsf{Rep})$.

1. $\mathsf{Gen}$ *is a probabilistic generation procedure, which on input* $w \in \mathcal{M}$ *outputs an "extracted" string* $R \in \{0,1\}^\ell$ *and a public string* $P$*. We require that for any distribution* $W$ *on* $\mathcal{M}$ *of min-entropy* $t$*, if* $\langle R, P\rangle \leftarrow \mathsf{Gen}(W)$*, then we have* $\mathbf{SD}\left(\langle R, P\rangle, \langle U_\ell, P\rangle\right) \leq \epsilon$.

2. $\mathsf{Rep}$ *is a deterministic reproduction procedure which allows one to recover* $R$ *from the corresponding public string* $P$ *and any vector* $w'$ *close to* $w$*: for all* $w, w' \in \mathcal{M}$ *satisfying* $\mathsf{dist}(w, w') \leq \tau$*, if* $\langle R, P\rangle \leftarrow \mathsf{Gen}(w)$*, then we have* $\mathsf{Rep}(w', P) = R$.

*The fuzzy extractor is* efficient *if* $\mathsf{Gen}$ *and* $\mathsf{Rep}$ *run in time polynomial in the representation size of a point in* $\mathcal{M}$.

**A Simple Construction** Recall that for secure sketches, we required that $Y(W) = S(W)$ be entropically secure. For fuzzy extractors, we will in fact require that the *pair* $Y(W) = \langle P, Z\rangle$ satisfy the definition of security. This is somewhat counter-intuitive: we think of $P$ as being published and $Z$ as being used as a secret key in some other application. However, we cannot guarantee that no information about $Z$ will be leaked in the other application (indeed, if $Z$ is used to encrypt a known string it may be leaked completely). Requiring that the pair $\langle P, Z\rangle$ be entropically secure protects against arbitrary information being revealed about $Z$.

Nevertheless, if we consider fuzzy extractors built from a sketch scheme and a hash family (as in [14]), then the requirement that $\langle Z, P\rangle$ be entropically secure reduces to the requirement that $S(W)$ be entropically secure. The following lemma follows from a standard hybrid argument:

**Lemma 14.** *Suppose that* $S$ *is a secure sketch with entropy loss* $t - t'$*, and* $H$ *is drawn from a 2-universal hash family from* $n$ *bits to* $t' - 2\log\left(\frac{1}{\epsilon}\right)$ *bits. Let* $P = \langle H, S(W)\rangle$ *and* $Z = H(W)$ *(as in [14]).*

*If* $Y_1(W) = S(W)$ *is* $(t, \epsilon)$*-indistinguishable, then* $Y_2(W) = \langle P, Z\rangle$ *is* $(t, 2\epsilon)$*-indistinguishable.*

Hence, it is sufficient to build secure sketch schemes which are entropically secure—the resulting fuzzy extractors will inherit the property.

## E.1 A Bound on Loss of Shannon Information

The argument that secure sketches must leak a lot of Shannon information, i.e. that $\mathbf{I}(W; S(W))$ must be high follows the lines of Shannon's noisy coding theorem, and is quite simple given the language of information theory.

The argument that fuzzy extractors must also leak a certain amount of Shannon information about their inputs is much more delicate. For simplicity, we restrict our attention to the uniform distribution, which is a valid min-entropy $t$ distribution for any $t$.[8]

The simplest consequence to take away from the result (Proposition 16, below) is that as soon as the number of errors $\tau$ to be tolerated becomes large (say $\sqrt{n}$), then the public part of the fuzzy extractor leaks $\Omega(n)$ bits of information about the secret input.

The proof uses the isoperimetric inequality on the hypercube $\{0,1\}^n$ (see [6], theorem 16.6), so we first introduce some notation. Given a set $S \in \{0,1\}^n$ and a number $\tau$, we let $Out_\tau(S) = \{y \mid \exists w \in S \text{ s.t. } \|w - y\| \leq \tau\}$ be the $\tau$-th *shadow* of $S$, i.e. the set of points of distance at most $\tau$ from some point in $S$. Then the isoperimetric inequality states that balls have the smallest outshadows, for every $\tau$. This allows one to lower bound $|Out_\tau(S)|$ in terms of $|S|$. Since we want to find a closed expression bounding $\mathbf{H}_{sh}(W \mid P)$ above, we will only use the following corollary of the isoperimetric inequality. Here $h_2$ is the binary entropy function, $h_2(p) = p\log(\frac{1}{p}) - (1-p)\log(\frac{1}{1-p})$.

**Fact 15.** *For every set $S \subset \{0,1\}^n$ such that $|Out_\tau(S)| \leq 2^{n-1}$, we have*

$$|S| \leq A_\tau \cdot |Out_\tau(S)|, \quad \textit{where} \quad A_\tau \leq \frac{\sum_{i=0}^{n/2-\tau-1} \binom{n}{i}}{2^{n-1}} \leq 2^{n(h_2(\frac{1}{2}-\frac{\tau}{n})-1)} \tag{4}$$

*In particular, when $\tau = \Omega(\sqrt{n})$, the ratio is exponentially small, i.e. $A_\tau = 2^{-\Omega(n)}$.*

**Proposition 16.** *Assume $(\mathsf{Gen}, \mathsf{Rep})$ is a $(n, t, \ell, \tau, \epsilon)$ fuzzy extractor, and let the output of the generation algorithm $\mathsf{Gen}(W)$ be $P, Z$, where $P$ is the public part and $Z$, the extracted key. Then for the uniform distribution $W \leftarrow \{0,1\}^n$, we have*

$$\mathbf{I}(W; P) \geq \log\left(\frac{1}{A_\tau}\right) - 2^{-\ell}n - \epsilon(n+\ell) \approx n\left(1 - h_2\left(\frac{1}{2} - \frac{\tau}{n}\right)\right)$$

*where $\alpha_\tau$ is as in Fact 15. If $\tau = \Omega(\sqrt{n})$, $\ell = \omega(1)$ and $\epsilon = o(1)$, then we can use the bounds on $A_\tau$ to conclude $P$ reveals $\Omega(n)$ bits of information about $W$.*

Since $\tilde{\mathbf{H}}_\infty(W \mid P) \leq \mathbf{H}_{sh}(W \mid P)$, the result also implies that average min-entropy of $W$ is reduced.

*Proof.* Since $W$ and $P$ determine $Z$, we have

$$\mathbf{H}_{sh}(W \mid P) = \mathbf{H}_{sh}(W, Z \mid P) = \mathbf{H}_{sh}(Z \mid P) + \mathbf{H}_{sh}(W \mid Z, P).$$

We will bound each of the two last terms separately. We begin with $\mathbf{H}_{sh}(Z|P)$. Let $g(x) = -x\log x$. Recall that the Shannon entropy of a distribution with probabilities $q_1, ..., q_L$ is $\sum_i g(q_i)$. We'll use a simple approximation, which can be derived by computing the derivative of $g()$: for $\delta \geq 0$, $g(2^{-\ell} + \delta) \leq g(2^{-\ell}) + \ell\delta$.

We expect the distribution of the pair $Z$ conditioned on most values $p$ of $P$ to be essentially uniform over $\{0,1\}^\ell$. In order to manipulate the small deviations from uniformity, we let

$$\delta_{p,r} = \max[\Pr(Z = r \mid P = p) - 2^{-\ell},\ 0].$$

[8] Even though our technique works for more general distributions, the particular bounds we get do not appear to be much stronger, while the exact estimates become intractable.

Since $\mathbf{SD}\left(\langle Z, P\rangle, \langle U_\ell, P\rangle\right) \leq \epsilon$, we have $\sum_{p,r} \delta_{p,r} \leq \epsilon$. Now, we can upper bound $\mathbf{H}_{sh}(Z \mid P)$ as follows:

$$\begin{aligned}
\mathbf{H}_{sh}(Z \mid P) &= \sum_p \Pr(P=p)\mathbf{H}_{sh}(Z \mid P=p) \\
&= \sum_p \Pr(P=p) \sum_r g(\Pr(Z=r \mid P=p)) \leq \sum_p \Pr(P=p) \sum_r g(2^{-\ell} + \delta_{p,r}) \\
&\leq \sum_p \Pr(P=p) \sum_r \left(g(2^{-\ell}) + \ell\delta_{p,r}\right) \leq \ell + \ell \sum_p \Pr(P=p) \sum_r \delta_{p,r} \\
&\leq \ell(1+\epsilon)
\end{aligned}$$

Next, given $p$ and $r$, denote by $S_{p,r}$ the set of $w$ for which $\mathsf{Rep}(w,p) = r$. Note that

$$\mathbf{H}_{sh}(W|P=p, Z=r) \leq \log|S_{p,r}|.$$

We wish to bound the size of the sets $S_{p,r}$. To do so, let $T_{p,r} = Out_\tau(S_{p,r})$ be the $\tau$-th shadow of $S_{p.r}$. For any $r_0 \neq r_1$, their $\tau$-th shadows must be disjoint (Why? If $w' \in T_{p,r_0} \cap T_{p,r_1}$, then error correction property of fuzzy extractors would imply that $\mathsf{Rep}(w',p)$ is equal to both $r_0$ and $r_1$, which is impossible.) This allows us to use the following lemma:

**Claim 17.** *For any $2^\ell$ subsets $S_1, ..., S_{2^\ell}$ of $\{0,1\}^n$, if the $\tau$-th shadows $Out_\tau(S_i)$ are mutually disjoint, then the product of the sizes is bounded above:*

$$\log(\prod_i |S_i|) \leq n + 2^\ell(\log(A_\tau) + n - \ell) \tag{5}$$

We will prove the claim below. For now, we can bound $\mathbf{H}_{sh}(W \mid P, Z)$:

$$\begin{aligned}
&\mathbf{H}_{sh}(W \mid P, Z) \\
&= \sum_p \Pr(P=p) \sum_r \Pr(Z=r \mid P=p)\mathbf{H}_{sh}(W \mid P=p, Z=r) \\
&\leq \sum_p \Pr(P=p) \sum_r (2^{-\ell} + \delta_{p,r}) \log|S_{p,r}| \\
&\leq n \sum_p \Pr(P=p)(\sum_r \delta_{p,r}) + 2^{-\ell} \sum_p \Pr(P=p) \log\left(\prod_r |S_{p,r}|\right)
\end{aligned}$$

The first of the terms in the last equation is at most $\epsilon$, since the probabilities $\Pr[P=p]$ are each bounded by 1, and the sum $\sum_{p,r} \delta_{p,r}$ is at most $\epsilon$. To bound the second term, we can apply the claim, once for each value of $p$, to the collection $\{S_{p,r}\}_{r \in \{0,1\}^\ell}$:

$$\begin{aligned}
\mathbf{H}_{sh}(W|P,Z) &\leq \epsilon n + 2^{-\ell} \sum_p \Pr[P=p](n + 2^\ell(\log(A_\tau) + n - \ell)) \\
&= n - \ell + \log(A_\tau) + n(2^{-\ell} + \epsilon)
\end{aligned}$$

Combining the bounds for $\mathbf{H}_{sh}(Z \mid P)$ and $\mathbf{H}_{sh}(W \mid P, Z)$, and replacing $n$ with $\mathbf{H}_{sh}(W)$, completes the proof. We get $\mathbf{H}_{sh}(W|P) \leq \mathbf{H}_{sh}(W) + \log(A_\tau) + n2^{-\ell} + \epsilon(n+\ell)$, which implies the main statement. □

*Proof of Claim 17.* By hypothesis, the $\tau$-th shadows $Out_\tau(S_i)$ are all disjoint, and hence at most one of them can have more than $2^{n-1}$ points. For all the remaining sets, we have $|S_i| \leq A_\tau |Out_\tau(S_i)|$. For the one "exceptional" set $_{i*}$ of large size, we can bound $\log |S_{i*}|$ by $n$. Thus

$$\log(\prod_{i=1}^{2^\ell} |S_i|) \leq n + \log(A_\tau^{2^\ell} \prod_i |Out_\tau(S_i)|) = n + 2^\ell \log(A_\tau) + \log(\prod_i |Out_\tau(S_i)|).$$

The sets $Out_\tau(S_i)$ are all disjoint, and so their sizes sum to at most $2^n$. If one has $2^\ell$ numbers $a_i$ whose sum is less than $2^n$, their product is maximized by setting all $a_i$ to $2^{n-\ell}$. This gives us $\log(\prod_{i=1}^{2^\ell} |S_i|) \leq n + 2^\ell \log(A_\tau) + 2^\ell \log(2^{n-\ell})$, as desired. □

# F Application: Perfectly One-Way Hash Functions

"Perfectly one-way" hash functions (POWFs) were introduced by Canetti [9] to attempt the formalize the common intuition that cryptographic hash functions reveal very little about their input. We will adopt the somewhat simplified version of the definition used in the subsequent paper of Canetti, Micciancio and Reingold [10]; see [9, 10] a discussion of the differences.

Informally, POWFs are *randomized* hash functions $w \mapsto H(w; R)$ which satisfy two properties, First, given $w$ and $y$, one can verify that $y = H(w; r)$ for some value of the randomness $r$. This means that a computationally bounded adversary cannot produce a pair $w' \neq w$ which would pass the same test. Second, if $R$ is random, then $H(w; R)$ reveals "no information" about $w$. The intuition that the hash leaks no information about the input was formalized in [10] using entropic security. Our results apply in two different ways:

**Noise Tolerance** We show how to construct "fuzzy"—that is, noise-resilient—perfect hash functions. The hash value for $w$ allows one to verify whether a candidate string $w'$ is close to $w$, but reveals nothing else about $w$. This is a significant departure from the approach of Canetti *et al.* The motivation behind [9, 10] was to formalize the properties of an ideal "random oracle" which might be achievable by a real computer program. In contrast, even given a random oracle, it is not at all clear how to construct a *proximity* oracle for a particular value $w$ (i.e. an oracle that accepts an input if and only if it is sufficiently close to $w$).

In that sense, the result is also about *code obfuscation:* noise-resilient POWFs might best be viewed as weakly obfuscated versions of a proximity oracle. This is all the more interesting since strong obfuscation is not possible, see [2].

**Improved Construction** We strengthen the results of [10] on information-theoretically-secure POWF's. We reduce the assumptions necessary for security: Canetti, Micciancio and Reingold [10] assume the existence of a collision-resistant hash function with an extra combinatorial property—*regularity* (a.k.a. balancedness)—in order for their proof of security to g through. We show how to modify the proof so the extra condition is unnecessary. We also improve the parameters of the [10] construction, roughly halving the requirement on the min-entropy of the input for the same level of security.

## F.1 Definition of Perfect One-way-ness

Recall the two informal conditions on PWOF's. Formalizing the first requirement is simple, though we note that the hash function requires a key in order to get full collision resistance. We denote by $R_n$ the space of random coins required by the hash, and by $K_n$ the space of keys (for input lengths $n$). A family of keyed randomized hash function $H^{(n)}$ with input length $n$ and output length $\ell(n)$ is a family of functions $\left\{H_k : \{0,1\}^n \times R_n \to \{0,1\}^{\ell(n)}\right\}_{k \in K_n}$. An ensemble of such functions $\mathcal{H} = \left\{H^{(n)}\right\}_{n \in \mathbb{N}}$ consists of one such family for every input length $n$.

**Definition 6 ([9, 10]).** *An ensemble of keyed randomized functions $\mathcal{H} = \{H_k\}_{k \in K_n, n \in \mathbb{N}}$ as above is* publicly verifiable *if there is a polynomial-time verification algorithm* Ver *such that*

- *For all keys $k \in K_n$, inputs $w \in \{0,1\}^n$, and strings $r \in R_n$, $\mathsf{Ver}(k, w, H_k(w; r)) = \text{ACC}$.*
- *For any PPT adversary $\mathcal{A}$, the probability over $k \in K_n$ that $\mathcal{A}(k)$ outputs a triple $(w, y, c)$ such that $\mathsf{Ver}(k, w, c) = \mathsf{Ver}(k, y, c) = \text{ACC}$ is negligible in $n$.*

The intuition that the hash leaks no information about the input was formalized using a definition almost identical to entropic security for predicates. Given the equivalence of entropic security with respect to predicates and functions, we formulate the definition in terms of functions.

The main difference was that in the definitions of [9, 10], the adversary's ability to predict a predicate $g(W)$ given the (randomized) hash value $H(w) = H_k(w; R))$ is compared the adversary's ability to predict $g(W)$ given only polynomially many accesses to an *identity oracle* $Id_w(\cdot)$ which answers outputs "yes" on input $w$ and "no" on any other input.

We'll say an adversary $\mathcal{A}^{\mathcal{O}(\cdot)}$ with access to an oracle $\mathcal{O}(\cdot)$ is poly-limited if there is some polynomial $p(\cdot)$ such that on inputs of length $n$, the adversary makes at most $p(n)$ queries to the oracle. A ensemble $\{W_n\}_{n \in \mathbb{N}}$ of $t(n)$-sources consists of distributions on $\{0,1\}^n$ with min-entropy at least $t(n)$.

**Definition 7 (Perfect One-Way-ness, [9, 10]).** *A ensemble of keyed randomized functions $\mathcal{H} = \{H_k\}_{k \in K_n, n \in \mathbb{N}}$ is $(t(n), \epsilon(n))$-perfectly one-way if for every adversary $\mathcal{A}$, for every ensemble $\{W_n\}_{n \in \mathbb{N}}$ of $t(n)$-sources, and for every function $f : \{0,1\}^* \to \{0,1\}^*$, there exists a poly-limited oracle adversary $\mathcal{A}_*$ such that, for every $n$ and $k \in K_n$:*

$$\Pr_{w \leftarrow W_n, r \leftarrow R_n}[\mathcal{A}(H_k(w; r)) = f(w)] - \Pr_{w \leftarrow W_n, r \leftarrow R_n}[\mathcal{A}^{Id_w(\cdot)}(1^n)) = f(w)] \leq \epsilon(n)$$

Note that adding the identity oracle makes no significant difference when the min-entropy of $W$ is very high and hence the chance that the adversary queries the oracle on $W$ is negligible. Hence, entropic security implies semantic security in the sense of [9, 10]. Despite this implication, the formulation in terms of the identity oracle makes sense in the context, since the public verifiability makes one able to verify if a particular value is indeed $w$. We retain the "oracle" flavor in the definition of noise-resilient POWFs.

### F.1.1 Noise-resilient POWFs

We now define the new primitive which we construct in this section. A *proximity* oracle $B_{w,\tau}(\cdot)$ accepts its input $w'$ if and only if the distance between $w$ and $w'$ is less than $\tau$. Implicit here is a measure of distance between strings. We will only discuss constructions for the Hamming distance, but we formulate the definitions in more generality. We will assume that the distance function $\mathsf{dist}(,)$ is a metric (that is, it satisfies the triangle inequality) on the space $\{0,1\}^*$. For simplicity we also assume that the distance between strings of different lengths is $+\infty$.

An ensemble of hash functions is called a *one-time* $(t(n), \epsilon(n), \tau(n))$-noise-resilient POWF (in the space $\mathsf{dist}(\cdot, \cdot)$) if it satisfies the following two conditions:

**Definition 8 (Proximity Verifiability).** *A ensemble of keyed randomized functions $\mathcal{H} = \{H_k\}_{k \in K_n, n \in \mathbb{N}}$ is* $(\mathsf{dist}(,), \tau(n))$-publicly proximity-verifiable *if there is a polynomial-time verification algorithm* Ver *such that*

- *For all pairs of inputs $w, w' \in \{0,1\}^n$ such that $\mathsf{dist}(w, w') \leq \tau(n)$, keys $k \in K_n$, and strings $r \in R_n$, $\mathsf{Ver}(k, w, H_k(w; r)) = \text{ACC}$.*
- *For any PPT adversary $\mathcal{A}$, the probability over $k \in K_n$ that $\mathcal{A}(k)$ outputs a triple $(w, \tilde{w}, c)$ such that $\mathsf{Ver}(k, w, c) = \mathsf{Ver}(k, \tilde{w}, c) = \text{ACC}$ and $\mathsf{dist}(w, \tilde{w}) \geq 2\tau(n)$ is negligible in $n$.*

**Definition 9 (Proximity-Semantic-Security).** *A ensemble of keyed randomized functions $\mathcal{H} = \{H_k\}_{k\in K_n, n\in\mathbb{N}}$ is $(t(n), \epsilon(n))$-semantically perfectly one-way for $(\mathsf{dist}(,), \tau(n))$ if for every adversary $\mathcal{A}$, for every ensemble $\{W_n\}_{n\in\mathbb{N}}$ of $t(n)$-sources, and for every function $f : \{0,1\}^* \to \{0,1\}^*$, there exists a poly-limited oracle adversary $\mathcal{A}^{\cdot}_*$ such that, for every $n$ and $k \in K_n$:*

$$\Pr_{w\leftarrow W_n, r\leftarrow R_n}[\mathcal{A}(H_k(w;r)) = f(w)] - \Pr_{w\leftarrow W_n, r\leftarrow R_n}[\mathcal{A}^{B_{w,\tau(n)}(\cdot)}(1^n) = f(w)] \leq \epsilon(n)$$

*where $B_{w,\tau}(\cdot)$ is the proximity oracle which accepts its input $w'$ iff $\mathsf{dist}(w, w') \leq \tau$.*

Unlike in the case of an identity oracle, proving that the proximity orace is not useful to the adverary requires much stronger bounds on the initial value of the min-entropy $t$. See the proof of security of the main construction, below.

## F.2 Constructing Noise-resilient POWFs

The basic idea of our main construction is simple: ntropically-secure secure sketches compose well with any ordinary POWF, as long as residual entropy of the secret given the sketch is higher than the entropy requirement for the POWF.

**Theorem 18 (Generic Construction).** *Suppose that*

- *$\{S_n\}_{n\in\mathbb{N}}$ is an ensemble of $(n, t-1, t', \tau)$ sketches which are $(t, \epsilon)$-entropically secure,*
- *$\{H_k\}_{k\in K_n, n\in\mathbb{N}}$ is a (ordinary) POWF as defined above which is $(t' - \log\left(\frac{1}{\epsilon}\right) + 1, \epsilon)$-perfectly one-way,*

*Then the ensemble $\{H'_k\}_{k\in K_n, n\in\mathbb{N}}$ of randomized hash functions given by*

$$\tilde{H}_k(w; \underbrace{r_1, r_2}_{r}) \stackrel{def}{=} S(w; r_1), H_k(w; r_2)$$

*is $\tau$-proximity-verifiable and $(t+1, 2\epsilon)$-perfectly one-way. (Here $t, t', \tau, \epsilon$ are functions of $n$.)*

*Proof.* The fact that the construction in the preceding theorem is *proximity-verifiable* is easy to check. Given a candidate string $\tilde{w}$, and a string $(s, c)$ which is a correctly generated hash of $w$, then the verification alogrithm $\mathsf{Ver}'(k, \tilde{w}, (s, c))$ does the following (a) Run the recovery procedure for $S()$ on the pair $(\tilde{w}, s)$, get back a candidate string $w'$ for $w$, and (b) check if $\mathsf{dist}(\tilde{w}, w') \leq \tau$ and $\mathsf{Ver}(k, w', c) = \text{ACC}$, where $\mathsf{Ver}()$ is the verification function for the original (nonfuzzy) POWF.

If $\tilde{w}$ is indeed close to $w$, then this test will always succeed. [9] On the other hand, if a poly-time adversary can produce a values $\tilde{w}, \tilde{z}$ which both pass verification with the same string $c$, then there are corresponding values $w, z$ within distance $\tau$ of $\tilde{w}$ (resp. $\tilde{z}$) such that $\mathsf{Ver}(k, w, c) = \mathsf{Ver}(k, z, c) = \text{ACC}$. By the verifiability of the original POWF-scheme, it must be that $w = z$, and so $\mathsf{dist}(\tilde{w}, \tilde{z}) \leq \mathsf{dist}(\tilde{w}, w) + \mathsf{dist}(z, \tilde{z}) \leq 2\tau$, as desired.

We now turn to the proof that the scheme is semantically perfectly one-way in the sense of Definition 9. We'll use the following general lemma on composing entropically-secure maps:

**Lemma 19.** *If (1): $Y_1()$ is a $(t, \epsilon)$-entropically-secure map, (2): for all distributions $W$ of min-entropy at least $t-1$ we have $\tilde{\mathbf{H}}_\infty(W \mid Y_1(W)) \geq t'$ and (3): $Y_2()$ is a $(t' - \log\left(\frac{1}{\epsilon}\right) + 1, \epsilon)$ secure map, then the map which outputs the pair $Y(w) = Y_1(w), Y_2(w)$ is $(t+1, 2\epsilon)$-entropically-secure.*

The lemma can be proven using a simple hybrid argument (see below). For now, we can use it to complete the proof of security of the noise-resilient POWF. Let $Y_1 = S()$ and $Y_2 = H_k()$. By the definition of a secure sketch and the hypotheses of the theorem statement, the conditions of the lemma are satisfied, and we get that the map $H'_k(\cdot; R)$ is $(t+1, 2\epsilon)$-entropically-secure. Entropic security implies semantic perfect one-way-ness with the same parameters. □

[9] Similarly, if the sketch only corrects errors with high probability, then the test will succeed with high probability, achieving a sligthly relaxed version of the definition of verifiability.

We can now prove the composition lemma used above:

*Proof of Lemma 19.* The proof follows a hybrid argument. In order to prove $t+1$-entropic security, we will prove $t-1$-indistinguishability and then apply the equivalence. Suppose that $W$ has min-entropy at least $t-1$. With probability $1-\epsilon$ over the values of $S(W)$, the min-entropy $\mathbf{H}_\infty(W \mid S(W))$ will be at least $te' - \log\left(\frac{1}{\epsilon}\right)$ (recall that $t' = \tilde{\mathbf{H}}_\infty(W \mid Y_1(W))$, so $2^{-t'}$ is the average value of $2^{-\mathbf{H}_\infty(W|Y_1(W))}$). Since $Y_1(W)$ is $t' - \log\left(\frac{1}{\epsilon}\right)+1$-entropically secure, it is $\left(t' - \log\left(\frac{1}{\epsilon}\right), 4\epsilon\right)$ indistinguishable and so with probability $1-\epsilon$ (over values of $Y_1(W)$), the statistical difference between $Y_1(W), Y_2(W)$ and $Y_1(W), Y_2(U_n)$ is at most $4\epsilon$. Hence, the overall statistical difference between the two distributions is at most $5\epsilon$. Finally, the distance between $Y_1(W), Y_2(U_n)$ and $Y_1(U'_n), Y_2(U_n)$ is at most $4\epsilon$ since $Y_1()$ is $(t-1, 4\epsilon)$-indistinguishable. By the triangle inequality, the distance between $Y_1(W), Y_2(W)$ and $Y_1(U'_n), Y_2(U_n)$ is at most $9\epsilon$, and so the scheme is $(t-1, 9\epsilon)$-indistinguishable. Applying the equivalence in the other direction completes the proof. □

### F.3 Improved Construction of Ordinary POWFs

Before we can apply the generic construction of the previous section, we need to constructions of ordinary, non-noise-resilient POWF's.

Canetti et al. [10] gave the following simple construction of perfect one-way hash functions which achieves (information-theoretic) entropic secrecy. Given a family of "regular" collision-resistant hash functions $\{\mathsf{crhf}_k\}_{k\in K_n}$, and a family of pairwise independent *permutations* $\{\pi_i\}_{i\in\mathcal{I}}$, we can define a probabilistic map

$$H_k(w; i) = i, \mathsf{crhf}_k(\pi_i(w)).$$

[10] proved that the construction is $(t,\epsilon)$-entropically secure as long as the output length $\ell(n)$ of the functions $\mathsf{crhf}_k$ satisfies $\ell(n) \leq (t - 2\log\left(\frac{1}{\epsilon}\right))/2$. Their analysis also required an additional assumption on the $\mathsf{crhf}$, namely that the functions be "regular" (a.k.a. balanced), that is for all $k$, every point in the image of $\mathsf{crhf}_k$ must have the same number of pre-images.

Here we improve on the analysis in several ways. First, we remove the assumption of regularity. This is based on a version of the left-over hash lemma in which a pairwise independent hash function is fed through an arbitrary function before producing output (Lemma 22). Second, we improve the parameters: we show that their construction only requires $\ell(n) \leq t - 2\log\left(\frac{1}{\epsilon}\right)$ (that is, we may leak twice as many bits about the input without compromising entropic security). Finally, we provide a stronger security guarantee, namely that the adversary may not learn any *function* of the input. We encapsulate these improvements in the following proposition.

**Proposition 20.** *Suppose that*

- $\{\mathsf{crhf}_k(\cdot)\}_{k\in K_n, n\in\mathbb{N}}$ *is a collision-resistant hash family from* $n$ *bits to* $\ell(n)$ *bits,*
- $\ell < t - 2\log\left(\frac{1}{\epsilon}\right)$,
- $\left\{\{\pi_i\}_{i\in\mathcal{I}}\right\}_{n\in\mathbb{N}}$ *is an ensemble of XOR-universal* permutations *of* $\{0,1\}^n$.

*Then the ensemble of randomized hash functions given by:* $H_k(w; i) = i, \mathsf{crhf}_k(\pi_i(w))$ *is* $(t,\epsilon)$*-entropically secure. (Here* $t, t', \tau, \ell, \epsilon$ *are all functions of* $n$*.)*

To prove entropic security, it suffices to prove that the scheme is indistinguishable. The statement follows directly from a variant of the left-over hash lemma (Lemma 22), which basically states that combining XOR-independent permutations with any arbitrary functions yields a "crooked" strong extractor: that is, the output may not be look random, but it will look the same for all input distributions of sufficiently high entropy. Contrary to intuition, this statement does *not* follow directly from the left-over hash lemma.

### F.4 Putting It All Together

We can now combine the results of this chapter so far. Our initial goal was a non-trivial family of noise-resilient POWF's. As mentioned above, these can be viewed as obfuscated code for proximity queries. We would like to combine Theorem 1 with the generic constructions of this section. For this purpose, we will use the fact that if there are length-reducing collision-resistant hash functions, then for for any output length $\ell(n) = \Omega(n)$, there exists a hash family $\{\mathsf{crhf}_k\}_{k \in K_n, n \in \mathbb{N}}$ with output length $\ell(n)$ for which no PPT adversary can find collisions with non-negligible probability. We obtain:

**Theorem 21.** *If collision-resistant hash functions exist, then for any initial entropy $t = \Omega(n)$, there exists a noise-resilient POWF ensemble which tolerates a linear number of errors $\tau = \Omega(n)$, is $(t, \epsilon)$-entropically-secure for $\epsilon = 2^{-\Omega(n)}$ and is proximity-publicly verifiable with negligible soundness error.*

## G Composing Hashing with Arbitrary Functions

This section states and proves the lemma needed for removing the regularity assumption from the construction of [10]. Below, $\mathbf{H}_2(X)$ refers to the Renyi entropy of a random variable $X$, $\mathbf{H}_2(X) \stackrel{\text{def}}{=} -\log \mathsf{Col}(X) = -\log \sum_x ]Pr[X = x]^2$.

**Lemma 22 (Composing with an arbitrary function).** *Let $f : \{0,1\}^N \to \{0,1\}^\ell$ be an arbitrary function. If $\{h_i\}_{i \in \mathcal{I}}$ is a family of pairwise independent hash functions from $n$ bits to $N$ bits and $X$ is a random variable in $\{0,1\}^n$ with Renyi entropy $\mathbf{H}_2(X) \geq \ell + \log\left(\frac{1}{\epsilon}\right) + 1$, then*

$$\langle I, f(h_i(X)) \rangle \approx_\epsilon \langle I, f(U_N) \rangle$$

*where $I \leftarrow \mathcal{I}$, $U_N \leftarrow \{0,1\}^N$ (both drawn uniformly), and $I$, $X$ and $U_N$ are independent.*

This lemma requires a fresh proof—it does not follow directly from the original left-over hash lemma: because $N$ may be much larger than $n$ and $\mathbf{H}_2(X)$, the distributions $\langle I, h_I(X) \rangle$ and $\langle I, U_N \rangle$ need not be indistinguishable. In fact, when $N > n$ they will have statistical distance almost 1.

The idea behind the proof is to show that for all non-zero strings $\alpha \in \{0,1\}^\ell$, the inner product modulo two $\alpha \odot f(h_I(X))$ is distributed almost identically to $\alpha \odot f(U_N)$. Elementary Fourier analysis then shows that the distributions $f(h_I(X))$ and $f(U_N)$ are close (even given $I$). Details follow.

*Proof.* The bias of a distribution $A$ over $\{0,1\}^\ell$ with respect to a string $\alpha$ is defined to be $\mathsf{bias}_\alpha(A) = \left| \mathbb{E}_A \left[ (-1)^{\alpha \odot A} \right] \right| = |2 \Pr[\alpha \odot A = 0] - 1|$.

The following fact about the hypercube $\{0,1\}^\ell$ will be useful below: For any random variables (distributions) $A$ and $B$ on $\{0,1\}^\ell$, we have:

$$\mathbf{SD}(A, B) \leq \sqrt{\sum_{\alpha \in \{0,1\}^\ell} (\mathsf{bias}_\alpha(A) - \mathsf{bias}_\alpha(B))^2}. \tag{6}$$

**Claim 23.** *For every $\alpha \in \{0,1\}^\ell$, the expectation, over $i \leftarrow \mathcal{I}$, of the expression*

$$(\mathsf{bias}_\alpha(f(h_i(X))) - \mathsf{bias}_\alpha(f(U_N)))^2$$

*is at most $\mathsf{Col}(X) = 2^{-\mathbf{H}_2(X)} \leq \epsilon^2 2^{-\ell}$.*

We first show that this claim implies the lemma, and then prove the claim further below. For every $i \in \mathcal{I}$, let $D_i = f(h_i(X))$. The first observation is that the distance we are seeking to bound is the average, taken over $i$, of the distance between $D_i$ and the target distribution $f(h_i(X))$.

$$\mathbf{SD}(\langle I, D_I \rangle, \langle I, f(U_N) \rangle) = \mathbb{E}_I [\mathbf{SD}(D_I, f(U_N))]$$

We can now bound the statistical difference using the biases (Eqn. 6):

$$\mathbf{SD}\left(\langle I, D_I\rangle\ ,\ \langle I, f(U_N)\rangle\right) \leq \mathbb{E}_I\left[\sqrt{\sum_{\alpha}\left(\mathsf{bias}_\alpha(D_I) - \mathsf{bias}_\alpha(f(U_N))\right)^2}\right]$$

For any random variable, $\mathbb{E}\left[\sqrt{X}\right] \leq \sqrt{\mathbb{E}\left[X\right]}$ (Jensen's inequality). Hence

$$\mathbf{SD}\left(\langle I, D_I\rangle\ ,\ \langle I, f(U_N)\rangle\right) \leq \sqrt{\sum_{\alpha}\mathbb{E}_I\left[\left(\mathsf{bias}_\alpha(D_I) - \mathsf{bias}_\alpha(f(U_N))\right)^2\right]}$$

By the main claim above, the term inside the square root sign is at most $\sum_\alpha \epsilon^2 2^\ell = \epsilon^2$, and so the statistical difference which we want to bound is at most $\epsilon^2$. □

To complete the proof above, we just have to prove the claim.

*Proof of Claim 23.* For $\alpha = 0^\ell$, the claim is trivial since the difference of biases is always 0. Fix $\alpha \neq 0^\ell$. Let

$$\mu = \mathsf{bias}_\alpha(f(U_N)) = \mathbb{E}_{U_N}\left[(-1)^{\alpha\odot f(U_N)}\right]$$

Let $p_x = \Pr[X = x]$. Then we can write $\mathsf{bias}\alpha(f(h_I(X))) - \mathsf{bias}_\alpha(f(U_N))$ as

$$\mathsf{bias}\alpha(f(h_I(X))) - \mathsf{bias}_\alpha(f(U_N)) = \sum_{x\in\{0,1\}^n} p_x(\underbrace{(-1)^{\alpha\odot f(h_I(x))} - \mu}_{Z_x})$$

Now let $Z_x$ be the random variable $(-1)^{\alpha\odot f(h_I(x))} - \mu$ (this is a function of $I$). Since $\{h_i\}$ is a pairwise independent family of hash functions, the expectation of $Z_x$ taken over $I$ is exactly 0 (that is, for any fixed $x$, $h_I(x)$ is uniformly distributed over $\{0,1\}^N$). Moreover, the variables $Z_x$ and $Z_y$ are independent for every pair of strings $x \neq y$, so that $\mathbb{E}_{Z_xZ_y}\left[=\right]0$. Thus

$$\mathbb{E}_I\left[\left(\mathsf{bias}\alpha(f(h_I(X))) - \mathsf{bias}_\alpha(f(U_N))\right)^2\right] = \sum_{x,y\in\{0,1\}^n} p_x p_y \mathbb{E}_I\left[Z_xZ_y\right] = \sum_x p_x^2 \mathbb{E}\left[Z_x^2\right]$$

The variance $\mathbb{E}_I\left[Z_x^2\right] = \mathsf{Var}\left[Z_x\right]$ is at most half of the range of $Z_x$, that is 1. Thus the expected square of the difference of biases is at most $\sum_x p_x^2 = \mathsf{Col}(X)$. □