Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets

Yevgeniy Dodis, Bhavana Kanukurthi, Jonathan Katz, Leonid Reyzin, and Adam Smith

***Abstract*—Consider two parties holding samples from correlated distributions W and W', respectively, where these samples are within distance t of each other in some metric space. The parties wish to agree on a close-to-uniformly distributed secret key R by sending a single message over an insecure channel controlled by an all-powerful adversary who may read and modify anything sent over the channel. We consider both the *keyless* case, where the parties share no additional secret information, and the *keyed* case, where the parties share a long-term secret $\mathsf{SK}_{\mathsf{Ext}}$ that they can use to generate a sequence of session keys $\{R_j\}$ using multiple pairs $\{(W_j, W'_j)\}$. The former has applications to, e.g., biometric authentication, while the latter arises in, e.g., the bounded-storage model with errors.**

We show solutions that improve upon previous work in several respects:

- **The best prior solution for the keyless case with no errors (i.e., $t = 0$) requires the min-entropy of W to exceed $2n/3$, where n is the bit-length of W. Our solution applies whenever the min-entropy of W exceeds the *minimal* threshold $n/2$, and yields a longer key.**
- **Previous solutions for the keyless case in the presence of errors (i.e., $t > 0$) required random oracles. We give the first constructions (for certain metrics) in the standard model.**
- **Previous solutions for the keyed case were stateful. We give the first stateless solution.**

I. Introduction

A number of works have explored the problem of *secret-key agreement based on correlated information*, by which two parties holding samples w, w' of correlated random variables W, W' communicate in order to generate a shared, secret, close-to-uniform key R. The problem has variously been called "information reconciliation" (especially when the challenge is to handle differences between the samples held by the parties), "privacy amplification" (especially in the case when $W = W'$ and the goal is to transform a nonuniform shared secret to a uniform one), or "fuzzy extraction." Early work [43], [5], [26], [3] assumed the parties could communicate over a *public* but *authenticated* channel or, equivalently, assumed a passive adversary. This assumption was relaxed in later work [29], [30], [42], [27], [33], which considered an active adversary who could modify all messages sent between the two parties.

The goal of the above works was primarily to explore the possibility of *information-theoretic* security, especially in the context of quantum cryptography; however, this is not the only motivation. The problem also arises in the context of using noisy data (such as biometric information, or observations of some physical phenomenon) for cryptographic purposes, even if computational security suffices. The same problem also arises in the context of the *bounded-storage model* (BSM) [28] in the presence of errors [14], [17]. We discuss each of these in turn.

A. Authentication Using Noisy Data

In the case of authentication/key agreement using noisy data, the random variables W, W' are *close* (with respect to some metric) but not *identical*. For simplicity, we assume the noisy data represents biometric information, though the same techniques apply to more general settings. In this context, two different scenarios have been considered:

"Secure authentication": Here, a trusted server stores some biometric data w of a user, obtained during an initial enrollment. Later, when the user and the server want to establish a secure communication session over an insecure channel, the user locally obtains a fresh biometric scan w' which is close, but not identical, to w. The user and the server then use w and w' to authenticate each other and agree on a key R.

"Key recovery": In this scenario, a user utilizes his biometric data w to generate a random key R along with some public information P, and then stores P on a (possibly untrusted) server. The key R is then used, for example, to encrypt some data for long-term storage. At a later point in time, the user obtains a fresh biometric scan w' along with the value P from the server; together, these values enable the user to recover R (and hence decrypt the encrypted data).

In the second setting the user is, in effect, running a key agreement protocol with *himself* at two points in time, with the (untrusted) server acting as the "communication channel" between these two instances of the user. This second scenario inherently requires a *noninteractive* (i.e., one-message) solution since w is no longer available at the later point in time. Note also that any solution for the second scenario also provides a solution for the first.

This is an expanded and corrected version of [15], [23].

Y. Dodis is with the Department of Computer Science, New York University, and was supported by NSF Grants #0133806, #0311095, and #0515121. E-mail: dodis@cs.nyu.edu.

B. Kanukurthi is with the Department of Computer Science, University of California, Los Angeles. This work was done while the author was at Boston University, and was supported by NSF grants #0311485, #0515100, #0546614, #0831281, #1012910, and #1012798. E-mail: bhavanak@cs.bu.edu.

J. Katz is with the Department of Computer Science, University of Maryland, and was supported by NSF grants #0310751, #0447075, and #0627306. E-mail: jkatz@cs.umd.edu.

L. Reyzin is with the Department of Computer Science, Boston University, and was supported by NSF grants #0311485, #0515100, #0546614, #0831281, #1012910, and #1012798. E-mail: reyzin@cs.bu.edu.

A. Smith is with the Computer Science and Engineering Department, Pennsylvania State University, This work was done while the author was at the Weizmann Institute of Science, and was supported by the Louis L. and Anita M. Perlman Fellowship. E-mail: asmith@cse.psu.edu.

Several protocols for key agreement using noisy data over an *authenticated* channel are known [5], [3], [22], [20], [16]. Most of the existing work for an *unauthenticated* channel, however, solves the problem only for two special cases [29], [30], [42], [27], [33]: (1) when $W = W'$, or (2) when W and W' consist of (arbitrarily many) independent realizations of the same random variable; i.e., $W = (W^{(1)}, W^{(2)}, \ldots)$ and $W' = (W'^{(1)}, W'^{(2)}, \ldots)$. In the case of biometric data, however, W, W' are not likely to be equal and we cannot in general obtain an unbounded number of samples.

Recently, there has been progress on the general case. Renner and Wolf [34] were the first to demonstrate that an *interactive* solution is possible. Their protocol was not efficient, but an efficient version was later given [24]. Boyen [8] showed (in the random oracle model) how to achieve *unidirectional* authentication, as well as a weak form of security for the second scenario (roughly, R remains secret but the user can be fooled into using an incorrect key R'). Boyen et al. [9] showed two solutions to the problem. Their first solution is noninteractive and thus applies to both scenarios above, but relies on random oracles. Their second solution is interactive, and relies on password-based key exchange as a primitive. This means that it provides *computational* rather than *information-theoretic* security; furthermore, given the current state-of-the-art for password-based key exchange, their solution is impractical without additional assumptions such as random oracles or the existence of public parameters.

B. The Bounded-Storage Model and the Keyed Case

Key agreement using correlated information arises also in the context of the *bounded-storage model* (BSM) [28] in the presence of errors [14], [17]. In the BSM, two parties share a long-term secret key $\mathsf{SK}_{\mathsf{BSM}}$. In each time period j, a long random string Z_j is broadcast to the parties (and the adversary); the assumption is that the length of Z_j is more than what the adversary can store. The parties use $\mathsf{SK}_{\mathsf{BSM}}$ and Z_j to generate a secret session key R_j in each period. This process should achieve "everlasting security" [1], meaning that even if $\mathsf{SK}_{\mathsf{BSM}}$ is revealed to the adversary in some time period n, all session keys $\{R_j\}_{j<n}$ remain independently and uniformly distributed from the perspective of the adversary.

A paradigm (formalized by [39]) for achieving the above is for $\mathsf{SK}_{\mathsf{BSM}}$ to contain a seed $\mathsf{SK}_{\mathsf{Sam}}$ for a sampler[1] and another seed $\mathsf{SK}_{\mathsf{Ext}}$ for a randomness extractor. The parties use $\mathsf{SK}_{\mathsf{Sam}}$ to sample some portion of Z_j in each period; in the absence of errors, this results in each party holding the same value w_j. Since the adversary may have some partial information about w_j, however, this shared value is not uniformly distributed from the point of view of the adversary, and the parties must therefore use a randomness extractor with the seed $\mathsf{SK}_{\mathsf{Ext}}$ to generate a uniform key R_j for the current period. In the presence of transmission errors in Z_j the problem is even more difficult, as the parties then hold correlated (but possibly unequal) strings w_j, w'_j after the initial sampling. The parallels to biometric authentication should be clear. Nevertheless, the problems are incomparable: in the case of the BSM with errors there is a stronger setup assumption (namely, that the parties share a long-term key $\mathsf{SK}_{\mathsf{BSM}}$) but the security requirements are more stringent since $\mathsf{SK}_{\mathsf{BSM}}$ needs to be reusable and everlasting security is required.

[1]A sampler [2] is a function that maps $\mathsf{SK}_{\mathsf{Sam}}$ to a set of bit positions. In fact, $\mathsf{SK}_{\mathsf{Sam}}$ may simply encode a set of randomly chosen bit positions, but better samplers — using shorter seeds — are available.

C. Our Contributions

We focus on the abstract problem of secret-key agreement between two parties holding instances w, w' of correlated random variables W, W' that are guaranteed to be close but not necessarily identical. Specifically, we assume that w and w' are within distance t in some underlying metric space. Our definitions as well as some of our results hold for arbitrary metric spaces, while other results assume specific metrics.

We restrict our attention to *noninteractive* protocols defined by procedures $(\mathsf{Gen}, \mathsf{Rep})$ that operate as follows. The first party, holding w, computes $(R, P) \leftarrow \mathsf{Gen}(w)$ and sends P to the second party; this second party computes $R' \leftarrow \mathsf{Rep}(w', P)$. (If the parties share a long-term key $\mathsf{SK}_{\mathsf{Ext}}$ then $\mathsf{Gen}, \mathsf{Rep}$ take this key as additional input.) The basic requirements, informally, are

Correctness: $R = R'$ if w' is within distance t of w.

Security: If the min-entropy of W is high, then R is uniformly distributed even given P.

So far, this gives exactly a *fuzzy extractor* as defined by Dodis et al. [16] (although we additionally allow the possibility of a long-term key). Since we are interested in the case when the parties communicate over an *unauthenticated* channel, however, we actually want to construct *robust* fuzzy extractors [9] that additionally protect against malicious modification of P. Robustness requires that if the adversary sends any modified value $\tilde{P} \neq P$, then with high probability the second player will reject (i.e., $\mathsf{Rep}(w', \tilde{P}) = \perp$). We distinguish between the notion of *pre-application robustness* and the stronger notion of *post-application robustness*, where in the latter case the adversary is given R before it generates $\tilde{P}$. Post-application robustness is needed in settings where the first party may begin using R before the second party computes R', and is also needed for the "key recovery" scenario discussed earlier (since previous usage of R may leak information about it).

We now summarize our results:

The case of no errors. Although our focus is on the case when W, W' are unequal, we obtain improvements also in the case when they are equal (i.e., $t = 0$) but nonuniform. Let m denote the min-entropy of W and let $n \geq m$ denote its bit-length. The best previous noninteractive solution in this setting is due to Maurer and Wolf [27] who show that when $m > 2n/3$ it is possible to achieve pre-application robustness and generate a shared key R of length $m - 2n/3$. On the other hand, results of [18], [19] imply that a non-interactive solution is impossible when $m \leq n/2$. (As shown in [27, Section III-C], interactive solutions can do better; in fact, it is possible for the length of R to be nearly m [33], [19], [11].)

We bridge the gap between known upper- and lower-bounds and show that whenever $m > n/2$ it is possible to achieve pre-application robustness and generate a shared key R of length

$2m - n$. This improves both the required min-entropy of W and the length of the resulting key. Moreover, we give the first solution satisfying *post-application* robustness. That solution also works as long as $m > n/2$, but extracts a key half as long (that is, of length $m - n/2$).

Handling errors. The only previously known construction of robust fuzzy extractors [9] relies on the random oracle model. We (partially) resolve the main open question of [9] by showing a construction of robust fuzzy extractors *in the standard model* for the specific cases of the Hamming and set-difference metrics.[2] (The solution in [9] is generic and applies to any metric admitting a good error-correcting code.) Our construction achieves post-application robustness.

The techniques of this paper were subsequently generalized in [12].

Using a shared long-term key. There are scenarios in which the two parties trying to derive R from w and w' already share a long-term secret key. Motivated by such settings, we define and construct a *keyed* robust fuzzy extractor for general metrics. In the process, we introduce a new primitive called an *extractor-MAC*: a one-time information-theoretic message authentication code whose output is independent of the key if the message has sufficient entropy.

Application to the BSM with errors. Prior work focusing on the BSM with errors [14], [17] showed a noninteractive (i.e., single-message) solution to the problem discussed in Section I-B when the samples w_j, w'_j of the parties have *constant* relative Hamming distance. The solution of [14] is stateful: the long-term key $\mathsf{SK}_{\mathsf{BSM}}$ is updated by both parties after each time period using information derived from Z_j. If a party misses a time period and is no longer synchronized with the other party, it is not clear how to recover. The solution of [17] is stateless; the parties keep the same long-term key $\mathsf{SK}_{\mathsf{BSM}}$ and can communicate even if one of them misses some Z_j. However, this solution assumes the parties can communicate over an authenticated channel. Building on keyed robust fuzzy extractors, we show a *stateless* solution for the BSM with errors (under the Hamming metric) using an *unauthenticated* channel.

II. Definitions and Preliminaries

For strings a and b, we use $a\|b$ to denote their concatenation and let $|a|$ denote the length of a. If S is a set, $x \leftarrow S$ means that x is chosen uniformly from S. If X is a probability distribution, then $x \leftarrow X$ means that x is chosen according to X. The notation $\Pr_X[x]$ denotes the probability assigned by X to the value x. (We often omit the subscript when the probability distribution is clear from context.) If A is a probabilistic algorithm and x is an input, $A(x;\omega)$ denotes the output of A running with random coins ω, and $A(x)$ is the random variable $A(x;\omega)$ for uniformly sampled ω. If X is a distribution, then $A(X)$ is the random variable obtained by sampling $x \leftarrow X$ and then running $A(x)$. We let U_ℓ denote the uniform distribution over $\{0,1\}^\ell$. All logarithms are base 2.

Let X_1, X_2 be two probability distributions over some set S. Their *statistical distance* is $\mathbf{SD}(X_1, X_2) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{s\in S}|\Pr_{X_1}[s] - \Pr_{X_2}[s]|$. If two distributions have statistical distance at most ε, we say they are ε-close and write $X_1 \approx_\varepsilon X_2$. Note that ε-close distributions cannot be distinguished with advantage better than ε by an adversary who gets a single sample, even if the adversary is computationally unbounded.

The *min-entropy* of a random variable X is defined as $\mathbf{H}_\infty(X) = -\log(\max_x \Pr_X[x])$. Following [16], we define the (average) conditional min-entropy of X given Y as

$$\widetilde{\mathbf{H}}_\infty(X \mid Y) = -\log\left(\mathbf{E}_{y\leftarrow Y}\left(2^{-\mathbf{H}_\infty(X|Y=y)}\right)\right)$$

(where the expectation is over y for which $\Pr[Y = y]$ is nonzero). This definition is suited for cryptographic purposes because the probability that an adversary can predict X when given the value of Y is $2^{-\widetilde{\mathbf{H}}_\infty(X|Y)}$.

Lemma 1 ([16, Lemma 2.2]) *Let Y have at most 2^λ elements in its support. Then $\widetilde{\mathbf{H}}_\infty(X \mid Y) \geq \mathbf{H}_\infty(X, Y) - \lambda$. (More generally, $\widetilde{\mathbf{H}}_\infty(X \mid Y, Z) \geq \widetilde{\mathbf{H}}_\infty(X, Y \mid Z) - \lambda$.)*

A. Hash Functions and Extractors

We recall the notion of almost-universal hashing [10], [36].

Definition 1 A family of efficient functions $\mathcal{H} = \left\{h_i : \{0,1\}^n \to \{0,1\}^\ell\right\}_{i\in I}$ is δ-almost universal if for all $x \neq x'$ we have $\Pr_{i\leftarrow I}[h_i(x) = h_i(x')] \leq \delta$. Families with $\delta = 2^{-\ell}$ are called universal. $\diamondsuit$

A simple universal family [36, Theorem 5.2] can be constructed by identifying I and $\{0,1\}^n$ with $GF(2^n)$ in the natural way, and defining $h_i(x)$ as the high-order ℓ bits of $i \cdot x$.

Extractors [31] yield a (close to) uniform string from a random variable with high min-entropy, using a uniform seed i. *Strong* extractors guarantee that the extracted string is uniform even conditioned on the seed. We consider only strong extractors in this paper, and thus often omit the qualifier "strong."

Definition 2 Let I be a set and the uniform distribution over that set. A function $\mathsf{Ext} : \{0,1\}^n \times I \to \{0,1\}^\ell$ is a strong (m, ε)-extractor if for all distributions X over $\{0,1\}^n$ with $\mathbf{H}_\infty(X) \geq m$ we have $\mathbf{SD}((\mathsf{Ext}(X; I), I), (U_\ell \times I)) \leq \varepsilon$. $\diamondsuit$

We refer to the second argument to Ext as the *seed*.

We need to strengthen the above definition to account for external information E an adversary knows that may be correlated with X. To do so, we generalize the min-entropy constraint on X to average min-entropy, and require the extracted string to be uniform even given E. Namely, we require that for any X, E such that $\widetilde{\mathbf{H}}_\infty(X \mid E) \geq m$ we have $\mathbf{SD}((\mathsf{Ext}(X; I), I, E), (U_\ell \times I \times E)) \leq \varepsilon$. Such extractors are called *average-case extractors*. Note that any $(m - \log\left(\frac{1}{\varepsilon}\right), \varepsilon')$-extractor is an $(m, \varepsilon + \varepsilon')$-average-case extractor, because $\Pr_{e\leftarrow E}[\mathbf{H}_\infty(X \mid e) \leq m - \log\left(\frac{1}{\varepsilon}\right)] \leq \varepsilon$ by Markov's inequality; Vahdan [40] proves the stronger

[2] A previous version of this work [15] contained an erroneous claim of a construction for edit distance, which proceeded by embedding edit distance into set difference using shingling (see [16]). That construction does not work, however, because the embedding fails to preserve the requirement that $m > n/2$.

statement that any (m, ε)-extractor for $m \leq n-1$ is also an $(m, 3\varepsilon)$-average-case extractor. However, the additional loss is not always necessary. Indeed, the Leftover Hash Lemma generalizes without any loss to the average-case setting. (Multiple versions of this lemma have appeared; we use the formulation of [37, Theorem 8.1], augmented by [16, Lemma 2.4] for the average case; see [21] and references therein for earlier formulations.)

Lemma 2 (Leftover Hash Lemma) *Fix $\ell, m, \varepsilon > 0$. If $\mathcal{H} = \{h_i : \{0,1\}^n \to \{0,1\}^\ell\}_{i \in I}$ is a $(2^{-\ell}(1+4\varepsilon^2) - 2^{-m})$-almost universal family, then $\mathcal{H}$ is a strong (m, ε)-average-case extractor (where the index of the hash function is the seed to the extractor). In particular, if $\mathcal{H}$ is universal and $\ell \leq m + 2 - 2\log\left(\frac{1}{\varepsilon}\right)$, then $\mathcal{H}$ is a strong (m, ε)-average-case extractor.*

The above holds even when $\mathcal{H}$ depends on E, i.e., when $\tilde{\mathcal{H}} = \{\mathcal{H}_e\}_{e \in E}$ is a collection of almost-universal families, one for each value of the external information E.

B. One-Time Message Authentication Codes

An (information-theoretic) one-time message authentication code (MAC) consists of polynomial-time algorithms $(\mathsf{Mac}, \mathsf{Vrfy})$. The first algorithm takes a key SK and a message $M \in \{0,1\}^n$ and outputs a tag t; we write this as $t = \mathsf{Mac}_{\mathsf{SK}}(M)$. The *verification algorithm* Vrfy takes as input a key SK, a message $M \in \{0,1\}^n$, and a tag t, and outputs either 1 or 0, with the former being interpreted as acceptance and the latter as rejection. Correctness requires that for all SK and all $M \in \{0,1\}^n$, we have $\mathsf{Vrfy}_{\mathsf{SK}}(M, \mathsf{Mac}_{\mathsf{SK}}(M)) = 1$. Security requires that when SK is chosen uniformly, an unbounded adversary cannot output a valid tag on a new message even after being given the tag on any message of its choice. Formally:

Definition 3 Message authentication code $(\mathsf{Mac}, \mathsf{Vrfy})$ is a δ-secure one-time MAC if for any adversary $\mathcal{A}$ and any message M, the probability that the following experiment outputs "success" is at most δ: Choose uniform key SK; let $t = \mathsf{Mac}_{\mathsf{SK}}(M)$; let $(M', t') \leftarrow \mathcal{A}(t)$; output "success" if $M' \neq M$ and $\mathsf{Vrfy}_{\mathsf{SK}}(M', t') = 1$. $\diamondsuit$

We next recall the notion of (almost) *strongly* universal hashing [41], [36].

Definition 4 A family of efficient functions $\mathcal{H} = \left\{h_i : \{0,1\}^n \to \{0,1\}^\ell\right\}_{i \in I}$ is δ-almost strongly universal if for all $x \neq x', y, y'$ it holds that: (a) $\Pr_{i \leftarrow I}[h_i(x) = y] = 2^{-\ell}$ and (b) $\Pr_{i \leftarrow I}[h_i(x) = y \wedge h_i(x') = y'] \leq \delta 2^{-\ell}$. Families with $\delta = 2^{-\ell}$ are called strongly universal or pairwise independent. $\diamondsuit$

A strongly universal family [36, Theorem 5.2] is obtained by identifying $\{0,1\}^n$ with GF_{2^n}, letting $I = GF(2^n) \times \{0,1\}^\ell$, and defining $h_{a,b}(x)$ as the high-order ℓ bits of $(a \cdot x) \oplus b$.

An almost strongly universal hash family can be used for information-theoretic authentication of a message M using a secret key i, by letting the tag be $t = h_i(M)$. The property of being δ-almost strongly universal implies that this is a δ-secure one-time MAC.

C. Secure Sketches and Fuzzy Extractors

We review the definitions of secure sketches and fuzzy extractors from [16]. Let $\mathcal{M}$ be a metric space with distance function dis. Informally, a secure sketch enables recovery of a string $w \in \mathcal{M}$ from any "close" string $w' \in \mathcal{M}$, without leaking too much information about w.

Definition 5 An $(m, \tilde{m}, t)$-secure sketch for $\mathcal{M}$ is a pair of efficient randomized algorithms $(\mathsf{SS}, \mathsf{SRec})$ such that:

1) The sketching procedure SS takes an input $w \in \mathcal{M}$ and outputs a string $s \in \{0,1\}^*$. The recovery procedure SRec takes as inputs an element $w' \in \mathcal{M}$ and a string $s \in \{0,1\}^*$, and returns an element of $\mathcal{M}$.
2) *Correctness:* If $\mathsf{dis}(w, w') \leq t$ then
$$\mathsf{SRec}(w', \mathsf{SS}(w)) = w.$$
3) *Security:* For any distribution W over $\mathcal{M}$ with $\mathbf{H}_\infty(W) \geq m$, we have $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq \tilde{m}$.

The quantity $m - \tilde{m}$ is called the entropy loss of the secure sketch. $\diamondsuit$

For the case of the Hamming metric on $\mathcal{M} = \{0,1\}^n$, we will make use of the syndrome construction from [16] (this construction also appeared as a component of earlier work, e.g., [4]). Here the sketch $s = \mathsf{SS}(w)$ consists of the k-bit syndrome[3] of w with respect to some (efficiently decodable) $[n, n-k, 2t+1]$-error-correcting code. We do not need any details of this construction other than the facts that s is a (deterministic) *linear function* of w and that the entropy loss is at most $|s| = k$. We also note that this construction can be extended to the set-difference metric [16].

As opposed to a secure sketch, whose goal is to recover the original input, a fuzzy extractor enables generation of a close-to-uniform string R from w, and subsequent reproduction of R from any w' close to w.

Definition 6 An $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ is a pair of efficient randomized algorithms $(\mathsf{Gen}, \mathsf{Rep})$ such that:

1) The generation procedure Gen takes input $w \in \mathcal{M}$ and outputs an extracted string $R \in \{0,1\}^\ell$ and a helper string $P \in \{0,1\}^*$. The reproduction procedure Rep takes as inputs an element $w' \in \mathcal{M}$ and a string $P \in \{0,1\}^*$, and returns a string in $\{0,1\}^\ell$.
2) *Correctness:* If $\mathsf{dis}(w, w') \leq t$ and (R, P) is output by $\mathsf{Gen}(w)$, then $\mathsf{Rep}(w', P) = R$.
3) *Security:* For any distribution W over $\mathcal{M}$ with min-entropy m, the string R is close to uniform conditioned on P. I.e., if $\mathbf{H}_\infty(W) \geq m$ and $(R, P) \leftarrow \mathsf{Gen}(W)$, then $\mathbf{SD}\left((R, P), (U_\ell \times P)\right) \leq \varepsilon$. $\diamondsuit$

Composing an $(m, \tilde{m}, t)$-secure sketch with an average-case $(\tilde{m}, \varepsilon)$-extractor $\mathsf{Ext}\colon \mathcal{M} \times I \to \{0,1\}^\ell$ yields a $(m, \ell, t, \varepsilon)$-fuzzy extractor with $P = (\mathsf{SS}(w), i)$ and $R = \mathsf{Ext}(w; i)$ (see [16, Lemma 4.1]).

Just as with ordinary extractors, a more general definition of fuzzy extractors accounts for external information E and

[3] If H is the parity matrix for a linear code C (i.e., $c \in C$ iff $cH^T = 0$), then the syndrome of a vector w is wH^T.

requires that for any W, E with $\widetilde{\mathbf{H}}_\infty(W \mid E) \geq m$ we have $\mathbf{SD}\left((R,P,E), U_\ell \times (P,E)\right) \leq \varepsilon$. A fuzzy extractor satisfying this definition is called an *average-case fuzzy extractor*, and all known constructions satisfy this more general definition.

In this work we will also use *keyed* fuzzy extractors where both Gen and Rep use the same key $\mathsf{SK}_{\mathsf{Ext}}$, which is uniform and independent of the input distribution W. Here we require the additional security property that $\mathsf{SK}_{\mathsf{Ext}}, R$ are independently uniform conditioned on P. This stronger requirement stems from the fact that $\mathsf{SK}_{\mathsf{Ext}}$ needs to be reusable; thus, it should remain uniform and independent of P, R in order to be useful next time. This requirement implies (by a hybrid argument) that keyed fuzzy extractors can be used multiple times (with the same key $\mathsf{SK}_{\mathsf{Ext}}$) to extract independent keys $\{R_j\}$ from independent $\{W_j\}$. It also implies that any extracted key R_j remains uniform even to an adversary who learns $\mathsf{SK}_{\mathsf{Ext}}$ and P_j (but not w_j).

Definition 7 An $(m, \ell, t, \varepsilon)$-keyed fuzzy extractor for $\mathcal{M}$ is a pair of efficient randomized algorithms (Gen, Rep) such that:

1) Algorithm Gen, on input a key $\mathsf{SK}_{\mathsf{Ext}}$ and $w \in \mathcal{M}$, outputs $R \in \{0,1\}^\ell$ and $P \in \{0,1\}^*$; we denote this by $(R,P) \leftarrow \mathsf{Gen}_{\mathsf{SK}_{\mathsf{Ext}}}(w)$. Algorithm Rep takes as input a key $\mathsf{SK}_{\mathsf{Ext}}$, an element $w' \in \mathcal{M}$, and a string $P \in \{0,1\}^*$, and returns a string in $\{0,1\}^\ell$; we denote this by $R' \leftarrow \mathsf{Rep}_{\mathsf{SK}_{\mathsf{Ext}}}(w', P)$.
2) *Correctness:* For any key $\mathsf{SK}_{\mathsf{Ext}}$, if $\mathsf{dis}(w, w') \leq t$ and (R,P) is output by $\mathsf{Gen}_{\mathsf{SK}_{\mathsf{Ext}}}(w)$, then it holds that $\mathsf{Rep}_{\mathsf{SK}_{\mathsf{Ext}}}(w', P) = R$.
3) *Security:* If $\mathsf{SK}_{\mathsf{Ext}}$ is uniform, distribution W over $\mathcal{M}$ is such that $\mathbf{H}_\infty(W) \geq m$, and $(R,P) \leftarrow \mathsf{Gen}_{\mathsf{SK}_{\mathsf{Ext}}}(W)$, then $\mathbf{SD}\left(\mathsf{SK}_{\mathsf{Ext}} \times (R,P),\, U_{|\mathsf{SK}_{\mathsf{Ext}}|} \times U_\ell \times P\right) \leq \varepsilon$. $\diamondsuit$

For some applications we need to impose the additional condition that, informally, P not reveal any information about the distribution W. Formally, the distribution P should be the same regardless of the distribution W, as long as W has sufficient min-entropy. It is easiest, though slightly more restrictive than necessary, to simply require P to be uniform (for any W with sufficient min-entropy). That is, we say that $(\mathsf{Gen}, \mathsf{Rep})$ has uniform helper strings if the security condition is strengthened to require $\mathbf{SD}\left(\mathsf{SK}_{\mathsf{Ext}} \times (R,P),\, U_{|\mathsf{SK}_{\mathsf{Ext}}|} \times U_\ell \times U_{|P|}\right) \leq \varepsilon$. This additional security condition was subsequently explored in the setting of interactive key agreement [7].

This additional requirement may seem strange: after all, security of a fuzzy extractor depends not on secrecy of the *distribution* W, but only on the fact that W has high min-entropy, which ensures that the specific sample w is secret. However, there are applications that need the distribution W to be kept secret, and the public output of the fuzzy extractor can harm them if this requirement is not satisfied. The specific application considered in this paper is to the bounded-storage model (introduced in Section I-B and addressed in detail in Section IV-C). In this application, the input distribution to the fuzzy extractor depends on the sampling seed $\mathsf{SK}_{\mathsf{Sam}}$, which needs to remain secret so that it can be reused.

D. Robust Fuzzy Extractors

Fuzzy extractors protect against a *passive* attack in which an adversary observes P and tries to learn something about the extracted key R. However, the definition says nothing about what happens if an adversary can modify P as it is sent to the user holding w'. That is, there are no guarantees about the output of $\mathsf{Rep}(w', \tilde{P})$ for $\tilde{P} \neq P$.

Boyen et al. [9] propose the notion of *robust* fuzzy extractors, which provide strong guarantees against such an attack. Specifically, Rep can now output either a key or a special value $\perp$ (denoting "fail"). The definition requires that with high probability any value $\tilde{P} \neq P$ produced by the adversary (after being given P) causes $\mathsf{Rep}(w', \tilde{P})$ to output $\perp$. Modified versions of the public information P will therefore be detected.

We consider two variants of this idea, depending on whether Gen and Rep additionally share a long-term key $\mathsf{SK}_{\mathsf{Ext}}$. (Boyen et al. considered only the keyless version.) Furthermore, we distinguish between two adversarial attacks, and thus two notions of robustness, depending on whether the adversary has access to R when modifying P. Indeed, if R is used (e.g., for encryption) and the adversary can observe some effect of this use (e.g., the ciphertext) before modifying P, then the notion of robustness from Boyen et al. (in which the adversary is given no information about R) is insufficient. Our stronger notion accounts for this by giving the adversary access to R in addition to P. This is a conservative choice that results in a broadly applicable definition: security holds regardless of how R is used and whether it remains hidden partially, computationally, or not at all. We call this stronger notion *post-application* robustness, and refer to the original notion (where R is not given to the adversary) as *pre-application* robustness. Pre-application robustness suffices if the adversary's ability to modify P ends prior to any observable use of R.

If W, W' are two (correlated) random variables over a metric space $\mathcal{M}$, we say $\mathsf{dis}(W, W') \leq t$ if the distance between W and W' is at most t with probability one. We call (W, W') a (t,m)*-pair* if $\mathsf{dis}(W, W') \leq t$ and $\mathbf{H}_\infty(W) \geq m$.

Definition 8 An $(m, \ell, t, \varepsilon)$-fuzzy extractor has post-application (resp., pre-application) robustness δ if for all (t,m)-pairs (W, W') and all adversaries $\mathcal{A}$, the probability that the following experiment outputs "success" is at most δ: Sample (w, w') from (W, W'); let $(R,P) \leftarrow \mathsf{Gen}(w)$; let $\tilde{P} \leftarrow \mathcal{A}(R,P)$ (resp., $\tilde{P} \leftarrow \mathcal{A}(P)$); output "success" if $\tilde{P} \neq P$ and $\mathsf{Rep}(w', \tilde{P}) \neq \perp$. $\diamondsuit$

The definition is illustrated in Figure 1. Note that the definition is interesting even when $w = w'$ (i.e., when $t = 0$), because ordinary extractors are not usually robust. We construct (keyless) robust fuzzy extractors in Section III, and keyed robust fuzzy extractors in Section IV.

The definition of robust extractors composes with itself in some situations. For example, a generalization of the above (used in [9]) allows the adversary to output $(\tilde{P}_1, \ldots, \tilde{P}_j)$; the adversary succeeds if there exists an i with $\mathsf{Rep}(w', \tilde{P}_i) \neq \perp$. A simple union bound shows that the success probability of an adversary in this case increases at most linearly in j.

Similarly, suppose two players (Alice and Bob) receive a sequence of pairs of random variables

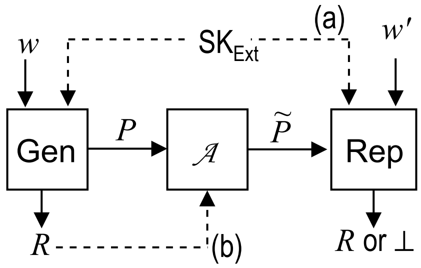

Fig. 1. Robust extractors (cf. Definition 8). Dashed lines indicate variations in the definition: (a) *Keyed extractors* take an additional input $\mathsf{SK}_{\mathsf{Ext}}$ shared by Gen and Rep. (b) For *pre-application* robustness, the adversary does not have access to the extracted key R.

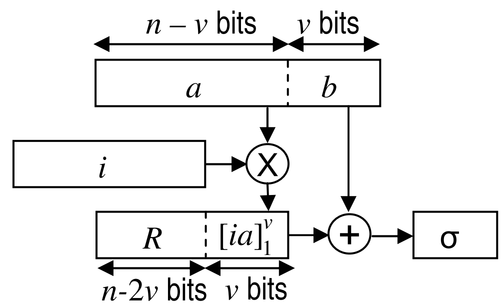

Fig. 2. Construction for the errorless case.

$(W_1, W_1'), (W_2, W_2'), \ldots, (W_j, W_j')$ (with Alice receiving the $\{W_i\}$ and Bob receiving the $\{W_i'\}$), such that $\mathsf{dis}(W_i, W_i') \le t$ for all i, and the entropy of W_i conditioned on the information $\{(W_k, W_k')\}_{k<i}$ from prior time periods is at least m. Alice and Bob can agree on random and independent keys $R_1, \ldots, R_j$ by having Alice apply Gen from a robust average-case fuzzy extractor to each W_i and then send P_i to Bob. The attacker's advantage in distinguishing the vector of unknown keys from random is at most $j\varepsilon$ (this follows by a hybrid argument that replaces extracted keys by random strings one a time, starting with the most recent one). The attacker's probability of forging a valid $\tilde{P}_i$ is at most δ in any given period i (this can be shown by simply giving the attacker $(W_1, W_1'), \ldots, (W_{i-1}, W_{i-1}')$); thus, the overall probability of forgery over all time periods is at most $j\delta$.

For *keyed* fuzzy extractors, robustness is defined exactly as in Definition 8 with the only difference being that Gen and Rep both use the same (uniform) key $\mathsf{SK}_{\mathsf{Ext}}$ (which is not given to the adversary); see Figure 1. At first glance, the addition of a long-term key may seem to trivialize the problem of constructing robust fuzzy extractors. For example, one might attempt to use $\mathsf{SK}_{\mathsf{Ext}}$ as a key for a message authentication code and, given output (R, P) from a fuzzy extractor, simply append to P the tag $\mathsf{Mac}_{\mathsf{SK}_{\mathsf{Ext}}}(P)$. While this may work in the computational setting, it will not suffice in the information-theoretic setting if we want to support an unbounded number of time periods (or if we want to use a key $\mathsf{SK}_{\mathsf{Ext}}$ whose length does not grow linearly in the number of time periods supported). Furthermore, such a construction will not satisfy the security property of Definition 7 because $\mathsf{SK}_{\mathsf{Ext}}$ will not be uniform conditioned on P and $\mathsf{Mac}_{\mathsf{SK}_{\mathsf{Ext}}}(P)$.

III. Constructing (Keyless) Robust Fuzzy Extractors

We begin by analyzing the case of no errors (i.e., $t = 0$), and then consider the more general case.

A. The Errorless Case ($w = w'$)

Consider the case where $\mathcal{M} = \{0,1\}^n$ and Alice and Bob hold the same sample $w \in \{0,1\}^n$ of a random variable W. In the presence of a *passive* adversary, Alice and Bob can agree on a uniform key using a strong extractor Ext. Phrased using the terminology of fuzzy extractors (with $t = 0$ here), Alice runs $\mathsf{Gen}(w)$ which simply samples a seed P for Ext, and sends P to Bob; both Alice and Bob then output the key $R = \mathsf{Rep}(w, P) = \mathsf{Ext}(w, P)$. This solution does not work if the adversary is *active*, which is why *robust* fuzzy extractors are interesting even in the errorless case. In particular, if an adversary forwards $\tilde{P} \ne P$ to Bob then there is no longer any guarantee on Bob's output $\mathsf{Ext}(w; \tilde{P})$; in fact, it is easy to show a construction of a strong extractor Ext with the property that a maliciously generated $\tilde{P}$ completely determines Bob's key $\tilde{R} = \mathsf{Ext}(w; \tilde{P})$. One idea to address this is for Alice to authenticate P using the key R she extracts, and then send the authentication tag along with P to Bob. In general this does not work either: if the adversary forwards $\tilde{P} \ne P$ to Bob, then it may be easy for the adversary to generate a forged tag with respect to the key $\tilde{R}$ that Bob derives. Instead, we use w itself to authenticate P and show that this approach works for a particular choice of strong extractor and message authentication code.

We define algorithms $\mathsf{Gen}, \mathsf{Rep}$ as follows. To compute $\mathsf{Gen}(w)$, parse w as two strings a and b of lengths $n-v$ and v, respectively, where $v < n/2$ is a parameter of the construction. View a as an element of $GF_{2^{n-v}}$ and b as an element of GF_{2^v} (the representation of field elements does not matter, as long as addition in the field corresponds to exclusive-or of bit strings). Choose random $i \in GF_{2^{n-v}}$, let $[ia]_{v+1}^{n-v}$ denote the most significant $n-2v$ bits of $ia \in GF_{2^{n-v}}$, and let $[ia]_1^v$ denote the remaining v bits of ia. View $[ia]_1^v$ as an element of GF_{2^v}. Then compute $\sigma = [ia]_1^v + b$, set $P = (i, \sigma)$, and let the extracted key be $R = [ia]_{v+1}^{n-v}$. See Figure 2.

$\mathsf{Rep}(w, \tilde{P})$, where $\tilde{P} = (i', \sigma')$, proceeds as follows. Parse w as two strings a and b as above. Then verify that $\sigma' = [i'a]_1^v + b$ and output $\perp$ if this is not the case. Otherwise, compute the extracted key $R' = [i'a]_{v+1}^{n-v}$.

The following theorem states the parameters for which $(\mathsf{Gen}, \mathsf{Rep})$ is a robust fuzzy extractor. (Since $t = 0$ here, the metric over $\{0,1\}^n$ is irrelevant.) Observe that extraction is possible as long as $\mathbf{H}_\infty(W) \stackrel{\text{def}}{=} m > n/2$, and in the case of pre-application robustness (which is the notion considered in [27]) we extract a key of length roughly $2m - n$. This improves on the result of Maurer and Wolf [27] who require $m > 2n/3$ and extract a key of length roughly $m - 2n/3$.

Theorem 3 *Fix v, and let $\ell = n - 2v$ be the length of the extracted key. Then:*

- *For any ε, δ satisfying*

$$\ell \le 2m - n - \max\left\{2\log\left(\tfrac{1}{\delta}\right), 4\log\left(\tfrac{1}{\varepsilon}\right)\right\},$$

$(\mathsf{Gen},\mathsf{Rep})$ *is an* $(m,\ell,0,\varepsilon)$*-fuzzy extractor with pre-application robustness* δ.

- *For any* ε,δ *satisfying*

$$\ell \le \min\left\{ \frac{2m-n-2\log\left(\frac{1}{\delta}\right)}{3}, \quad 2m-n-4\log\left(\frac{1}{\varepsilon}\right)\right\},$$

$(\mathsf{Gen},\mathsf{Rep})$ *is an* $(m,\ell,0,\varepsilon)$*-fuzzy extractor with post-application robustness* δ.

Proof: We show that $R\in\{0,1\}^\ell$ is close to uniform conditioned on P, and then argue robustness.

Extraction. We begin by showing that $\mathcal{H}=\{h_i : h_i(a,b)\stackrel{\text{def}}{=}(\sigma,R)\}$ is a universal hash family. Indeed, for $(a,b)\neq(a',b')$ we have

$$\Pr_i\left[h_i(a,b)=h_i(a',b')\right]=$$
$$\Pr_i\left[[ia]_1^v-[ia']_1^v=b'-b \bigwedge [ia]_{v+1}^{n-v}=[ia']_{v+1}^{n-v}\right].$$

This is equivalent to $\Pr_i\left[i(a-a')=0^{n-2v}\|(b'-b)\right]$, where "$\|$" denotes concatenation (this is because we insisted that addition/subtraction in the finite fields corresponds to bitwise exclusive-or). If $a=a'$ then we must have $b\neq b'$ and so the probability is 0. If $a\neq a'$, then there is a unique i that satisfies the equality. Thus, the probability is at most $1/|GF_{2^{n-v}}|=2^{v-n}$.

Using the above and the leftover hash lemma (Lemma 2) we see that $(R,P)=(R,(i,\sigma))$ is $2^{((\ell+v)-m-2)/2}\le\varepsilon/2$-close to $(U_\ell\times U_{n-v}\times U_v)$ or, put differently, that $\mathbf{SD}\left((R,P),U_\ell\times U_n\right)\le\epsilon/2$. This implies $\mathbf{SD}\left((R,P),U_\ell\times P\right)\le\epsilon$ using the triangle inequality.

Pre-application robustness. We prove the stronger result that robustness holds for worst-case choice of i. Fix i and $\mathcal{A}$, and let Succ be the event that $\mathcal{A}$ succeeds. Since $\mathcal{A}$ is unbounded, we may assume it is deterministic. Upon observing σ, the adversary outputs $\mathcal{A}(\sigma)=(i',\sigma')\neq(i,\sigma)$. If $i'=i$, then Rep will reject unless $\sigma'=\sigma$; therefore, we need only consider the case $i'\neq i$. By definition, $\mathcal{A}$ succeeds only if $\sigma'=[i'a]_1^v+b$.

Call a triple (σ,i',σ') a *transcript*, and say it is *possible* if $\mathcal{A}(\sigma)=(i',\sigma')$. For any possible transcript $\mathsf{tr}=(\sigma,i',\sigma')$ the following holds (in the probability expressions below, $a\|b$ are chosen according to the distribution W conditioned on tr or, equivalently, conditioned on σ):

$$\Pr[\mathsf{Succ}\mid\mathsf{tr}]=\Pr_{a\|b}\left[\,[ia]_1^v+b=\sigma\bigwedge[i'a]_1^v+b=\sigma'\right]$$
$$=\Pr_{a\|b}\left[\,[ia]_1^v-[i'a]_1^v=\sigma-\sigma'\bigwedge b=\sigma-[ia]_1^v\right]$$
$$=\Pr_{a\|b}\left[\,[(i-i')a]_1^v=\sigma-\sigma'\bigwedge b=\sigma-[ia]_1^v\right],$$

where the final equality holds because we insisted that addition/subtraction in our fields corresponds to bitwise exclusive-or. The term $(i-i')\cdot a$ takes on each possible value in $GF_{2^{n-v}}$ exactly once as a varies; therefore, there are $2^{n-v}/2^{|\sigma|}=2^{n-2v}$ values of a for which $[a(i-i')]_1^v=\sigma-\sigma'$. For each such value of a, there is a unique value of b that satisfies $b=\sigma-[ia]_1^v$. Each (a,b) pair occurs with probability at most $2^{-\mathbf{H}_\infty(W|\sigma)}$. Thus,

$$\Pr[\mathsf{Succ}\mid\mathsf{tr}]\le 2^{n-2v}\cdot 2^{-\mathbf{H}_\infty(W|\sigma)}.$$

The overall success probability of $\mathcal{A}$ is given by

$$\begin{aligned}\mathbf{E}_{\mathsf{tr}}\left[\Pr[\mathsf{Succ}\mid\mathsf{tr}]\right] &\le 2^{n-2v}\cdot\mathbf{E}_{\mathsf{tr}}\left[2^{-\mathbf{H}_\infty(W|\sigma)}\right]\\ &= 2^{n-2v}\cdot 2^{-\widetilde{\mathbf{H}}_\infty(W|\sigma)}.\end{aligned}$$

Since $|\sigma|=v$, we have $\widetilde{\mathbf{H}}_\infty(W\mid\sigma)\ge m-v$ and we conclude that $\Pr[\mathsf{Succ}]\le 2^{n-v-m}\le\delta$.

Post-application robustness. Because $|R|=\ell$, providing R to the adversary can increase its success probability by a multiplicative factor of at most 2^ℓ as compared to pre-application robustness.[4] Thus, if $3\ell\le 2m-n-2\log\left(\frac{1}{\delta}\right)$ the adversary's success probability (in the post-application robustness game) is at most $2^\ell\cdot 2^{n-v-m}=2^\ell\cdot 2^{(n+\ell-2m)/2}\le\delta$. ■

B. Improved Post-Application Robustness for the Errorless Case

In this section, we present a construction of an extractor with post-application robustness that extracts a key of length $m-n/2-\log\left(\frac{1}{\delta}\right)$, an improvement by a factor of $3/2$ as compared to the construction given above.

Assume n is even for simplicity. To compute $\mathsf{Gen}(w)$, let a and b denote the first and last halves of w, respectively, and view a and b as elements of $GF_{2^{n/2}}$. Choose a random $i\in GF_{2^{n/2}}$ and compute $y=ia+b$. Let σ be the first v bits of y, where $v<n/2$ is a parameter of the scheme, and let R be the remainder of y; i.e., $\sigma=[y]_1^v$ and $R=[y]_{v+1}^{n/2}$. Output $P=(i,\sigma)$.

$\mathsf{Rep}(w,\tilde{P})$, where $\tilde{P}=(i',\sigma')$, proceeds in the obvious way: Parse w as two strings a,b as above. Then verify that $\sigma'=[i'a+b]_1^v$ and output $\perp$ if this is not the case. Otherwise, compute the extracted key $R'=[i'a+b]_{v+1}^{n/2}$.

Before giving the formal proof, we provide some intuition as to why this construction has better post-application robustness. Recall that in the previous construction w is parsed as two strings a and b of lengths $n-v$ and v, respectively, and the values σ,R are computed as $\sigma=[ia]_1^v+b$ and $R=[ia]_{v+1}^{n-v}$. Increasing v improves robustness but decreases the number of extracted bits. For pre-application robustness, setting $v=n-m+\log\left(\frac{1}{\delta}\right)$ suffices, and thus the construction extracts nearly $(2m-n)$ bits. For post-application robustness, however, a larger v must be used and consequently the number of extracted bits is decreased.

The post-application robustness game reveals more information to the adversary $\mathcal{A}$ about w than the pre-application robustness game. This additional information—namely, R itself—may make it easier for $\mathcal{A}$ to guess σ'. The key to our improvement is to use the pairwise-independent function $h_i(a,b)=ia+b$ to compute both σ and R. Because of pairwise independence, the value (σ,R) of $h_i(a,b)$ leaks nothing about the value $(\sigma',R')=h_{i'}(a,b)$ for any $i'\neq i$. (This holds when (a,b) is uniform; when (a,b) has min-entropy m, then $\mathcal{A}$ may have up to $n-m$ bits of information about σ'.) In contrast, in the previous construction only σ was computed using a pairwise-independent hash function. This works better

[4] One might hope to improve this analysis, but we show in Appendix A that the analysis here is essentially tight.

for pre-application robustness (because b can be taken shorter), but worse for post-application robustness.

Theorem 4 *Fix v, and let $\ell = n/2 - v$ be the length of the extracted key. Then for any ε, δ satisfying*

$$\begin{aligned} \ell &\leq m - n/2 - \log\frac{1}{\delta} \\ m &\geq n/2 + 2\log\frac{1}{\varepsilon}, \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ *is an $(m, \ell, 0, \varepsilon)$-fuzzy extractor with post-application robustness δ.*

Proof: We first show that $R \in \{0,1\}^\ell$ is nearly uniform given P. The proof proceeds along the lines of the analogous proof for Theorem 3. As before, we begin by showing that $\mathcal{H} = \{h_i : h_i(a,b) = (\sigma, R)\}$ is universal. Indeed, for $(a,b) \neq (a',b')$ we have

$$\begin{aligned} \Pr_i[h_i(a,b) = h_i(a',b')] &= \Pr_i[ia + b = ia' + b'] \\ &= \Pr_i[i(a - a') = (b - b')],. \end{aligned}$$

If $a = a'$ then $b \neq b'$ and so $\Pr_i[i(a-a') = (b-b')] = 0$. If $a \neq a'$, then there is a unique i for which $i(a-a') = (b-b')$, and so $\Pr_i[i(a-a') = (b-b')] = 2^{-n/2}$.

The above and Lemma 2 imply that (i, R, σ) is $2^{(n/2-m)/2-1}$-close to $U_{n/2} \times U_\ell \times U_v$. As in the previous proof, and recalling that $P = (i, \sigma)$, this means that $\mathbf{SD}\left((R,P), U_\ell \times P\right) \leq 2^{(n/2-m)/2} \leq \epsilon$.

Post-application robustness. As in the previous proof, we prove that robustness holds for worst-case choice of i. Fix i and $\mathcal{A}$, and let Succ be the event that $\mathcal{A}$ succeeds. Since $\mathcal{A}$ is unbounded, we may assume it is deterministic. Thus, upon observing σ, R the adversary outputs $(i', \sigma') \neq (i, \sigma)$; the adversary succeeds if $[i'a + b]_1^v = \sigma'$. Note that if $i' = i$ then Rep will reject unless $\sigma' = \sigma$; therefore, we need only consider the case $i' \neq i$.

We now let a *transcript* be a tuple $\mathsf{tr} = (\sigma, R, i', \sigma')$, and say it is *possible* if $\mathcal{A}(\sigma, R) = (i', \sigma')$. For any possible transcript $\mathsf{tr} = (\sigma, R, i', \sigma')$ we have the following (in the probability expressions below, $a\|b$ are chosen according to the distribution W conditioned on tr or, equivalently, conditioned on σ):

$$\begin{aligned} &\Pr[\mathsf{Succ} \mid \mathsf{tr}] \\ &\quad = \Pr_{a\|b}\left[\left(ia + b = \sigma \| R\right) \bigwedge \left([i'a+b]_1^v = \sigma'\right)\right] \\ &= \sum_{R' \in \{0,1\}^\ell} \Pr_{a\|b}\left[\left(ia + b = \sigma\|R\right) \bigwedge \left(i'a + b = \sigma'\|R'\right)\right]. \end{aligned}$$

For any fixed R', there is a unique value (a,b) for which $ia + b = \sigma\|R$ and $i'a + b = \sigma'\|R'$. Each (a,b) pair occurs with probability at most $2^{-\mathbf{H}_\infty(W|\sigma,R)}$. We thus see that

$$\Pr[\mathsf{Succ} \mid \mathsf{tr}] \leq 2^\ell \cdot 2^{-\mathbf{H}_\infty(W|\sigma,R)}.$$

The overall success probability of $\mathcal{A}$ is given by

$$\mathbf{E}_{\mathsf{tr}}\left[\Pr[\mathsf{Succ} \mid \mathsf{tr}]\right] \leq 2^\ell \cdot 2^{-\widetilde{\mathbf{H}}_\infty(W|\sigma,R)}.$$

Since $|\sigma| + |R| = n/2$, we have $\widetilde{\mathbf{H}}_\infty(W \mid \sigma, R) \geq m - n/2$ and so $\Pr[\mathsf{Succ}] \leq 2^{\ell - m + n/2} \leq \delta$. ■

C. Authenticating a Message While Extracting

Each of the constructions given previously uses the parties' input w to authenticate the extractor seed i. Each construction can be extended to additionally authenticate a message M, i.e., to be simultaneously a robust fuzzy extractor and an information-theoretic one-time MAC. In this setting, both Gen and Rep will take an additional input M, and it should be difficult for an adversary to cause Rep to accept a different M. (We are being informal here since this is merely a stepping stone to the results of the following section.) This could be done naively by using (a part of) R as a key for a MAC, but this would correspondingly reduce the final number of extracted bits. In contrast, the approach presented here (almost) does not reduce the length of R at all.

We show how to extend the original construction given at the beginning of Section III-A; the construction of Section III-B can be extended similarly. We adapt a standard technique [6], [13], [38] for authenticating messages using polynomial-based almost-universal hash functions. Let $|M| = L \cdot (n - v)$, where L is known to both parties in advance. Split M into L chunks $M_0, \ldots, M_{L-1}$, each $n - v$ bits long, and view these as coefficients of a polynomial $M(x) \in GF_{2^{n-v}}[x]$ of degree $L - 1$. To compute $\mathsf{Gen}(w, M)$, parse w as $a\|b$, choose random $i \in GF_{2^{n-v}}$, compute $\sigma = [a^2 M(a) + ia]_1^v + b$, and set $P = (i, \sigma)$. As before, the extracted key is $R = [ia]_{v+1}^{n-v}$.

The procedure Rep, given w, M', and $\tilde{P} = (i', \sigma')$, verifies that $|M'| = L \cdot (n-v)$ and that $\sigma' = [a^2 \cdot M'(a) + i'a]_1^v + b$. If so, it accepts M' as valid and additionally outputs $R = [i'a]_{v+1}^{n-v}$.

Extraction and robustness (which here means that neither i nor M can be modified without detection) are proved in a manner very similar to the proof of Theorem 3. Fix arbitrary M, known to the adversary. To argue that R is nearly uniform given $P = (i, \sigma)$, we will show that $\mathcal{H} = \{h_i : h_i(a,b) \stackrel{\text{def}}{=} (\sigma, R)\}$ is universal. Indeed, for $(a,b) \neq (a',b')$, we have

$$\Pr_i\left[h_i(a,b) = h_i(a',b')\right] = \Pr_i\Bigg[i \cdot (a - a') = \Bigg(0^{n-2v} \, \| \\ \Bigg(\Big[(a')^2 \cdot M(a') - a^2 \cdot M(a)\Big]_1^v + b' - b\Bigg)\Bigg)\Bigg],$$

If $a = a'$ then $b \neq b'$ and the above equality cannot be satisfied; if $a \neq a'$, there is a unique i satisfying the equality. This proves universality. The rest of the proof proceeds as before.

For (pre-application) robustness, fix arbitrary M and i (known to $\mathcal{A}$) and proceed as before. The only difference is that we now need to compute the number of values of a for which

$$[a^2 M(a) + ia - a^2 M'(a) - i'a]_1^v = \sigma - \sigma'. \tag{1}$$

The crucial property is that the polynomial $x^2 M(x) + ix - x^2 M'(x) - i'x$ is nonconstant if $(M, i) \neq (M', i')$. A nonconstant polynomial of degree at most $L+1$ can take on a given value at most $L+1$ times; hence, there are at most $(L+1)2^{n-2v}$ values of a satisfying Eq. (1). The probability that the adversary succeeds (in changing either i or M without being detected) is thus at most $(L+1)\cdot 2^{n-v-m}$. Note that the resulting forgery probability is affected only by a multiplicative factor of $(L+1)$; since we expect $(L+1) \ll 1/\delta$ in practice, the impact is small.

D. Adding Error-Tolerance ($w \neq w'$)

We now consider settings when the input w' held by the second party is close, but not identical to, the input w used by the first party. An obvious first attempt is to include a secure sketch $s = \mathsf{SS}(w)$ along with (i, σ), and to authenticate s using the message-authentication technique discussed in the previous section; s would allow recovery of w from w', and then verification could proceed as before. Unfortunately, this does not quite work: if the adversary modifies the sketch s, then a different value $w^* \neq w$ may be recovered; however, the results of the previous section apply only when the receiver uses the same w as the sender. In effect, we have a circularity: the receiver uses w to verify that s was not modified, but the receiver computes w (from w') using a possibly modified s.

We show how to break this circularity using a modification of the message-authentication technique from earlier. The key idea is to exploit algebraic structure in the metric space, and to change the message authentication code so that it remains secure *even when the adversary can influence the key* (this is sometimes referred to as "security against related-key attacks"; our approach was generalized in [12]). Specifically, we first treat the case where the distance between w and w' is small in the Hamming metric; in Section III-G we extend the approach to the set-difference metric.

Another problem arises from the fact that the performance of our previous constructions degrades not only when the min-entropy m of the input decreases, but also when the entropy gap $g = n - m$ increases (for example, Theorem 3 can extract roughly $m - g$ bits with pre-application robustness). Because s reveals information about w, the entropy of w from the adversary's point of view decreases, and the entropy gap increases. An important idea is to limit this increase by using the (shorter) part of w that is independent of s.

E. Tolerating Binary Hamming Errors

We begin by extending the construction presented at the beginning of Section III-A to tolerate binary hamming errors; we then extend the construction from Section III-B.

Our metric space is $\mathcal{M} = \{0,1\}^n$ and the distance between two strings is Hamming distance—i.e., the number of bit positions in which they differ. Suppose the input W is a distribution of min-entropy m over $\mathcal{M}$, and that w' is guaranteed to be within distance t of w. Our starting point is to use a deterministic, linear, secure sketch $s = \mathsf{SS}(w)$ that is k bits long; let $n' = n - k$ and note that $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq m - k$. We assume that SS is a surjective, linear function (this is the case for the syndrome sketch for the Hamming metric), and so there exists a $k \times n$ matrix S of rank k such that $\mathsf{SS}(w) = S \cdot w$. Let $S^\perp$ be an $n' \times n$ matrix such that the $n \times n$ matrix $\left(\frac{S}{S^\perp}\right)$ has full rank. We let $\mathsf{SS}^\perp(w) \stackrel{\text{def}}{=} S^\perp w$. One can view $\mathsf{SS}^\perp(w)$ as the information remaining in w once $\mathsf{SS}(w)$ has been learned by the adversary.

We define $\mathsf{Gen}, \mathsf{Rep}$ as follows. Gen, on input w, begins by computing $s = \mathsf{SS}(w)$ and $c = \mathsf{SS}^\perp(w)$. It then parses $c \in \{0,1\}^{n'}$ as two strings a, b with $|a| = n' - v$ and $|b| = v$, where $v \leq n'/2$ (so $|a| \geq |b|$) is a parameter of the construction. Letting $L = 2\lceil \frac{k}{2(n'-v)} \rceil$, it pads s with 0s to length $L(n'-v)$ and parses the resulting string as $s_{L-1} \| s_{L-2} \| \cdots \| s_0$ with $s_i \in GF_{2^{n'-v}}$. It chooses random $i \leftarrow GF_{2^{n'-v}}$, and defines $f_{s,i}(x) = x^{L+3} + x^2 \cdot (s_{L-1}x^{L-1} + s_{L-2}x^{L-2} + \cdots + s_0) + ix$. Finally, it sets $\sigma = [f_{s,i}(a)]_1^v + b$, and outputs $R = [ia]_{v+1}^{n'-v}$ and $P = (s, i, \sigma)$.

Rep, on inputs w' and $\tilde{P} = (s', i', \sigma')$, first computes $w^* = \mathsf{SRec}(w', s') \in \{0,1\}^n$. It checks that $\mathsf{dis}(w^*, w') \leq t$ and $\mathsf{SS}(w^*) = s'$; if not, then it outputs $\perp$. Otherwise, let $c' = \mathsf{SS}^\perp(w^*)$ and parse c' as $a' \| b'$ with $|a'| = n' - v$ and $|b'| = v$. Check that $\sigma' = [f_{s',i'}(a')]_1^v + b'$: if not, output $\perp$; otherwise output $R' = [i'a']_{v+1}^{n'-v}$.

Before turning to the detailed analysis, we note that the polynomial $f_{s,i}$ defined above differs from the message-authentication technique in the previous section only in the leading term x^{L+3} (and the forcing of L to be even). It has the property that for any pair $(s', i') \neq (s, i)$, and for any fixed offset Δ_a, the polynomial $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$ is a non-constant polynomial of degree at most $L+2$: this is easy to see for $\Delta_a = 0$; if $\Delta_a \neq 0$, then the leading term is $\Delta_a \cdot x^{L+2}$ (recall we are working in a field of characteristic 2 and L is even). Our analysis will show that $f_{s,i}(a)$ amounts to a message authentication code (where the shared key a is used to authenticate s, i) that is provably secure against a class of related-key attacks where the adversary can force the receiver to use a key shifted by an offset known to the adversary.

Theorem 5 *Let $\mathcal{M}$ denote $\{0,1\}^n$ under the Hamming metric, let SS be the $(m, m-k, t)$-secure syndrome sketch for $\mathcal{M}$, and let B denote the volume of the ball of radius t in $\mathcal{M}$. Fix v, and let $\ell = n - k - 2v$ be the length of the extracted key. Then:*

- *For any ε, δ satisfying*

$$\ell \leq 2m - n - k - 2\max\left\{ \log B + \log\left(2\left\lceil \frac{k}{n-k} \right\rceil + 2\right) + \log\left(\tfrac{1}{\delta}\right),\ 2\log\left(\tfrac{1}{\varepsilon}\right) \right\}$$

$$\leq 2m - n - k - 2\max\left\{ \log B + \log\frac{2n}{\delta},\ 2\log\left(\tfrac{1}{\varepsilon}\right) \right\},$$

  $(\mathsf{Gen}, \mathsf{Rep})$ *is an $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ with pre-application robustness δ.*

- *For any ε, δ satisfying*

$$\ell \leq \min\left\{ \frac{1}{3}\left(2m - n - k - 2\left(\log B + \log\left(2\left\lceil \frac{k}{n-k} \right\rceil + 2\right) + \log\left(\tfrac{1}{\delta}\right)\right)\right),\ 2m - n - k - 4\log\left(\tfrac{1}{\varepsilon}\right) \right\}$$

$$\leq \min\left\{ \frac{1}{3}\left(2m - n - k - 2\log B - 2\log\frac{2n}{\delta}\right), \right.$$

$$2m - n - k - 4\log\left(\tfrac{1}{\varepsilon}\right) \Bigg\},$$

(Gen, Rep) *is an* $(m, \ell, t, \varepsilon)$*-fuzzy extractor for* $\mathcal{M}$ *with post-application robustness* δ.

Note that $\log B \le nH_2(t/n)$ if $t \le n/2$, where $H_2(x)$ is the binary entropy function [25, Chapter 10, §11, Lemma 8], and $\log B \le t\log(n+1)+1$ always.[5]

Before giving the proof, we briefly discuss the parameters obtained. The bound on ℓ differs in two main terms from the bound in the errorless case of Theorem 3. First, we lose the length k of the sketch. This is not surprising, since the sketch may reduce the min-entropy of W by up to k bits. Second, we lose another additive factor of $2\log B$. In general this is (to some extent) inherent, since the min-entropy of W' may be as low as $\mathbf{H}_\infty(W) - \log B$. Looking at it slightly differently, in our analysis we start by giving the attacker $\Delta = w' - w$ "for free", which can reduce the min-entropy of W by $\log B$. We can prove a generalization of the above result where the term $2m - 2\log B$ is replaced by $2\widetilde{\mathbf{H}}_\infty(W \mid \Delta)$. Thus, for example, if errors are independent of w then the term $\log B$ is no longer present.

Proof: That the construction satisfies the functionality of a robust fuzzy extractor is clear, and we thus turn to proving security. The argument that R is nearly uniform given P is similar to the errorless case, except that the entropy loss due to the sketch s has to be taken into account. For every s, the family $\mathcal{H} = \{h_i : h_i(c) \stackrel{\text{def}}{=} (\sigma, R)\}$ is universal because for every $c \ne c'$, there is at most one i such that $h_i(c) = h_i(c')$. Since $\widetilde{\mathbf{H}}_\infty(c \mid \mathsf{SS}(W)) = \widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \ge m - k$, applying Lemma 2 and proceeding as in the proof of Theorem 3 gives $\mathbf{SD}\left((R,P), U_\ell \times P\right) \le 2^{(v+\ell-(m-k))/2} \le \epsilon$.

Pre-application robustness. We prove the stronger result that robustness holds for worst-case choice of i, and even if the adversary is given $\Delta = w' - w$. Fix i and $\mathcal{A}$, and let Succ be the event that $\mathcal{A}$ succeeds. Since $\mathcal{A}$ is unbounded, we may assume it is deterministic. Upon observing s, σ, Δ, the adversary outputs $\mathcal{A}(s,\sigma,\Delta) = (s', i', \sigma') \ne (s, i, \sigma)$. If $(s', i') = (s, i)$ then Rep will reject unless $\sigma' = \sigma$; thus, we need only consider the case $(s', i') \ne (s, i)$.

Call a tuple $(s, \sigma, \Delta, s', i', \sigma')$ a *transcript* and denote it by tr. Call a transcript *feasible* if $\mathcal{A}(s,\sigma,\Delta) = (s', i', \sigma')$. For some fixed feasible transcript, the adversary's success depends only on the choice of w conditioned on the given values of s, σ, Δ, R. (Note w' is determined by w and Δ.)

Recall that w^*, a', b' denote the values reconstructed during the course of applying Rep to w' and s', i', σ'. We claim that for any feasible transcript there is at most one value w^* for which Rep will not reject. Indeed, say there are two distinct values w_1^*, w_2^* for which Rep does not reject; this means $\mathsf{dis}(w', w_1^*), \mathsf{dis}(w', w_2^*) \le t$ and $\mathsf{SS}(w_1^*) = \mathsf{SS}(w_2^*) = s'$. But then $w_1^* = \mathsf{SRec}(w', \mathsf{SS}(w_1^*)) = \mathsf{SRec}(w', s') = \mathsf{SRec}(w', \mathsf{SS}(w_2^*)) \ne w_2^*$, violating correctness of the secure sketch. This implies there is also at most one value for each of a', b' for which Rep will not reject. $\mathcal{A}$ may be unable to compute w^*, a', b' (since it does not know w'); however, we claim that $\mathcal{A}$ *can* compute the differences $\Delta_a = a' - a$ and $\Delta_b = b' - b$. Let $\Gamma \stackrel{\text{def}}{=} w^* - w' = w^* - w - \Delta$, and recall the weight of Γ is at most t. By linearity of SS, we have

$$\mathsf{SS}(\Gamma) = \mathsf{SS}(w^*) - \mathsf{SS}(w) - \mathsf{SS}(\Delta) = s' - s - \mathsf{SS}(\Delta).$$

The right-hand side of the above equation is known to $\mathcal{A}$, and an argument as above shows that there is at most one Γ with weight at most t that satisfies the above equation. Thus, Γ can be computed by $\mathcal{A}$. Linearity of $\mathsf{SS}^\perp$ means that $\mathcal{A}$ can also compute

$$\Delta_a \| \Delta_b = \mathsf{SS}^\perp(w^*) - \mathsf{SS}^\perp(w) = \mathsf{SS}^\perp(\Gamma) + \mathsf{SS}^\perp(\Delta).$$

Next, we prove that for any feasible transcript $\mathsf{tr} = (s, \sigma, \Delta, s', i', \sigma')$, we have

$$\Pr_{w \leftarrow W}[\mathsf{Succ} \mid \mathsf{tr}] \le (L+2) \cdot 2^{n'-2v} \cdot 2^{-\mathbf{H}_\infty(W \mid s,\sigma,\Delta)}. \quad (2)$$

To see this, note that $\mathcal{A}$ succeeds only if $\sigma' = [f_{s',i'}(a')]_1^v + b'$, which is the same as requiring that a be a solution to the equation $[f_{s,i}(a) - f_{s',i'}(a+\Delta_a)]_1^v = \sigma - \sigma' + \Delta_b$. (Recall from above that we may assume Δ_a, Δ_b are known to $\mathcal{A}$.) But for any distinct pairs $(s,i) \ne (s',i')$ and for any Δ_a, the polynomial $f_{s,i}(x) - f_{s',i'}(x+\Delta_a)$ is non-constant and has degree at most $L+2$. (If $\Delta_a = 0$ this is immediate; if $\Delta_a \ne 0$, then the leading term is $(L+3)\cdot\Delta_a \cdot x^{L+2}$, which is non-zero since L is even and we are working in a field of characteristic 2.) Thus, for any $X \in \{0,1\}^{n'-2v}$ the number of values of a for which $f_{s,i}(a) - f_{s',i'}(a+\Delta_a) = X \| \Delta_b + \sigma - \sigma'$ is at most $L+2$, and so the number of values of a that satisfy $[f_{s,i}(a) - f_{s',i'}(a+\Delta_a)]_1^v = \Delta_b + \sigma - \sigma'$ is at most $(L+2)\cdot 2^{n'-2v}$. Each such value occurs with probability at most $2^{-\mathbf{H}_\infty(a \mid s,\sigma,\Delta)}$ (where we let a also stand for the random variable describing the distribution of a), giving the bound $\Pr_{w \leftarrow W}[\mathsf{Succ} \mid \mathsf{tr}] \le (L+2)\cdot 2^{n'-2v} \cdot 2^{-\mathbf{H}_\infty(a \mid s,\sigma,\Delta)}$. Note that

$$\mathbf{H}_\infty(a \mid s,\sigma,\Delta) = \mathbf{H}_\infty(a, s, \sigma \mid s,\sigma,\Delta) = \mathbf{H}_\infty(a, s, b \mid s,\sigma,\Delta),$$

because $b = \sigma - [f_{s,i}(a)]_1^v$; finally,

$$\mathbf{H}_\infty(a, s, b \mid s,\sigma,\Delta) = \mathbf{H}_\infty(W \mid s,\sigma,\Delta)$$

since $w = \left(\frac{S}{S^\perp}\right)^{-1} \cdot (s \| a \| b)$. This completes the proof of Eq. (2).

We may now easily prove the theorem. We have

$$\begin{aligned}\Pr_{w,\Delta}[\mathsf{Succ}] &= \mathbf{E}_{\mathsf{tr}}\Big[\Pr_w[\mathsf{Succ} \mid \mathsf{tr}]\Big] \\ &\le \mathbf{E}_{\mathsf{tr}}\Big[(L+2)\cdot 2^{n'-2v}\cdot 2^{-\mathbf{H}_\infty(W \mid s,\sigma,\Delta)}\Big] \\ &= (L+2)\cdot 2^{n'-2v}\cdot 2^{-\widetilde{\mathbf{H}}_\infty(W \mid s,\sigma,\Delta)},\end{aligned}$$

using Eq. (2). Since $|s| + |\sigma| + |\Delta| \le k + v + \log B$, Lemma 1 gives

$$\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W), \sigma, \Delta) \ge m - k - v - \log B.$$

Observe that $L = 2\lceil k/2(n-k-v)\rceil \le 2\lceil k/(n-k)\rceil$ (because $v \le (n-k)/2$). We conclude that the success probability of $\mathcal{A}$ is at most

[5]Note $B = 1 + \sum_{i=1}^{t}\binom{n}{i}$. The second bound is achieved by noting that every point in the ball centered at 0 can be represented by up to t strings of length $\log(n+1)$ each, where each string represents the position of a 1 or indicates "the end" in case the weight of a point is less than t.

$$B \cdot (L+2) \cdot 2^{n'-2v-m+k+v}$$
$$= B \cdot (L+2) \cdot 2^{n-m-v} \leq B \cdot 2\lceil k/(n-k) \rceil \cdot 2^{n-m-v} \leq \delta.$$

Post-application robustness. Because the extracted key R is of length $n-k-2v$, providing it to the adversary can increase its success probability by at most a factor of 2^{n-k-2v}. The rest of the analysis remains the same. ∎

F. Improved Post-Application Robustness for the Hamming Metric

In this section we extend the construction from Section III-B to tolerate binary Hamming errors. The space $\mathcal{M}$ is still $\{0,1\}^n$ with Hamming distance. $\mathsf{Gen}(w)$ is similar to the one in the previous construction except that now a and b are obtained by splitting c into two equal parts (we assume for simplicity that n' is even) and computing $\sigma = [f_{s,i}(a)+b]_1^v$ and $R = [f_{s,i}(a)+b]_{v+1}^{n'/2}$.

Theorem 6 *Let $\mathcal{M}$ denote $\{0,1\}^n$ under the Hamming metric, let SS be the $(m, m-k, t)$-secure syndrome sketch for $\mathcal{M}$, and let B denote the volume of the ball of radius t in $\mathcal{M}$. Fix v, and let $\ell = (n-k)/2 - v$ be the length of the extracted key. Then for any ε, δ satisfying*

$$\ell \leq m - \frac{1}{2}(n+k) - \log B - \log(2\left\lceil \frac{k}{n-k} \right\rceil + 2) - \log\left(\tfrac{1}{\delta}\right)$$
$$\text{and } m \geq \frac{1}{2}(n+k) + 2\log\left(\tfrac{1}{\varepsilon}\right),$$

$(\mathsf{Gen}, \mathsf{Rep})$ is a $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ with post-application robustness δ.

Proof: We first show that R is nearly uniform given $P = (s, i, \sigma)$. For every s the family $\mathcal{H} = \{h_i : h_i(c) = (\sigma, R)\}$ is universal. Since $\widetilde{\mathbf{H}}_\infty(c \mid \mathsf{SS}(W)) = \widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq m - k$, applying the Leftover Hash Lemma (Lemma 2) and proceeding as in the proof of Theorem 3 shows that $\mathbf{SD}\left((R,P), U_\ell \times P\right) \leq 2^{(n'/2-m+k)/2} \leq \epsilon$.

Post-application robustness. We prove the stronger result that robustness holds for worst-case choice of i, and even if the adversary is given $\Delta = w' - w$. Fix i and $\mathcal{A}$, and let Succ be the event that $\mathcal{A}$ succeeds. Since $\mathcal{A}$ is unbounded, we may assume it is deterministic. Upon observing s, σ, Δ, R, the adversary outputs $\mathcal{A}(s, \sigma, \Delta, R) = (s', i', \sigma') \neq (s, i, \sigma)$. If $(s', i') = (s, i)$ then Rep will reject unless $\sigma' = \sigma$; thus, we need only consider the case $(s', i') \neq (s, i)$.

Call a tuple $(s, \sigma, \Delta, R, s', i', \sigma')$ a *transcript* and denote it by tr. Call a transcript *feasible* if $\mathcal{A}(s, \sigma, \Delta, R) = (s', i', \sigma')$. For some fixed feasible transcript, the adversary's success depends only on choice of w (conditioned on the given values of s, σ, Δ, R).

As in the proof of Theorem 5, for any feasible transcript there is at most one value for each of a', b' for which Rep will not reject, and moreover the values $\Delta_a = a' - a$ and $\Delta_b = b' - b$ can be computed by $\mathcal{A}$. Following an argument exactly as in the proof of that theorem, for any feasible transcript $\mathsf{tr} = (s, \sigma, \Delta, R, s', i', \sigma')$ we have

$$\Pr_{w \leftarrow W}[\mathsf{Succ} \mid \mathsf{tr}] \leq (L+2) \cdot 2^{n'/2-v} \cdot 2^{-\mathbf{H}_\infty(W \mid s,\sigma,\Delta,R)},$$

and so $\Pr[\mathsf{Succ}] =$

$$\mathbf{E}_{\mathsf{tr}}\left[\Pr_{w \leftarrow W}[\mathsf{Succ} \mid \mathsf{tr}]\right] \leq (L+2) \cdot 2^{n'/2-v} \cdot 2^{-\widetilde{\mathbf{H}}_\infty(W \mid s,\sigma,\Delta,R)}.$$

Since $\widetilde{\mathbf{H}}_\infty(W \mid s, \sigma, \Delta, R) \geq m - (|s| + |\sigma| + |R| + |\Delta|) = m - (k + n'/2 + \log B)$, we obtain $\Pr_{w \leftarrow W}[\mathsf{Succ} \mid \mathsf{tr}] \leq B \cdot (L+2) \cdot 2^{n-v} \cdot 2^{-m} \leq \delta$. ∎

G. Construction for the Set-Difference Metric

The constructions from the previous two sections rely heavily on the linearity of the secure sketch used in the protocol and on the structure of the Hamming space. Using the techniques from [16], however, they can be extended to handle errors under the set-difference metric.

In the *set-difference* metric, elements of $\mathcal{M}$ are sets of at most r elements chosen from some fixed universe of size N; the distance between two sets $a, b \in \mathcal{M}$ is the size of their symmetric difference: $\mathsf{dis}(a,b) = |\{x : x \in a \cup b \text{ and } x \notin a \cap b\}|$. Noting that elements of $\mathcal{M}$ can be represented by characteristic vectors of length N, we see that the set-difference metric is equivalent to the Hamming metric; this is inefficient, however, since elements of $\mathcal{M}$ can be represented using at most $r \log N$ bits. Algorithms here should, ideally, run in time $\mathsf{poly}(r \log N)$ rather than time $\mathsf{poly}(N)$.

In order to extend the analysis of the previous sections to handle this different representation of the input, we need a pair of functions $\mathsf{SS}, \mathsf{SS}^\perp$ that take sets and output strings of length k and $r\log(N+1) - k$, respectively. A set w of size at most r should be uniquely determined by the pair $(\mathsf{SS}(w), \mathsf{SS}^\perp(w))$, and the functions should be linear in the following sense: the addition/removal of a particular element should correspond to adding/subtracting a particular bit vector. In other words, $\mathsf{SS}()$ and $\mathsf{SS}^\perp()$ should be linear in the characteristic vector of their input set. The $\mathsf{SS}()$ function of the BCH secure sketch of Dodis et al. [16, Section 6.3] (called "PinSketch") is, in fact, linear: it outputs t values of $\log(N+1)$ bits each in order to correct up to t errors, thus producing sketches of length $k = t\log(N+1)$. We will see in a moment how to construct $\mathsf{SS}^\perp$ corresponding to this SS. For the PinSketch construction the universe must be viewed as nonzero elements of a binary field GF_{2^α} for some α and thus $N = 2^\alpha - 1$.

The constructions of Gen and Rep are the same as in the previous sections, but using different SS, SRec, and $\mathsf{SS}^\perp$ functions. In addition, Rep should check that the recovered value w^* is a set with elements in $GF^*_{2^\alpha}$. (Note, however, that it is not necessary to check that w^* has size at most r; the constructions work correctly even if w' has more than r elements, so long as $\mathsf{dis}(w, w') \leq t$.)

The analysis is the same as in the previous sections. The volume B of the ball of radius t remains the same as in the binary Hamming case; here, N is very large compared to t and so we use $\log B \leq t\log(N+1) = t\alpha$ in our formulas since this is now a close approximation. Using $k = t\log(N+1) = t\alpha$

and $n = r\log(N+1) = r\alpha$, we obtain the following as corollaries of Theorems 5 and 6, respectively.

Corollary 7 *Let $\mathcal{M}$ be the set-difference metric on sets of size at most r over the universe $GF^*_{2^\alpha}$. Using $(\mathsf{Gen}, \mathsf{Rep})$ from Section III-E with $\mathsf{SS}, \mathsf{SS}^\perp, \mathsf{SRec}$ as described above, fix v and then let $\ell = (r-t)\alpha - 2v$ be the length of the extracted key. Then:*

- *For any ε, δ satisfying*

$$\ell \le 2m - r\alpha - t\alpha - 2\max\left\{t\alpha + \log\frac{2r\alpha}{\delta},\ 2\log\left(\tfrac{1}{\varepsilon}\right)\right\},$$

$(\mathsf{Gen}, \mathsf{Rep})$ is an $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ with pre-application robustness δ.

- *For any ε, δ satisfying*

$$\ell \le \min\left\{\frac{1}{3}\left(2m - r\alpha - 3t\alpha - 2\log\frac{2r\alpha}{\delta}\right),\ 2m - (r+t)\alpha - 4\log\left(\tfrac{1}{\varepsilon}\right)\right\},$$

$(\mathsf{Gen}, \mathsf{Rep})$ is an $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ with post-application robustness δ.

Corollary 8 *Let $\mathcal{M}$ be the set-difference metric on sets of size at most r over the universe $GF^*_{2^\alpha}$. Using $(\mathsf{Gen}, \mathsf{Rep})$ from Section III-F with $\mathsf{SS}, \mathsf{SS}^\perp, \mathsf{SRec}$ as described above, fix v and then let $\ell = (r-t)\alpha/2 - v$ be the length of the extracted key. Then for any ε, δ satisfying*

$$\begin{aligned} \ell &\le m - \frac{1}{2}r\alpha - \frac{3}{2}t\alpha - \log\left(\frac{2n}{\delta}\right) \\ m &\ge \frac{1}{2}(t+r)\alpha + 2\log\left(\tfrac{1}{\varepsilon}\right), \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ is a $(m, \ell, t, \varepsilon)$-fuzzy extractor for $\mathcal{M}$ with post-application robustness δ.

It remains to describe $\mathsf{SS}^\perp$. For self-containment, we include a description of SS as well. To compute $\mathsf{SS}(w)$ and $\mathsf{SS}^\perp(w)$ on input $w \subseteq GF^*_{2^\alpha}$, let $s_i \stackrel{\text{def}}{=} \sum_{x\in w} x^i$ (computations in GF_{2^α}) and, viewing s_i values as bit strings, output $\mathsf{SS}(w) = s_1\|s_3\|s_5\|...\|s_{2t-1}$ and $\mathsf{SS}^\perp(w) = s_{2t+1}\|s_{2t+3}\|...\|s_{2r-1}$. Given any set of r points, these two vectors are easy to compute in $O(r^2)$ operations in GF_{2^α}. Moreover, given $s_1, ..., s_{2r-1}$ one can recover w. (Simply observe that $(\mathsf{SS}(w), \mathsf{SS}^\perp(w))$ is the syndrome of the characteristic vector of w with respect to the binary BCH code of distance $2r+1$, and that the weight of this vector is at most r. See [16, Lemma 6.2], setting $n = 2^\alpha - 1$, $k = n - r\alpha$ and $\delta = 2r+1$.) Algorithms $\mathsf{SS}, \mathsf{SS}^\perp$ have the desired linearity property since adding or removing an element y from w corresponds to adding y^i to each component s_i (and we require addition in binary fields to correspond to bitwise exclusive-or).

IV. Keyed Robust Fuzzy Extractors and Their Applications

In this section we show that the addition of a very short, long-term, shared secret key $\mathsf{SK}_{\mathsf{Ext}}$ allows us to achieve considerably better parameters when constructing *keyed* robust fuzzy extractors. The parameters we obtain are optimal up to constant factors.

To motivate our construction, recall the naive transformation from fuzzy extractors to keyed robust fuzzy extractors discussed in Section II-D. Suppose we start from the generic construction of a fuzzy extractor from [16, Lemma 4.1]: here $P = (s, i)$, where $s \leftarrow \mathsf{SS}(w)$ for a secure sketch SS, and the extracted key is $R = \mathsf{Ext}(w; i)$. In an attempt to make this construction robust, we may set $\sigma = \mathsf{Mac}_{\mathsf{SK}_{\mathsf{Ext}}}(s, i)$ and include σ as part of P. This is fine for one-time use, but leaks information about $\mathsf{SK}_{\mathsf{Ext}}$ so cannot be used an unbounded number of times. Formally, this construction does not satisfy Definition 7 since $\mathsf{SK}_{\mathsf{Ext}}$ is not uniform given P.

We can change the scheme to avoid this. Note that Rep must recover $w = \mathsf{Rec}(w', s)$ before computing R. Thus, we can add w to the authenticated message: that is, set $\sigma = \mathsf{Mac}_{\mathsf{SK}_{\mathsf{Ext}}}(w, s, i)$. The tag can be verified by Rep after recovering w. This does not strengthen the robustness property, which was already satisfied by the original scheme. However, it does help with the problem of revealing $\mathsf{SK}_{\mathsf{Ext}}$, since now the attacker $\mathcal{A}$ *does not know the entire message being authenticated*, so the entropy of the message can be used to hide $\mathsf{SK}_{\mathsf{Ext}}$. Thus, we see that we need to construct an information-theoretic MAC whose secret key is independent of the tag as long as the authenticated message has high min-entropy. Observe that in strong randomness extractors, the output is independent of the seed. Thus, it suffices to ensure that Mac is simultaneously a message authentication code and a strong randomness extractor when the key is viewed as the seed. (Note that we do not need the guarantee, provided by the extractor property, that the tag output by Mac is itself uniform; nevertheless, uniform tags are easy enough to achieve.) This is the problem we turn to in the next section.

A. Extractor-MACs

Definition 9 A family $\{\mathsf{Mac}_{\mathsf{SK}} : \{0,1\}^n \to \{0,1\}^v\}$ of functions is a strong (m, ε, δ) (average-case) extractor-MAC if it is δ-almost strongly universal and an (m, ε) (average-case) strong extractor. $\diamondsuit$

When constructing MACs, one typically tries to minimize the tag length v (to approach the bound $\log\left(\frac{1}{\delta}\right)$), while for extractors one tries to maximize the output length v (to approach the bound $\tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right)$). In our setting, the extractor constraint is merely a convenient way to argue key reuse, so we will in fact try to minimize v. Naturally, we also want to minimize the min-entropy threshold m.

Our construction of extractor-MACs follows from the observation that almost strongly-universal hash functions are MACs and, as universal hash functions, also extractors. (In fact, this observation was used to get extractors with short seeds in [35, Section 3].) We exemplify our construction with the family constructed in [6, Section 4]. Specifically, we compose two hash families as follows. Let $\{p_\beta\}$ be a $(\delta\varepsilon^2/2)$-almost universal hash family mapping $\tilde{n}$-bit inputs to u-bit outputs (for some u to be determined later), and let $\{f_\alpha\}$ be a strongly universal hash family mapping u-bit inputs to v-bit outputs, where $v = \log\left(\frac{1}{\delta}\right) + 1$ (i.e., $2^{-v} = \frac{\delta}{2}$). Set $\mathsf{Mac}_{\alpha,\beta}(w) = f_\alpha(p_\beta(w))$. By [36, Theorem 5.5], $\{MAC_{\alpha,\beta}\}$ is a δ-almost strongly universal hash family, since $\delta\varepsilon^2/2 + 2^{-v} \le \delta$. This means it can be used for message authentication. Furthermore, by [36, Theorem 5.4] it is $(\delta\varepsilon^2/2+2^{-v}) = (1+\varepsilon^2)2^{-v}$-almost universal, since $\{f_\alpha\}$ is 2^{-v}-almost universal. By the Leftover Hash Lemma (Lemma 2), this means it is an (m,ε)-extractor with $m = \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$.

We will set $\{f_\alpha\}$ to be the family from [36, Theorem 5.2] (described following Definition 4) with keys of length $u+v$. It remains to set u so that we can construct a convenient almost-universal hash family $\{p_\beta\}$. We use the polynomial-based construction from [6], [13], [38]. The key β is a point in GF_{2^u}, and the message x is split into $c = \tilde{n}/u$ pieces $(x_0, \ldots, x_{c-1})$, each of which is viewed as an element of GF_{2^u}. Then $p_\beta(x_0 \ldots x_c) = x_{c-1}\beta^{c-1} + \ldots + x_1\beta + x_0$. This family is $(c-1)/2^u$-almost universal with key length u (because two distinct degree-$(c-1)$ polynomials agree on at most $c-1$ points). We can set $u = v + \log(\frac{\tilde{n}}{\varepsilon^2}) = 1 + \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right) + \log\tilde{n}$ to make $(c-1)/2^u < \tilde{n}/2^u = \delta\varepsilon^2/2$. This gives key length $2u+v$, and we obtain:

Theorem 9 *For any δ, ε, and $m \ge \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$, there exists a (m,ε,δ)-extractor-MAC for messages of length n, with key length $\kappa = 3 + 2\log n + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right)$ and tag length $v = \log\left(\frac{1}{\delta}\right) + 1$.*

This construction has both short keys and short tags. One can reuse the key SK as long as the min-entropy of the authenticated message is above the threshold $\log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$. The tag length is within one bit of optimal, since it is impossible to obtain δ-almost strong universality with tags shorter than $\log\left(\frac{1}{\delta}\right)$. Known bounds on extractors [32, Theorem 1.9] (reinterpreted for *strong* extractors by viewing the seed as part of the extractor output), imply that the key length is optimal up to a constant factor and the entropy threshold is optimal up to an additive constant.

B. Constructing Keyed Robust Fuzzy Extractors

We now apply extractor-MACs to build keyed robust fuzzy extractors. We start with a generic construction and set the parameters below.

Assume $(\mathsf{SS}, \mathsf{SRec})$ is an $(m, \tilde{m}, t)$-secure sketch with sketch length k; Ext is an average-case $(\tilde{m}, \varepsilon)$-extractor with n-bit inputs, ℓ-bit outputs, and d-bit seeds; and Mac is an average-case $(\tilde{m} - \ell, \varepsilon, \delta)$-extractor-MAC from $\tilde{n} = n + k + d$ bits to v bits having a key SK of length κ. We now define a keyed robust fuzzy extractor with secret key $\mathsf{SK}_{\mathsf{Ext}}$, which is simply the extractor-MAC secret key SK:

- $\mathsf{Gen}_{\mathsf{SK}}(w)$: compute sketch $s \leftarrow \mathsf{SS}(w)$, sample i at random, set key $R = \mathsf{Ext}(w; i)$, tag $\sigma = \mathsf{Mac}_{\mathsf{SK}}(w, s, i)$, $P = (s, i, \sigma)$ and output (R, P).
- $\mathsf{Rep}_{\mathsf{SK}}(w', (s', i', \sigma'))$: Let $\bar{w} = \mathsf{SRec}(w', s')$. If $\mathsf{Mac}_{\mathsf{SK}}(\bar{w}, s', i') = \sigma'$, then $R = \mathsf{Ext}(\bar{w}; i)$; else $R = \perp$.

Theorem 10 *The above construction is a $(m, \ell, t, 4\varepsilon)$-keyed fuzzy extractor with post-application robustness δ, which uses a secret key $\mathsf{SK}_{\mathsf{Ext}}$ of length κ and outputs public information P of length $k + d + v$.*

Proof: We need to show correctness, security, and unforgeability. Correctness follows immediately from the correctness of the secure sketch. To show security (that is, extraction), we need to argue that for any W of min-entropy m, we have

$$(\mathsf{SK}, R, P) \approx_{4\varepsilon} U_{|\mathsf{SK}|} \times U_\ell \times P\,,$$

or, equivalently,

$$(\mathsf{SK}, R, s, i, \sigma) \approx_{4\varepsilon} U_{|\mathsf{SK}|} \times U_\ell \times (s, i, \sigma)\,.$$

Indeed,

$$(R, s, i) \approx_\varepsilon U_\ell \times \mathsf{SS}(W) \times U_d$$

because $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \ge \tilde{m}$ and Ext is an average-case $(\tilde{m}, \varepsilon)$-extractor. This trivially implies that

$$U_{|\mathsf{SK}|} \times (R, s, i) \times U_v \approx_\varepsilon U_{|\mathsf{SK}|} \times U_\ell \times \mathsf{SS}(W) \times U_d \times U_v\,.$$

On the other hand,

$$(\mathsf{SK}, R, s, i, \sigma) \approx_\varepsilon U_{|\mathsf{SK}|} \times (R, s, i) \times U_v$$

because $\widetilde{\mathbf{H}}_\infty(W \mid R, s, i) \ge \widetilde{\mathbf{H}}_\infty(W, R, i \mid s) - \ell - d \ge \widetilde{\mathbf{H}}_\infty(W, i \mid \mathsf{SS}(W)) - \ell - d \ge \widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) + d - \ell - d = \tilde{m} - \ell$ (the first inequality follows from Lemma 1 and the last inequality follows by independence of i), and Mac is a $(\tilde{m} - \ell, \varepsilon)$ average-case extractor.

By the triangle inequality, therefore, we obtain

$$(\mathsf{SK}, R, s, i, \sigma) \approx_{2\varepsilon} U_{|\mathsf{SK}|} \times U_\ell \times \mathsf{SS}(W) \times U_d \times U_v\,.$$

Using the triangle inequality again we obtain the desired result.

To show robustness, suppose $\mathcal{A}$ outputs $\tilde{P} = (s', i', \sigma') \ne (s, i, \sigma)$. First consider the case when $(s, i) = (s', i')$. In this case, $\mathsf{dis}(w, w') \le t$ implies $\mathsf{SRec}(w', s') = w$, and thus $\mathsf{Mac}_{\mathsf{SK}}(w^*, s, i) = \sigma$. Therefore, unless $\sigma' = \sigma$ and $\tilde{P} = P$, Rep will output $\perp$. Now consider the case when $(s, i) \ne (s', i')$. Then, in order for Rep not to reject, $\mathcal{A}$ must correctly guess the tag of a new message with a uniformly chosen key SK, which cannot be done with probability higher than δ by the δ-almost strong universality of Mac. Note that this implies post-application robustness: it does not hurt to reveal R (or even w itself) to $\mathcal{A}$, because the security of Mac relies on the secrecy of SK only. ■

The price of authentication. We compare the parameters of Theorem 10 to the original (non-robust, non-keyed) constructions of [16]. First, note that the choice of a sketch and strong extractor can be done in the same manner as for non-robust fuzzy extractors. Assume we use the construction of Theorem 9 for Mac. Then the secret key $\mathsf{SK}_{\mathsf{Ext}}$ is just the MAC key, whose length is $2\log n + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right) + O(1)$ as long as $d = O(n)$ and $k = O(n)$ (which is the case with typical extractor and secure sketch constructions), so that

$\tilde{n} = O(n)$. For the extractor-MAC of Theorem 9 to work, we need $\tilde{m} - \ell \geq \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$ or $\ell \leq \tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right) - \log\left(\frac{1}{\delta}\right)$. This means that the key R is only $\log\left(\frac{1}{\delta}\right) + 2$ bits shorter than for non-robust extractors, which can extract $\ell = \tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right) + 2$ bits [16, Lemma 4.3]. Finally, the length of P increases only by the tag length $v = \log\left(\frac{1}{\delta}\right) + 1$.

C. Uniform Helper Strings and Application to the Bounded-Storage Model with Errors

Keyed robust fuzzy extractors allow us to remove the need for an authenticated channel between the honest parties in the bounded-storage model (BSM) with errors. As explained following Definition 7, the first step is to construct such extractors with uniform helper strings. We then show in more detail how they apply to the BSM.

Keyed robust extractors with uniform helper strings. Examining the keyed construction in Theorem 10, we see that the only place where the value $P = (s, i, \sigma)$ depends on (the distribution of) w is in the sketch $s \leftarrow \mathsf{SS}(w)$. Indeed, the seed i is chosen uniformly at random, and the value σ is close to uniform (even conditioned on i, s, w, and $\mathsf{SK}_{\mathsf{Ext}}$) by the properties of the extractor-MAC. Thus, to solve our problem we only need to build an $(m, \tilde{m}, t)$-secure sketch SS such that $\mathsf{SS}(W)$ is statistically close to uniform whenever W has sufficient min-entropy. (Note that such sketches cannot be deterministic.) Such sketches were studied by Dodis and Smith [17], where they were used to solve the noisy-BSM problem even in the authenticated-channel case. In particular, Dodis and Smith show such sketches for the binary Hamming metric with parameters that are only a constant factor worse than those of regular sketches.

Theorem 11 ([17, Theorem 1]) *For any min-entropy $m = \Omega(n)$, there exists an efficient $(m, \tilde{m}, t)$-secure sketch for the Hamming metric over $\{0,1\}^n$ that is also an (m, ε)-extractor, where $\tilde{m}$, t, and $\log\left(\frac{1}{\varepsilon}\right)$ are all $\Omega(n)$, and the length of the sketch is $k = O(n)$.*

Using such sketches in the construction of Section IV-B gives us the following theorem.

Theorem 12 *Using the sketch of Theorem 11 in the construction of Section IV-B gives an $(m, \ell, t, 3\varepsilon)$-keyed fuzzy extractor with uniform helper strings and post-application robustness δ.*

Proof: Correctness and unforgeability are shown the same way as in Theorem 10. To show security (that is, extraction) with uniform helper strings, we need to argue that for any W with min-entropy m we have

$$(\mathsf{SK}, R, P) \approx_{3\varepsilon} U_{|\mathsf{SK}|} \times U_\ell \times U_{|P|}$$

or, equivalently,

$$(\mathsf{SK}, R, s, i, \sigma) \approx_{3\varepsilon} U_{|\mathsf{SK}|} \times U_\ell \times U_k \times U_d \times U_v\,.$$

Indeed,

$$(R, s, i) \approx_\varepsilon U_\ell \times \mathsf{SS}(W) \times U_d$$

for the same reason as in Theorem 10. On the other hand, $\mathsf{SS}(W) \approx_\varepsilon U_k$ by Theorem 11 and therefore

$$U_\ell \times \mathsf{SS}(W) \times U_d \approx_\varepsilon U_\ell \times U_k \times U_d$$

which, by the triangle inequality, implies

$$(R, s, i) \approx_{2\varepsilon} U_\ell \times U_k \times U_d\,.$$

The rest of the proof proceeds as in Theorem 10. ■

Application to the bounded-storage model To explain the application, we first briefly recall the key elements of the bounded-storage model [28] with errors [14], [17], concentrating only on the *stateless variant* of [17]. Our discussion will be specific to Hamming distance.

In the bounded-storage model with errors, two parties (say, Alice and Bob) start by sharing a long-term secret key $\mathsf{SK}_{\mathsf{BSM}}$. At each time period j, Alice (resp., Bob) has access to a noisy version X_j (resp., X'_j) of a random string Z_j (of length N). We assume a bound on the Hamming distance of X_j and X'_j. Both the honest parties and the attacker $\mathcal{A}$ are limited in storage to considerably fewer than N bits. More specifically, we assume that $\mathcal{A}$ can look at the entire Z_j but store only γN bits of (arbitrary) information about Z_j, for $\gamma < 1$. After $\mathcal{A}$ has stored its information about Z_j, it cannot see Z_j again; this means that Z_j has average min-entropy $(1-\gamma)N$ from the adversary's point of view by Lemma 1. The honest parties are even more limited in their storage, but they can use their shared secret key to gain an advantage over the adversary and communicate securely without the need for computational assumptions (they can even achieve *everlasting security* [1]).

Prior work [14], [17] assumed that the communication channel between Alice and Bob was authenticated or, equivalently, that the adversary does not modify the messages between Alice and Bob. This authenticated channel was used to reconcile the differences between (the relevant portions of) X_j and X'_j received by the two parties. In this work, we remove the need for the authenticated channel.

The basic idea underlying prior work is to use fuzzy extractors to derive a key R_j from X_j and X'_j that is unknown to $\mathcal{A}$. For example, in "sample-and-extract" protocols [39] one part of $\mathsf{SK}_{\mathsf{BSM}}$ consists of a key $\mathsf{SK}_{\mathsf{Sam}}$ for an *oblivious sampler* [2], [39]. This key specifies n locations in the N-bit string X_j (resp., X'_j) which Alice (resp., Bob) will read to obtain an n-bit substring w_j (resp., w'_j). The properties of the sampler ensure that (a) with high probability w_j and w'_j are still close (say, within Hamming distance t from each other); and (b) with high probability, $\mathcal{A}$ still has some uncertainty (min-entropy $m \approx (1-\gamma)n$) about w_j and w'_j. (Note that it is crucial that $\mathcal{A}$ does not know $\mathsf{SK}_{\mathsf{Sam}}$ at the time Z_j is broadcast, so $\mathcal{A}$ is unable to store information that is specifically correlated to w_j, w'_j.) Fuzzy extractors can then be used to derive R_j from w_j and w'_j, with Alice running $\mathsf{Gen}(w_j)$ to obtain (P_j, R_j) and sending the helper string P_j to Bob over the authenticated channel, and then Bob running $\mathsf{Rep}(w'_j, P)$ to get R_j [14], [17].

To remove the need for an authenticated channel, Alice and Bob can use a *robust* fuzzy extractor instead. Because they are already in the shared-key setting, they can use a *keyed* robust fuzzy extractor, storing its secret key $\mathsf{SK}_{\mathsf{Ext}}$ as part of their long-term secret key $\mathsf{SK}_{\mathsf{BSM}}$ (in addition to $\mathsf{SK}_{\mathsf{Sam}}$). There is, however, a subtle problem which already caused difficulties even in the case of authenticated channels and nonrobust extractors [14], [17].

The problem arises due to the reuse of $\mathsf{SK}_{\mathsf{BSM}}$. As discussed in Sections II-C and II-D, $\mathsf{SK}_{\mathsf{Ext}}$ can be reused safely, but only if the input to the fuzzy extractor has sufficient min-entropy (from the adversary's point of view). In the current setting, however, a potential problem is that $\mathcal{A}$ may use information gleaned from P_j in order to reduce the entropy of w_{j+1}. Specifically, if P_j is correlated with w_j, then P_j may reveal information about the sampler key $\mathsf{SK}_{\mathsf{Sam}}$ that was used to sample w_j. In other words, by observing P_j, $\mathcal{A}$ may learn something about *the locations* in the large random string Z_j that were used to obtain w_j. While it is too late for $\mathcal{A}$ to observe those locations in Z_j (because of the bounded-storage assumption), $\mathcal{A}$ may be able to observe the same locations in the next string Z_{j+1}, thus reducing the min-entropy of w_{j+1}, which will be obtained from those locations.

We can solve this problem by making sure that P_j reveals nothing about $\mathsf{SK}_{\mathsf{Sam}}$. This is precisely what is guaranteed by keyed robust fuzzy extractors *with uniform helper strings*, as constructed in Theorem 12, since P_j is distributed the same way (up to a small statistical distance) regardless of what $\mathsf{SK}_{\mathsf{Sam}}$ is. (To use Theorem 12, we need to ensure that the input to the extractor has sufficient min-entropy. This holds with overwhelming probability, even conditioned on $\mathsf{SK}_{\mathsf{Sam}}$ and the knowledge of $\mathcal{A}$, because $\mathcal{A}$ is unlikely to have stored much useful information about the locations sampled by $\mathsf{SK}_{\mathsf{Sam}}$.) Thus, using extractors with uniform helper strings ensures that the public value P hides the entire $\mathsf{SK}_{\mathsf{BSM}} = (\mathsf{SK}_{\mathsf{Sam}}, \mathsf{SK}_{\mathsf{Ext}})$, and not just $\mathsf{SK}_{\mathsf{Ext}}$, and therefore allows for the reuse of $\mathsf{SK}_{\mathsf{BSM}}$.

Using such robust fuzzy extractors in place of nonrobust fuzzy extractors allows us to remove the need for authenticated channels in [17]; the security argument (omitted here) is similar to the one there. Now Alice and Bob no longer need to trust that their message goes unmodified: they will (with probability $1-\delta$) detect any modification to the helper string. The price is that Alice and Bob have to additionally share a (short) extractor-MAC key SK, compute the tag $\sigma = \mathsf{Mac}_{\mathsf{SK}}(w, s, i)$, and send this (short) tag together with the rest of the information. Thus, we obtain a stateless protocol in the BSM without assuming authenticated channels, which tolerates a linear fraction of Hamming errors, requires a long-term shared secret key of size $O(\log N + \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right))$, and requires Alice and Bob to read $O(\ell)$ bits of the source, and to send a single message of size $O(\ell)$ per time period in order to extract ℓ bits that are ε-close to uniform. These parameters are optimal up to constant factors.

Acknowledgment

We thank Hoeteck Wee for his collaboration during the early stages of this work, and Raef Bassily and anonymous referees for comments on the manuscript.

References

[1] Y. Aumann, Y. Ding, and M. Rabin. Everlasting security in the bounded storage model. *IEEE Trans. Information Theory*, 48(6):1668–1680, 2002.

[2] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 276–287. IEEE, 1994.

[3] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplification. *IEEE Trans. Information Theory*, 41(6):1915–1923, 1995.

[4] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska. Practical quantum oblivious transfer. In *Advances in Cryptology — Crypto '91*, volume 576 of *LNCS*, pages 351–366. Springer, 1992.

[5] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[6] J. Bierbrauer, T. Johansson, G. Kabatianskii, and B. Smeets. On families of hash functions via geometric codes and concatenation. In *Advances in Cryptology — Crypto '93*, volume 773 of *LNCS*, pages 331–342. Springer, 1994.

[7] N. J. Bouman and S. Fehr. Secure authentication from a weak key, without leaking information. *Advances in Cryptology — Eurocrypt 2011*, volume 6632 of *LNCS*, pages 246–265. Springer, 2011.

[8] X. Boyen. Reusable cryptographic fuzzy extractors. In *11th ACM Conf. on Computer and Communications Security*, pages 82–91. ACM Press, 2004.

[9] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 147–163. Springer, 2005.

[10] L. Carter and M. N. Wegman. Universal classes of hash functions. *J. Computer and System Sciences*, 18(2):143–154, 1979.

[11] N. Chandran, B. Kanukurthi, R. Ostrovsky, and L. Reyzin. Privacy amplification with asymptotically optimal entropy loss. In *42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 785–794. ACM Press, 2010.

[12] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *Advances in Cryptology — Eurocrypt 2008*, volume 4965 of *LNCS*, pages 471–488. Springer, 2008.

[13] B. den Boer. A Simple and Key-Economical Unconditional Authentication Scheme. *Journal of Computer Security*, 2:65–72, 1993.

[14] Y. Z. Ding. Error correction in the bounded storage model. In *2nd Theory of Cryptography Conference — TCC 2005*, volume 3378 of *LNCS*, pages 578–599. Springer, 2005.

[15] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In *Advances in Cryptology — Crypto 2006*, volume 4117 of *LNCS*, pages 232–250. Springer, 2006.

[16] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[17] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 654–663. ACM Press, 2005.

[18] Y. Dodis and J. Spencer. On the (non)universality of the one-time pad. In *43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 376–387. IEEE, 2002.

[19] Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 601–610. ACM Press, 2009.

[20] N. Frykholm and A. Juels. Error-tolerant password recovery. In *8th ACM Conf. on Computer and Communications Security*, pages 1–9. ACM Press, 2001.

[21] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[22] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *6th ACM Conf. on Computer and Communications Security*, pages 28–36. ACM Press, 1999.

[23] B. Kanukurthi and L. Reyzin. An improved robust fuzzy extractor. In *6th Intl. Conf. on Security and Cryptography for Networks (SCN)*, volume 5229 of *LNCS*, pages 156–171. Springer, 2008.

[24] B. Kanukurthi and L. Reyzin. Key agreement from close secrets over unsecured channels. In *Advances in Cryptology — Eurocrypt 2009*, volume 5479 of *LNCS*, pages 206–223. Springer, 2009.

[25] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Elsevier Science, 1977.

[26] U. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Information Theory*, 39(3):733–742, 1993.

[27] U. Maurer and S. Wolf. Secret-key agreement over unauthenticated public channels III: Privacy amplification. *IEEE Trans. Information Theory*, 49(4):839–851, 2003.

[28] U. M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[29] U. M. Maurer. Information-theoretically secure secret-key agreement by NOT authenticated public discussion. In *Advances in Cryptology — Eurocrypt '97*, volume 1233 of *LNCS*, pages 209–225. Springer, 1997.

[30] U. M. Maurer and S. Wolf. Privacy amplification secure against active adversaries. In *Advances in Cryptology — Crypto '97*, volume 1294 of *LNCS*, pages 307–321. Springer, 1997.

[31] N. Nisan and D. Zuckerman. Randomness is linear in space. *J. Computer and System Sciences*, 52(1):43–53, 1996.

[32] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Mathematics*, 13(1):2–24, 2000.

[33] R. Renner and S. Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In *Advances in Cryptology — Crypto 2003*, volume 2729 of *LNCS*, pages 78–95. Springer, 2003.

[34] R. Renner and S. Wolf. The exact price for unconditionally secure asymmetric cryptography. In *Advances in Cryptology — Eurocrypt 2004*, volume 3027 of *LNCS*, pages 109–125. Springer, 2004.

[35] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[36] D. R. Stinson. Universal hashing and authentication codes. *Designs, Codes, and Cryptography*, 4(4):369–380, 1994.

[37] D. R. Stinson. Universal hash families and the leftover hash lemma, and applications to cryptography and computing. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 42:3–31, 2002.

[38] R. Taylor. An integrity check value algorithm for stream ciphers. In *Advances in Cryptology — Crypto '93*, volume 773 of *LNCS*, pages 40–48. Springer, 1994.

[39] S. P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, 2004.

[40] S. P. Vadhan. *Pseudorandomness*. To appear in *Foundations and Trends in Theoretical Computer Science*. Now Publishers.

[41] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. Computer and System Sciences*, 22(3):265–279, 1981.

[42] S. Wolf. Strong security against active attacks in information-theoretic secret-key agreement. In *Advances in Cryptology — Asiacrypt '98*, volume 1514 of *LNCS*, pages 405–419. Springer, 1998.

[43] A. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.

APPENDIX

We argue that the construction from Section III-A cannot extract more than the stated number of bits if post-application robustness is desired.

For post-application robustness, the concern is that R can reveal information to the adversary about σ' for a cleverly chosen i'. Here we show an adversarial strategy that does exactly this and succeeds in the post-application robustness game with probability $\delta/2$. In our attack we fix a particular (and somewhat unusual) representation of field elements. (Recall that the theorem was claimed to work for any representation of field elements, so long as addition of field elements corresponds to the exclusive-or of bit strings.) Typically, one views $GF_{2^{n-v}}$ as $GF_2[x]/(p(x))$ for some irreducible polynomial p of degree $n-v$, and represents elements as GF_2-valued vectors in the basis $(x^{n-v-1}, x^{n-v-2}, ..., x^2, x, 1)$. We will do the same, but reorder the basis elements so as to separate the even and odd powers: $(x^{n-v-1}, x^{n-v-3}, \dots, x, x^{n-v-2}, x^{n-v-4}, \dots, 1)$ (assuming, for concreteness, that $n-v$ is even). Letting x denote the field element corresponding to the polynomial x, the property of this representation we use is that the bits of the left half of any value $z \in F_{2^{n-v}}$ with last bit 0 are equal to the right half of the bits of z/x.

Recall $w = a\|b$. Suppose the distribution W on w is such that the top $n-m$ bits of b are 0 (the rest of the bits of w are uniform). Given σ and R, the adversary gets to see the top $\ell + (n-m)$ bits of ia. Therefore, the adversary knows $\ell + (n-m)$ bits from the bottom half of ia/x as long as the last bit of ia is 0, which happens with probability $1/2$. To use this knowledge, the adversary will simply ensure that the difference between σ' and σ is $[ia/\mathsf{x}]_1^v$, by letting $i' = i + i/\mathsf{x}$.

In detail, the adversarial strategy is as follows: let $i' = i + i/\mathsf{x}$; let τ consist of R concatenated with the top $n-m$ bits of σ and $\log\left(\frac{1}{\delta}\right) = v - \ell - (n-m)$ random bits, and let $\sigma' = \sigma + \tau$. The adversary wins whenever $\tau = [ia/\mathsf{x}]_1^v$, which happens with probability $2^{v-\ell-(n-m)}/2 = \delta/2$, because all but $\log\left(\frac{1}{\delta}\right)$ bits of τ are correct as long as the last bit of ia is 0.