Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets[*]

Yevgeniy Dodis [†] Jonathan Katz[‡] Leonid Reyzin[§] Adam Smith[¶]

July 22, 2010

Abstract

Consider two parties holding samples from correlated distributions W and W', respectively, that are within distance t of each other in some metric space. These parties wish to agree on a uniformly distributed secret key R by sending a single message over an insecure channel controlled by an all-powerful adversary. 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 SK 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.

Our results improve upon previous work in several respects:

- The best previous 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 min-entropy of W exceeds the *minimal possible* 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.

1 Introduction

A number of works have explored the problem of *secret key agreement based on correlated information* by which two parties holding instances of correlated random variables W and W' communicate and thereby generate a shared, secret uniformly-random key R. The random variables W and W' are not assumed to be individually uniform; this may be because of the parties' limited abilities to generated randomness, or because of the adversarial ability to observe some information correlated with W and W', thus making W and W' nonuniform from the adversary's point of view. The problem is sometimes known as "information reconciliation" (to handle differences between the

[*] An expanded and corrected version of [DKRS06]

[†] Dept. of Computer Science, New York University. dodis@cs.nyu.edu. This research was supported by NSF Grants #0515121, #0133806, #0311095.

[‡] Dept. of Computer Science, University of Maryland. jkatz@cs.umd.edu. This research was supported by NSF Trusted Computing grant #0310751 and NSF CAREER award #0447075.

[§] Dept. of Computer Science, Boston University. reyzin@cs.bu.edu. Work supported by NSF grants #0311485, #0515100, #0546614, #0831281, and #1012910.

[¶] Computer Science & Engineering Department, Pennsylvania State University. asmith@cse.psu.edu. Work done while at the Weizmann Institute of Science. Supported by the Louis L. and Anita M. Perlman Fellowship.

instances of W and W') and "privacy amplification" (to handle nonuniformity), and sometimes as "fuzzy extraction."

Early work [Wyn75, BBR88, Mau93, BBCM95] assumed that the parties could communicate over a *public* but *authenticated* channel or, equivalently, assumed a passive adversary. This assumption was relaxed in later work [Mau97, MW97, Wol98, MW03, RW03], which considered an active adversary who could modify all messages sent between the two parties.

The motivation of the above works was primarily to explore the possibility of *information-theoretic* security; however, this is not the only motivation. The problem also arises in the context of using noisy data (such as biometric information) for cryptographic purposes, even if computational security suffices. The problem also arises in the context of the *bounded storage model* (BSM) [Mau92] in the presence of errors [Din05, DS05]. We discuss each of these in turn.

Authentication using noisy data. In the case of authentication 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": In this scenario, a trusted server stores some biometric data w of a user obtained during an initial enrollment. At a later time, when the user and the server want to agree on a key R for a secure communication session, the user obtains a fresh biometric scan w' which is close, but not identical, to w; the server and the user use w and w' to agree on R.

"Key recovery": In this scenario, a user (on his own) uses 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 be 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 recovery of R (and hence enable decryption of the 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) key agreement protocol since w is no longer available at the later point in time. Note also that any solution for the second scenario is also a solution for the first.

Solutions for achieving secret key agreement using noisy data and an *authenticated* channel are known [BBR88, BBCM95, JW99, FJ01, LT03, DORS08]. Most of the existing work for an *unauthenticated* channel, such as [Mau97, MW97, Wol98, MW03, RW03], solves the problem only for two special cases: (1) when $W = W'$ and (2) when W and W' consist of (arbitrarily-many) independent realizations of the same random experiment; 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.

There has been recent progress on the general case. Renner and Wolf [RW04] were the first to demonstrate that the first scenario is solvable in principle with a multi-round protocol, but using a nonconstructive solution, which means that parties need more than polynomial time to complete the protocol (it was made constructive later in [KR09]). Boyen [Boy04] showed, in the random oracle model, how to achieve *unidirectional* authentication with noisy data, as well as a weak form of security for the second scenario (essentially, R remains secret but the user can be fooled into using an incorrect key R'). Work of Boyen et al. [BDK+05] showed two solutions. The first is noninteractive and thus applies to both scenarios, but relies on random oracles. The second is interactive, and thus cannot be used for the key recovery scenario; it relies on an underlying password-based key-exchange protocol, which means it provides *computational* rather than *information-theoretic* security; furthermore, given the current state-of-the-art for password-based key exchange [BPR00, BMP00, KOY01, GL01, GL03], it is impractical without additional assumptions, such as random oracles, ideal ciphers, or public parameters.

THE BOUNDED STORAGE MODEL AND THE "KEYED" CASE. Key agreement using correlated information arises also in the context of the *bounded storage model* (BSM) [Mau92] in the presence of errors [Din05, DS05]. In the BSM, two parties share a long-term secret key sk. In each of time period $j = 1, \ldots$, a long random string Z_j is broadcast to the parties (and observed by an adversary); the assumption is that the length of Z_j is more than what the adversary can store. The parties use sk and Z_j to generate a secret session key R_j, with $|R_j| \gg |sk|$, in each period. This process should achieve "everlasting security" [ADR02], meaning that even if sk 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 typical paradigm for achieving the above is for the parties to sample (using sk) shorter strings from the random string Z_j in each period (this way, they don't need to store Z_j, which is considered impossible even for the adversary). In the presence of transmission errors, these shorter strings will be similar, but not identical—thus, one party will have w_j and the other will have w'_j. The parties will then use sk to generate R_j. The parallels to the case of biometric authentication, as discussed earlier, should now be clear. Nevertheless, the problems are incomparable: in the case of the BSM with errors there is a stronger setup assumption (the parties share a long-term key sk), but the security requirements are more stringent, in particular because sk needs to be reusable.

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 with respect to some underlying metric. Our definitions as well as some of our results hold for arbitrary metric spaces, while others assume the Hamming or set difference metrics.

We consider only *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 SK then $\mathsf{Gen}, \mathsf{Rep}$ take this as additional input.) The basic requirements, informally, are

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

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

So far, this gives exactly a *fuzzy extractor* as defined by Dodis et al. [DORS08] (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 [BDK$^+$05] 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., $\mathsf{Rep}(w', P) = \perp$). We distinguish between the weaker *pre-application* robustness and the stronger *post-application* robustness, depending on whether the adversary has access to R when trying to modify 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 usages of R may leak information about it).

The case of no errors. Although our focus is on the case when W, W' are *close*, 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 n denote its bit-length. The best previous non-interactive solution in this setting is due to Maurer and Wolf [MW03] who show that for 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 [DS02, DW09] imply that a non-interactive solution is impossible when $m \leq n/2$. (Interactive solutions can do much better, as shown in [MW03, Section IIIC]; in fact, the length of R can get very close to m [RW03, DW09, CKOR10].)

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 as well as the length of the resulting key. Moreover, we give the first solution satisfying *post-application* robustness which still works as long as $m > n/2$ (but extracts a key that is three time shorter; an improvement in the key length was subsequently achieved in [KR08]).

Handling errors. The only previously known construction of robust fuzzy extractors [BDK$^+$05] relies on the random oracle model. (This solution is generic and applies to any metric admitting a good error-correcting code.) We (partially) resolve the main open question of [BDK$^+$05] by showing a construction of post-application robust fuzzy extractors *in the standard model*, for the case of the Hamming and set difference metrics.[1] The techniques of this paper were subsequently generalized and applied in the shared parameters setting in [CDF$^+$08].

Relying on shared secret keys. There are scenarios in which the two parties trying to derive R from w and w' already share a secret key. (Naturally, one may ask why derive R when they already have a shared secret; the answer is that the shared-secret may be long-term, while R is derived for a particular session or time period, as in the bounded storage model (BSM) described above.) For those scenarios, we define and construct a *keyed* robust fuzzy extractor (for general metrics). In the process, we introduce a new primitive, possibly of broader interested, called "extractor-MAC": a one-time information-theoretic message authentication code whose output is close to uniform.

Application to BSM with errors. Prior work focusing on the BSM with errors [Din05, DS05] showed that a constant relative Hamming distance between the samples w_j and w'_j by the two parties can be tolerated using a single-message protocol. The solution of [Din05] is stateful (i.e., the long-term key sk has to be updated between each time period) while the solution of [DS05] requires the parties to communicate over an authenticated channel (i.e., the adversary is assumed not to modify the message). Building on keyed robust fuzzy extractors, we show a *stateless* BSM solution (for the Hamming metric) using an *unauthenticated* channel. The cost of removing the authenticated channel assumption is very low.

2 Background

For strings a and b, $a\|b$ denotes their concatenation, and $|a|$ denotes the length of a. If S is a set, $x \leftarrow S$ means that x is chosen uniformly from S. If X is probability distribution, then $x \leftarrow X$ means that x is chosen according to distribution 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 an input, $A(x;\omega)$ denotes the output of A running with random coins ω, and $A(x)$ denotes the random variable $A(x;\omega)$ for uniformly sampled ω. If X is a random variable, then $A(X)$ is the random variable obtained by sampling $x \leftarrow X$ and sampling $A(x)$. All logarithms are base 2 unless noted otherwise.

[1] The conference version of the current paper [DKRS06] contained an erroneous claim about a construction for the edit distance, which proceeded by embedding edit distance into set difference through shingling (see [DORS08]). That construction did not work, essentially because the set description became more than twice the length of the input, thus violating the $m > n/2$ bound.

Min-entropy. The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr_X[x])$. Following [DORS08], define the (average) conditional min-entropy of X given Y as $\widetilde{\mathbf{H}}_\infty(X \mid Y) = -\log(\mathbf{E}_{y\leftarrow Y}(2^{-\mathbf{H}_\infty(X|Y=y)}))$ (here the expectation is taken over y for which $\Pr[Y = y]$ is nonzero). This definition is convenient for cryptographic purposes, because the probability that the adversary will predict X given Y is $2^{-\widetilde{\mathbf{H}}_\infty(X|Y)}$. It also works well with extractors, to be defined below. Finally, we will use [DORS08, Lemma 2.2], which states that $\widetilde{\mathbf{H}}_\infty(X \mid Y) \ge \mathbf{H}_\infty((X,Y)) - \lambda$, where 2^λ is the number of elements in Y (more generally, $\widetilde{\mathbf{H}}_\infty(X \mid (Y,Z) \ge \widetilde{\mathbf{H}}_\infty((X,Y) \mid Z) - \lambda)$.

Hash functions and extractors. We recall the notion of almost-universal hashing [CW79, Sti94]:

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 distinct x, x' we have $\Pr_{i\leftarrow I}[h_i(x) = h_i(x')] \le \delta$. Families with $\delta = 2^{-\ell}$ are called *universal.* $\diamondsuit$

A simple universal family [Sti94, Theorem 5.2] can be constructed by identifying I and $\{0,1\}^n$ with $\mathbb{F}(2^n)$ and defining $h_i(x)$ as the topmost ℓ bits of $i \cdot x$ (it doesn't matter how the elements of $\mathbb{F}(2^n)$ are represented as long as the representation takes n bits and addition in $\mathbb{F}(2^n)$ corresponds to exclusive-or).

Let X_1, X_2 be two probability distributions over S. Their *statistical distance* is $\mathbf{SD}(X_1, X_2) \overset{\text{def}}{=} \frac{1}{2}\sum_{s\in S} |\Pr_{X_1}[s] - \Pr_{X_2}[s]|$. 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.

Extractors [NZ96] yield a close-to-uniform string from a random variable with high min-entropy, using a uniformly random seed i as a kind of catalyst. Strong extractors are ones in which the extracted string looks random even in the presence of the seed; in particular, this means that the seed can in some cases be public and reusable. We consider only strong extractors in this paper, and thus often omit the adjective "strong."

Definition 2 For some set I let $\mathsf{Ext} : \{0,1\}^n \times I \to \{0,1\}^\ell$ be a function. It is a strong (m, ε)-extractor if for all distributions X over $\{0,1\}^n$ with $\mathbf{H}_\infty(X) \ge m$ and for the uniform distribution over I we have $\mathbf{SD}((\mathsf{Ext}(X; I), I), U_\ell \times I) \le \varepsilon$. A sample from I is called *seed.* $\diamondsuit$

We will need to strengthen the definition of extractors slightly to account for any external information E that may be correlated with X. To do so, we relax the min-entropy constraint on X to average min-entropy, and require the extracted string remain uniform even in the presence of E: that $\widetilde{\mathbf{H}}_\infty(X \mid E) \ge m$ should imply $\mathbf{SD}((\mathsf{Ext}(X; I), I, E), U_\ell \times I \times E) \le \varepsilon$. We will call such extractors average-case extractors. Note that any $(m - \log\left(\frac{1}{\varepsilon}\right), \varepsilon')$ extractor is automatically an $(m, \varepsilon + \varepsilon')$ average-case extractor, because $\Pr_{e\in E}[\mathbf{H}_\infty(X \mid e) \le m - \log\left(\frac{1}{\varepsilon}\right)] \le \varepsilon$, by Markov's inequality. However, this additional loss is not always necessary.

Indeed, the Leftover Hash Lemma, which says that almost universal hash functions are extractors, generalizes without any loss to the average case. (Multiple versions of this lemma have appeared; we use the formulation of [Sti02, Theorem 8.1], augmented by [DORS08, Lemma 2.4] for the average case; see [HILL99] and references therein for earlier formulations.)

Lemma 1 (Leftover Hash Lemma) *For $\ell, m, \varepsilon > 0$, 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, $\mathcal{H}$ is a strong (m, ε) average-case extractor if $\mathcal{H}$ is universal and $\ell \le m + 2 - 2\log\left(\frac{1}{\varepsilon}\right)$.*

This Lemma holds even when $\mathcal{H}$ is allowed to depend on the extra information E about the input X, i.e., when $\widetilde{\mathcal{H}} = \{\mathcal{H}_e\}_{e \in E}$ is a family of almost-universal families, one for each value of the extra information.

Strong universality and one-time message authentication codes (MACs).

While universal hash functions are good for randomness extraction, strongly universal hash functions [WC81, Sti94], defined below, can be used for message authentication, as well.

Definition 3 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 distinct $x \neq x', y, y'$ (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 simple strongly universal family [Sti94, Theorem 5.2] can be constructed by identifying $\{0,1\}^n$ with $\mathbb{F}(2^n)$, letting $I = \mathbb{F}(2^n) \times \{0,1\}^\ell$, viewing i as a pair (a, b) and defining $h_{a,b}(x)$ as the topmost ℓ bits of $a \cdot x$ exclusive-ored with b (it doesn't matter how the elements of $\mathbb{F}(2^n)$ are represented as long as the representation takes n bits and addition in $\mathbb{F}(2^n)$ corresponds to exclusive-or).

An almost strongly universal can be used for information-theoretic authentication of a message x using a secret key i shared in advance, by transmitting a tag $y = h_i(x)$ along with x. The property of being δ-almost strongly universal guarantees that the probability of a successful forgery after the adversary receives a tag on a message of its choice [WC81, p. 271] is at most δ [Sti94, Theorem 3.2].

Secure Sketches and Fuzzy Extractors. We start by reviewing the definitions of secure sketches and fuzzy extractors from [DORS08]. Let $\mathcal{M}$ be a metric space with distance function dis (we will generally denote by n the length of each element in $\mathcal{M}$). 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 4 An $(m, \tilde{m}, t)$-secure sketch is a pair of efficient randomized procedures $(\mathsf{SS}, \mathsf{SRec})$ s.t.:

1. The sketching procedure SS on input $w \in \mathcal{M}$ returns a bit string $s \in \{0,1\}^*$. The recovery procedure SRec takes an element $w' \in \mathcal{M}$ and $s \in \{0,1\}^*$.

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 min-entropy m, the (average) min-entropy of W conditioned on s does not decrease very much. Specifically, if $\mathbf{H}_\infty(W) \geq m$ then $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq \tilde{m}$.

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 [DORS08] (this construction appeared as a component of protocols earlier, e.g., in [BBCS91]). In it, the sketch $s = \mathsf{SS}(w)$ consists of the k-bit syndrome[2] of w with respect to some (efficiently decodable) $[n, n-k, 2t+1]$-error-correcting code C. 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 should also note that this construction extends to the set difference metric through sublinear-time encoding and decoding [DORS08].

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 its subsequent reproduction given any w' close to w.

[2]For a linear code with parity check matrix H, the syndrome of w is $wH^\top$.

Definition 5 An $(m, \ell, t, \varepsilon)$-$\mathsf{fuzzy\ extractor}$ is a pair of efficient randomized procedures $(\mathsf{Gen}, \mathsf{Rep})$ with the following properties:

1. The generation procedure Gen, on input $w \in \mathcal{M}$, outputs an extracted string $R \in \{0,1\}^\ell$ and a helper string $P \in \{0,1\}^*$. The reproduction procedure Rep takes an element $w' \in \mathcal{M}$ and a string $P \in \{0,1\}^*$ as inputs.

2. *Correctness:* If $\mathsf{dis}(w, w') \le t$ and $(R, P) \leftarrow \mathsf{Gen}(w)$, 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 even conditioned on the value of P. Formally, if $\mathbf{H}_\infty(W) \ge m$ and $(R, P) \leftarrow \mathsf{Gen}(W)$, then we have $\mathbf{SD}\left((R, P), U_\ell \times P\right) \le \varepsilon$. $\diamondsuit$

Just like with ordinary extractors, a slightly more general definition of fuzzy extractors accounts for any external information E that may be correlated with W, and requires that $\widetilde{\mathbf{H}}_\infty(W \mid E) \ge m$ should imply $\mathbf{SD}\left((R, P, E), U_\ell \times (P, E)\right) \le \varepsilon$. This is known as an $\mathsf{average\text{-}case}$ fuzzy extractor; all known constructions satisfy this more general definition.

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 [DORS08, Lemma 4.1]. In that case $P = (\mathsf{SS}(w), i)$ and $R = \mathsf{Ext}(w; i)$.

3 New Definitions: Robust Fuzzy Extractors

Fuzzy extractors, defined above, 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 $\tilde{P} \ne P$.

Boyen *et al.* [BDK+05] propose the notion of *robust* fuzzy extractors, which provides strong guarantees against such an attack. Specifically, Rep can output either a key or a special value $\perp$ ("fail"). We require that any value $\tilde{P} \ne P$ produced by the adversary given P causes $\mathsf{Rep}(w', \tilde{P})$ to output $\perp$. Modified versions of the correct public information P will therefore be detected.

We consider two variants of this idea, depending on whether Gen and Rep additionally share a (short) long-term key SK. Boyen *et al.* considered the keyless primitive; this is what we define first. Further below, we adjust the definitions to the case of a shared key.

Furthermore, we distinguish between two adversarial attacks, and thus two notions of robustness, depending on whether the adversary has access to R when trying to modify P. Indeed, if R has been used (e.g., for encryption) and the adversary can observe some effect of this use (e.g., the ciphertext) before attempting its modification of P, then the notion of robustness from Boyen *et al.*, which gives the adversary no information about R, is insufficient. Our stronger notion accounts for this by the giving the adversary access to all of 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 the original one, where R is not given to the adversary, *pre-application* robustness. We should emphasizes that pre-application robustness still suffices in scenarios where adversarial ability to modify P expires prior to any observable use of R.

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

Definition 6 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 $(R, P) = \mathsf{Gen}(w)$; let $\tilde{P} = \mathcal{A}(R, P)$ (resp., $\tilde{P} = \mathcal{A}(P)$); output "success" if $\tilde{P} \neq P$ and $\mathsf{Rep}(w', \tilde{P}) \neq \perp$. $\diamondsuit$

The definition is illustrated in Fig. 1. Note that the adversary we consider is computationally unbounded. The definition applies equally to average-case fuzzy extractors. We construct examples of keyless robust fuzzy extractors in Section 4. Observe that this definition is interesting even when $w = w'$ (i.e., in the non-fuzzy case of $t = 0$), because ordinary extractors are not usually robust.

Re-using Robust Extractors. The definition of robust extractors composes with itself in some situations. For example, a generalization of the above (used in [BDK+05]) 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 that two players (Alice and Bob) receive a sequence of correlated pairs of random variables $(W_1, W_1'), (W_2, W_2') \ldots$, such that each pair is at distance at most t and the entropy of W_i conditioned on information from other time periods $\left\{(W_j, W_j')\right\}_{j \neq i}$ is at least m (we call such a sequence (t, m)-*correlated*). Once again, a simple hybrid argument shows that Alice and Bob can agree on (essentially) random and uncorrelated keys $R_1, R_2, \ldots$, by having Alice apply Gen from a robust fuzzy extractor to each W_i and send P_i to Bob (here we need an average-case fuzzy extractor, because W_j may be correlated to W_i). Namely, after j periods the attacker's advantage at distinguishing the vector of unknown keys from random is at most $j\varepsilon$, and her probability of forging a valid message $\tilde{P}_i$ is at most δ in each period.

Thus, multiple-use security comes for free with keyless robust extractors. The situation is more delicate in the keyed case, because information about the key may be disclosed by the extractor. This problem can be overcome by a strong security requirement, as we discuss next.

Keyed Robust Fuzzy Extractors. In some scenarios, such as the bounded storage model, the parties running Gen and Rep can additionally share a short, truly random key to help them extract a (long) session key R from close variables W and W'. Syntactically, this simply means that Gen and Rep now also take an extra input SK: namely, we have $(R, P) \leftarrow \mathsf{Gen}_{\mathsf{SK}}(w)$, $R' = \mathsf{Rep}_{\mathsf{SK}}(w', P)$, and require that for any SK we have $R = R'$ whenever $\mathsf{dis}(w, w') \leq t$.

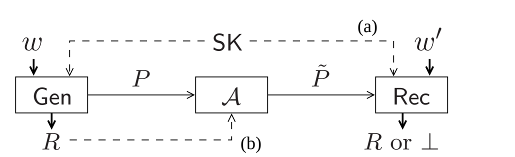

Figure 1: Robustness of extractors (Def. 6). Dotted lines indicate variations in the definition. (a) *Keyed extractors* take an additional input SK shared by Gen and Rep. (b) For *pre-application* robustness, the adversary does not have access to the extracted key R.

The robustness property of keyed fuzzy extractors (Def. 6) does not change with the addition of SK: in particular, the attacker does not get the secret key SK in the unforgeability experiment of Def. 6. At first glance, this appears to trivialize the problem of constructing keyed robust fuzzy extractors. For example, one might attempt the following transformation: use SK as a MAC key and, given an output (R, P) of a fuzzy extractor, simply append to P the tag $\mathsf{MAC}_{\mathsf{SK}}(P)$. This transformation is not sufficient, however, because we require SK to be reusable, and information-theoretic MAC keys are not. Namely, our definition implies that a single fixed-length key SK can be used to *extract an unbounded number of independent keys* $\{R_j\}$ from different (t, m)-pairs (W_j, W_j'). In fact, we require something stronger: we require that SK look uniform even given the extracted R and P (which, of course, implies that it can be reused). In particular, the above transformation will not satisfy our definition, because it leaks information on SK, making it no longer uniform given the adversary's view.

Definition 7 A keyed $(m, \ell, t, \varepsilon)$-fuzzy extractor with post-application (resp., pre-application) robustness δ is an $(m, \ell, t, \varepsilon)$-fuzzy extractor with post-application (resp., pre-application) robustness δ in which both Gen and Rep take an additional input SK (chosen uniformly from all binary strings of a given length), correctness is required to hold only if they have the same SK, robustness is required to hold only if they have the same SK which is not given to the adversary $\mathcal{A}$, and the security requirement is strengthened as follows: for any distribution W over $\mathcal{M}$ with min-entropy m, the string (SK, R) is close to a pair of fresh uniform random strings, even conditioned on the value of P: if $\mathbf{H}_\infty(W) \geq m$ and $(R, P) \leftarrow \mathsf{Gen}_{\mathsf{SK}}(W)$, then $(\mathsf{SK}, R, P) \approx_\varepsilon U_{|\mathsf{SK}|} \times U_\ell \times P$. We say that it has *uniform helper strings* if the security requirement is replaced with the following: $(\mathsf{SK}, R, P) \approx_\varepsilon U_{|\mathsf{SK}|} \times U_\ell \times U_{|P|}$. $\diamondsuit$

This definition ensures that value of SK is essentially independent of the attacker's view. This security condition is important for two reasons: first, it means that the session key R remains secure even if the long-term key SK is revealed in the future; second, the long-term key can be reused (e.g., for future authentication). If Alice and Bob are given a sequence of j (t, m)-correlated pairs (as discussed above), then $\mathcal{A}$ has advantage at most $j\varepsilon$ in distinguishing the vector of unknown session keys from random. Similarly, the adversary's probability of forging a valid $\tilde{P}_j$ in the j-th execution of the robustness experiment (Def. 6) is at most $j\varepsilon + \delta$.

Finally, we note that some settings require the more stringent security requirement of uniform helper strings from Def. 7. In this case the adversary's view hides both the long-term key SK and the exact distribution of W (since P is distributed almost independently of W). The bounded storage model, discussed in Section 5, is an example of such a setting.

4 Constructing Keyless Robust Fuzzy Extractors

Keyless robust fuzzy extractors solve the problem of secret key generation over a completely insecure channel. We begin by analyzing the case of no errors (i.e., $t = 0$) and then consider the more challenging case where errors may occur.

4.1 The Errorless Case ($w = w'$)

When the adversary is *passive* and there are no errors, than the only problem left is the nonuniformity of W; in that case, an ordinary strong extractor suffices for applications such as key agreement. Indeed, if Alice and Bob both hold the same value w of sufficient minentropy, then Alice's $\mathsf{Gen}(w)$ procedure can choose a seed i for a strong extractor and set $R = \mathsf{Ext}(w; i)$ and $P = i$. Bob's procedure $\mathsf{Rep}(w, P)$ simply outputs $R = \mathsf{Ext}(w; i)$.

Unfortunately, this solution does not work if the adversary is *active*, which is why *robust* extractors are interesting even in the errorless (nonfuzzy) case. In particular, if $i' \neq i$ there is no longer any guarantee on the output $\mathsf{Ext}(w; i')$ (and it is easy to show counterexamples where a malicious i' completely determines $\mathsf{Ext}(w; i')$ even if w is uniform). One idea is to somehow authenticate i using the extracted key R, and add the authenticator to P; in general, this does not work either, because if i changed, then R will be also changed. It turns out, however, that w itself can be used to authenticate i, at least for a particular choice of message authentication and a particular strong extractor. Details follow.

Construction. We assume the input w is in $\{0, 1\}^n$. To compute $\mathsf{Gen}(w)$, parse w as two strings a and b of lengths $n - v$ and v, respectively (the value of v will be determined later). View a as an element of $\mathbb{F}_{2^{n-v}}$ and b as an element of $\mathbb{F}_{2^v}$ (the construction is not sensitive to the choice of representation of field elements as bit strings, and works as long as addition in the field corresponds to exclusive-or of bit strings). Choose a random $i \in \mathbb{F}_{2^{n-v}}$, compute $ia \in \mathbb{F}_{2^{n-v}}$, and let $[ia]_{v+1}^{n-v}$ denote the most significant $n-2v$ bits of ia and $[ia]_1^v$ denote the remaining v bits. View $[ia]_1^v$ as an element of $\mathbb{F}_{2^v}$. Then compute $\sigma = [ia]_1^v + b$, set $P = (i, \sigma)$, and let the extracted key be $R = [ia]_{v+1}^{n-v}$.

$\mathsf{Rep}(w, \tilde{P})$, 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 $R = [i'a]_{v+1}^{n-v}$.

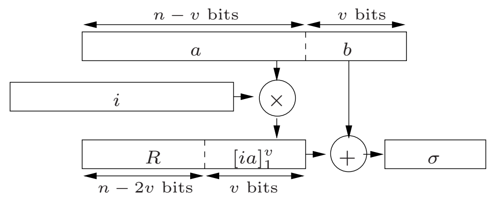

Figure 2: Construction for errorless case.

Theorem 2 *Let $\mathcal{M} = \{0,1\}^n$. Setting $v = (n-\ell)/2$, the above construction is an $(m, \ell, 0, \varepsilon)$ fuzzy extractor with robustness δ, for any $m, \ell, \varepsilon, \delta$ satisfying*

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

for pre-application robustness and

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

for post-application robustness.

To help understand the formula for ℓ, let the "entropy gap" $g = n - m$, and ignore terms in ϵ, δ. For nonrobust extraction, the most we can hope to extract is the entropy of the input $m = n - g$. Pre-application robustness imposes an additive cost of g (since $2m - n = n - 2g$), essentially because σ has to be a bit over g bits long (else it is possible that σ would be constant regardless of w and thus forgeable). Requiring post-application robustness further reduces the length of R by a factor of 3, essentially because of the need to insure that R does not leak so much information that σ becomes forgeable.

Observe that extraction is possible as long as $\mathbf{H}_\infty(W) \stackrel{\text{def}}{=} m > n/2$. Furthermore, in the case of pre-application robustness (which is the notion of security considered by Maurer and Wolf [MW03]) we extract a key of length roughly $2m - n$, thus improving on the [MW03] requirement of entropy $m > 2n/3$ and extraction length $m - 2n/3$.

Proof We first show that R is close to uniform even conditioned on P, and then argue robustness.

Extraction. We show that R is nearly uniform given $P = (i, \sigma)$. To do so, we first show that $h_i(a, b) \stackrel{\text{def}}{=} (\sigma, R)$, 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 our fields corresponds to bitwise exclusive-or). If $a = a'$ then $b \neq b'$, and thus this probability is 0. If $a \neq a'$, then there is a unique i that satisfies the equality. Thus, the probability is at most $1/|\mathbb{F}_{2^{n-v}}| = 2^{v-n}$. Applying Lemma 1, we see that the distribution of $(R, P) = (R, (i, \sigma))$ is $2^{(n-v-m)/2-1}$-close to $(U_{n-2v} \times U_{n-v} \times U_{2v})$.

We now need a simple Lemma.

Lemma 3 *If $\mathbf{SD}\left((A,B), C \times D\right) \le \alpha$, then $\mathbf{SD}\left((A,B), C \times B\right) \le 2\alpha$.*

Proof $\mathbf{SD}(C \times D, C \times B) = \mathbf{SD}(D, B) \le \mathbf{SD}(C \times D, (A, B)) \le \alpha$. The claim follows by the triangle inequality applied to $\mathbf{SD}((A, B), C \times D)$ and $\mathbf{SD}(C \times D, C \times B)$. ■

Applying this lemma to $A = R$, $B = P$, $C = U_{n-2v}$, $D = U_{n-v} \times U_{2v}$, we get that (R, P) is ε-close to $U_{n-2v} \times P$, for $\varepsilon = 2^{(n-v-m)/2}$.

Pre-application robustness. Assume $\mathcal{A}$ is deterministic; since we allow an unbounded adversary, this is without loss of generality. Fix arbitrary i known to $\mathcal{A}$ (this makes the result stronger, since we will show that robustness holds for worst-case choice of i). Upon observing σ, the adversary outputs $\mathcal{A}(\sigma) = (i', \sigma') \neq (i, \sigma)$. Note that if $i' = i$, then Rep will reject unless $\sigma' = \sigma$; so, we need only consider the case $i' \neq i$. By construction, $\mathcal{A}$ succeeds if $\sigma' = [i'a]_1^v + b$.

Call the triple (σ, i', σ') a *transcript* and denote it by tr. Call a transcript *possible* if $\mathcal{A}(\sigma) = (i', \sigma')$. Note that the number of possible transcripts is at most $2^{|\sigma|} = 2^v$. Let Succ be the event that $\mathcal{A}$ succeeds. Note that, given our conventions above, this now depends only on the choice of $w = a\|b$; for each w, we can evaluate whether the adversary will succeed by summing over all possible transcripts (this sum will be either 0 or 1). In the formulas below and let $[\![\textit{expression}]\!]$ equal 1 if *expression* is true, and 0 otherwise. We have:

$$\begin{aligned}
\Pr[\mathsf{Succ}] &\le \sum_{a,b} \Pr_W[a\|b] \sum_{\mathsf{tr}} [\![\, [ia]_1^v + b = \sigma \wedge \mathcal{A}(\sigma) = (i', \sigma') \wedge [i'a]_1^v + b = \sigma' \,]\!] \\
&= \sum_{\mathsf{tr}} [\![\, \mathcal{A}(\sigma) = (i', \sigma') \,]\!] \sum_{a,b} \Pr_W[a\|b] \cdot [\![\, [ia]_1^v + b = \sigma \wedge [i'a]_1^v + b = \sigma' \,]\!] \\
&= \sum_{\text{possible}\,\mathsf{tr}} \Pr_{a\|b \leftarrow W}\left[[ia]_1^v - [i'a]_1^v = \sigma - \sigma' \wedge b = \sigma - [ia]_1^v \right].
\end{aligned}$$

Now, $[ia]_1^v - [i'a]_1^v = [ia - i'a]_1^v = [a(i - i')]_1^v$ (again, this is because we insisted that addition/subtraction in our fields corresponds to bitwise exclusive-or). For fixed tr, the term $a(i - i')$ takes on each possible value in $\mathbb{F}_{2^{n-v}}$ exactly once as a varies; therefore, there are 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 $b = \sigma - [ia]_1^v$. Since the min-entropy of W is at least m, each such (a, b) pair occurs with probability at most 2^{-m} and so $\Pr_{a\|b \leftarrow W}[[ia]_1^v - [i'a]_1^v = \sigma - \sigma' \wedge b = \sigma - [ia]_1^v] \le 2^{n-2v-m}$. We conclude that $\Pr[\mathsf{Succ}] \le \sum_{\text{possible}\,\mathsf{tr}} 2^{n-2v-m} \le 2^{n-v-m}$.

Setting v. If we want to achieve pre-application robustness δ and an extracted key that is ε-close to uniform, we obtain the following constraints on v:

- $v < n/2$ (because the construction achieves $|R| = n - 2v > 0$);
- $v \ge n - m + \log\left(\frac{1}{\delta}\right)$ (to achieve $2^{n-v-m} \le \delta$); and
- $v \ge n - m + 2\log\left(\frac{1}{\varepsilon}\right)$ (to achieve $2^{(n-v-m)/2} \le \varepsilon$).

We extract $n - 2v$ bits, so we'd like to set v as small as possible to extract the largest possible key. Thus, we set $v = n - m + \max\left(\log\left(\frac{1}{\delta}\right), 2\log\left(\frac{1}{\varepsilon}\right)\right)$ to extract

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

bits, and extraction is possible as long as $m > n/2$.

Post-Application Robustness. Because $|R| = n - 2v$ bits, providing R to the adversary can increase the adversary's success probability by a multiplicative factor of at most 2^{n-2v} as compared to pre-application robustness (indeed, if the increase was greater, then the adversary could break pre-application robustness with higher probability by simply guessing the value of R).

Thus, if we want robustness δ and our extracted key to be ε-close to uniform, we obtain the following constraints on v:

- $v < n/2$ (as before);
- $v \geq (2n - m + \log\left(\frac{1}{\delta}\right))/3$ (to achieve $2^{2n-m-3v} \leq \delta$); and
- $v \geq n - m + 2\log\left(\frac{1}{\varepsilon}\right)$ (as before).

If $\delta < \varepsilon^6 2^{2m-n}$, setting $v = (2n - m + \log\left(\frac{1}{\delta}\right))/3$ allows us to extract

$$\ell = \frac{2m - n - 2\log\left(\frac{1}{\delta}\right)}{3}$$

bits, or about a third of what was possible if we required only pre-application robustness. If $\delta \geq \varepsilon^6 2^{2m-n}$, we can extract as many bits as for pre-application robustness, because the uniformity constraint dominates the robustness constraint. Again, extraction is possible as long as $m > n/2$. ■

4.2 Authenticating a Message While Extracting

The above construction uses the input w to authenticate the extractor seed i. It can be extended to additionally authenticate a (bounded-length) message M; i.e., to be simultaneously a robust fuzzy extractor and an information-theoretic one-time MAC. In this case, 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.) Naturally, this could be done easily 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 adapt a standard technique [BJKS93, dB93, Tay93] for authenticating long messages using polynomial-based almost universal hash functions.

Construction. Assume $|M| = L \cdot (n - v)$, where L is known to all 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 $M(x) \in \mathbb{F}_{2^{n-v}}[x]$ of degree $L - 1$. Modify the construction of the previous section as follows: to compute $\mathsf{Gen}(w, M)$, parse w as $a\|b$, choose random $i \in \mathbb{F}_{2^{n-v}}$, compute $\sigma = [a^2M(a) + ia]_1^v + b$, 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 $|M'| = L(n - v)$ and that $\sigma' = [a^2M'(a) + i'a]_1^v + b$. If so, it computes $R = [i'a]_{v+1}^{n-v}$.

Analysis. Extraction and robustness (which here implies that neither i nor M can be modified without being detected) are proved in a manner very similar to the proof of Theorem 2.

Fix arbitrary M, known to the adversary. To argue that R is nearly uniform given $P = (i, \sigma)$, we will show that the function $h_i(a, b) \stackrel{\text{def}}{=} (\sigma, R)$ is universal. Indeed, for $(a, b) \neq (a', b')$, $\Pr_i\left[h_i(a, b) = h_i(a', b')\right] = \Pr_i\left[i(a - a') = 0^{n-2v}\|y\right]$, where $y \in \{0,1\}^v$ is equal to $[(a')^2M(a') - a^2M(a)]_1^v + b' - b$. If $a = a'$ then $b \neq b'$; hence $y \neq 0$ and the equality cannot be satisfied. If $a \neq a'$, then there is a unique i that satisfies this equality. The rest of the proof proceeds as before.

For 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

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

The crucial property is that the polynomial $x^2M(x)+ix-x^2M'(x)-i'x$ 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). As before, then, the adversary's success (in changing either i or M without being detected) is at most $(L+1)\cdot 2^{n-v-m}$.

The optimal setting of v can be computed as before. We remark that the resulting parameters are affected only by an addition of $\log(L+1)$ to $\log\left(\frac{1}{\delta}\right)$; in particular, since L can be expected to be much less than $1/\delta$, this impact is small.

4.3 Adding Error-Tolerance

We now consider settings when the input w' to Rep is close, but not identical to, the value w used in Gen. An obvious first attempt, given the scheme just discussed, is to include a secure sketch $s = \mathsf{SS}(w)$ along with (i,σ), and to authenticate s using the message authentication technique discussed previously; 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 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 used by the sender (and so in particular we can no longer claim that the adversary could not have modified s without being detected). 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 used 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* (sometimes called security against related-key attacks). Specifically, we will assume that the distance between w and w' is small in the Hamming metric, and that we are given a deterministic, linear secure sketch (for example, the syndrome construction described following Def. 4). In Section 4.3.2 we will 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 entropy m of the input decreases, but also when the entropy gap $g = n-m$ increases (for example, Theorem 2 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 will decrease, and the gap will increase. An important idea is to decrease the gap by using the (shorter) part of w that is independent of s.

4.3.1 Construction for Binary Hamming Errors

Suppose the input W is a distribution of min-entropy m on n-bit string. Our starting point is a *deterministic, linear* secure sketch $s = \mathsf{SS}(w)$ that is k bits long; let $n' = n-k$; note that $\tilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq m-k$. 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 an $k \times n$ matrix S of rank k such that $\mathsf{SS}(w) = Sw$ (see footnote 2). 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.

Gen(w):

1. Set $s = \mathsf{SS}(w),\ c = \mathsf{SS}^{\perp}(w),\ k = |s|,\ n' = |c|$.
   - Parse c as $a\|b$ with $|a| = n' - v$ and $|b| = v$.
   - Let $L = 2\lceil \frac{k}{2(n'-v)} \rceil$. Pad s with 0s to length $L(n'-v)$.
     Parse the padded s as $s_{L-1}\|s_{L-2}\|\ldots\|s_0$, for $s_i \in GF(2^{n'-v})$.
2. Select $i \leftarrow GF(2^{n'-v})$.
   - Define $f_{s,i}(x) = x^{L+3} + x^2(s_{L-1}x^{L-1} + s_{L-2}x^{L-2} + \cdots + s_0) + ix$.
3. Set $\sigma = [f_{s,i}(a)]_1^v + b$ and output $R = [ia]_{v+1}^{n'-v}$ and $P = (s, i, \sigma)$.

To reproduce R given w' and $\tilde{P} = (s', i', \sigma')$, first compute $w^* = \mathsf{SRec}(w', s')$; make sure $w^* \in \{0,1\}^n$, $\mathsf{SS}(w^*) = s'$ and $\mathsf{dis}(w^*, w') \leq t$ (these conditions are not guaranteed for arbitrary inputs by the definition of SRec); if not, output $\perp$. Let $c' = \mathsf{SS}^{\perp}(w^*)$; parse c' as $a'\|b'$. Compute σ^* as above using s', a', b', i', and check that this matches the value σ' received. If so, output $R = [i'a']_{v+1}^{n'-v}$, else output $\perp$.

Note that Rep and Gen have to pre-agree on k (the length of s) and $n' = n - k$. We view these values as being fixed. It does not matter at what end s is padded with zeroes as long Gen and Rep do it the same way.

Analysis. 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 $((L+3) \bmod 2)\Delta_a x^{L+2}$). In our analysis, we use the linearity of the scheme to understand the offset $\Delta_a = a' - a$, and conclude that the adversary succeeds only if it can guess the last v bits of $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$, which happens with low probability. Note that this definition of $f_{s,i}$ amounts to a message authentication code (MAC) provably secure against a class of related key attacks where the adversary can force the receiver to use a key shifted by an offset of the adversary's choice. We obtain:

Theorem 4 *Let $\mathcal{M} = \{0,1\}^n$ with the Hamming metric. Assume SS is a linear $(m, m-k, t)$-secure sketch for $\mathcal{M}$. Setting $v = (n - \ell - k)/2$, the above construction is an $(m, \ell, t, \varepsilon)$ fuzzy extractor with robustness δ, for any $m, \ell, \varepsilon, \delta$ satisfying*

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

for pre-application robustness and

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

for post-application robustness, where b is the logarithm of the volume of the Hamming ball of radius t in $\{0,1\}^n$. (For $t \leq n/2$, $b \leq nH_2(t/n) = t\log\frac{n}{t} + (n-t)\log\frac{n}{n-t}$.) The exact bounds can be improved somewhat; see Theorem 10.

The proof of this theorem, along with a more detailed statement, is given in Appendix A. We briefly discuss the parameters in the statement. The bound on ℓ differs in two large terms from the bound in the errorless case of Theorem 2. First, we lose the length of the sketch k. This is not surprising, since we need to publish the sketch in order to correct errors.[3] Second, we lose twice the logarithm of the volume of the Hamming ball of radius t. This is due to our analysis of robustness, which essentially starts by giving the attacker the error pattern $\Delta = w' \oplus w$ "for free." This in the worst case can reduce the min-entropy w by the logarithm of volume of the Hamming ball of radius t. Our analysis can, in fact, yield a more general result: if $\hat{m} = \widetilde{\mathbf{H}}_\infty(W \mid \Delta)$, then $2(m-b)$ in the above bounds on ℓ simply gets replaced with $2\hat{m}$. For instance, when knowing the error pattern $w' \oplus w$ does not reduce the entropy of w (say, the errors are independent of w, as in the work of Boyen [Boy04]), then the term b disappears from the bounds.

The analysis gives away Δ since we can then use the linearity of the sketch to conclude that the adversary knows the difference between the original input w and the value w^* that $\mathsf{Rep}(w', \tilde{P})$ reconstructs (see the proof of Theorem 10). This means the adversary knows $\Delta_a = a' - a$, and we can use the properties of $f_{s,i}$ to bound the forgery probability.

4.3.2 Extension to Set Difference Metric

The construction from the previous section relies heavily on the linearity of the secure sketch used in the protocol and on the structure of the Hamming space. Using the techniques from [DORS08], however, it can be extended to set difference.

In the *set difference* metric, each element of $\mathcal{M}$ is a set of at most r elements chosen from a large universe of N elements; the distance between two such sets is the size of their symmetric difference: $\mathsf{dis}(a, b) = |\{x : x \in a \cup b \text{ and } x \notin a \cap b\}|$. This is geometrically identical to the Hamming metric, since we can represent sets as characteristic vectors in $\{0,1\}^N$ (by ordering the N-element universe and placing a bit into every location indicating whether it is in the sent). However, the efficiency requirement is much stricter: for set difference, we require that operations take time polynomial in the description length of the inputs, which is only $r \log N$, not N. (To be precise, sets of size *exactly* r take $r \log N$ bits to describe; sets of size *up to* r can be described in $r \log(N+1)$ bits by adding a special character for "absent").

In order to extend the analysis of the previous section to handle this different representation of the input, we need a pair of functions $\mathsf{SS}(), \mathsf{SS}^\perp()$ that take sets and output bit strings of length k and $r\log(N+1) - k$, respectively. A set w of size up to r should be unique given the pair $(\mathsf{SS}(w), \mathsf{SS}^\perp(w))$, and the functions should possess the following linearity property: the addition or removal of a particular element in the set should correspond to adding a particular bit vector to the output. 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. [DORS08, 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 build $\mathsf{SS}^\perp$ for it. For the PinSketch construction the universe must be viewed as nonzero elements of a binary field $\mathbb{F}_{2^\alpha}$ for some α and thus $N = 2^\alpha - 1$.

The construction of Gen and Rep remains the same as in the previous section, with the exception of different SS, SRec, and $\mathsf{SS}^\perp$ functions. In addition, Rep needs to replace the check that $w^* \in \{0,1\}^n$ with the check that the recovered w^* is a set with elements in $\mathbb{F}^*_{2^\alpha}$ (note, however, that it is not necessary to check the sizes of w' and w^*; the construction works correctly even if w' has more than r elements, as long as $\mathsf{dis}(w, w') \le t$).

[3] In fact, a more naive construction would lose $2k$, since the sketch reduces the min-entropy m by k, and blindly applying the errorless bound $2m - n$ would double this loss. The use of $\mathsf{SS}^\perp$ is precisely what allows us not to lose the value k twice.

The analysis is the same as in the previous section. The volume of the ball of radius t remains the same as in the binary Hamming case; however, since N is large compared to t, $t\log(N+1)$ is a good approximation for it (it is an upper bound because every point in the ball can be described by up to t values), and we can use it to simplify the formulas. Replacing both k and b with $t\log(N+1) = t\alpha$, and n with $r\log(N+1) = r\alpha$, we obtain the following statement.

Corollary 5 *Let $\mathcal{M}$ be the set difference metric on sets of size up to r over the universe $\mathbb{F}^*_{2^\alpha}$. The above construction is an $(m, \ell, t, \varepsilon)$ fuzzy extractor with robustness δ, for any $m, \ell, \varepsilon, \delta$ satisfying*

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

for pre-application robustness and

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

for post-application robustness. The exact bounds can be improved somewhat; see Theorem 10.

We need to describe $\mathsf{SS}^\perp$. To compute $\mathsf{SS}(w)$ and $\mathsf{SS}^\perp(w)$ on input $w \subseteq \mathbb{F}^*_{2^\alpha}$, let $s_i \stackrel{\text{def}}{=} \sum_{x\in w} x^i$ (computations in $\mathbb{F}_{2^\alpha}$) 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 $\mathbb{F}_{2^\alpha}$. Moreover, together they contain all the entropy of the original input w, since given $s_1, ..., s_{2r-1}$ one can recover w. (Simply observe that $(\mathsf{SS}(w), \mathsf{SS}^\perp(w))$ is the syndrome, with respect to the binary BCH code of distance $2r+1$, of the characteristic vector of w, and that the weight of this vector is at most r. For more details, see [DORS08, Lemma 6.2], setting $n = 2^\alpha - 1$, $k = n - r\alpha$ and $\delta = 2r+1$.) Finally, the pair $(\mathsf{SS}, \mathsf{SS}^\perp)$ has the desired linearity property, since adding or removing an element y from w corresponds to adding x^i each component s_i (and we require addition in binary fields to correspond to bitwise exclusive-or).

5 Keyed Robust Fuzzy Extractors and Their Applications

In this section we show that the addition of a very short secret key SK allows us to achieve considerably better parameters when constructing *keyed* robust fuzzy extractors. The parameters are optimal up to constant factors.

To motivate our construction, let us recall the naive transformation from fuzzy extractors to keyed robust fuzzy extractors discussed in Section 3. Suppose we start from the generic construction of a fuzzy extractor from [DORS08, Lemma 4.1]: $P = (s, i)$, $R = \mathsf{Ext}(w; i)$, where $s \leftarrow \mathsf{SS}(w)$ for a secure sketch SS, Ext is an average-case extractor, and i is its seed. In an attempt to make this construction robust, we set $\mathsf{SK} = \mu$ and $\sigma = \mathsf{MAC}_\mu(s, i)$, and redefine P to also include the tag σ. The problem is that the value σ allows the attacker to distinguish the real key μ from a random key $U_{|\mu|}$, since the attacker knows the authenticated message (s, i). Thus this scheme fails to meet the security requirement of Def. 7.

We can change the scheme to avoid this. First, note that Rep must recover the input $w = \mathsf{Rec}(w', s)$ before computing R. Thus, we can add w to the authenticated message: that is, set $\sigma = \mathsf{MAC}_\mu(w, s, i)$. Rep will be able to verify it after recovering w. This does not strengthen the robustness property, which was already satisfied by the naive scheme. However, it does help satisfy security. In the naive scheme the attacker $\mathcal{A}$ *knows the message* (s, i) *we are authenticating*. In contrast, W has high entropy from $\mathcal{A}$'s point of view, even given $\mathsf{SS}(W)$ and R (for appropriate parameters). Thus, to make the pair (P, R) independent of $\mathsf{SK} = \mu$, it suffices to construct information-theoretic MACs whose key μ looks independent from the tag, as long as the authenticated message has high min-entropy. In other words, if we can ensure that the MAC is simultaneously a strong randomness extractor (where μ is the seed), we can solve our problem, because in strong extractors the seed and the extracted output look jointly uniform.

5.1 Extractor-MACs

Definition 8 A family of functions $\left\{\mathsf{MAC}_\mu : \{0,1\}^{\tilde{n}} \to \{0,1\}^v\right\}$ is a strong $(\tilde{m}, \varepsilon, \delta)$ (average-case) extractor-MAC if it is δ-almost strongly universal (Def. 3) and a $(\tilde{m}, \varepsilon)$ (average-case) strong extractor (Def. 2, where the key μ is the seed). $\diamondsuit$

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 minimal min-entropy threshold $\tilde{m}$.

Our construction of extractor-MACs follows from the observation that almost strongly universal hash functions are MACs and, as universal hash functions, also extractors by the leftover hash lemma Lemma 1. (In fact, this observation was used to get extractors with short seeds in [SZ99, Section 3].) Any family of almost strongly universal hash functions will work. We exemplify our construction with the family constructed in [BJKS93, Section 4]. Specifically, we compose two hash families, as follows. Let $\{p_\beta\}$ be a $(\delta\varepsilon^2/2)$-almost universal hash family mapping $\tilde{n}$ bits to u bits (for some u to be determined later), and let $\{f_\alpha\}$ be a strongly universal hash family mapping u to v bits, 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 [Sti94, Theorem 5.5], $\{MAC_{\alpha,\beta}\}$ is a δ-almost strongly universal hash family, since $\delta\varepsilon^2/2 + 2^{-v} \leq \delta$. This means it can be used for message authentication. And by [Sti94, 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. This means, by the leftover hash lemma (Lemma 1), that it is a (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 [Sti94, Theorem 5.2] (described following Def. 3 above) 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 can use the polynomial-based construction of [BJKS93, dB93, Tay93], also used in previous sections. The key β is a point in $\mathbb{F}_{2^u}$, 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 $\mathbb{F}_{2^u}$. Now, set $p_\beta(x_0 \ldots x_c) = x_{c-1}\beta^{c-1} + \ldots + x_1\beta + x_0$. 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 6 *For any δ, ε and $\tilde{m} \geq \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$, there exists a $(\tilde{m}, \varepsilon, \delta)$-extractor-MAC with key length $\kappa = 3 + 2\log\tilde{n} + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right)$ and tag length $v = \log\left(\frac{1}{\delta}\right) + 1$.*

This construction has both short keys and short tags (linear in the security parameters $\log\left(\frac{1}{\delta}\right)$, $\log\left(\frac{1}{\varepsilon}\right)$). One can reuse the key μ as long as min-entropy of the 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 (one can't obtain δ-almost strong universality with tags shorter than $\log\left(\frac{1}{\delta}\right)$). This implies, from bounds on extractors [RT00, Theorem 1.9] (the bounds need to be reinterpreted for strong extractors considered here, by viewing the seed as part of the extractor output), that the key length is optimal up to a constant factor and the entropy threshold is optimal up an additive constant.

5.2 Building Keyed Robust Fuzzy Extractors

We now apply the extractor-MACs to build keyed robust fuzzy extractors for $\mathcal{M}$ (which we assumed for simplicity is $\{0,1\}^n$). We start with a generic construction and set the parameters below.

Assume $(\mathsf{SS}, \mathsf{SRec})$ is a $(m, \tilde{m}, t)$-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 μ of length κ. We now define a keyed robust fuzzy extractor with secret key $\mathsf{SK} = \mu$:

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

We note that a variant of this construction (whose security is proven analogously) would move the extractor seed i into the secret key SK. Namely, set $\mathsf{SK} = (\mu, i)$, $\sigma = \mathsf{MAC}_\mu(w, S)$ and $P = (S, \sigma)$. The main advantage of this variant is that the scheme becomes non-interactive in the case of no errors (i.e., $t = 0$) or requires only a short message for small t. However, in order to keep the length of SK low one must choose the extractor Ext carefully to make sure it has short seeds.

Theorem 7 *The above construction is a $(m, \ell, t, 4\varepsilon)$-keyed fuzzy extractor with post-application robustness δ, which uses a secret key SK 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,

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

Indeed,

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

because $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq \tilde{m}$, and Ext used to extract R 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,

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

because $\widetilde{\mathbf{H}}_\infty(W \mid R, s, i) \geq \widetilde{\mathbf{H}}_\infty(W, R, i \mid s) - \ell - d \geq \widetilde{\mathbf{H}}_\infty(W, i \mid \mathsf{SS}(W)) - \ell - d \geq \widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) + d - \ell - d = \tilde{m} - \ell$ (the first inequality follows from [DORS08, Lemma 2.2], as described in Section 2 above, and the last inequality follows by independence of i), and MAC is a $(\tilde{m} - \ell, \varepsilon)$ average-case extractor.

By triangle inequality, therefore, we obtain

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

Finally, applying Lemma 3 (here $A = (\mu, R)$, $B = P = (s, i, \sigma)$, $C = U_{|\mathsf{SK}|} \times U_\ell$, $D = \mathsf{SS}(W) \times U_d \times U_v$), we obtain the desired result.

To show robustness, suppose $\mathcal{A}$ gives $\tilde{P} = (s', i', \sigma')$ different from $P = (s, i, \sigma)$. 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 $\mathsf{MAC}_\mu(w^*, s, i) = \sigma$. Therefore, unless $\sigma' = \sigma$ and $\tilde{P} = P$, Rep will output $\perp$. Now consider the case when $(s, i) \neq (s', i')$. Then, in order for Rep not to reject, $\mathcal{A}$ must correctly guess the tag of a new message with a uniformly chosen key μ, which cannot be done with probability higher than δ by the δ-almost strong universality of MAC. Note that this shows 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 7 to the original (non-robust, non-keyed) constructions of [DORS08]. 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 6 for MAC. Then the secret key SK is the just the MAC key μ, whose length will be $2\log n + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right) + O(1)$, as long as $d = O(n)$ and $k = O(n)$ (which is the case with typical extractor and secure sketch constructions), so that $\tilde{n} = O(n)$. In order for the extractor-MAC of Theorem 6 to work, we need $\tilde{m} - \ell \ge \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$, or $\ell \le \tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right) - \log\left(\frac{1}{\delta}\right)$. This means that the 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 [DORS08, Lemma 4.3]. Finally, the length of P increases increases only by the tag length $v = \log\left(\frac{1}{\delta}\right) + 1$.

5.3 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 with errors. To explain this result, we first briefly recall the key elements of the bounded storage model (BSM) [Mau92] with errors [Din05, DS05], concentrating only on the *stateless variant* of [DS05]. Our discussion will be specific to *Hamming errors.*

In the bounded storage model with errors, the parties (say, Alice and Bob) have a long-term secret key sk, and at each time period j have access to two noisy versions X_j and X'_j of a very long random string Z_j (of length N). We assume that noise is Hamming noise, i.e., there is 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, so Z_j has average min-entropy $(1-\gamma)N$ from its point of view by [DORS08, Lemma 2.2]. 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 something called *everlasting* security [ADR02]).

Prior work [Din05, DS05] assumed that the communication channel between Alice and Bob is 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. In this work, we remove the need for the authenticated channel.

The basic idea underlying prior work is to use sk to extract a short-term key R_j from X_j and X_j' that will be unknown to $\mathcal{A}$. For example, in "sample-and-extract" protocols [Vad04], one part of sk consists of a key sam for an *oblivious sampler* [BR94, Vad04]. The key sam specifies n secret locations in the N-bit string X_j (respectively, X_j') which Alice (respectively, Bob) will read, obtaining n-bit substring w_j (respectively, 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'. Fuzzy extractors can be used to the derive R_j from w_j and w_j', with Alice running $\mathsf{Gen}(w_j)$ to obtain (P_j, R_j), sending the helper string P_j to Bob over the authenticated channel, and Bob running $\mathsf{Rep}(w_j', P)$ to get R_j [Din05, DS05].

In order to remove the need for an authenticated channel, Alice and Bob can use a keyed robust fuzzy extractor instead, storing its secret key SK as part of their long-term secret key sk. However, there is a subtle problem here, which already caused difficulties even in the case of authenticated channels and nonrobust extractors [Din05, DS05].

The problem is in the reuse of key sk. Even though robust fuzzy extractors and their keyed variants can be reused, as discussed Section 3, they can be reused only when the distribution is guaranteed to have sufficient entropy (from the adversary's point of view). In the current setting, however, $\mathcal{A}$ can use information gleaned from P_j in order to reduce the entropy of w_{j+1} (conditioned on $\mathcal{A}$'s storage), because $\mathcal{A}$ can adaptively decide what information about Z_{j+1} to store and choose to store information that is related to w_{j+1}. This can happen if P_j reveals something about the sampler key sam, thus enabling $\mathcal{A}$ to determine which positions in X_{j+1} Alice will read in order to create w_{j+1}.

Even though our definition of keyed robust extractors (Def. 7) is strong enough to ensure that the public value P_j is statistically independent of the key SK (which means that SK can be reused), it still allows the value P_j to depend on the w_j (which, in turn, depends on the sampling key sam), thus making it unsafe to reuse sam. We can solve this problem by using keyed robust fuzzy extractors with uniform helper strings (also defined in Def. 7). Using them will ensure the public value P not only hides the secret key SK, but even *the distribution* of w_j: $(\mathsf{SK}, R, P_j) \approx_\varepsilon U_{|\mathsf{SK}|} \times U_\ell \times U_{|P_j|}$. This will be sufficient for our setting by an argument (omitted here) similar to the authenticated case considered in [DS05].

Keyed Robust Extractors with Uniform Helper Strings. Examining the keyed construction in Theorem 7, 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 at random, and the value σ is close to uniformly random (even conditioned on i, s, w, and SK) 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 enough min-entropy: $\mathsf{SS}(W) \approx_\varepsilon U_{|\mathsf{SS}(W)|}$ (note that such sketches cannot be deterministic, because a deterministic SS would give half of the w values the same first bit). Such sketches were studied by Dodis and Smith [DS05], since they were already needed to solve the noisy BSM problem even in the authenticated channel case. In particular, [DS05] built such sketches binary Hamming metric with parameters that are only a constant factor worse than those of regular sketches.

Theorem 8 ([DS05, Theorem 1]) *For any min-entropy $m = \Omega(n)$, there exists efficient $(m, \tilde{m}, t)$-secure sketches for the Hamming metric over $\{0,1\}^n$ that are also (m, ε)-extractors, 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 5.2 gives us the following theorem.

Theorem 9 *Using sketches of Theorem 8 in the construction of Section 5.2 gives a $(m, \ell, t, 3\varepsilon)$-keyed fuzzy extractor with uniform helper strings and post-application robustness δ.*

Proof Correctness and unforgeability are shown the same way as in Theorem 7. To show security (that is, extraction) with uniform helper strings, we need to argue that for any W of min-entropy m, we have

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

or, equivalently,

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

Indeed,

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

for the same reason as in Theorem 7. On the other hand, $\mathsf{SS}(W) \approx_\varepsilon U_k$ by Theorem 8 and therefore

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

which, by triangle inequality, implies

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

The rest of the proof proceeds the same way as in Theorem 7, except the application of Lemma 3 is not needed. ■

Application to the BSM. Using the construction of Theorem 9 instead of the nonrobust fuzzy extractor allows us to remove the need for authenticated channels in [DS05]. 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 pass 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.

Acknowledgments. We would like to thank Hoeteck Wee for his collaboration at the early stages of this work.

References

[ADR02] Y. Aumann, Y. Ding, and M. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[BBCS91] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In J. Feigenbaum, editor, *Advances in Cryptology—CRYPTO '91*, volume 576 of *LNCS*, pages 351–366. Springer-Verlag, 1992, 11–15 August 1991.

[BBR88] Charles Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 147–163. Springer-Verlag, 2005.

[BJKS93] Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben Smeets. On Families of Hash Functions via Geometric Codes and Concatenation. In Stinson [Sti93], pages 331–342.

[BMP00] V. Boyko, P. MacKenzie, and S. Patel. Provably-secure password-authenticated key exchange using Diffie-Hellman. In Preneel [Pre00], pages 156–171.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *11th ACM Conference on Computer and Communication Security*. ACM, October 25–29 2004.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Preneel [Pre00], pages 139–155.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287. IEEE, 1994.

[CDF+08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT08*, pages 471–488. Springer-Verlag, 2008.

[CKOR10] Nishanth Chandran, Bhavana Kanukurthi, Rafail Ostrovsky, and Leonid Reyzin. Privacy amplification with asymptotically optimal entropy loss. In Leonard J. Schulman, editor, *STOC*, pages 785–794. ACM, 2010.

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[dB93] Bert den Boer. A Simple and Key-Economical Unconditional Authentication Scheme. *Journal of Computer Security*, 2:65–71, 1993.

[Din05] Yan Zong Ding. Error correction in the bounded storage model. In Joe Kilian, editor, *2nd Theory of Cryptography Conference — TCC 2005*, volume 3378 of *LNCS*. Springer-Verlag, 2005.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *LNCS*, pages 232–250. Springer-Verlag, 20–24 August 2006.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. arXiv:cs/0602007.

[DS02] Y. Dodis and J. Spencer. On the (non-)universality of the one-time pad. In *43rd Annual Symposium on Foundations of Computer Science*, pages 376–385. IEEE, 2002.

[DS05] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 654–663. ACM, 2005.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pages 601–610, Bethesda, Maryland, 31 May–2 June 2009.

[FJ01] Niklas Frykholm and Ari Juels. Error-tolerant password recovery. In *Eighth ACM Conference on Computer and Communication Security*, pages 1–8. ACM, November 5–8 2001.

[GL01] Oded Goldreich and Yehuda Lindell. Session-key generation using human passwords only. In Joe Kilian, editor, *Advances in Cryptology—CRYPTO 2001*, volume 2139 of *LNCS*, pages 408–432. Springer-Verlag, 2001.

[GL03] Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In Eli Biham, editor, *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 524–543. Springer-Verlag, 2003.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36. ACM, November 1999.

[KOY01] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In Birgit Pfitzmann, editor, *Advances in Cryptology—EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 475–494. Springer-Verlag, 2001.

[KR08] Bhavana Kanukurthi and Leonid Reyzin. An improved robust fuzzy extractor. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *Sixth Conference on Security and Cryptography in Networks SCN '08*, volume 5229 of *LNCS*, pages 206–223, September 2008.

[KR09] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. In Antoine Joux, editor, *Advances in Cryptology—EUROCRYPT 2009*, volume 5479 of *LNCS*. Springer, 2009. Full version available at `http://eprint.iacr.org/2008/494`.

[LT03] J.-P. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA*, pages 393–402, 2003.

[Mau92] Ueli Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[Mau93] Ueli Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[Mau97] Ueli Maurer. Information-theoretically secure secret-key agreement by NOT authenticated public discussion. In Walter Fumy, editor, *Advances in Cryptology—EUROCRYPT 97*, volume 1233 of *LNCS*, pages 209–225. Springer-Verlag, 1997.

[MS77] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Elsevier Science, 1977.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski, Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *LNCS*, pages 307–321. Springer-Verlag, 1997.

[MW03] Ueli Maurer and Stefan Wolf. Secret-key agreement over unauthenticated public channels — Part III: Privacy amplification. *IEEE Trans. Info. Theory*, 49(4):839–851, 2003.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–53, 1996.

[Pre00] Bart Preneel, editor. *Advances in Cryptology—EUROCRYPT 2000*, volume 1807 of *LNCS*. Springer-Verlag, 2000.

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[RW03] Renato Renner and Stefan Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*, pages 78–95. Springer-Verlag, 2003.

[RW04] Renato Renner and Stefan Wolf. The exact price for unconditionally secure asymmetric cryptography. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 109–125. Springer-Verlag, 2004.

[Sti93] Douglas R. Stinson, editor. *Advances in Cryptology—CRYPTO '93*, volume 773 of *LNCS*. Springer-Verlag, 22–26 August 1993.

[Sti94] D. R. Stinson. Universal hashing and authentication codes. *Designs, Codes, and Cryptography*, 4(4):369–380, 1994.

[Sti02] 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. Available at `http://www.cacr.math.uwaterloo.ca/~dstinson/publist.html`.

[SZ99] Aravind Srinivasan and David Zuckerman. Computing with very weak random sources. *SIAM J. Computing*, 28(4):1433–1459, 1999.

[Tay93] Richard Taylor. An Integrity Check Value Algorithm for Stream Ciphers. In Stinson [Sti93], pages 40–48.

[Vad04] S. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1), 2004.

[WC81] M.N. Wegman and J.L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

[Wol98] Stefan Wolf. Strong security against active attacks in information-theoretic secret-key agreement. In Kazuo Ohta and Dingyi Pei, editors, *Advances in Cryptology—ASIACRYPT '98*, volume 1514 of *LNCS*, pages 405–419. Springer-Verlag, 1998.

[Wyn75] A.D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.

A Proof of Theorem 4 and Corollary 5

We first give a more detailed statement of the result:

Theorem 10 *Assume* SS *is a deterministic linear $(m, m-k, t)$-secure sketch for the Hamming metric on $\{0,1\}^n$. Setting $v = (n-\ell-k)/2$, the construction of Section 4.3.1 is an $(m, \ell, t, \varepsilon)$ fuzzy extractor with robustness δ, for any $m, \ell, \varepsilon, \delta$ satisfying*

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

for pre-application robustness and

$$\begin{aligned}\ell \quad \le \min\Bigg(\quad & \frac{1}{3}\left(2m - n - k - 2\left(\log B + \log\left(2\left\lceil\frac{k}{n-k}\right\rceil + 2\right) + \log\left(\tfrac{1}{\delta}\right)\right)\right), \\ & 2m - n - k - 4\log\left(\tfrac{1}{\varepsilon}\right) \Bigg)\end{aligned}$$

for post-application robustness, where B denotes the size of the Hamming ball of radius t in $\{0,1\}^n$.

The construction of Section 4.3.2 is a robust fuzzy extractor for the set difference metric on sets of size up to r over the universe $\mathbb{F}^*_{2^\alpha}$ for the same parameters, except B now denotes the size of the Hamming ball of radius t in $\{0,1\}^{2^\alpha-1}$, $n = r\alpha$ and $k = t\alpha$

Note that the volume of the Hamming ball of radius t in $\{0,1\}^n$ is $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t}$, and that for $t \le n/2$, its logarithm is at most $nH_2(t/n)$ [MS77, Chapter 10, §11, Lemma 8], where $H_2(x) \stackrel{\text{def}}{=} -x\log x - (1-x)\log_2(1-x) \le -2x\log x$ is the binary entropy function. Another bound on the logarithm of the volume is $t\log(n+1)$, achieved simply by noting that every point in this ball (when centered at 0) can 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. Also note that $2\left\lceil\frac{k}{n-k}\right\rceil + 2 < n$. These observations give the statements of Theorem 4 and Corollary 5.

Proof We will prove the theorem for the Hamming metric; the proof set difference follows, as discussed in Section 4.3.2.

We need to argue robustness (that if P is modified, Rep will likely reject), and that the extracted key R is uniform even given P. We will give the adversary $\mathcal{A}$ very strong control over w': the value of w' can be any function $\mathrm{Err}(w, P)$ as long as it is within the error-correcting distance of the secure sketch. Denote by B the volume of the Hamming ball of radius t in $\{0,1\}^n$.

Pre-application Robustness. Assume that the adversary $\mathcal{A}$ is deterministic (since she is unbounded, this is without loss of generality): upon observing $P = (s, i, \sigma)$, $\mathcal{A}$ outputs $\tilde{P} = (s', i', \sigma')$, where (s', i', σ') is some function $\mathcal{A}(s, i, \sigma)$. Fix arbitrary i known to $\mathcal{A}$ (this makes the result stronger, since we will show that robustness holds for worst-case choice of i). Now, the only remaining probability comes from the choice of w and w'; also, s and σ determine the entire transcript $\mathsf{tr} = (i, s, \sigma, s', i', \sigma')$. If $i = i'$ and $s = s'$, then Rep will compute $c' = c$, so for verification to succeed, σ' must equal σ, which means that P cannot be modified at all. Hence, assume $(i, s) \neq (i', s')$.

For the purposes of the analysis, we assume that $\mathcal{A}$ learns $\Delta = w' - w$ "for free" (that is, we will condition on Δ even though it may not be necessary). Recall that w^* is the putative value of w reconstructed by SRec, and a', b' are the putative values of a, b: $w^* = a' \| b'$.

Let Succ be the event that $\mathcal{A}$ succeeds (that is, Rep accepts $\tilde{P}$). We will need two claims to bound the probability of Succ.

1. *Given $\Delta = w' - w$, and the sketches s, s', $\mathcal{A}$ can compute $\Delta_a = a' - a$ and $\Delta_b = b' - b$, or determine that* Rep *will reject before computing a' and b'.*

2. *For any* tr *and difference Δ, we have*

$$\Pr_w(\mathsf{Succ} \mid \mathsf{tr}, \Delta) \leq (L+2) 2^{n' - 2v} 2^{-\mathbf{H}_\infty(W | s, \sigma, \Delta)}.$$

The first claim says that given the information at hand, $\mathcal{A}$ knows the offset between the authentication key used by Gen to compute σ and the key which Rep will use to verify σ'. The second claim is that she still cannot succeed as long as W has enough min-entropy, even knowing that offset.

Before proving the claims, let us see why the second one suffices. Recall that adversarial success depends only on w, w', or, equivalently, on w, Δ, and that the transcript tr is determined by s and σ. We can thus expand the probability of success conditioned on the values s, σ, Δ.

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

The last equality follows from the definition of the average conditional min-entropy $\widetilde{\mathbf{H}}_\infty(W \mid s, \sigma, \Delta)$, and the fact that s, σ determine s', i', σ'. The information on which W is conditioned can be represented by $k + v + \log B$ bits, where B is the volume of the Hamming ball of radius t in $\{0,1\}^n$ (that is, the number of possibilities for Δ), and therefore by [DORS08, Lemma 2.2],

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

The forgery probability is thus $\Pr(\mathsf{Succ}) \leq B(L+2)2^{n'-2v-m+k+v} = B(L+2)2^{n-m-v}$. We will see how to set v further below. For now, note that the probability of forgery differs from the errorless case by a factor of $B(L+2)$, and is low when m and v are large.

It remains to prove the claims above. For the first claim, observe that by correctness of SRec, Rep will produce w^* such that $\mathsf{dis}(w^*, w') \leq t$ and $\mathsf{SS}(w^*) = s'$ if such a w^* exists; and if it does not exist, Rep will output $\perp$ before proceeding to compute $c' = a' \| b'$. Let $\Gamma = w^* - w' = w^* - w - \Delta$; the conditions on w^* are equivalent to saying that there exists Γ of weight at most t such that $\mathsf{SS}(\Gamma) = s' - s - \mathsf{SS}(\Delta)$ (by linearity of SS). Such Γ can be found by $\mathcal{A}$, if it exists, by simply running $\mathsf{SRec}(0^n, s' - s - \mathsf{SS}(\Delta))$ and testing if the result has the correct weight and SS (by correctness of the secure sketch scheme at input Γ, which is at most t away from 0^n, this procedure must find Γ if it exists). Furthermore, if Γ exists, $w^* - w = \Gamma + \Delta$. By linearity,

$c' - c = \mathsf{SS}^{\perp}(w^*) - \mathsf{SS}^{\perp}(w) = \mathsf{SS}^{\perp}(w^* - w) = \mathsf{SS}^{\perp}(\Gamma + \Delta)$. Thus, $\mathcal{A}$ can find whether Rep will output $\perp$ before producing a' and b' by checking if Γ exists, and if so, can compute the desired quantities by computing $\mathsf{SS}^{\perp}(\Gamma + \Delta)$ and parsing it into $n' - v$ bits of Δ_a and v bits of Δ_b.

We can use the first claim to prove Claim 2 above. Given a particular transcript tr and Δ, $\mathcal{A}$ succeeds only when $\sigma' = [f_{s',i'}(a')]_1^v + b'$, which is the same as requiring that a be a solution to the equation $[f_{s,i}(a) - f_{s',i'}(a + \Delta_a)]_1^v = \Delta_b + \sigma - \sigma'$. By Claim 1 above, this equation is fixed given tr and Δ.

Now we claim that for any distinct pairs $(s, i) \neq (s', i')$ and for any offset Δ, the polynomial $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$ is non-constant of degree at most $L + 2$. There are two cases in the proof: (a) if $\Delta_a \neq 0$, then the coefficient of order $L + 2$ is $(L + 3)\Delta_a$, which is non-zero (in $GF(2^v)$) since L is even; (b) if $\Delta_a = 0$ then the polynomial $f_{s,i}(x) - f_{s',i'}(x)$ is non-constant (a coefficient of order 2 or higher is nonzero if $s \neq s'$, or the coefficient of order 1 is nonzero if $i \neq i'$).

Thus, the number of a values that satisfy $[f_{s,i}(a) - f_{s',i'}(a + \Delta_a)]_1^v = \Delta_b + \sigma - \sigma'$ is at most $(L + 2)2^{n'-2v}$ (a goes through two mappings: the polynomial, which is at most $(L + 2)$-to-1, and truncation, which is $2^{n'-2v}$-to-1). Each such value occurs with probability $2^{-\mathbf{H}_\infty(a|\mathsf{tr},\Delta)}$, giving the bound $(L + 2)2^{n'-2v}2^{-\mathbf{H}_\infty(a|\mathsf{tr},\Delta)}$ on the probability of forgery for a particular tr and Δ.

Since all the other parts of tr are deterministic functions of s and σ, we can replace tr with (s, σ) in the formula. Finally, $\mathbf{H}_\infty(a \mid s, \sigma, \Delta) = \mathbf{H}_\infty(a, s, \sigma \mid s, \sigma, \Delta) = \mathbf{H}_\infty(a, s, b \mid s, \sigma, \Delta)$, because $b = \sigma - [f_{s,i}(a)]_1^v$; and $\mathbf{H}_\infty(a, s, b \mid s, \sigma, \Delta) = \mathbf{H}_\infty(W \mid s, \sigma, \Delta)$, because $w = \left(\frac{S}{S^{\perp}}\right)^{-1}(s\|a\|b)$. Thus, the success probability is at most $(L + 2)2^{n'-2v}2^{-\mathbf{H}_\infty(W|s,\sigma,\Delta)}$, as desired.

Extraction. The argument that R is nearly uniform given P is similar to the errorless case, except s has to be taken into account. For every s, the function $H_i(c) \stackrel{\text{def}}{=} (\sigma, R)$ is universal, because for every c_1, c_2, there is at most one i such that $H_i(c_1) = H_i(c_2)$ (because $i(a_1 - a_2)$ is fixed, like in the errorless case). Because $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \geq m - k$, applying Lemma 1, we see that the distribution of $(R, P) = (R, (i, \sigma, s))$ is $2^{(n'-v-(m-k))/2-1} = 2^{(n-v-m)/2-1}$-close to $(U_{n'-2v} \times U_{n'-v} \times U_{2v} \times \mathsf{SS}(W))$.

Applying Lemma 3 to $A = R$, $B = P$, $C = U_{n'-2v}$, $D = U_{n'-v} \times U_{2v} \times \mathsf{SS}(W)$, we get that (R, P) is ε-close to $U_{n'-2v} \times P$, for $\varepsilon = 2^{(n'-v-m)/2}$.

Setting v. We thus obtain the following constraints on v:

- $v \leq (n - k)/2$ (because the construction requires $|a| \geq |b| = v$);
- $v \geq n - m + \log B + \log(L + 2) + \log\left(\frac{1}{\delta}\right)$ (because $B(L + 2)2^{n-m-v} \leq \delta$); and
- $v \geq n - m + 2\log\left(\frac{1}{\varepsilon}\right)$ (because $2^{(n-v-m)/2} \leq \varepsilon$).

Observe that $L = 2\lceil k/(n - k - v)\rceil \leq 2\lceil k/(n - k)\rceil$ (because $v \leq (n - k)/2$). Thus, we set

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

to extract a key of length

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

Post-Application Robustness. Because the extracted key R contains $n - k - 2v$ bits, providing it to Eve increasing her ability to forge the σ by at most a factor of 2^{n-k-2v} (indeed, if the increase was greater, she could break the original MAC with higher probability by simply guessing the value of R). The rest of the analysis remains the same. Thus, the second constraint on v becomes

- $v \geq \frac{1}{3}\left(2n - k - m + \log B + \log\left(2\left\lceil\frac{k}{n-k}\right\rceil + 2\right) + \log\left(\frac{1}{\delta}\right)\right)$

If the second constraint dominates the third, which happens whenever

$$\delta < \varepsilon^6 B\left(2\left\lceil\frac{k}{n-k}\right\rceil + 2\right)2^{2m-n-k},$$

then we can extract

$$\ell = \frac{1}{3}\left(2m - n - k - 2\left(\log B + \log\left(2\left\lceil\frac{k}{n-k}\right\rceil + 2\right) + \log\left(\frac{1}{\delta}\right)\right)\right)$$

bits, or about a third of what was possible if we required only pre-application robustness. If larger δ is acceptable in an application, then we can set v according to the third constraint to extract as many bits as in the case of pre-application robustness. ■