Key Agreement from Close Secrets over Unsecured Channels

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

January 27, 2009

Abstract

We consider information-theoretic key agreement between two parties sharing somewhat different versions of a secret w that has relatively little entropy. Such key agreement, also known as information reconciliation and privacy amplification over unsecured channels, was shown to be theoretically feasible by Renner and Wolf (Eurocrypt 2004), although no protocol that runs in polynomial time was described. We propose a protocol that is not only polynomial-time, but actually practical, requiring only a few seconds on consumer-grade computers.

Our protocol can be seen as an interactive version of robust fuzzy extractors (Dodis et al., Crypto 2006). While robust fuzzy extractors, due to their noninteractive nature, require w to have entropy at least half its length, we have no such constraint. In fact, unlike in prior solutions, in our solution the entropy loss is essentially unrelated to the length or the entropy of w, and depends only on the security parameter.

1 Introduction

We consider the problem of information-theoretic key agreement between two parties that initially possess only correlated weak secrets. At the start of the protocol, Alice has a string w, Bob has w' that is similar, but not identical, to w, and the adversary Eve's information about w is incomplete. The goal is for Alice and Bob to agree on a shared secret key k about which Eve has no information. Security has to hold even in the case of active Eve, i.e., one who can perform the (wo)man-in-the-middle attack. It is important that the output k be as long as possible given the entropy of w (the difference between the length of k and the entropy of w is known as the entropy loss).

This setting arises, for example, when Alice and Bob have access to a (possibly) noisy channel that can be partially eavesdropped by Eve; or when a trusted server (Alice) stores the biometric of a user (Bob), and the user subsequently uses his fresh biometric reading to authenticate himself to the server; or when Alice and Bob are mobile nodes wanting to authenticate each other based on the fact that their knowledge of a location is greater than Eve's (e.g., if they are much closer to a particular location than Eve, and thus are able to observe it at higher resolution).

Renner and Wolf [RW04] proposed the first protocol to solve this problem. This protocol (described in [RW04, Corollary 9]) is very general: it does not require proximity between w and w', but only requires, roughly, that information that w and w' contain about each other is more than the information that Eve has about them. However, the price for this generality is that the protocol is not practical as presented: the round complexity is quite high, and the local running time of each party is not even polynomial.

Renner and Wolf mention briefly, however [RW04, Section 2.2], that local running time can be made polynomial through the use of error-correcting techniques when w and w' "are bitstrings which differ in a certain limited number of positions" (that is, are close in the Hamming metric). Indeed, subsequently, Dodis et al. [DKRS06] used error-correcting techniques to propose a protocol that is computationally efficient not only for the Hamming metric, but also for the set difference metric. Moreover, their protocol has just a single message from Alice to Bob.

Unfortunately, the price for such high efficiency is high entropy loss: if the length of w is n and its entropy (after the error-correcting information) is m, then the protocol of Dodis et al. cannot output k longer than $m - (n - m)$. In particular, if the entropy of w is less than half its length, it achieves nothing (this is unavoidable in all single-message protocols [DKRS06, KR08a], as pointed out in [DKRS06] and shown in [DW08]).

Our Contribution We build on the results of [RW04] and [DKRS06] by proposing a protocol that is efficient for both parties and has both lower round complexity and lower entropy loss than the protocol of [RW04]. Our analysis decouples security from the length n of w, thus offering a flexible tradeoff between security and performance. Without going into details of all the parameters, for security 2^{-L}, the length of k in our protocol is about $m - L^2/2 - O(L \log L + L \log n)$ and the number of messages exchanged between Alice and Bob is $L + \log n + 5$. More details and a more careful performance comparison are provided in Section 2.

Our protocol is more general than the work of [DKRS06] not only in the entropy requirement, but also in the kinds of differences between w and w' it can handle. Specifically, it can handle any metric that has secure sketches [DORS08] (see Section 3.2) that do not lose too much entropy (in particular, therefore, our protocol tolerates Hamming, set difference, edit distance [DORS08] and point-set difference [CL06] errors). Thus, while Renner and Wolf showed feasibility of key agreement from correlated information, and Dodis et al. showed its practicality for certain restricted settings, we demonstrate its practicality for a broad class of settings.

Implementation Results We implemented our protocol using Shoup's NTL [Sho01], although we have not performed careful code optimization and did not include any improvements of Section 4.2. The protocol was tested for $L = 80$ and $n = 100,000$ on a LAN with Alice and Bob running on a 2.4Ghz Intel Pentium 4 and a WAN with Alice running on a 2.4Ghz Intel Xeon instead. The running times over a WAN and LAN were nearly the same, both less than 5 seconds. Of the total running time, approximately 1.5 seconds were spent by each party on computation and an additional 1 second was spent in total communication costs. The improvements in Section 4.2 will reduce the running time further (although the impact of these improvements on the number of rounds and the amount of computation is easy to understand, it is difficult to say how much the actual total running times will decrease).

Other related work Variants of this problem have been extensively studied, under the names "information reconciliation," "privacy amplification," and "fuzzy extractors." Without providing an exhaustive overview of the literature, we note here the most closely related work. Information-theoretic security against active Eve was achieved by Maurer, Renner, and Wolf [Mau97, MW97, Wol98, MW03, RW03] in the restricted setting when $w = w'$ or when w, w', and Eve's information come from repeated independent identically distributed experiments. Boyen et al [BDK+05] removed those restrictions, instead requiring that w and w' be close in some metric that has secure

sketches, but achieved only computational security. One of their solutions relies on the random oracle model, and the other on computational assumptions necessary to enable password-base authenticated key agreement.

The starting point for our work is the same as for [RW04]: a protocol, also by Renner and Wolf [RW03], designed for the case of $w = w'$. We modify it for the case of $w \neq w'$ in a way that improves it even for the case of $w = w'$, and provide a more careful, concrete security analysis for it ([RW03] provides only an asymptotic analysis that works when $n \to \infty$).

2 Overview of the Result

Notation, Distributions, Entropy 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

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

(they are said to be ε-close if $\mathbf{SD}(X_1, X_2) \leq \varepsilon$). 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 $2^{-\widetilde{\mathbf{H}}_\infty(W|E)}$.

Throughout this paper, for any string x, we use the notation λ_x to denote its length and h_x to denote its entropy (i.e, $\mathbf{H}_\infty(X)$).

Model We now define our goal, by modifying the noninteractive robust fuzzy extractor definition of [DKRS06]. An *Interactive Robust Fuzzy Extractor* protocol allows two parties, Alice and Bob, holding instances w, w' of correlated random variables W, W' that are guaranteed to be close but not identical, to agree on a secret key. We assume that w and w' are within distance at most η in some underlying metric space. The correctness of the protocol guarantees that when the protocol is executed in the presence of a passive adversary (one who does not interfere with messages between Alice and Bob), the parties end up agreeing on the same secret key, as long as $\mathsf{dis}(w, w') \leq \eta$.

The security of the protocol guarantees that even when the protocol is executed in the presence of an active adversary, who interferes with messages arbitrarily, if both parties accept, then they agree on a key that is uniformly random from the adversary's point of view. Moreover, if only one party accepts, then its key is still uniformly random from the adversary's point of view. (As was observed in, for example, [Wol98] and [Sho99], we cannot require that if one party rejects, then so does the other party, because an active adversary can always replace the last message with an invalid one—by that time, the sender of that message must have already accepted, while the recipient will reject.)

We now present the formal definition of an interactive robust fuzzy extractor. Let $w, w' \in \{0,1\}^n$ chosen according to distributions W, W' be the secret values held by Alice and Bob respectively.

Call three correlated random variables (W, W', E) (where W and W' range over some metric space $\mathcal{M}$) *suitable* if $\widetilde{\mathbf{H}}_\infty(W \mid E) \geq h_W$ and $\Pr_{(w,w')\leftarrow(W,W')}[\mathsf{dis}(w, w') \leq \eta] = 1$.

Let Protocol (A, B) be executed in the presence of an active adversary Eve. Let C_a be the random variable describing A's view of the communication when (A, B) is executed in the presence of Eve.Likewise, define C_b. (We will use c_a, c_b to denote specific values of these variables.) We denote the private coins of Alice and Bob by r_a and r_b respectively. Alice's output will be denoted by $\mathsf{k_A} = \mathsf{A}(w, c_a, r_a)$, and Bob's by $\mathsf{k_B} = \mathsf{B}(w', c_b, r_b)$ (if successful, both will be of length λ_k; rejection will be denoted by a special symbol $\perp$). Let $C = C_a \cup C_b \cup E$ be Eve's view of the protocol; because Eve is computationally unbounded, we can assume she is deterministic.

Definition 1. An interactive protocol (A, B) played by Alice and Bob on a communication channel fully controlled by an adversary Eve, is an $(\mathcal{M}, h_W, \lambda_k, \eta, \delta, \epsilon)$-interactive robust fuzzy extraction protocol if it satisfies the following properties whenever (W, W', E) are suitable:

1. Correctness. If Eve is passive, then $\Pr[\mathsf{k_A} = \mathsf{k_B}] = 1$.
2. Robustness. The probability that the following experiment outputs "Eve wins" is at most δ: sample (w, w', e) from (W, W', E); let c_a, c_b be the communication upon execution of (A, B) with $Eve(e)$ actively controlling the channel, and let $\mathsf{A}(w, c_a, r_a) = \mathsf{k_A}, \mathsf{B}(w', c_b, r_b) = \mathsf{k_B}$. Output "Eve wins" if $(\mathsf{k_A} \neq \mathsf{k_B} \wedge \mathsf{k_A} \neq \perp \wedge \mathsf{k_B} \neq \perp)$.
3. Extraction. Letting C denote an active Eve's view of the protocol,

$$\mathbf{SD}\left((\mathsf{k_A}, C, E \mid \mathsf{k_A} \neq \perp), (U_{\lambda_k}, C, E)\right) \leq \epsilon \text{ and } \mathbf{SD}\left((\mathsf{k_B}, C, E \mid \mathsf{k_B} \neq \perp), (U_{\lambda_k}, C, E)\right) \leq \epsilon\,.$$

Our Protocol Before going into details in subsequent sections, we present here a high-level overview of our protocol. We start with an authentication sub-protocol Auth presented in [RW03] that achieves the following: using the secret w that is common to Alice and Bob, it allows Alice to send to Bob an authentic (but nonsecret) message M of length λ_M bit-by-bit in $2\lambda_M$ messages. Alice and Bob [RW03] can use this sub-protocol in order to agree on a key k as follows: they use Auth to get an extractor seed s from Alice to Bob, and then extract k from w using s.[1]

We modify this protocol by using Auth to authenticate a MAC key instead of an extractor seed. The MAC key, in turn, is used to authenticate the extractor seed s (which can be done very efficiently using simple information-theoretic MACs). This seems counterintuitive, because Auth reveals what is being authenticated, while MAC keys need to remain secret. The insight is to *use* the MAC key *before* Auth begins.[2] Our modification is beneficial for three reasons. First, MAC keys can be made shorter than extractor keys, so Auth is used on a shorter string, thus reducing the number of rounds and the entropy loss. Second, this modification allows us to use the same MAC key to authenticate not only the extractor seed s, but also the error-correction information (the so-called "secure sketch" of w [DORS08]) in the case Bob's w' is different from Alice's w. Third, because there are MACs that are secure even against (limited) key modification [DKRS06, CDF$^+$08], we can lower the security parameters in Auth, further increasing efficiency and reducing entropy loss.

The rest of the paper is devoted to filling in the details of the above overview, including smaller improvements not discussed here, and proving the following theorem.

[1] For technical reasons, since the adversary can modify message of Auth, she may have some information about the string extracted from w; this problem is easily handled, see Section 4.

[2] This idea has been used before in several contexts; to the best of our knowledge it was first used in [Che97] in the context of secure link state routing.

Theorem 1. *Given an $[n, \kappa, 2\eta+1]_2$ linear error correcting code, the protocol presented in Section 4 is an $(\mathcal{M}, h_W, \lambda_k, \eta, \delta, \epsilon)$-*interactive robust fuzzy extraction protocol*, where $\mathcal{M}$ is the Hamming space over $\{0,1\}^n$ with the following parameters: Setting security $\delta = 2^{-L}$, the protocol can extract $\lambda_k = h_W - (n-\kappa) - 2\log\frac{1}{\varepsilon} - (L^2/2 + O(L(\log n + \log L)))$ bits (assuming $n < 2^L$ and $\lambda_k - (n-\kappa) + 2\log\frac{1}{\varepsilon} > 10L$). The protocol involves an exchange of $L + \log n + 5$ messages between the two parties.*

The constant hidden by the O in the entropy loss is small, with $O(L(\log n + \log L))$ really being less than $3L\log 2L + \frac{1}{2}L\log n + 3(\log 8L)(\log 16n)$.

We obtain similar results for other metric spaces, with the only difference being that $n-\kappa$ in the entropy loss gets replaced by the entropy loss of the secure sketch for that metric space (see Section 3.2).

Comparison with Prior Work When no error-correction is needed (i.e., $w = w'$ and $\eta = 0$), then $n - \kappa = 0$, and we get an improvement of the result of [RW03]. The result of [RW03] sets $L = \Theta(\sqrt{n}/\log n)$ and loses $\Theta(n/\log n)$ bits of entropy. This can be seen in the description of protocol Auth in [RW03], which has $\Theta(\sqrt{n})$ rounds, each losing $\Theta(L)$ bits. Our protocol has only $\Theta(L)$ rounds, with each also losing $\Theta(L)$ bits. Thus, our result is a $\Theta(\log n)$-factor improvement in efficiency and entropy loss for the same security (moreover, the constant hidden by Θ, although difficult to compute exactly, is substantial, likely bigger than $\log n$ in real applications).

A precise comparison with [RW04], which uses [RW03] as a building block and adds error-correction, is even more complicated. Our advantage in the number of rounds remains the same, though the constant factor improves even further. To compare the entropy loss, we can fix the secure sketch code used in our protocol (which can be based on any linear error-correcting code) to the one implicitly used in [RW04]. In that case, the entropy loss due to added error-correction is asymptotically the same for our protocol and for the protocol of [RW04], though the constant in our protocol is substantially lower. On the other hand, an important advantage of our protocol is that we can choose a code that is efficiently decodable, in which case the entropy loss due to error-correction may increase, but the protocol will run in polynomial-time.

We now compare our result to the construction of [DKRS06]. The advantage of the [DKRS06] construction is that it takes only a single message and the entropy loss is linear in L rather than quadratic. The disadvantage is that it loses additional $n - h_W$ bits of entropy, which means that it is most effective when W has very high entropy. In particular, it becomes inapplicable when $h_W \le n/2$.

3 Building Blocks

3.1 Extractors

Because in this paper Eve is always 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 2. 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 $\widetilde{\mathbf{H}}_\infty(W \mid E) \ge m$, we have $\mathbf{SD}\left((\mathsf{Ext}(W; X), X, E), (U_l, X, E)\right) \le \varepsilon$ where X is the uniform distribution over $\{0,1\}^r$.

We should note that some strong extractors (in particular, universal hashing [CW79, HILL99]) are already average-case extractors, and any strong extractor can be made average-case with a slight increase in input entropy [DORS08, Section 2.5].

The following (new) lemma shows that strings extracted by average-case extractors have high average min-entropy, even given the seed. The proof is in Appendix A.

Lemma 1. *Let* Ext *be a an average-case* (n, m, l, ε)*-strong extractor. Then if* $\widetilde{\mathbf{H}}_\infty(W \mid E) \geq m$*, and* W *consists of* n*-bit strings,* $\widetilde{\mathbf{H}}_\infty(\mathsf{Ext}(W, X) \mid X, E) \geq \min\left(l, \log \frac{1}{\varepsilon}\right) - 1$.

3.2 A Variation on Secure Sketches

So far, we have presented the error-correcting information that Alice sends to Bob in the first message as a secure sketch. Actually, we need a slight variant on secure sketches, one that provides some resilience even when the sketch is modified. The requires a different definition than the definition of [DORS08], though it turns out that known constructions need to be modified only slightly to satisfy it.

Secure sketches, defined in [DORS08], provide two algorithms: "generate" (Gen) that takes an input w and produces a sketch P and "recover" (Rec) that outputs w from the sketch P and any w' sufficiently close to w. Their security guarantees that some entropy remains in w even given P. Secure sketches provide no guarantees when P has been tampered with, while we need to make sure that the output of Rec still has entropy. Thus, we need to add a weak form of robustness (i.e., resilience to active attack) to secure sketches. At the same time, we do not need a full recovery of the original w: we will be satisfied if both Gen and Rec produce some string R that preserves some of the entropy of w. In that way, our new primitive is like a fuzzy extractor, except we do not require that R be uniform, merely that it have entropy. In keeping with extractor literature terminology [CRVW02], we call the primitive a *weakly robust fuzzy conductor* because it conducts entropy from w to R and is robust against active attacks on P. Because we no longer recover the original w but rather reproduce the same R, we rename Rec into Rep.

Let $\mathcal{M}$ be a metric space with distance function dis. Suppose $(\mathsf{Gen}, \mathsf{Rep})$ are two procedures, where $\mathsf{Gen}(w)$, for $w \in \mathcal{M}$, outputs an extracted string $R \in \{0,1\}^*$ and a helper string $P \in \{0,1\}^*$, and $\mathsf{Rep}(w', P')$, for $w' \in \mathcal{M}, P' \in \{0,1\}^*$, outputs $R' \in \{0,1\}^*$.

Definition 3. The procedures $(\mathsf{Gen}, \mathsf{Rep})$ are an $(\mathcal{M}, h_W, h_R, h_{R'}, \eta)$-weakly robust fuzzy conductor if they satisfy the following properties:

1. Error-Tolerance. If $\mathsf{dis}(w, w') \leq \eta$ and R, P were generated by $(R, P) \leftarrow \mathsf{Gen}(w)$, then $\mathsf{Rep}(w', P) = R$.
2. Security of Gen. For any suitable (W, W', E), the string R has high entropy even for those who observe P and E: if $(R, P) \leftarrow \mathsf{Gen}(W)$, then $\widetilde{\mathbf{H}}_\infty(R \mid E, P) \geq h_R$.
3. Security of Rep. Even if the adversary modifies P, the string produced by Rep has high entropy: for all (adversarial) functions A and suitable (W, W', E), if $(R, P) \leftarrow \mathsf{Gen}(W)$, $P' \leftarrow A(P, E)$, and $R' \leftarrow \mathsf{Rep}(W', P')$, then $\widetilde{\mathbf{H}}_\infty(R' \mid E, P) \geq h_{R'}$.

We can build weakly robust fuzzy conductors out of any secure sketch $(\mathsf{SS}, \mathsf{Rec})$. (Secure sketches, defined in [DORS08], allow the recovery of w from a close string w'). We use the secure sketch constructions of [DORS08] to build weakly robust fuzzy conductors for Hamming, set difference, and edit distance metrics. Namely, in Appendix B, we easily obtain

- for Hamming distance over an alphabet of size F, given an $[n, \kappa, 2t+1]$ linear error-correcting code for the alphabet, we get $h_R = h_W - (n-\kappa)\log F$, $h_{R'} = h_W - 2(n-\kappa)\log F$, and $\eta = t$.
- for set difference, with sets whose elements come from a universe of size U, we get $h_R = h_W - \eta\log(U+1)$ and $h_{R'} = h_W - 2\eta\log(U+1)$ for any η.
- for edit distance over an alphabet of size we get $h_R = h_W - \lceil \frac{n}{c} \rceil \log(n-c+1) - \alpha$, and $h_{R'} = h_W - \lceil \frac{n}{c} \rceil \log(n-c+1) - 2\alpha$, where $\alpha = (2c-1)\eta\lceil\log(F^c+1)\rceil$, for any constant c and η.

3.3 One-time message authentication codes (MACs).

One-time MACs allow information-theoretic authentication of a message using a key shared in advance.

Definition 4. A function family $\left\{\mathsf{MAC}_k : \{0,1\}^{\lambda_M} \to \{0,1\}^{\lambda_\sigma}\right\}$ is a δ-secure one-time deterministic MAC for messages of length λ_M with tags of length λ_σ if for any $M \in \{0,1\}^{\lambda_M}$ and any function (adversary) $A : \{0,1\}^{\lambda_\sigma} \to \{0,1\}^{\lambda_M} \times \{0,1\}^{\lambda_\sigma}$,

$$\Pr_k \left[\mathsf{MAC}_k(M') = \sigma' \wedge M' \neq M \mid (M', \sigma') = A(\mathsf{MAC}_k(M))\right] \leq \delta\,.$$

MAC Construction. We will use the following standard MAC technique [dB93, Tay93, BJKS93]. View the key k as two values, a and b, of λ_σ bits each. Split the message M into c chunks $M_0, \ldots, M_{c-1}$, each λ_σ bits long, and view these as coefficients of a polynomial $\tilde{M}(x) \in \mathbb{F}_{2^{\lambda_\sigma}}[x]$ of degree $c-1$. Then $\mathsf{MAC}_k(M) \stackrel{\text{def}}{=} a\tilde{M}(a) + b$. This is a $\lceil \lambda_M/\lambda_\sigma \rceil\, 2^{-\lambda_\sigma}$-secure message authentication code.

This construction has two properties that are particularly important to us. First, its key length is close to optimal (it is not hard to show that $\lambda_\sigma \geq \log\frac{1}{\delta}$—else, adversary could simply guess a tag; and $|k| \geq 2\log\frac{1}{\delta}$—else, there would be fewer than $\frac{1}{\delta}$ tags for M' given one of the $\frac{1}{\delta}$ tags for M). Second, it is secure even when the adversary knows something about the key, with security degrading according to the amount of information adversary knows (this kind of security was first addressed in [MW97]). Intuitively, the security of this MAC is roughly the entropy of the key minus half the key length. More formally,

Proposition 1. Let (K, E) be a joint distribution. Then for all (adversarial) functions M with λ_M-bit outputs and A,

$$\Pr_{(k,e)\leftarrow(K,E)} \left[\mathsf{MAC}_k(M') = \sigma' \wedge M' \neq M \mid (M', \sigma') = A(\mathsf{MAC}_k(M(e)), e)\right] \leq \left\lceil \frac{\lambda_M}{\lambda_\sigma} \right\rceil 2^{\lambda_\sigma - \widetilde{\mathbf{H}}_\infty(K|E)}\,.$$

We defer the proof of this proposition to Appendix C. We note that its proof becomes simpler (than similar prior proofs) if we use the notion of average min-entropy. In particular, we will use following lemma [DORS08] that states that average min-entropy of a variable from the point of view of an adversary doesn't decrease by more than the number of bits (correlated with the variable) observed by the adversary.

Lemma 2. *If B has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(A \mid B, E) \geq \widetilde{\mathbf{H}}_\infty(A, B \mid E) - \lambda \geq \widetilde{\mathbf{H}}_\infty(A \mid E) - \lambda$.*

Alice and Bob share a string R. Alice wishes to authentically send Bob $M = M_1 \dots M_{\lambda_M}$ of λ_M bits. The value λ_M and the number of ones in M is known to Bob.
For i = 1 to λ_M:

1. Alice sends Bob challenge $x_i \in_r \{0,1\}^q$.

2. Bob receives x'_i, sends $b'_i = \mathsf{Ext}(R; x'_i)$, and challenge $y'_i \in_r \{0,1\}^q$

3. Alice receives b_i, y_i, verifies that $b_i = \mathsf{Ext}(R; x_i)$ and aborts if not.
She sends $(1, a_i = \mathsf{Ext}(R; y_i))$ if $M_i = 1$, and $(0, \perp)$ otherwise.

4. Bob receives b'_i, a'_i aborts if $b'_i = 1$ and $a'_i \neq \mathsf{Ext}(R; y'_i)$
and accepts otherwise.
If $i = \lambda_M$, Bob verifies that the number of ones in the received string match the expected number of ones; aborts otherwise.

Note that step 3 and 4 of each iteration are combined with steps 1 and 2, respectively, of the next iteration.

Figure 1: Protocol Interactive Message Authentication Auth

3.4 A Modification of Renner-Wolf Interactive Authentication

The [RW03] authentication protocol allows two parties who share the same string R to authenticate a message M, even if R has very little entropy.

We generalize this protocol slightly (to use general extractors instead of the specific polynomial authentication function) and present it in Figure 1. We assume that Ext is an average-case extractor that takes seeds of length q, and outputs $L+1$-bit strings that are 2^{-L-1}-close to uniform as long as the input has sufficient entropy h (in particular, $h \geq 3L+1$ suffices if one is using universal hashing as the extractor). For our purposes, it suffices to assume that the length of M and the number of ones in it (i.e., its Hamming weight $\mathsf{wt}(M)$) are known to Bob. If $|M|$ is known but $\mathsf{wt}(M)$ is not, M can be first encoded as a balanced string (i.e., a string with the same number of zeros and ones), by encoding, for example, a 0 as 01 and a 1 as a 10. This doubles the length of M.[3]

We note that [RW03] present a technique that can be used even if $|M|$ is unknown (namely, encoding M as a string that becomes balanced only at the end), but we will not need it here.

Each round of the protocol reveals $L+1$ bits of information correlated to R if $M_i = 0$, and $2L+1$ bits of information of information correlated to R if $M_i = 1$. Hence, by Lemma 2, the

[3]More efficient methods for encoding M as a balanced string are, of course, also possible. The length of M can be increased by less than $\log_2 |M|$ through the use of algorithms from Bos and Chaum [BC92] or Reyzin and Reyzin [RR02]. These algorithms compute a bijection between integers in $[1, \binom{n}{n/2}]$ and subsets of size $n/2$ of a set of size n. Any such subset can be viewed as a balanced string (where the i^{th} bit is set to 1 iff the i^{th} element is in the subset). Therefore, to balance a string M, it can be viewed as integer, and the subset produced by one of the above algorithms can be viewed as its balanced encoding.

adversary's uncertainty about R will be sufficient for the extractor to work until the last round as long as $\widetilde{\mathbf{H}}_\infty(R|E) \geq 3L+1+(L+1)(\lambda_M + \mathsf{wt}(M))$, and by Lemma 1 the a_i and b_i values will have entropy L from the adversary's point of view.

The intuition for the security of this protocol is that Eve cannot answer a random query x_i or y_i with probability greater than 2^{-L} because of the entropy of the answers, and hence can neither remove zero bits (because challenges to Bob keep him synchronized) nor insert one bits (because Alice is required to answer a challenge for each one). She can insert zero bits and change zeros to ones, but that is taken care of by the assumption that Bob knows λ_M and $\mathsf{wt}(M)$.

We do not formally define or prove security of this protocol, as the proof is essentially the same as in [RW03]. The probability that Eve succeeds in transmitting $M' \neq M$ to Bob and Bob does not reject (or Alice rejects and Bob accepts) is at most 2^{-L}.

We note the following security property observed in [RW04]. Consider a setting where, because of Eve's malicious interference, Bob does not have the same R as Alice does, but instead some (possibly correlated) R'. The protocol may not be complete, of course. However, it still secure, in the sense that Eve's chances of authenticating a message $M' \neq M$ are not more than when R is the same for Alice and Bob, as long as R' also has sufficient entropy ($\geq 3L+1+(L+1)(\lambda_M + \mathsf{wt}(M))$).

An additional security property (neither mentioned nor needed before) is that no information about the message M being authenticated is revealed to Eve until Bob receives the first message of the protocol. This holds with probability at least $1 - 2^{-L}$ even when Eve is active, because she cannot get Alice to reveal even the first bit M_1 without answering her challenge x_i, which she is unlikely to do without Bob's help.

4 Our Protocol

We propose the following privacy amplification protocol, in which Alice starts with w and Bob with w' such that $\mathsf{dis}(w, w') \leq \eta$. Below, MAC refers to the construction from Lemma 3.3 and Ext refers to an arbitrary average-case extractor (the choice of extractor will affect security only marginally, and will mostly affect efficiency, as we discuss below; in particular, extractors as simple as universal hashing can be used). Lengths that are currently undefined (such as of MAC keys and extractor seeds) will be set in subsequent sections in order to achieve desired security levels. In the protocol description below, extractor outputs of varying lengths and distance from uniform are needed at different stages of the protocol. We account for this variation by using two different extractors, denoted by Ext_1, Ext_2.

1. Alice generates a random MAC key k_1 and extractor seed s_1, computes $(R, P) \leftarrow \mathsf{Gen}(w)$, $\sigma_1 = \mathsf{MAC}_{k_1}(s_1, P)$, and sends $((s_1, P), \sigma_1)$ to Bob.

2. Alice initiates the Renner-Wolf message authentication protocol(Auth) for the message k_1 (suitably converted to a balanced string as indicated in Section 3.4), using R as the shared secret value.

3. Bob receives $((s_1', P'), \sigma_1')$, and computes $R' = \mathsf{Rep}(w', P')$. He responds to Alice's Auth protocol, using the string R' as the shared secret value.

4. Upon completion of Alice's side of Auth (if she has not yet rejected), Alice

   - extracts $k_2 = \mathsf{Ext}_1(R; s_1)$;

- generates a random seed s_2;
- sends Bob s_2 and $\sigma_2 = \mathsf{MAC}_{k_2}(s_2)$;
- outputs her final key $\mathsf{k_A} = k_3 = \mathsf{Ext}_2(R; s_2)$.

5. Upon completion of Bob's side of the Auth with the received message k_1', and receipt of s_2', σ_2' from Alice, Bob
   - verifies the first MAC, $\mathsf{Verify}_{k_1'}((s_1', P'), \sigma_1')$ (if fail, rejects);
   - computes the key for the second MAC, $k_2' = \mathsf{Ext}_1(R'; s_1')$;
   - verifies the second MAC, $\mathsf{Verify}_{k_2'}(s_2', \sigma_2')$ (if fail, rejects);
   - outputs his final key $\mathsf{k_B} = k_3' = \mathsf{Ext}_2(R'; s_2')$.

The intuition behind the security of this protocol is in the ordering of events. First, Alice authenticates a message (s_1, P) to Bob using a MAC with a truly random key k_1 which is unknown to Eve. This ensures that Eve cannot (except with negligible probability) succeed in modifying the message while preserving the integrity of the tag. However, Bob does not know k_1, either—which means he must defer the verification of the tag σ until a later stage.

Second, after she is sure that Bob has received the message and the tag (and thus it is too late for Eve to try modifying them), Alice transmits k_1 to Bob using the Renner-Wolf authentication protocol. The protocol reveals all of k_1 to Eve, but at this point k_1 is completely useless to her, because it is too late to try to modify the message and the tag. She cannot modify k_1 (except with negligible probability), by the security of the authentication protocol. It is crucial here that the authentication protocol is secure *even if* Eve modified P (such modification would not be detected until later), giving Alice and Bob different secrets R and R', because both R and R' have sufficiently high entropy. This enables Bob to verify the correctness of the MAC (and hence ensure that $R = R'$) at the end of the protocol.

The last steps of the protocol are a bit confusing, because instead of just outputting k_2 as the final key, Alice adds a level of indirection, using k_2 as a key to authenticate another extractor seed s_2, which is then used to extract the output key. This is similar to [RW03] and is needed because k_2, computed as $\mathsf{Ext}(w; s_1)$, is guaranteed to be close to uniform only when s_1 is a random seed independent of Eve's view. However, s_1 is revealed to Eve before Auth and an active Eve can modify the challenges within Auth (which are extractor seeds) to be correlated to s_1. By the time $\mathsf{Ext}(w; s_1)$ is computed after Auth, s_1 is not necessarily independent of Eve's view. Thus, k_2 is not necessarily suitable for the final output, although it is possible to show that it still has entropy and is therefore suitable as a MAC key. In Section 4.2 we show how to reduce the length of k_2 (and thus the entropy loss) as compared to the protocol of [RW03].

4.1 Analysis

The security parameter for our protocol is L. Through out this section, as with the rest of the paper, for any string x we use λ_x to denote the length of the x and h_x to denote its entropy (i.e, $\mathbf{H}_\infty(x)$).

Robustness We can view the protocol as consisting of two phases.

- Phase 1: Agreeing on k_2 from close secrets w, w'

- Phase 2: Using k_2 to agree final string k_3

Security of Phase 1.

Suppose Eve succeeds in an active attack against Phase 1, i.e., $k_2 \neq k_2'$. There are two possibilities.

1. $k_1 = k_1'$ (Eve does not attack protocol Auth). Therefore, in order for $k_2 \neq k_2'$, either $s_1 \neq s_1'$ or $P \neq P'$. Because Bob verifies the first MAC, Eve needs to come up with a valid $((s_1', P_1'), \sigma_1')$, which she has to do when she forwards Bob his very first message. This case again gives rise to two possible options, depending on when Eve sends to Bob her guess for $((s_1', P_1'), \sigma_1')$:
   - If Eve sends it right after Alice sends $((s_1, P_1), \sigma_1)$ and her first challenge x_1 to Bob, then this is equivalent to an active attack on a MAC, because she needs to produce her "guess" for $((s_1', P_1'), \sigma_1')$, before she sees any information correlated with k_1. We denote this probability by $\Pr[\mathsf{mac}]$. For an appropriate setting of length of k_1 (namely, $2L + 2\log \lambda_{(s_1,P)}/L$, where $\lambda_{(s_1,P)}$ is the length of s_1 and P) using the MAC construction from Section 3.3, we can show that $\Pr[\mathsf{mac}] \leq 2^{-L}$.
   - If Eve sends it later, then she needs to respond to x_1. We denote this probability by $\Pr[\mathsf{random-challenge}]$. From Section 3.4, $\Pr[\mathsf{random-challenge}] \leq 2^{-L}$.
2. $k_1 \neq k_1'$: In this case, Eve has to authenticate $k_1' \neq k_1$, using $\mathsf{Protocol\ Auth}$ in order to succeed. Therefore, her chances of success in this case are bounded by her chances of succeeding in an active attack against Auth. We denote this probability by $\Pr[\mathsf{Auth}]$. Again, if we run Auth on the security parameter $L+1$, we can show that $\Pr[\mathsf{auth}] \leq 2^{-L}$.

Security of Phase 2. This analysis is essentially the same as in [RW03]; we improve it in the next section. The key $k_2 = \mathsf{Ext}(R, s_1)$ agreed upon by the parties at the end of Phase 1 is used in Phase 2 to authenticate a fresh extractor seed s_2 (of length λ_{s_2}) using the single message MAC scheme of Section 3.3. However, the authentication protocol of Phase 1 gives Eve the ability to query the parties and get some information about $\mathsf{Ext}(w, s_1)$, decreasing the entropy of k_2. Knowing that this decrease will be no more than the amount communicated about R during Phase 1 (which is $\Theta(L^2)$ bits), we will set the length of k_2 to be twice that plus $2L + 2\log \lambda_{s_2}/L$ to get the desired 2^{-L} security for the second MAC.

It is easy to verify by counting the entropy lost during each round that the protocol, as presented here, gives us Theorem 1 up to constant factors. (More precisely, it proves the following modification of Theorem 1: in the expression for λ_k, increase the coefficient of L^2 from 1/2 to 9, and increase the number of messages by a factor of 4.) In the next section we present a number of improvements that reduce the entropy loss by (significant) constant factors, proving Theorem 1.

4.2 Constant-Factor Improvements

In this section we propose improvements that reduce the round complexity by a factor of 4 and the entropy loss by a factor of up to 18, making this protocol considerably more practical.

Reducing the length of the extracted MAC key k_2 Note that choosing the length of k_2 as above increases the entropy loss of the protocol by almost a factor of 3. By reworking the analysis of Phase 1 using the notion of average min-entropy (similar to the analysis in Appendix C), we can show that requiring k_2 to be longer than twice the communication in Phase 1, as discussed above, is unnecessary. Using the same notation that we used in the protocol description, we let σ_2 denote the tag of the MAC. To succeed in forging it, the adversary Eve needs to successfully change σ_2 to σ_2'. In addition, in Phase 1 she is also allowed to query Alice and Bob, say, T times. Protocol Auth implicitly imposes the constraint that Eve needs to also respond to T such queries. Let us denote her queries by $(q_1, \ldots, q_T)$ and responses by $(q_1', \ldots, q_T')$. We analyze the security of phases I and II jointly by looking at the average min-entropy of $(\sigma_2', (q_1', \ldots, q_T'))$ given $(\sigma_2, (q_1, \ldots, q_T))$. It turns out to be roughly $\lambda_{k_2} - T - \lambda_{\sigma_2}$, which makes the likelihood that Eve to completes phase I and comes up with σ_2' is no more than 2^{-L} if $\lambda_{k_2} > 2L + T$.

Working Base 4. Recall that in ith round of Auth, Bob sends Alice an extractor seed sufficient to extract $L + 1$ bits, and Alice responds with either nothing or the extracted string, depending on the value of the ith bit of the message being transmitted. We improve this by encoding the message transmitted by Auth (namely, the MAC key k_1) in base 4 rather than in base 2. Bob will send Alice an extractor seed sufficient to extract $3L + 1$ bits, and Alice will respond with nothing, the first $L+1$ bits, the first $2L+1$ bits, or all $3L+1$ bits depending on the ith digit of the message. This protocol works for strings that are "balanced" in base 4: i.e., messages M of length κ_M whose base-4 digits whose digits add up to $1.5\kappa_M$. It takes κ_M rounds and loses $2.5L\kappa_M$ bits of entropy, while maintaining the same security. This improves the number of rounds by a factor 2 and the entropy loss by a factor of $3/2.5 = 1.2$, because κ_M is half of the length that M would have if written in binary (the techniques used to balance a message in base 2 are also applicable in base 4, and increase the length by essentially the same ratio).

Working in Parallel. We further improve Auth by having Alice and Bob authenticate two halves of M to each other. Namely, Alice authenticates half of M to Bob at the same time as Bob authenticates half of M to Alice. Since M was initially chosen by Alice, she first has to sends half of M to Bob to he can authenticate it back to her. This has to occur in Alice's second message, because we need to make sure that M remains secret until after Bob's first message. Note that Bob's messages for authenticating his half of M can be combined with his answers to Alice's challenges. Namely, in each round, Alice will send Bob an extractor seed sufficient to extract $4L+1$ bits, and Bob will respond with the first $L + 1$ bits, the first $2L + 1$ bits, or the first $3L + 1$ bits, or all $4L + 1$ bits depending on the appropriate digit of M.

Note that the security proof goes through without adding any new challenges from Bob to Alice (i.e., Alice's responses remain $0, L + 1, 2L + 1$ or $3L + 1$ extracted bits long).

This improvement cuts the number of rounds essentially by a factor of 2 (except for the fact that Bob ends up one round behind), and cuts the entropy loss by a factor of 5/4=1.25 (because there are no challenges from Bob to Alice, only from Alice to Bob, and now there are half as many of those).

Not converting to/from a balanced string. Because MACs work even when the key does not have full entropy, Alice can simply choose a random balanced string for the MAC key k_1 instead of choosing fully random k_1, converting it to a balanced string for Auth, and then having Bob convert

it back. The security of the MAC will remain essentially 2^{-L} if k_1 is a random string of length $2L + 2\log L$ bits that is balanced when viewed in base 4 (because its entropy $\mathbf{H}_\infty(k_1)$ will be at least $2L + \log L$ by bounds on the central quadrinomial coefficient).

While this by itself is not a big improvement (on Alice's side, choosing a random balanced string is about as hard as choosing a random string of length $2L$ and converting to a balanced one of length $2L + \log L$; so the savings are mainly on Bob's side), it enables the next one.

Lowering the number of extracted bits in Auth. If we lower the bit length of the extracted strings exchanged in Auth, we increase the probability that the adversary succeeds in making a few changes to k_1. However, by using a MAC secure against related key attacks, we can actually tolerate a few such changes. Cramer et al. [CDF+08, Corollary 2], following a construction in [DKRS06], present a MAC that is essentially as efficient as the one we use, and is secure even if the adversary is allowed to change the key by exclusive-or with an adversarially chosen string. Thus, we need to make sure that Eve's changes to k_1 can be characterized as exclusive-or with a particular string.

Namely, suppose that instead of using responses of length $0, L+1, 2L+1$, or $3L+1$, Alice uses responses of length $0, \mu+1, 2\mu+1$, or $3\mu+1$, and instead of using responses of length $L+1, 2L+1, 3L+1$, or $4L+1$, Bob uses responses of length $L+1, L+\mu+1, L+2\mu+1$, or $L+3\mu+1$, for some $\mu < L$. The fact that Bob's responses are of length at least $L+1$ ensures that Eve cannot insert or delete digits from k_1 but with probability 2^{-L}. She can, however, increase a digit with probability $2^{-\mu}$. Because we require a balanced string, any decreased digit must be compensated by an increased digit; thus, if the total sum of digit changes is γ, then the probability that Eve does not get caught is $2^{-\gamma\mu/2}$.

We are working base 4, but using MACs that are based on bit-string keys. We will convert from base 4 to base 2 using the Gray code: $0 \to 00, 1 \to 01, 2 \to 11, 3 \to 10$. This will ensure that Hamming distance between the key sent by Alice and the key received by Bob is at most γ. (If we did not use the working-in-base-4 improvement, then, of course, this would not be necessary.)

The MAC of [CDF+08] is secure when the adversary chooses the modification to the key without seeing the key. Namely, for any string Δ, the probability that the adversary can forge a MAC with the key $\mathsf{k}_1 \oplus \Delta$ is low, where the probability is taken over a random k_1. In our setting, the adversary does get to see the key, because Auth does not hide k_1. However, what helps is that Δ likely has low Hamming weight, because to achieve high Hamming weight, Eve would have to successfully respond to a number of random challenges which does not hold. Therefore, the number of possible Δ values is small, which is almost as good as having a fixed Δ.

More precisely, let π be the security of the MAC for any fixed Δ, and α be the length of the MAC key. Then there are at most $\alpha^2/2$ values of Δ of Hamming weight 2, and by the union bound, the probability that will succeed with a forgery of the MAC by changing k_1 by two bits is at most $\pi\alpha^2/2$. At the same time, the probability that Eve will then be able to change k_1 by two bits is at most $2^{-\mu}$. Letting $\mu = 2\log\alpha$, we get that Eve's probability of success overall is $\pi/2$. Similarly, there are at most $\alpha^3/3!$ values of Δ of Hamming weight 3, and the overall probability of success using any such Δ is $\pi/3!$. Continuing in this manner, we get that overall probability of Eve's success through modification of k_1 is less than $\pi(1/2! + 1/3! + 1/4! + \dots) = \pi(e-2) < \pi$ (we are using here that Δ of Hamming weight 1 is impossible because the string $k_1 \oplus \Delta$ that Bob receives must be balanced).

Now to achieve MAC security $\pi = 2^{-L}$, we need to set the length of the MAC key, to be $\alpha \le 2L + 2\log(n/L+3) + 2$, and $\mu = 2\log(\alpha) = 2\log(2L + 2\log(n/L+3) + 2)$. This follows

from [CDF+08, Corollary 2].

Careful counting of round complexity and entropy loss, taking into account the improvements above, gives us the statement Theorem 1.

Acknowledgments

We thank Yevgeniy Dodis and Daniel Wichs for several insightful discussions. We also thank Renato Renner and Stefan Wolf for sharing with us their thoughts on the encoding of a string into a balanced string. We are grateful to MIT's Theory of Computation Group, where some of this work was performed, for its hospitality. This work was supported in part by the U.S. National Science Foundation grants CCF-0515100 and CNS-0546614.

References

[BC92] Jurjen N. E. Bos and David Chaum. Provably unforgeable signatures. In Ernest F. Brickell, editor, *Advances in Cryptology—CRYPTO '92*, volume 740 of *LNCS*, pages 1–14. Springer-Verlag, 1993, 16–20 August 1992.

[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.

[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.

[Che97] Steven Cheung. An efficient message authentication scheme for link state routing. In *In 13th Annual Computer Security Applications Conference*, pages 90–98, 1997.

[CL06] Ee-Chien Chang and Qiming Li. Hiding secret points amidst chaff. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 59–72. Springer-Verlag, 2006.

[CRVW02] Michael R. Capalbo, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *IEEE Conference on Computational Complexity*, page 15, 2002.

[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.

[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.

[DW08] Yevgeniy Dodis and Daniel Wichs. One-round authenticated key agreement from weak secrets. Technical Report 2008/503, Cryptology ePrint archive, `http://eprint.iacr.org`, 2008.

[HILL99] J. Hastad, 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.

[KR08a] 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*, September 2008.

[KR08b] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. Technical Report 2008/494, Cryptology ePrint archive, `http://eprint.iacr.org`, 2008.

[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.

[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.

[RR02] Leonid Reyzin and Natan Reyzin. Better than BiBa: Short one-time signatures with optimal signing and verifying. In Lynn Batten and Jennifer Seberry, editors, *ACISP 2002: 7th Australasian Conference on Information Security and Privacy*, volume 2384 of *LNCS*, pages 144–153, July 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, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 109–125. Springer-Verlag, 2004.

[Sho99] Victor Shoup. On formal models for secure key exchange. Technical Report RZ 3120 (#93166), IBM Zurich Research Lab, 1999. http://eprint.iacr.org/1999/012.

[Sho01] Victor Shoup. Ntl: A library for doing number theory (version 5.4.2), 2001. http://www.shoup.net/ntl.

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

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

[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.

A Proof of Lemma 1

Proof.

$$\begin{aligned}
\widetilde{\mathbf{H}}_\infty(\mathsf{Ext}(W,X)|X,E) &= -\log\left(\mathop{\mathbf{E}}_{x,e}\left(\max_{s\in\{0,1\}^l}\Pr[\mathsf{Ext}(W,X)=s|X=x,E=e]\right)\right)\\
&= -\log\left(\sum_{x,e}\Pr[X=x,E=e]\max_{s\in\{0,1\}^l}\Pr[\mathsf{Ext}(W,X)=s|X=x,E=e]\right)\\
&= -\log\left(\sum_{x,e}\max_{s}\Pr[\mathsf{Ext}(W,X)=s,X=x,E=e]\right)
\end{aligned}$$

For each (x,e) let s_{xe} denote the extractor output that maximizes $\Pr[\mathsf{Ext}(W,X)=s,X=$

$x, E = e]$. Let $T = \{(s_{xe}, x, e)\}$. Then it follows from the definition of statistical distance that:

$$\begin{aligned}
\widetilde{\mathbf{H}}_\infty(\mathsf{Ext}(W,X)|X,E) &= -\log \Pr[(\mathsf{Ext}(W,X),X,E) \in T]) \\
&\qquad \text{(by definition of } T) \\
&\geq -\log(\Pr[(U_l,X,E) \in T] + \varepsilon) \\
&\qquad \text{(by definition of } \mathsf{Ext} \text{ and } \mathbf{SD}\,()) \\
&= -\log\left(\sum_{x,e} \Pr[U_l = s_{xe}, X = x, E = e] + \varepsilon\right) \\
&\qquad \text{(because each pair } (s,e) \text{ occurs once in } T \\
&= -\log\left(\sum_{x,e} \Pr[U_l = s_{xe}] \Pr[X = x] \Pr[E = e] + \varepsilon\right) \\
&= -\log\left(\sum_{x,e} 2^{-l} \Pr[X = x] \Pr[E = e] + \varepsilon\right) \\
&= -\log(2^{-l} + \varepsilon) \\
&\geq \min\left(l, \log\tfrac{1}{\varepsilon}\right) - 1\,.
\end{aligned}$$

□

B Building Weakly Robust Fuzzy Conductors

As mentioned before, weakly robust fuzzy conductors can be built trivially out of any secure sketch $(\mathsf{SS}, \mathsf{Rec})$ defined in [DORS08]. For the sake of completeness, we review the definition below.

Definition 5. An $(m, \tilde{m}, t)$-secure sketch is a pair of efficient randomized procedures $(\mathsf{SS}, \mathsf{Rec})$ s.t.:

1. The sketching procedure SS on input $w \in \mathcal{M}$ returns a bit string $s \in \{0,1\}^*$. The recovery procedure Rec takes an element $w' \in \mathcal{M}$ and $s \in \{0,1\}^*$.
2. *Correctness:* If $\mathsf{dis}(w, w') \leq t$ then $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$.
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. ♢

To build a weakly robust fuzzy conductor from a secure sketch, simply let $\mathsf{Gen}(w) = (w, \mathsf{SS}(w))$, and $R = \mathsf{Rep}(w', P') = \mathsf{Rec}(w', P')$ unless $\mathsf{Rec}(w', P')$ fails to produce a value that is within η of w' (which can happen only if $P' \neq P$), in which case let $\mathsf{Rep}(w', P') = w'$. If the sketch length is λ, then this construction is an $(\mathcal{M}, h_W, h_W - \lambda, h_W - 2\lambda, \eta)$-weakly robust fuzzy conductor. This can be seen as follows: $\widetilde{\mathbf{H}}_\infty(R|E,P) \geq h_W - \lambda$ by Lemma 2. If $\mathsf{Rec}(w', P') = R'$, then w can be recovered from R' if one knows $\mathsf{SS}(w')$ and $\mathsf{SS}(w)$, by computing $\mathsf{Rec}(R', \mathsf{SS}(w'))$ to get w' and then $\mathsf{Rec}(w', \mathsf{SS}(w))$ to get w. Hence, $\widetilde{\mathbf{H}}_\infty(R'|E, \mathsf{SS}(w), \mathsf{SS}(w')) \geq \widetilde{\mathbf{H}}_\infty(w|E, \mathsf{SS}(w), \mathsf{SS}(w')) \geq h_W - 2\lambda$, again by Lemma 2.

This produces weakly robust fuzzy conductors for Hamming distance and set difference, using deterministic secure sketches of Constructions 3, 5, and 6 of [DORS08]. In particular, for Hamming distance over an alphabet of size F, given an $[n, k, 2t+1]$ linear error-correcting code for the alphabet, this gives $h_R = h_W - (n-k)\log F$, $h_{R'} = h_W - 2(n-k)\log F$, and $\eta = t$. For set difference, with sets whose elements come from a universe of size n, this gives $h_R = h_W - \eta\log(n+1)$ and $h_{R'} = h_W - 2\eta\log(n+1)$ for any η. Construction 9 of [DORS08], for edit distance, needs to modified slightly by omitting the information required to reverse the embedding, and letting R be the embedded version of w, $R = \mathsf{SH}_c(w)$; this produces a weakly robust fuzzy conductor for edit distance over alphabet of size F with $\lambda = (2c-1)\eta\lceil\log(F^c+1)\rceil$, $h_R = h_W - \lceil\frac{n}{c}\rceil\log(n-c+1) - \lambda$, and $h_{R'} = h_W - \lceil\frac{n}{c}\rceil\log(n-c+1) - 2\lambda$, for any choice of positive integer c and any η.

C Proof of MAC Construction (Proposition 1)

Proof. To prove security of the MAC construction, fix M and A; assume without loss of generality that A never outputs $M' = M$. A receives e and $\sigma = \mathsf{MAC}_k(M(e))$, and needs to predict $\sigma' = \mathsf{MAC}_k(M')$ for a new message M' of its choice. Note that M and M' are computed as deterministic functions of σ and e. Let (Σ', Σ) denote the distribution on σ' and σ (induced by K, E).

Recall that the likelihood of outputting the correct value of some variable X by an adversary who knows the value of a correlated Y is $2^{-\widetilde{\mathbf{H}}_\infty(X|Y)}$. We can therefore prove the security of the MAC by analyzing the average min-entropy of the Σ' conditioned on E and Σ. By Lemma 2, this min-entropy is $\widetilde{\mathbf{H}}_\infty(\Sigma' \mid \Sigma, E) \geq \widetilde{\mathbf{H}}_\infty(\Sigma', \Sigma \mid E) - \lambda_\sigma$.

To estimate $\widetilde{\mathbf{H}}_\infty(\Sigma', \Sigma \mid E)$, notice that for our MAC construction, knowing M, M', σ, σ' gives us an equation for a of the form $a(\tilde{M}(a) - \tilde{M}'(a)) - \sigma' + \sigma = 0$ (and b is then uniquely determined by a, M and σ). This is a nonzero polynomial in a of degree $\left\lceil\frac{\lambda_M}{\lambda_\sigma}\right\rceil$ and therefore has at most $\left\lceil\frac{\lambda_M}{\lambda_\sigma}\right\rceil$ roots. Let the roots be ordered lexicographically and let r denote the position of a in this lexicographic ordering. Knowledge of k given e is equivalent to the knowledge of σ, σ', r given e, from which $M = M(e), M' = M(\sigma, e)$, the polynomial, its roots, and a, b can be uniquely computed. Let R denote the distribution on r (induced by (K, E)). Therefore,

$$\widetilde{\mathbf{H}}_\infty(\Sigma, \Sigma' \mid E) \geq \widetilde{\mathbf{H}}_\infty(\Sigma, \Sigma', R \mid R, E) = \widetilde{\mathbf{H}}_\infty(K \mid R, E) \geq \widetilde{\mathbf{H}}_\infty(K \mid E) - \log\left\lceil\frac{\lambda_M}{\lambda_\sigma}\right\rceil$$

(by Lemma 2). Thus, $\widetilde{\mathbf{H}}_\infty(\Sigma'|\Sigma, E) \geq \widetilde{\mathbf{H}}_\infty(K|E) - \lambda_\sigma - \log\left\lceil\frac{\lambda_M}{\lambda_\sigma}\right\rceil$. Raising 1/2 to this power gives the desired result. □