An Improved Robust Fuzzy Extractor

Bhavana Kanukurthi and Leonid Reyzin
Boston University Computer Science
`http://cs-people.bu.edu/bhavanak, http://www.cs.bu.edu/~reyzin`

July 24, 2008

Abstract

We consider the problem of building robust fuzzy extractors, which allow two parties holding similar random variables W, W' to agree on a secret key R in the presence of an active adversary. Robust fuzzy extractors were defined by Dodis et al. in Crypto 2006 to be noninteractive, i.e., only one message P, which can be modified by an unbounded adversary, can pass from one party to the other. This allows them to be used by a single party at different points in time (e.g., for key recovery or biometric authentication), but also presents an additional challenge: what if R is used, and thus possibly observed by the adversary, before the adversary has a chance to modify P. Fuzzy extractors secure against such a strong attack are called post-application robust.

We construct a fuzzy extractor with post-application robustness that extracts a shared secret key of up to $(2m - n)/2$ bits (depending on error-tolerance and security parameters), where n is the bit-length and m is the entropy of W. The previously best known result, also of Dodis et al., extracted up to $(2m - n)/3$ bits (depending on the same parameters).

1 Introduction

Consider the following scenario. A user Charlie has a secret w that he wants to use to encrypt and authenticate his hard drive. However, w is not a uniformly random key; rather, it is a string with some amount of entropy from the point of view of any adversary $\mathcal{A}$. Naturally, Charlie uses an extractor [NZ96], which is a tool for converting entropic strings into uniform ones. An extractor Ext is an algorithm that takes the entropic string w and a uniformly random seed i, and computes $R = \mathsf{Ext}(w; i)$ that is (almost) uniformly random even given i.

It may be problematic for Charlie to memorize or store the uniformly random R (this is in contrast to w, which can be, for example, a long passphrase already known to Charlie, his biometric, or a physical token, such as a physical one-way function [PRTG02]). Rather, in order to decrypt the hard drive, Charlie can use i again to recompute $R = \mathsf{Ext}(w; i)$. The advantage of storing i rather than R is that i need not be secret, and thus can be written, for example, on an unencrypted portion of the hard drive.

Even though the storage of i need not be secret, the authenticity of i is very important. If $\mathcal{A}$ could modify i to i', then Charlie would extract some related key R', and any guarantee on the integrity of the hard drive would vanish, because typical encryption and authentication schemes do not provide any security guarantees under related-key attacks. To authenticate i, Charlie would need to use some secret key, but the only secret he has is w.

This brings us to the problem of building *robust* extractors: ones in which the authenticity of the seed can be verified at reconstruction time. A robust extractor has two procedures: a randomized $\mathsf{Gen}(w)$, which generates (R, P) such that R is uniform even given P (think of P as containing the seed i as well as some authentication information), and $\mathsf{Rep}(w, P')$, which reproduces R if $P' = P$ and outputs $\perp$ with high probability for an adversarially produced $P' \neq P$.

Note that in the above scenario, the adversary $\mathcal{A}$, before attempting to produce $P' \neq P$, gets to see the value P and how the value R is used for encryption and authentication. Because we want robust fuzzy extractors to be secure for a wide variety of applications, we do not wish to restrict how R is used and, therefore, what information about R is available to $\mathcal{A}$. Rather, we will require that $\mathcal{A}$ has low probability of getting $\mathsf{Rep}(w, P')$ to not output $\perp$ *even* if $\mathcal{A}$ is given both P and R. This strong notion of security is known as *post-application* robustness.

An additional challenge may be that the value w when Gen is run is slightly different from the value w' available when Rep is run: for example, the user may make a typo in a long passphrase, or a biometric reading may differ slightly. Extractors that can tolerate such differences and still reproduce R exactly are called *fuzzy* [DORS08]. Fuzzy extractors are obtained by adding error-correcting information to P, to enable Rep to compensate for errors in w'. The specific constructions depend on the kinds of errors that can occur (e.g., Hamming errors, edit distance errors, etc.).

Robust (fuzzy) extractors are useful not only in the single-party setting described above, but also in interactive settings, where two parties are trying to derive a key from a shared (slightly different in the fuzzy case) secret w that either is nonuniform or about which some limited information is known to the adversary $\mathcal{A}$. One party, Alice, can run Gen to obtain (R, P) and send P to the other party, Bob, who can run Rep to also obtain R. However, if $\mathcal{A}$ is actively interfering with the channel between Alice and Bob and modifying P, it is important to ensure that Bob detects the modification rather than derives a different key R'. Moreover, unless Alice can be sure that Bob truly received P before she starts using R in a communication, post-application robustness is needed.

Prior Work. Fuzzy extractors, defined in [DORS08], are essentially the noninteractive variant of privacy amplification and information reconciliation protocols, considered in multiple works, including [Wyn75, BBR88, Mau93, BBCM95]. Robust (fuzzy) extractors, defined in [BDK$^+$05, DKRS06], are the noninteractive variant of privacy amplification (and information reconciliation) secure against active adversaries [Mau97, MW97, Wol98, MW03, RW03, RW04].

Let the length of w be n and the entropy of w be m. Post-application robust fuzzy extractors cannot extract anything out of w if $m < n/2$, because an extractor with post-application robustness implies an information-theoretically secure message authentication code (MAC) with w as the key[1], which is impossible if $m < n/2$ (see [DS02] for impossibility of deterministic MACs if $m < n/2$ and its extension by [Wic08] to randomized MACs). Without any set-up assumptions, the only previously known post-application robust extractor, due to [DKRS06], extracts R of length $\frac{2}{3}(m - n/2 - \log\frac{1}{\delta})$ (or even less if R is required to be very close to uniform), where δ is the probability that the adversary violates robustness. Making it fuzzy further reduces the length of R by an amount related to the error-tolerance. (With set-up assumptions, one can do much better: the construction of [CDF$^+$08] extracts almost the entire entropy m, reduced by an amount related to security and, in the fuzzy case, to error-tolerance. However, this construction assumes that a nonsecret uniformly random string is already known to both parties, and that the distribution on

[1] The MAC is obtained by extracting R, using it as a key to any standard information-theoretic MAC (e.g., [WC81]), and sending P along with the tag to the verifier

w, including adversarial knowledge about w, is independent of this string.)

OUR RESULTS. The robust extractor construction of [DKRS06] is parameterized by a value v that can be decreased in order to obtain a longer R. In fact, as shown in [DKRS06], a smaller v can be used for *pre-application* robustness (a weaker security notion, in which $\mathcal{A}$ gets P but not R). We show in Theorem 2 that the post-application-robustness analysis of [DKRS06] is essentially tight, and if v is decreased, the construction becomes insecure.

Instead, in Section 3, we propose a new construction of an extractor with post-application robustness that extracts R of length $m - n/2 - \log \frac{1}{\delta}$, improving the previous result by a factor of $3/2$ (more if R is required to be very close to uniform). While this is only a constant-factor increase, in scenarios where secret randomness is scarce it can make a crucial difference. Like [DKRS06], we make no additional set-up assumptions. Computationally, our construction is slightly more efficient than the construction of [DKRS06]. Our improved robust extractor translates into an improved robust fuzzy extractor using the techniques of [DKRS06], with the same factor of 3/2 improvement.

In addition, we show (in Section 3.2) a slight improvement for the pre-application robust version of the extractor of [DKRS06], applicable when the extracted string must be particularly close to uniform.

2 Preliminaries

NOTATION. For binary strings a, b, $a||b$ denotes their concatenation, $|a|$ denotes the length of a. For a binary string a, for we denote by $[a]_i^j$, the substring $b = a_i a_{i+1} \dots a_j$. If S is a set, $x \leftarrow S$ means that x is chosen uniformly from S. If X is a probability distribution (or a random variable), then $x \leftarrow X$ means that x is chosen according to distribution X. If X and Y are two random variables, then $X \times Y$ denotes the product distribution (obtained by sampling X and Y *independently*). All logarithms are base 2.

RANDOM VARIABLES, ENTROPY, EXTRACTORS. Let U_l denote the uniform distribution on $\{0,1\}^l$. Let X_1, X_2 be two probability distributions over some set S. Their *statistical distance* is

$$\mathbf{SD}\left(X_1, X_2\right) \stackrel{\text{def}}{=} \max_{T \subseteq S}\{\Pr[X_1 \in T] - \Pr[X_2 \in T]\} = \frac{1}{2}\sum_{s \in S}\left|\Pr_{X_1}[s] - \Pr_{X_2}[s]\right|$$

(they are said to be ε-close if $\mathbf{SD}\left(X_1, X_2\right) \le \varepsilon$). We will use the following lemma on statistical distance that was proven in [DKRS08]:

Lemma 1. *For any joint distribution (A, B) and distributions C and D over the ranges of A and B respectively, if $\mathbf{SD}\left((A, B), C \times D\right) \le \alpha$, then $\mathbf{SD}((A, B), C \times B) \le 2\alpha$.*

MIN-ENTROPY. The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) = -\log(\max_w \Pr[W = w])$ (all logarithms are base 2, unless specified otherwise). Following [DORS08], for a joint distribution (W, E), define the (average) conditional min-entropy of W given E as

$$\widetilde{\mathbf{H}}_\infty(W \mid E) = -\log(\mathop{\mathbf{E}}_{e \leftarrow E}(2^{-\mathbf{H}_\infty(W|E=e)}))$$

(here the expectation is taken over e for which $Pr[E = e]$ is nonzero). A computationally unbounded adversary who receives the value of E cannot find the correct value of W with probability greater than $2^{-\widetilde{\mathbf{H}}_\infty(W|E)}$. We will use the following lemma from [DORS08]:

Lemma 2. *Let A, B, C be random variables. If B has at most 2^λ possible values, then the following holds: $\widetilde{\mathbf{H}}_\infty(A|B,C) \geq \widetilde{\mathbf{H}}_\infty((A,B)|C) - \lambda \geq \widetilde{\mathbf{H}}_\infty(A|C) - \lambda$. In particular, $\widetilde{\mathbf{H}}_\infty(A|B) \geq \mathbf{H}_\infty((A,B)) - \lambda \geq \mathbf{H}_\infty(A) - \lambda$.*

Because in this paper the adversary is sometimes assumed to have some external information E about Alice and Bob's secrets, we need the following variant, defined in [DORS08, Definition 2], of the definition of strong extractors of [NZ96]:

Definition 1. Let $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^l$ be a polynomial time probabilistic function that uses r bits of randomness. We say that Ext is an average-case (n, m, l, ε)-strong extractor if for all pairs of random variables (W, E) such that $w \in W$ is an n-bit string and $\widetilde{\mathbf{H}}_\infty(W \mid E) \geq m$, we have $\mathbf{SD}\left((\mathsf{Ext}(W; X), X, E), (U_l, X, E)\right) \leq \varepsilon$, where X is the uniform distribution over $\{0,1\}^r$.

Any strong extractor can be made average-case with a slight increase in input entropy [DORS08, Section 2.5]. We should note that some strong extractors, such as universal hash functions [CW79, HILL99] discussed next, generalize without any loss to average-case.

The Leftover Hash Lemma We first recall the notion of universal hashing [CW79]:

Definition 2. A family of efficient functions $\mathcal{H} = \left\{h_i : \{0,1\}^n \to \{0,1\}^\ell\right\}_{i \in I}$ is *universal* if for all distinct x, x' we have $\Pr_{i \leftarrow I}[h_i(x) = h_i(x')] \leq 2^{-l}$.

$\mathcal{H}$ is *pairwise independent* if for all distinct x, x' and all y, y' it holds that $\Pr_{i \in I}[h_i(x) = y \wedge h_i(x') = y'] \leq 2^{-2\ell}$. $\diamondsuit$

Lemma 3 (Leftover Hash Lemma, average-case version [DORS08]). *For $\ell, m, \varepsilon > 0$, $\mathcal{H}$ is a strong (m, ε) average-case extractor (where the index of the hash function is the seed to the extractor) if $\mathcal{H}$ is universal and $\ell \leq m + 2 - 2\log\frac{1}{\varepsilon}$.*

This Lemma easily generalizes to the case when $\mathcal{H}$ is allowed to depend on the extra information E about the input X. In other words, every function in $\mathcal{H}$ takes an additional input e, and the family $\mathcal{H}$ is universal for every fixed value of e.

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 3. 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. $\diamondsuit$

In this paper, we will construct a robust fuzzy extractor for the binary Hamming metric using secure sketches for the same metric. We will briefly review the syndrome construction from [DORS08, Construction 3] that we use (see also references therein for its previous incarnations). Consider an efficiently decodable $[n, n-k, 2t+1]$ linear error-correcting code C. The sketch $s = \mathsf{SS}(w)$ consists of the k-bit syndrome w with respect to C. We will use the fact that s is a (deterministic) linear function of w and that the entropy loss is at most $|s| = k$ bits in the construction of our robust fuzzy extractor for the Hamming metric.

We note that, as was shown in [DKRS06], the secure sketch construction for the set difference metric of [DORS08] can be used to extend the robust fuzzy extractor construction in the Hamming metric to the set difference metric.

While a secure sketch enables recovery of a string w from a close string w', a fuzzy extractor extracts a close-to-uniform string R and allows the precise reconstruction of R from any string w' close to w.

Definition 4. An $(m, \ell, t, \varepsilon)$-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 $\mathsf{dis}(w, w') \leq t$ 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 $\mathbf{H}_\infty(W) \geq m$ and $(R, P) \leftarrow \mathsf{Gen}(W)$, then we have $\mathbf{SD}\left((R, P), U_\ell \times P\right) \leq \varepsilon$. $\diamondsuit$

Note that fuzzy extractors allow the information P to be revealed to an adversary without compromising the security of the extracted random string R. However, they provide no guarantee when the adversary is active. Robust fuzzy extractors defined (and constructed) in [DKRS06] formalize the notion of security against active adversaries. We review the definition below.

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

Definition 5. 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., $\tilde{P} = A(P)$); output "success" if $\tilde{P} \neq P$ and $\mathsf{Rep}(w', \tilde{P}) \neq \perp$. $\diamondsuit$

We note that the above definitions can be easily extended to give *average-case* fuzzy extractors (where the adversary has some external information E correlated with W), and that our constructions satisfy those stronger definitions, as well.

3 The New Robust Extractor

In this section we present our new extractor with post-application robustness. We extend it to a robust *fuzzy* extractor in Section 5. Our approach is similar to that of [DKRS06]; a detailed comparison is given in Section 4.

STARTING POINT: KEY AGREEMENT SECURE AGAINST A PASSIVE ADVERSARY. Recall that a strong extractor allows extraction of a string that appears uniform to an adversary even given the presence of the seed used for extraction. Therefore, a natural way of achieving key agreement in the errorless case is for Alice to pick a random seed i for a strong extractor and send it to Bob (in the clear). They could then use $R = \mathsf{Ext}(w; i)$ as the shared key. As long as the adversary is passive, the shared key looks uniform to her. However, such a protocol can be rendered completely insecure when executed in the presence of an active adversary because $\mathcal{A}$ could adversarially modify i to i' such that R' extracted by Bob has no entropy. To prevent such malicious modification of i we will require Alice to send an authentication of i (along with i) to Bob. In our construction, we authenticate i using w as the key and then extract from w using i as the seed. Details follow.

CONSTRUCTION. For the rest of the paper we will let $w \in \{0,1\}^n$. We will assume that n is even (if not, drop one bit of w, reducing its entropy by at most 1). To compute $\mathsf{Gen}(w)$, let a be the first half of w and b the second: $a = [w]_1^{n/2}, b = [w]_{n/2+1}^n$. View a,b as elements of $\mathbb{F}_{2^{n/2}}$. Let $v = n - m + \log \frac{1}{\delta}$, where δ is the desired robustness. Choose a random $i \in \mathbb{F}_{2^{n/2}}$. Compute $y = ia + b$. Let σ consist of the first v bits of y and the extracted key R consist of the rest of y: $\sigma = [y]_1^v$, $R = [y]_{v+1}^{n/2}$. Output $P = (i, \sigma)$.

$\mathsf{Gen}(w)$:
1. Let $a = [w]_1^{n/2}, b = [w]_{n/2+1}^n$
2. Select a random $i \leftarrow \mathbb{F}_{2^{n/2}}$
3. Set $\sigma = [ia + b]_1^v$, $R = [ia + b]_{v+1}^{n/2}$ and output $P = (i, \sigma)$

$\mathsf{Rep}(w, P' = (i', \sigma'))$:
1. Let $a = [w]_1^{n/2}, b = [w]_{n/2+1}^n$
2. If $\sigma' = [i'a + b]_1^v$ then compute $R' = [i'a + b]_{v+1}^{n/2}$ else output $\perp$

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

If ε is so low that the constraint $m \ge n/2 + 2\log \frac{1}{\varepsilon}$ is not satisfied, then the construction can be modified as shown in Section 3.1.

Proof. EXTRACTION. Our goal is to show that R is nearly uniform given P. To do so, we first show that the function $h_i(a, b) = (\sigma, R)$ is a universal hash family. Indeed, for $(a, b) \ne (a', b')$ consider

$$\begin{aligned}
\Pr_i[h_i(a,b) = h_i(a',b')] &= \Pr_i[ia + b = ia' + b'] \\
&= \Pr_i[i(a - a') = (b - b')] \\
&\le 2^{-n/2}.
\end{aligned}$$

To see the last inequality recall that $(a, b) \ne (a', b')$. Therefore, if $a = a'$, then $b \ne b'$ making the $\Pr_i[i(a - a') = (b - b')] = 0$. If $a \ne a'$, then there is a unique $i = (b - b')/(a - a')$ that satisfies the equality. Since i is chosen randomly from $\mathbb{F}_{2^{n/2}}$, the probability of the specific i occurring is $2^{-n/2}$.

Because $|(R,\sigma)| = n/2$, Lemma 3 gives us $\mathbf{SD}\left((R,P), U_{|R|} \times U_{|P|}\right) \le \varepsilon/2$ as long as $n/2 \le m + 2 - 2\log\frac{2}{\varepsilon}$, or, equivalently, (R,P) is $2^{(n/2-m)/2-1}$-close to $U_{|R|} \times U_{|P|}$. Applying Lemma 1 to $A = R$, $B = P$, $C = U_{\frac{n}{2}-v}$, $D = U_{\frac{n}{2}} \times U_v$, we get that (R,P) is ε-close to $U_{(\frac{n}{2})-v} \times P$, for $\varepsilon = 2^{(n/2-m)/2}$. From here it follows that for extraction to be possible, $m \ge n/2 + 2\log\frac{1}{\varepsilon}$.

Post-Application Robustness. In the post-application robustness security game, the adversary $\mathcal{A}$ on receiving $(P = (i,\sigma), R)$ (generated according to procedure Gen) outputs $P' = (i', \sigma')$, and is considered successful if $(P' \ne P) \wedge [i'a+b]_1^v = \sigma'$. In our analysis, we will assume that $i' \ne i$. We claim that this does not reduce $\mathcal{A}$'s success probability. Indeed, if $i' = i$ then, for $P' \ne P$ to hold, $\mathcal{A}$ would have to output $\sigma' \ne \sigma$. However, when $i' = i$, Rep would output $\perp$ unless $\sigma' = \sigma$.

In our analysis, we allow $\mathcal{A}$ to be deterministic. This is without loss of generality since we allow an unbounded adversary. We also allow $\mathcal{A}$ to arbitrarily fix i. This makes the result only stronger since we demonstrate robustness for a worst-case choice of i.

Since i is fixed and $\mathcal{A}$ is deterministic, (σ, R) determines the transcript $\mathsf{tr} = (i, \sigma, R, i', \sigma')$. For any particular tr, let $\mathsf{Succ_{tr}}$ be the event that the transcript is tr and $\mathcal{A}$ wins, i.e., that $ia + b = \sigma || R \wedge [i'a+b]_1^v = \sigma'$. We denote by $\mathsf{Bad_{tr}}$ the set of $w = a||b$ that make $\mathsf{Succ_{tr}}$ true. For any tr, $\Pr_w[\mathsf{Succ_{tr}}] \le |\mathsf{Bad_{tr}}| 2^{-m}$, because each w in $\mathsf{Bad_{tr}}$ occurs with probability at most 2^{-m}. We now partition the set $\mathsf{Bad_{tr}}$ into 2^ℓ disjoint sets, indexed by $R' \in \{0,1\}^\ell$:

$$\begin{aligned} \mathsf{Bad}_{\mathsf{tr}}^{R'} &\stackrel{\text{def}}{=} \{w \mid w \in \mathsf{Bad_{tr}} \wedge [i'a+b]_{v+1}^{\ell} = R'\} \\ &= \{w \mid (ia+b = \sigma||R) \wedge (i'a+b = \sigma'||R')\} \end{aligned}$$

For a particular value of (tr, R'), $w = a||b$ is uniquely determined by the constraints that define the above set Therefore, $|\mathsf{Bad}_{\mathsf{tr}}^{R'}| = 1$. Since $\mathsf{Bad_{tr}} = \bigcup_{R' \in \{0,1\}^\ell} \mathsf{Bad}_{\mathsf{tr}}^{R'}$, we get $|\mathsf{Bad_{tr}}| \le 2^\ell = 2^{n/2-v}$. From here it follows that

$$\Pr[\mathsf{Succ_{tr}}] \le |\mathsf{Bad_{tr}}| 2^{-m} \le 2^{n/2-v-m}.$$

$\Pr[\mathsf{Succ_{tr}}]$ measures the probability that the transcript is tr and $\mathcal{A}$ succeeds. To find out the probability that $\mathcal{A}$ succeeds, we need to simply add $\Pr[\mathsf{Succ_{tr}}]$ over all possible tr. Since a transcript is completely determined by σ, R, the total number of possible transcripts is $2^{|\sigma|+|R|} = 2^{n/2}$ and, therefore, $\mathcal{A}$'s probability of success is at most 2^{n-v-m}.

To achieve δ-robustness, we need to set v to at least $n - m + \log\frac{1}{\delta}$. From here it follows that $\ell = \frac{n}{2} - v \le \frac{1}{2}(2m - n - 2\log\frac{1}{\delta})$. □

3.1 Getting Closer to Uniform

If ε is so low that the constraint $m \ge n/2 + 2\log\frac{1}{\varepsilon}$ is not satisfied, then in our construction we can simply shorten R by $\beta = n/2 + 2\log\frac{1}{\varepsilon} - m$ bits, as follows: keep $v = n - m + \log\frac{1}{\delta}$ (regardless of ℓ), and let $R = [ia+b]_{v+1}^{\ell+v}$, for any $\ell \le 2m - n - \log\frac{1}{\delta} - 2\log\frac{1}{\varepsilon}$. This keeps σ the same, but shortens R enough for the leftover hash lemma to work. The proof remains essentially the same, except that to prove robustness, we will give the remaining bits $[ia+b]_{\ell+v+1}^{n/2}$ for free to $\mathcal{A}$.

3.2 Improving the construction of [DKRS06] When the Uniformity Constraint Dominates

The construction of Dodis et al. [DKRS06] parses w as two strings a and b of lengths $n - v$ and v, respectively. The values σ, R are computed as $\sigma = [ia]_1^v + b$ and $R = [ia]_{v+1}^n$; $P = (i, \sigma)$. In order to get R to be uniform given P, the value v is increased until the leftover hash lemma can be applied to (R, σ). However, we observe that this unnecessarily increases the length of σ (i.e., for every bit added to v, two bits are subtracted from R). Instead, we propose to improve this construction with essentially the same technique as we use for our construction in Section 3.1. The idea is to simply shorten R without increasing the length of σ. This improvement applies to both pre- and post-application robustness.

For post-application robustness, suppose the uniformity constraint dominates, i.e., $2\log\frac{1}{\varepsilon} > (2m - n + \log\frac{1}{\delta})/3$. Modify the construction of [DKRS06] by setting $v = (2n - m + \log\frac{1}{\delta})/3$ and $R = [ia]_{v+1}^{n-v-\beta}$, where $\beta = 2\log\frac{1}{\varepsilon} - (2m - n - \log\frac{1}{\delta})/3$. This will result in an extracted key of length $\ell = (4m - 2n - \log\frac{1}{\delta})/3 - 2\log\frac{1}{\varepsilon}$. However, even with the improvement, the extracted key will be always shorter than the key extracted by our scheme, as explained in Section 4.2

In contrast, this improvement seems useful in the case of pre-application robustness. Again, suppose the uniformity constraint dominates, i.e., $2\log\frac{1}{\varepsilon} > \log\frac{1}{\delta}$. Modify the construction of [DKRS06] by setting $v = n - m + \log\frac{1}{\delta}$ and $R = [ia]_{v+1}^{n-v-\beta}$, where $\beta = 2\log\frac{1}{\varepsilon} - \log\frac{1}{\delta}$. This will result in an extracted key of length $\ell = 2m - n - 2\log\frac{1}{\varepsilon} - \log\frac{1}{\delta}$, which is $2\log\frac{1}{\varepsilon} - \log\frac{1}{\delta}$ longer than the key extracted without this modification.

4 Comparison with the construction of [DKRS06]

4.1 When the Robustness Constraint Dominates

Recall that the construction of Dodis et al. [DKRS06] parses w as two strings a and b of lengths $n-v$ and v, respectively. The values σ, R are computed as $\sigma = [ia]_1^v + b$ and $R = [ia]_{v+1}^n$; $P = (i, \sigma)$. Notice that, like in our construction, increasing v improves robustness and decreases the number of extracted bits. For pre-application robustness, setting $v = n - m + \log\frac{1}{\delta}$ suffices, and thus the construction extracts nearly $(2m-n)$ bits. However, for post-application robustness, a much higher v is needed, giving only around $\frac{1}{3}(2m - n)$ extracted bits.

The post-application robustness game reveals more information to $\mathcal{A}$ about w than the pre-application robustness game. This additional information—namely, R—may make it easier for $\mathcal{A}$ to guess σ' for a well-chosen i'. The key to our improvement is in the pairwise independence of the function $ia + b$ that computes both σ and R: because of pairwise independence, the value (σ, R) of the function on input i tells $\mathcal{A}$ nothing about the value (σ', R') on another input i'. (This holds, of course, for uniformly chosen key (a, b); when (a, b) has entropy m, then $\mathcal{A}$ can find out $n - m$ bits of information about σ'.)

In contrast, in the construction of [DKRS06], only σ is computed using a pairwise independent hash function. This works well (in fact, better than our construction, because b can be shorter) for pre-application robustness, where $\mathcal{A}$ does not find out R. But it makes it possible for R to decrease $\mathcal{A}$'s uncertainty about σ' by as much as $\ell = |R|$, thus necessitating the length v of σ' (and hence σ) to be $v > \ell + (n - m)$ (the $(n - m)$ term is the amount of entropy already potentially "missing" from σ' because of the nonuniformity of w). See Section 4.3 for a detailed description of an adversarial strategy that utilizes R to obtain σ' in the [DKRS06] construction.

Another way to see the differences between the two constructions is through the proof. In the proof of post-application robustness, the transcript tr includes R, which makes for 2^ℓ times more transcripts than in the proof of pre-application robustness. However, the fact that this R imposes an additional constraint of w, thus reducing the size of the set $\mathsf{Bad}_{\mathsf{tr}}$, can compensate for this increase.

It turns out that for the construction of [DKRS06], this additional constraint can be redundant if the adversary is clever about choosing i' and σ', and the size of $\mathsf{Bad}_{\mathsf{tr}}$ doesn't decrease. Using a pairwise-independent function for computing R in our construction ensures that this additional constraint decreases the size of $\mathsf{Bad}_{\mathsf{tr}}$ by 2^ℓ. Thus, our construction achieves the same results for pre- and post-application robustness.

4.2 When the Uniformity Constraint Dominates

It should be noted that there may be reasonable cases when the uniformity constraint ε on R is strong enough that the construction of [DKRS06] extracts even fewer bits, because it needs to take $v \geq n - m + 2\log\frac{1}{\varepsilon}$ to ensure near-uniformity of R given P. In that case, as long as $m \geq n/2 + 2\log\frac{1}{\varepsilon}$, our construction will extract the same amount of bits as before, thus giving it an even bigger advantage. And when $m < n/2 + 2\log\frac{1}{\varepsilon}$, our construction still extracts at least $3/2$ times more bits than the construction of [DKRS06], even with the improvement of Section 3.2 applied (this can be seen by algebraic manipulation of the relevant parameters for the post-application robustness case).

4.3 Why the construction of [DKRS06] cannot extract more bits

Recall that the robust fuzzy extractor of [DKRS06] operates as follows: parse w as two strings a, b of lengths $n - v, v$ respectively and compute $\sigma = [ia]_1^v + b$ and $R = [ia]_{v+1}^n$; $P = (i, \sigma)$.

For post-application robustness, the concern is that R can reveal information to the adversary about σ' for a cleverly chosen i'. Because the length of σ' is v and $\ell + (n - m)$ bits of information about σ' may be available (the ℓ term comes from $|R|$, and $(n - m)$ term comes from the part of w which has no entropy), this leads to the requirement that $v \geq \ell + n - m + \log\frac{1}{\delta}$ to make sure the adversary has to guess at least $\log\frac{1}{\delta}$ bits about σ'. Plugging in $\ell = n - 2v$, we obtain $\ell \leq \frac{2}{3}(m - n/2 - \log\frac{1}{\delta})$, which is the amount extracted by the construction.

Here we show an adversarial strategy that indeed utilizes R to obtain information about σ' to succeed with probability $\delta/2$. This demonstrates that the analysis in [DKRS06] is tight up to one bit. To do so we have to fix a particular (and somewhat unusual) representation of field elements. (Recall that any representation of field elements works for constructions here and in [DKRS06], as long as addition of field elements corresponds to the exclusive-or of bit strings.) Typically, one views $\mathbb{F}_{2^{n-v}}$ as $\mathbb{F}_2[x]/(p(x))$ for some irreducible polynomial p of degree $n - v$, and represents elements as $\mathbb{F}_2$-valued vectors in the basis $(x^{n-v-1}, x^{n-v-2}, ..., x^2, x, 1)$. We will do the same, but will reorder the basis elements so as to separate the even and the odd powers of x: $(x^{n-v-1}, x^{n-v-3}, \ldots, x, x^{n-v-2}, x^{n-v-4}, \ldots, 1)$ (assuming, for concreteness, that $n-v$ is even). The advantage of this representation for us is that the top half of bits of some value $z \in F_{2^{n-v}}$ is equal to the bottom half of the bits of z/x, as long as the last bit of z is 0.

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

Thus, the adversarial strategy is as follows: let $i' = i + i/x$; let τ consist of the ℓ bits of R, the top $n - m$ bits of σ, and $\log\frac{1}{\delta} = v - \ell - (n - m)$ randomly guessed bits, and let $\sigma' = \sigma + \tau$.

The adversary wins whenever $\tau = [ia/x]_1^v$, which happens with probability $2^{v-\ell-(n-m)}/2 = \delta/2$, because all but $\log \frac{1}{\delta}$ bits of τ are definitely correct as long as the last bit of ia is 0.

The above discussion gives us the following result.

Theorem 2. *There exists a basis for $GF(2^{n-v})$ such that for any integer m there exists a distribution W of min-entropy m for which the post-application robustness of the construction from [DKRS06, Theorem 3] can be violated with probability at least $\delta/2$, where v is set as required for robustness δ by the construction (i.e., $v = (n-\ell)/2$ for $\ell = (2m - n - 2\log \frac{1}{\delta})/3$).*

Note that our lower bound uses a specific representation of field elements, and hence does not rule out that for some particular representation of field elements, a lower value of v and, therefore, a higher value of ℓ is possible. However, a security proof for a lower value of v would have to then depend on the properties of that particular representation and would not cover the construction of [DKRS06] in general.

5 Tolerating Binary Hamming Errors

We now consider the scenario where Bob has a string w' that is close to Alice's input w (in the Hamming metric). In order for them to agree on a random string, Bob would first have recover w from w'. To this end, Alice could send the secure sketch $s = \mathsf{SS}(w)$ to Bob along with (i, σ). To prevent an undetected modification of s to s', she could send an authentication of s (using w as the key) as well. The nontriviality of making such an extension work arises from the fact that modifying s to s' also gives the adversary the power to influence Bob's verification key $w^* = \mathsf{SRec}(w', s')$. The adversary could perhaps exploit this circularity to succeed in an active attack (the definition of standard authentication schemes only guarantee security when the keys used for authentication and verification are the same).

We break this circularity by exploiting the algebraic properties of the Hamming metric space, and using authentication secure against algebraic manipulation [DKRS06, CDF$^+$08]. The techniques that we use are essentially the same as used in [DKRS06], but adapted to our construction. We present the construction here and then discuss the exact properties that we use in the proof of security.

Construction. Let $\mathcal{M}$ be the Hamming metric space on $\{0,1\}^n$. Let W be a distribution of min-entropy m over $\mathcal{M}$. Let $s = \mathsf{SS}(w)$ be a deterministic, linear secure sketch; let $|s| = k$, $n' = n - k$. Assume that SS is a surjective linear function (which is the case for the syndrome construction for the Hamming metric mentioned in Section 2). Therefore, there exists a $k \times n$ matrix S of rank k such that $\mathsf{SS}(w) = \mathsf{S}w$. Let $\mathsf{S}^\perp$ be an $n' \times n$ matrix such that $n \times n$ matrix $\left(\frac{\mathsf{S}}{\mathsf{S}^\perp}\right)$ has full rank. We let $\mathsf{SS}^\perp(w) = \mathsf{S}^\perp(w)$.

To compute $\mathsf{Gen}(w)$, let $s = \mathsf{SS}(w)$, $c = \mathsf{SS}^\perp(w)$; $|c| = n'$. We assume that n' is even (if not, drop one bit of c, reducing its entropy by at most 1). Let a be the first half of c and b the second. View a, b as elements of $\mathbb{F}_{2^{n'/2}}$. Let $L = 2\lceil \frac{k}{n'} \rceil$ (it will important for security that L is even). Pad s with 0s to length $Ln'/2$, and then split it into L bit strings $s_{L-1}, \ldots, s_0$ of length $n'/2$ bits each, viewing each bit string as an element of $\mathbb{F}_{2^{n'/2}}$. Select $i \leftarrow \mathbb{F}_{2^{n'/2}}$. Define $f_{s,i}(x) = x^{L+3} + x^2(s_{L-1}x^{L-1} + s_{L-2}x^{L-2} + \cdots + s_0) + ix$. Set $\sigma = [f_{s,i}(a) + b]_1^v$, and output $P = (s, i, \sigma)$ and $R = [f_{s,i}(a) + b]_{v+1}^{n'/2}$.

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

$\mathsf{Rep}(w', P' = (s', i', \sigma'))$:
1. Compute $w^* = \mathsf{SRec}(w', s')$
   - Verify that $\mathsf{dis}(w^*, w') \le t$ and $\mathsf{SS}(w^*) = s'$. If not, output $\perp$.
2. Let $c' = \mathsf{SS}^{\perp}(w^*)$. Parse c' as $a'||b'$.
3. Compute $\sigma^* = [f_{s',i'}(a') + b']_1^v$.
   - Verify that $\sigma^* = \sigma'$. If so, output $R = [f_{s',i'}(a') + b']_{v+1}^{n'/2}$, else output $\perp$.

In the theorem statement below, let B denote the volume of a Hamming ball or radius t in $\{0,1\}^n$ ($\log B \le nH_2(t/n)$ [MS77, Chapter 10, §11, Lemma 8] and $\log B \le t\log(n+1)$ [DKRS06]).

Theorem 3. *Assume* SS *is a deterministic linear* $(m, m-k, t)-$*secure sketch of output length* k *for the Hamming metric on* $\{0,1\}^n$*. Setting* $v = (n-k)/2 - l$*, the above construction is an* (m, l, t, ε) *fuzzy extractor with robustness* δ *for any* m, l, t, ε *satisfying* $l \le m - n/2 - k - \log B - \log\left(2\left\lceil \frac{k}{n-k} \right\rceil + 2\right) - \log\frac{1}{\delta}$ *as long as* $m \ge \frac{1}{2}(n+k) + 2\log\frac{1}{\varepsilon}$.

Again, if $m < \frac{1}{2}(n+k) + 2\log\frac{1}{\varepsilon}$, the construction can be modified, as shown in Section 5.1.

Proof. Extraction. Our goal is to show that R is nearly uniform given $P = (i, s, \sigma)$. To do so, we first note that for every s, the function $h_i(c) = (\sigma, R)$ is a universal hash family. Indeed for $c \ne c'$ there is a unique i such that $h_i(c) = h_i(c')$ (since $i(a - a')$ is fixed, like in the errorless case). We also note that $\widetilde{\mathbf{H}}_\infty(c \mid \mathsf{SS}(W)) \ge \widetilde{\mathbf{H}}_\infty(c, \mathsf{SS}(W)) - k = \mathbf{H}_\infty(W) - k = m - k$ by Lemma 2. Because $|(R, \sigma)| = n'/2$, Lemma 3 (or, more precisely, its generalization mentioned in the paragraph following the lemma, needed here because h_i depends on s) gives us

$$\mathbf{SD}\left((R, P), U_{|R|} \times \mathsf{SS}(W) \times U_{n'/2} \times U_v\right) \le \varepsilon/2$$

for $n'/2 \le m - k + 2 - 2\log(2/\varepsilon)$. This is equivalent to saying that (R, P) is $2^{(n'/2-m+k)\frac{1}{2}-1}$-close to $U_{|R|} \times \mathsf{SS}(W) \times U_{n'/2} \times U_v$.

Applying Lemma 1 to $A = R$, $B = P$, $C = U_{n'/2-v}$, $D = \mathsf{SS}(w) \times U_{n'/2} \times U_v$, we get that (R, P) is ε-close to $U_{\frac{n'}{2}-v} \times P$, for $\varepsilon = 2^{(\frac{n'}{2}-m+k)/2}$.

From here it follows that for extraction to be possible, $m \ge \frac{1}{2}(n+k) + 2\log\frac{1}{\varepsilon}$.

Post-Application Robustness. In the post-application robustness security game, the adversary $\mathcal{A}$ on receiving $(P = (s, i, \sigma), R)$ (generated according to procedure Gen) outputs $P' = (s', i', \sigma')$, and is considered successful if $(P' \ne P) \wedge \mathsf{Rep}(w', s') \ne \perp$. In our analysis, we will assume that $(i', s') \ne (i, s)$. We claim that this does not reduce $\mathcal{A}$'s success probability. Indeed, if $(i', s') = (i, s)$ then, c' computed within Rep will equal c. So, for $P' \ne P$ to hold, $\mathcal{A}$ would have to output $\sigma' \ne \sigma$.

However, when $(i', c', s') = (i, c, s)$, Rep would compute $\sigma^* = \sigma$, and therefore would output $\perp$ unless $\sigma' = \sigma$.

In our analysis, we allow $\mathcal{A}$ to be deterministic. This is without loss of generality since we allow an unbounded adversary. We also allow $\mathcal{A}$ to arbitrarily fix i. This makes the result only stronger since we demonstrate robustness for a worst-case choice of i.

Since i is fixed and $\mathcal{A}$ is deterministic, the $\mathsf{tr} = (i, s, \sigma, R, i', s', \sigma')$ is determined completely by (s, σ, R). Recall that the prime challenge in constructing a robust fuzzy extractor was that $\mathcal{A}$ could somehow relate the key used by Rep to verify σ' to the authentication key that was used by Gen to come up with σ. As was done in [DKRS06], we will argue security of our construction by showing that the MAC scheme implicitly used in our construction remains unforgeable even when $\mathcal{A}$ could force the verification key to be at an offset (of her choice) from the authentication key. We will formalize such an argument by assuming that $\mathcal{A}$ learns $\Delta = w' - w$. Recall that $w^* = \mathsf{SRec}(w', s')$ and $c' = a'||b' = \mathsf{SS}^{\perp}(w^*)$. The following claim that was proven in [DKRS06] states that given (Δ, s), $\mathcal{A}$ can compute the offsets $\Delta_a = a' - a, \Delta_b = b' - b$ induced by her choice of s'.

Claim 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', b'.*

In other words, she can compute the offset between the authentication key that Gen used to come up with σ and the verification key that Rep will use to verify σ'. We will now argue that as long as W has sufficient min-entropy, even knowing the offset does not help $\mathcal{A}$ succeed in an active attack. Recall that since i is arbitrarily fixed by $\mathcal{A}$, $\mathcal{A}$'s success depends on w, w', or, alternatively, on w, Δ. Fix some Δ. For any particular tr, let $\mathsf{Succ}_{\mathsf{tr},\Delta}$ be the event that the transcript is tr and $\mathcal{A}$ wins, i.e., that $f_{s,i}(a) + b = \sigma||R \wedge [f_{s',i'}(a') + b']_1^v = \sigma' \wedge \mathsf{SS}(w) = s$, conditioned on the fact that $w' - w$ is Δ. We denote by $\mathsf{Bad}_{\mathsf{tr},\Delta}$ the set of w that make $\mathsf{Succ}_{\mathsf{tr},\Delta}$ true. We now partition the set $\mathsf{Bad}_{\mathsf{tr},\Delta}$ into 2^ℓ disjoint sets, indexed by $R' \in \{0,1\}^\ell$:

$$\begin{aligned}
\mathsf{Bad}^{R'}_{\mathsf{tr},\Delta} &\stackrel{\text{def}}{=} \{w \,|\, w \in \mathsf{Bad}_{\mathsf{tr},\Delta} \wedge [f_{s',i'}(a') + b']^{\ell}_{v+1} = R'\} \\
&= \{w \,|\, (f_{s,i}(a) + b = \sigma||R) \wedge (f_{s',i'}(a') + b' = \sigma'||R') \wedge \mathsf{SS}(w) = s\}.
\end{aligned}$$

By Claim 1, fixing (tr, Δ), also fixes Δ_a, Δ_b. It follows that every $w \in \mathsf{Bad}^{R'}_{\mathsf{tr},\Delta}$ needs to satisfy

$$f_{s,i}(a) - f_{s',i'}(a + \Delta_a) = (\Delta_b + \sigma - \sigma')||(R - R') \wedge \mathsf{SS}(w) = s.$$

For a given tr, Δ, R', the right hand side of the first equation takes a fixed value. Let us now focus on the polynomial $f_{s,i}(a) - f_{s',i'}(a + \Delta_a)$. We will consider two cases:

- $\Delta_a = 0$: In this case, $f_{s,i}(x) - f_{s',i'}(x)$ is a polynomial in which a coefficient of degree 2 or higher is nonzero if $s \neq s'$ and a coefficient of degree 1 or higher is nonzero if $i \neq i'$.

- $\Delta_a \neq 0$: Observe that the leading term of the polynomial is $((L+3) \bmod 2)\Delta_a x^{L+2}$. Since we forced L to be even, the coefficient of the leading term is nonzero, making $f_{s,i}(x) - f_{s',i'}(x+\Delta_a)$ a polynomial of degree $L + 2$.

Therefore, in either case, the $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$ is a nonconstant polynomial of degree at most $L + 2$. A nonconstant polynomial of degree d can take on a fixed value at most d times. It, therefore, follows that there are at most $L + 2$ values of a such that $f_{s,i}(a) - f_{s',i'}(a + \Delta_a) = (\Delta_b + \sigma - \sigma')||(R - R')$. Each such a uniquely determines $b = (\sigma||R) - f_{s,i}(a)$. And w is uniquely determined by $c = a||b = \mathsf{SS}^{\perp}(w)$ and $s = \mathsf{SS}(w)$. Therefore, there are at most $L+2$ values of w in the set $\mathsf{Bad}^{R'}_{\mathsf{tr},\Delta}$ i.e, $|\mathsf{Bad}^{R'}_{\mathsf{tr},\Delta}| \leq L+2$. Since $\mathsf{Bad}_{\mathsf{tr},\Delta} = \bigcup_{R' \in \{0,1\}^\ell} \mathsf{Bad}^{R'}_{\mathsf{tr},\Delta}$, we get $|\mathsf{Bad}_{\mathsf{tr},\Delta}| \leq (L+2)2^\ell = (L+2)2^{n'/2-v}$. Thus, $\Pr_w[\mathsf{Succ}_{\mathsf{tr},\Delta}] \leq |\mathsf{Bad}_{\mathsf{tr}}| 2^{-\mathbf{H}_\infty(w|\Delta)} \leq (L+2)2^{n'/2-v-\mathbf{H}_\infty(w|\Delta)}$.

To find out the probability $\Pr_w[\mathsf{Succ}_\Delta]$ that $\mathcal{A}$ succeeds conditioned on a particular Δ, we need to add up $\Pr_w[\mathsf{Succ}_{\mathsf{tr},\Delta}]$ over all possible transcripts. Recalling that each transcript is determined by σ, R and s and hence there are $2^{n'/2+k}$ of them, and that $n'+k = n$, we get $\Pr_w[\mathsf{Succ}_\Delta] \leq (L+2)2^{n-v-\mathbf{H}_\infty(w|\Delta)}$.

Finally, the probability of adversarial success it at most

$$\mathop{\mathbf{E}}_{\Delta} \Pr_{w}[\mathsf{Succ}_\Delta] \leq (L+2)2^{n-v-\widetilde{\mathbf{H}}_\infty(w|\Delta)}\,.$$

In particular, if the errors Δ are independent of w, then $\widetilde{\mathbf{H}}_\infty(w|\Delta) = \mathbf{H}_\infty(w) = m$, and the probability of adversarial success is at most $(L+2)2^{n-v-m}$. In the worst case, however, the entropy of w may decrease at most by the number of bits needed to represent Δ. Let B be the volume of the hamming ball of radius t in $\{0,1\}^n$. Then, Δ can be represented in $\log B$ bits and $\widetilde{\mathbf{H}}_\infty(w|\Delta) \geq m - \log B$, by Lemma 2. From here it follows that

$$\Pr[\mathcal{A}'s\ success] \leq B(L+2)2^{n-v-m}$$

To achieve $\delta-$robustness, we want $B(L+2)2^{n-v-m} \leq \delta$ i.e., $v \geq n - m + \log B + \log(L+2) + \log\frac{1}{\delta}$. Setting $v = n - m + \log B + \log(L+2) + \log\frac{1}{\delta}$, and using $L = 2\lceil\frac{k}{n-k}\rceil$ it follows that

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

□

5.1 Getting Closer to Uniform

If ε is so low that $m \geq \frac{1}{2}(n+k) + 2\log\frac{1}{\varepsilon}$ does not hold, we can modify our construction just as we did in section 3.1, by shortening R by $\beta = \frac{1}{2}(n+k) + 2\log\frac{1}{\varepsilon} - m$. That is, keep $v = n - m + \log B + \log(L+2) + \log\frac{1}{\delta}$ fixed and let $R = [f_{s,i}(a)+b]_{v+1}^{\ell+v}$, where $\ell \leq n/2 - v - \beta$.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation grants CCF-0515100 and CNS-0546614.

References

[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.

[BBR88] C. Bennett, G. Brassard, and J. 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.

[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.

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

[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.

[DKRS08] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. Manuscript, 2008.

[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.

[HILL99] J. Hrastad, 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.

[Mau93] Ueli Maurer. Protocols for secret key agreement by public discussion based on common information. In Douglas R. Stinson, editor, *Advances in Cryptology—CRYPTO '93*, volume 773 of *LNCS*, pages 461–470. Springer-Verlag, 22–26 August 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] U. Maurer and S. 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] U. Maurer and S. 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.

[PRTG02] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297:2026–2030, 20 September 2002.

[RW03] R. Renner and S. 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, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 109–125. Springer-Verlag, 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.

[Wic08] Daniel Wichs. Private Communication, 2008.

[Wol98] S. 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.