Privacy Amplification with Asymptotically Optimal Entropy Loss

Nishanth Chandran*
Department of Computer Science
UCLA
nishanth@cs.ucla.edu

Bhavana Kanukurthi†
Department of Computer Science
Boston University
bhavanak@cs.bu.edu

Rafail Ostrovsky‡
Department of Computer Science and Mathematics
UCLA
rafail@cs.ucla.edu

Leonid Reyzin†
Department of Computer Science
Boston University
reyzin@cs.bu.edu

ABSTRACT

We study the problem of "privacy amplification": key agreement between two parties who both know a weak secret w, such as a password. (Such a setting is ubiquitous on the internet, where passwords are the most commonly used security device.) We assume that the key agreement protocol is taking place in the presence of an active computationally unbounded adversary Eve. The adversary may have partial knowledge about w, so we assume only that w has some entropy from Eve's point of view. Thus, the goal of the protocol is to convert this non-uniform secret w into a uniformly distributed string R that is fully secret from Eve. R may then be used as a key for running symmetric cryptographic protocols (such as encryption, authentication, etc.).

Because we make no computational assumptions, the entropy in R can come only from w. Thus such a protocol must minimize the entropy loss during its execution, so that R is as long as possible. The best previous results have entropy loss of $\Theta(\kappa^2)$, where κ is the security parameter, thus requiring the password to be very long even for small values of κ. In this work, we present the first protocol for information-theoretic key agreement that has entropy loss **linear** in the security parameter. The result is optimal up to constant factors. We achieve our improvement through a somewhat surprising application of error-correcting codes for the edit distance.

Categories and Subject Descriptors

E.3 [**Data Encryption**]; E.4 [**Coding and Information Theory**]

General Terms

Theory, Security

Keywords

Privacy amplification, Information-theoretic key agreement, Entropy loss

1. INTRODUCTION

The classical problem of privacy amplification, introduced by Bennett, Brassard, and Robert [1], considers the setting in which two parties, Alice and Bob, start out knowing a common string w that is partially secret. Following [2], we make no assumption on the distribution of w beyond a lower bound on its entropy. The goal of Alice and Bob is to perform key agreement: to agree on a string R that is fully secret. Formally, R should be statistically close to uniform from the point of view of the adversary Eve (we will let $2^{-\kappa}$ be the statistical distance between the distribution of R and the uniform distribution, and call κ the *security parameter*). Such R can then be used as a key for symmetric cryptography.

We make no computational assumptions and therefore model Eve as being computationally unbounded. This requirement implies that all the entropy in R comes from w. Thus, one of the most important requirements of such a protocol is that the output length of R must be as long as possible given the entropy of w. The difference between the entropy of w and the length of R is called the *entropy loss*.

We can model the communication channel between Alice and Bob as authenticated or unsecured. Equivalently, we can model Eve as passive or active. If Eve is passive (i.e.,

*Research supported in part by NSF grants 0716835, 0716389, 0830803, 0916574.

†Research supported in part by National Science Foundation grants CNS-0831281 and CNS-0546614

‡Research supported in part by an IBM Faculty Award, Xerox Innovation Group Award, the Okawa Foundation Award, Intel, Teradata, NSF grants 0716835, 0716389, 0830803, 0916574, BSF grant and U.C. MICRO grant.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.
Copyright 2010 ACM 978-1-4503-0050-6/10/06 ...$10.00.

merely listens in on the messages between Alice and Bob but cannot modify them), strong extractors [17] provide an optimal solution to this problem. Alice simply sends an extractor seed to Bob, they both extract R from w using the seed, and the entropy loss can be as low as 2κ.

However, if Eve is active (i.e., can modify the messages), the problem becomes considerably harder. In particular, one needs to worry not only about achieving R that is $2^{-\kappa}$-close to uniform, but also about ensuring that Alice and Bob output *the same* R with probability at least $1 - 2^{-\kappa}$. While the specific focus of our paper is to obtain R of maximum length, there are other parameters that one might wish to optimize. For instance, one might wish to minimize the round complexity or to minimize the entropy required in w for the protocol to work. Renner and Wolf [19], building on a series of works by Maurer, Renner and Wolf [14, 15, 22, 16], presented the first protocol that is secure against active Eve and works even when w is less than half-entropic (i.e., $h_W \leq \lambda_w/2$ where h_W and λ_w denote the entropy and the length of w). Moreover, by the use of extractors with asymptotically optimal seed length [8] and through the analysis of Kanukurthi and Reyzin [12], it can be seen that the protocol of [19] achieves entropy loss of $\Theta(\kappa^2)$ and takes $\Theta(\kappa)$ rounds of communication. (The work of [12], which builds upon [19], achieves the same asymptotic parameters but considerably improves the constants hidden inside Θ by eliminating the need for complicated extractors.) Dodis and Wichs [7] reduce the number of messages in the protocol from $\Theta(k)$ to just 2, but do not improve the entropy loss. They also present a two-message non-constructive (in other words, non-polynomial-time) protocol with an entropy loss of $\Theta(\kappa)$. While this result shows the theoretical feasibility of achieving such a low entropy loss, the only efficient solution that matches this entropy loss relies on Random Oracles [3]—i.e., an "assumption" that a publicly known truly random function is available. (It should be noted that single-message protocols for the same problem exist [15, 5, 10]; however, they have entropy loss $\lambda_w - h_W + \Theta(\kappa)$ and require thus require w to be at least half-entropic.)

Achieving a *efficient* privacy amplification protocol with entropy loss $\Theta(\kappa)$ and without resorting to the Random Oracle Model has, until now, been an open question.

Our Contribution.

We construct the first efficient protocol for privacy amplification over unsecured channels whose entropy loss (and number of rounds) is linear in the security parameter κ. This security loss is optimal up to constant factors, because extractor bounds require entropy loss at least $2\kappa - O(1)$ even in the case of authenticated channels [18]. We thus demonstrate that, up to constant factors, privacy amplification over unsecured channels can be as entropy-efficient as over authenticated ones.

Extension to Information Reconciliation/Fuzzy Extractors.

Consider the following generalization: Alice starts out with w, but Bob starts out with w' that is close to w in some metric space. Their goal is still the same: to agree on the same uniform string R. This problem is known as privacy amplification with information reconciliation [1] or as fuzzy extractors [6]. Constructions secure against active Eve appeared in [20, 5, 10, 12].

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).

Using the same approach as in [12], our protocol extends to this setting, as well.

A Related Problem.

Ishai, Kushilevitz, Ostrovsky, and Sahai [9] consider the problem of constructing a two-party protocol that extracts m "clean" instances of a joint distribution (X, Y) from $\mathcal{O}(m)$ "dirty" instances. This task can be viewed as a generalization of randomness extraction (the special case is when X and Y are identical bits). However, the techniques of [9] do not directly apply to the case when we have only *one* instance of a distribution. Furthermore, the entropy loss achieved in their work is significantly greater (constant factor times m) than in our case, where we obtain entropy loss independent of the entropy or length of the secret and linear in the security parameter.

Construction Techniques.

Our starting point is the protocol for interactive authentication from [12], which generalizes the authentication protocol of Renner and Wolf [19]. We focus only on the case when Alice and Bob share the same secret w; i.e., when $w = w'$ (as mentioned, the more general case can be addressed in the exact same way as in [12]). Using known techniques from the works of [19] and [12], it suffices to construct a message authentication protocol that can authentically transfer a message m from Alice to Bob using w.

The authentication protocol of [19, 12] works by authenticating bits of m one by one. For each bit of m, Bob sends Alice a random extractor seed, and, if the bit is equal to 1, Alice responds with the output of the extractor on input w using Bob's seed. Alice also sends Bob a random extractor seed of her own, to which he always responds (applying the extractor to w using Alice's seed), regardless of the bit of m. Each extractor output is κ bits long. This results in $\Theta(\kappa)$ entropy loss for every bit authenticated, and $\Theta(\kappa^2)$ entropy loss overall, because the message being authenticated needs to be $\Theta(\kappa)$ bits long (it is, actually, a MAC key in the protocol of [12]). The security proof shows that to succeed in breaking such an authentication protocol, the active adversary Eve must respond with *at least one* extractor output on her own without interacting with either Alice or Bob. Because the extractor output is a nearly-uniform κ-bit string, Eve cannot succeed with probability much higher than $2^{-\kappa}$.

The high level intuition for our protocol is as follows. If we were to shorten the length of the extractor output (in the authentication protocol) to be a constant number of bits, then we only lose $\Theta(1)$ bits of entropy for every bit of m and obtain an $\Theta(\kappa)$ entropy loss overall. On the other hand, the success probability of an adversary is a constant (by the same proof as before). If we could instead now ensure that Eve must respond with *several* (namely, $\Theta(\kappa)$) extractor outputs on her own, then we could show that the success

probability is $2^{-\kappa}$. This can be done by encoding m in a special error-detecting code of distance $\Theta(\kappa)$ and ensuring that to introduce $\Theta(\kappa)$ errors required to avoid detection, Eve must come up with $\Theta(\kappa)$ extractor outputs on her own.

It turns out that the code we need is a code for the edit distance [13], for the following reasons. We observe that, since the authentication of m is done bit-by-bit, Eve can change m by inserting individual bits, deleting them, or changing them from 0 to 1 or from 1 to 0. Deletions and changes from 0 to 1 require Eve to guess an extractor output on a fresh random seed. Because in the context of [19, 12] the length of m and the number of 1s in it are known a priori to both Alice and Bob, insertions and changes of 1 to 0 must be accompanied by deletions and changes of 0 to 1. Thus, Eve can create edit errors in the message, but at least a quarter of the errors introduced (namely, deletions and changes of 0 to 1) require her to find an extractor output on her own. So, if instead of authenticating the bits of the m, Alice first encoded m in an error-detecting code for 4κ edit errors, Eve would have to respond with at least κ extractor outputs on her own. Of course, the length of the codeword must still remain linear in the length of m. The codes of Schulman and Zuckerman [21] have this property (though we need to modify them to ensure the number of 1s is the same for every codeword of a given length).

Proof Techniques.

While using an error-detecting code and shortening the extractor outputs intuitively may seem to work in a straightforward manner, the proofs turn out to be quite tricky. In particular, we will need to use a proof technique that is completely different from the one used in [19] and [12], for the following reason. In the authentication protocols of [19, 12], Alice authenticates bits of the message one at a time. The proofs make use of induction on the length of the message received so far by Bob to show that if Eve was successful in changing any bit(s) of the message, then Eve must have responded to *one* extractor output on a random seed on her own. We, on the other hand, cannot use such an induction argument since we need to precisely characterize *how many* extractor outputs Eve must have given in relation to the number of bits modified. Instead, we use a new proof technique wherein we view the entire protocol transcript from the point of view of Eve as a string of literals, where each literal represent an interaction either with Alice or Bob. Using combinatorial arguments, we show that Eve cannot interleave these literals to her advantage without having to respond to many extractor outputs.

Next, one might like to claim that if Eve were to respond with multiple extractor outputs to random, independent seeds, then since the seeds are independent, her success in giving the right response for each of the seeds is also independent. Unfortunately, this intuition does not quite hold and there are subtleties in proving the theorem in this manner. This is because, while average min-entropy (introduced in [6]) gives us a guarantee that on average the entropy in w does not get reduced by too much, it may be the case that there may be a particular bad run (which occurs with very small probability) in which all information about w is revealed. This, in turn, destroys all independence in the probabilities of Eve's success. To counter this, we take the approach of considering two separate cases — the case when the run does not reveal too much about w and where we can argue "sufficient independence" (which happens with high probability), and the case when the run might reveal too much about w (perhaps all of it) and we cannot argue independence (which happens with low probability). Now, if we make an assumption that w begins with $\Theta(\kappa)$ bits of entropy more than what would be needed otherwise, we can show that the probability with which independence does not hold is low enough for our theorem to be true.

Organization of the paper.

We introduce notation and define our security model in Section 2. In Section 3, we briefly describe some of the existing tools that we require for our construction. Our main construction is given in Section 4. We give the proof of our main theorem in 5, providing complete details in the full version [4].

2. PRELIMINARIES

Notation.

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 $\mathbf{SD}\left(X_1, X_2\right) \leq \varepsilon$). The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) = -\log(\max_w \Pr[W = w])$. For a joint distribution (W, E), define the (average) conditional min-entropy of W given E [6] 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 give the correct value of W (in a single attempt) 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, $\mathbf{H}_\infty(X)$). We make use of the following lemma [6, Lemma 2.2b], which states that the average min-entropy of a variable from the point of view of an adversary does not decrease by more than the number of bits (correlated with the variable) observed by the adversary.

LEMMA 1. *If B has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(A \mid B) \geq \mathbf{H}_\infty(A, B) - \lambda \geq \mathbf{H}_\infty(A) - \lambda$.*

We also use the following lemma [6, Lemma 2.2a], which says that min-entropy is t bits less than the average min-entropy with probability at most 2^{-t}.

LEMMA 2. *For any $\delta > 0$, the conditional entropy $\mathbf{H}_\infty(A \mid B = b)$ is at least $\widetilde{\mathbf{H}}_\infty(A \mid B) - \log(\frac{1}{\delta})$ with probability at least $1 - \delta$ over the choice of b.*

Model and Security definition.

We now present the formal definition of a Privacy Amplification Protocol. Our definition is actually a modification

of the definition from [12, Definition 1] (which focuses on the case where Alice and Bob have correlated secrets, while we restrict our attention to the case where they have identical secrets). Let $w \in \{0,1\}^n$ be chosen according to distribution W be the secret value held by Alice and Bob respectively. 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 $C = C_a \cup C_b$ 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 a* $(h_W, \lambda_k, \delta, \epsilon)$-privacy amplification protocol *if it satisfies the following properties whenever* $\mathbf{H}_\infty(W) \geq h_W$*:*

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* $2^{-\delta}$*: sample* w *from* W*; 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,*

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

and

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

An important building block that we will construct is an interactive authentication protocol. In an authentication protocol, Alice additionally takes as input the message m to be authenticated to Bob. We now present the formal definition. (The definition we use is just an interactive variant of one-time message authentication codes. See [12, Definition 4] for one such definition.)

DEFINITION 2. *An interactive protocol* (A, B) *played by Alice and Bob on a communication channel fully controlled by an adversary Eve, is a* (h_W, κ)-interactive authentication protocol *if it satisfies the following properties whenever* $\mathbf{H}_\infty(W) \geq h_W$*:*

1. *Correctness. If Eve is passive, then* $\Pr[m_A = m_B = m] = 1$.
2. *Robustness. The probability that the following experiment outputs "Eve wins" is at most* $2^{-\kappa}$*: sample* w *from* W*; let* c_a, c_b *be the communication upon execution of* (A, B) *with Eve actively controlling the channel, and let* $\mathsf{A}(w, c_a, r_a, m) = m_A, \mathsf{B}(w, c_b, r_b) = m_B$*. Output "Eve wins" if* $(m_B \neq m_A \wedge m_A \neq \perp \wedge m_B \neq \perp)$.

3. BUILDING BLOCKS

We begin by presenting the building blocks needed for our main construction of a privacy amplification protocol with optimal entropy loss.

3.1 Extractors

Extractors [17] yield a close-to-uniform string from a random variable with high min-entropy, using a uniformly random seed i as a kind of catalyst. Strong extractors are ones in which the extracted string looks random even in the presence of the seed.

DEFINITION 3. *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* (n, t, l, ε)-strong extractor *if for all pairs of random variables* W *such that* $w \in W$ *is an* n*-bit string and* $\mathbf{H}_\infty(W) \geq t$*, we have* $\mathbf{SD}\left((\mathsf{Ext}(W; X), X), (U_l, X)\right) \leq \varepsilon$*, where* X *is the uniform distribution over* $\{0,1\}^r$.

For the purposes of this work, we would like to argue that extractor outputs are hard to predict for an adversary even when she has some external information $E = e$. The following lemma [12, Lemma 1] (the proof can be found in the full version [11]) shows that strings extracted by extractors have high average min-entropy, even given the seed (as long the entropy of the secret w is high even when conditioned on the external information e).

LEMMA 3. *Let* Ext *be a* (n, t, l, ε)*-strong extractor. Then if* $\mathbf{H}_\infty(W \mid E = e) \geq t$*, and* W *consists of* n*-bit strings,* $\widetilde{\mathbf{H}}_\infty(\mathsf{Ext}(W|e, X) \mid X, E) \geq \min\left(l, \log \frac{1}{\varepsilon}\right) - 1$.

3.2 Edit distance codes

Codes for insertion and deletion errors were first considered in the work of Levenshtein [13]. The first polynomial-time encodable and decodable codes that have constant rate and can correct a constant fraction of errors were given by Schulman and Zuckerman in [21]. For our application, we only require the code to be polynomial-time encodable and not necessarily polynomial-time decodable (this will be sufficient to get polynomial-time error detection, which is all we need). Let m be a message of length λ_m. For any two strings c and c' of length λ_c, let $\mathsf{EditDis}(c, c')$ denote the edit distance between c and c'; i.e., the number of single-bit insert and delete operations required to change string c to c' is $\mathsf{EditDis}(c, c')$.

DEFINITION 4. *Let* $m \in \{0,1\}^{\lambda_m}$*. For some constant* $0 < e_A < 1$*, a function* $\mathsf{Edit}(\cdot) : \{0,1\}^{\lambda_m} \to \{0,1\}^{\lambda_c}$*, is a* (λ_m, e_A, ρ)-error-detecting code *for edit errors, if* $\rho\lambda_c = \lambda_m$ *and satisfies the following properties:*

- $c = \mathsf{Edit}(m)$ *can be computed in polynomial (in* λ_m*) time, given* m*, for all* $m \in \{0,1\}^{\lambda_m}$*;*
- *For any* $m, m' \in \{0,1\}^{\lambda_m}$ *with* $m \neq m'$*,* $\mathsf{EditDis}(c', c) > e_A\lambda_c$*, where* $c = \mathsf{Edit}(m)$ *and* $c' = \mathsf{Edit}(m')$.

$\rho = \frac{\lambda_m}{\lambda_c}$*, is called the* rate *of the code.*

The theorem from Schulman and Zuckerman [21] is as follows:

THEOREM 1. *Let $0 < e_A < 1$ be a constant. Then, for some constant ρ, there exists a (λ_m, e_A, ρ)-error-detecting code for edit errors.*

Remark. We actually require an error-detecting code $\mathsf{Edit}(\cdot)$ for edit distances, in which the Hamming weight (number of 1s) of every codeword is the same. Such codes can be constructed easily[1] using the same construction of Schulman and Zuckerman [21], without losing out on the constant parameters ρ and e_A.

3.3 Interactive authentication protocol

We use the interactive authentication protocol from [12] that is used by Alice and Bob to send bits authentically to each other. Let the security parameter be κ. Let Alice and Bob share an n-bit secret $w \in W$ with min-entropy h_W. Let the message that Alice wishes to authenticate be $m = m_1 \dots m_{\lambda_m}$. Assume that Bob knows λ_m and the number of ones in m (say $\mathsf{wt}(\mathsf{m})$). Let Ext be a $(\lambda_w, t, \kappa + 1, 2^{-\kappa-1})$-strong extractor with seed length q bits. That is, Ext takes seeds of input q, outputs $\kappa + 1$-bit strings that are $2^{-\kappa-1}$-close to uniform, as long as the input has sufficient entropy t. (In particular, if $t \geq 3\kappa + 1$ is sufficient if one is using universal hashing.) The authentication protocol from [12], which is a modification of the scheme from [19], is presented below.

Protocol $\mathsf{Auth}(w, m)$**:**
For $i = 1$ to λ_m

1. Alice sends Bob a fresh random challenge $x_i \in_r \{0,1\}^q$.
2. Bob receives x_i, sends $r_{x_i} = \mathsf{Ext}(w; x_i)$, and a fresh random challenge $y_i \in_r \{0,1\}^q$.
3. Alice receives r_{x_i}, y_i, verifies that $r_{x_i} = \mathsf{Ext}(w; x_i)$ and aborts if not.
   She sends $(1, r_{y_i} = \mathsf{Ext}(w; y_i))$ if $m_i = 1$, and $(0, \perp)$ otherwise.
4. Bob receives m_i, r_{y_i} aborts if $m_i = 1$ and $r_{y_i} \neq \mathsf{Ext}(w; y_i)$ and accepts otherwise.
   If $i = \lambda_m$, Bob verifies that the number of ones in the received string $= \mathsf{wt}(\mathsf{m})$; aborts otherwise.

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

We say that Eve *responded to a fresh random challenge* sent by Alice (Bob), if Alice (Bob) sends a challenge and Eve responds to it without having received a response to an extractor challenge (on any, not necessarily the same, seed) in the meantime. (Note that if Eve sends $\perp$ in response to Bob's challenge, thereby authenticating a zero bit, it is not considered as a response to a fresh random challenge.) The intuition for the security of the above protocol is as follows. Note that Eve can insert 0 bits (this does not require a response from Alice) and can change a 1 bit to a 0 bit. Since, Bob knows λ_m as well as $\mathsf{wt}(\mathsf{m})$, if Eve were to insert 0 bits or change a 1 bit to a 0, then she must also either remove 0 bits or insert 1 bits. Now, removing 0 bits sent by Alice or inserting 1 bits require answering a random challenge of Alice or Bob. By Lemma 3, we have that the r_{x_i} and r_{y_i} values have entropy κ from Eve's point of view. Since the responses (r_{x_i} and r_{y_i}) have entropy κ bits, Eve cannot respond to a fresh random challenge by Alice or Bob with probability $> 2^{-\kappa}$. Hence, the probability of Eve's success can be shown to be at most $2^{-\kappa}$.

To analyze the entropy loss, we see that every round of interaction reveals $\kappa + 1$ bits of information correlated to w in the case when $m_i = 0$ and $2\kappa+2$ bits of information correlated to w in the case when $m_i = 1$. Therefore, by Lemma 1, as long as $\mathbf{H}_\infty(W) \geq t + (\kappa + 1)(\lambda_m + \mathsf{wt}(\mathsf{m}))$, (where t is the threshold entropy that is needed by the specific extractor used for the extraction to be secure), the entropy in w is sufficient for the extractor to work until the last round (to be precise, we need here an extractor that works for *average* min-entropy; see [12] for details). This leads to an entropy loss of $\Theta(\lambda_m \kappa)$ in the authentication protocol, which translates to an entropy loss of $\Theta(\kappa^2)$ for the key agreement protocol (because the key agreement protocol of [12] needs to authenticate a message—namely, a MAC key—of length $\Theta(\kappa)$).

4. MAIN CONSTRUCTION

Our main construction of a privacy amplification protocol is obtained by building an improved authentication protocol with low entropy loss.

In particular, our main theorem is:

THEOREM 2. *Let κ be the security parameter. Let $\mathsf{Edit}(\cdot)$ be a $(4(\kappa+1), e_A, \rho)$-error-detecting code for constants $0 < e_A, \rho < 1$. Let Ext be a $(h_W, t, \tau, 2^{-\tau})$-strong extractor with $\tau = \frac{\rho}{e_A} + 1$. Then there exists an efficient (h_W, κ)-interactive authentication protocol for messages[2] of length $4(\kappa+1)$ with entropy loss $\frac{8\tau(\kappa+1)}{\rho}$. The protocol works as long $h_W \geq \frac{8\tau(\kappa+1)}{\rho} + t + \kappa + 1$.* [3]

Given that e_A, ρ and τ are constants, using the result of [12] on converting a message authentication protocol to a protocol for privacy amplification, we obtain the following corollary to Theorem 2:

COROLLARY 1. *There exists an efficient $(h_W, \lambda_k, 2^{-\kappa}, \epsilon)$ privacy amplification protocol with entropy loss $\mathcal{O}(\kappa)$. The length of the extracted key $\lambda_k = h_W - 2\log\frac{1}{\epsilon} - \mathcal{O}(\kappa)$.*

The rest of the paper will focus on proving Theorem 2. Below, we present our improved authentication protocol. We present the proof of security and entropy loss for our authentication protocol in Section 5.

Improved Authentication Protocol.

We start with the authentication protocol described in Section 3.3 and decrease the length of the extractor output in each round to be a constant τ (instead of $\kappa+1$). This gives

[1] Schulman and Zuckerman [21] build a two-layered code constructing a greedy code for edit errors on logarithmic size blocks of messages. By considering the space of codewords for this greedy code to be such that the Hamming weight of all codewords is the same, one can get the required construction.

[2] We note that the message length can be $z(\kappa+1)$, for arbitrary constant z. In this case, we require $\tau = \frac{4\rho}{ze_A} - 1$.

[3] It is easy to see that if we universal hashing as our extractor, t can be set to be $3\tau + 1$ thereby giving us a protocol that works as long as $h_W = \Theta(\kappa)$.

us an $\Theta(\lambda_m)$ entropy loss for the authentication protocol as desired. Unfortunately, the security of this protocol no longer holds. The security proof in [12] shows that in order to get Bob to accept any message $m' \neq m$, Eve must respond to at least *one* fresh random challenge from either Alice or Bob. The probability with which Eve could respond to a fresh random challenge from either Alice or Bob is $2^{-\tau+1}$, which not high enough if τ is a constant.

To rectify this problem, we ensure that in order to make Bob accept a different message, Eve must respond to *many* (namely, $\Theta(\kappa)$) fresh random challenges, which translates into a success probability of only $2^{-\kappa}$ as desired. To do so, we have Alice transmit the message $c = \mathsf{Edit}(m)$ (see Definition 4) and Bob verify that c is a valid codeword (or, equivalently, since we do not require that codeword validity be efficiently verifiable, Alice can send m to Bob in the clear and Bob can re-encode it to check if he gets c).

We now describe the authentication protocol precisely. Let the security parameter be κ. Let Alice and Bob share an λ_w-bit secret $w \in W$ with min-entropy h_W. Let the message that Alice wishes to authenticate be $m \in \{0,1\}^{\lambda_m}$. Let Ext be a average-case $(\lambda_w, t, \tau, 2^{-\tau})$-strong extractor with seed length q bits for some constants tand τ. Let $\mathsf{Edit}(\cdot)$ be a (λ_m, e_A, ρ)-error-detecting code for constants $0 < e_A, \rho < 1$, such that the Hamming weight of all codewords is the same (call it $\mathsf{wt}(\mathsf{c})$).

Protocol $\mathsf{NewAuth}(w, m)$**:**

1. Alice sends Bob the message m. Let the message received by Bob be m'.
2. Alice and Bob execute protocol $\mathsf{Auth}(w, c)$ for $c = \mathsf{Edit}(m)$, using extractor Ext for all the responses.
3. Let the string received by Bob be denoted by c'. Bob computes $\mathsf{Edit}(m')$. If $c' \neq \mathsf{Edit}(m')$, then Bob rejects. Otherwise, Bob accepts m' as the message received.

Intuition.

The key to our improvement is to show that protocol Auth gives Eve an edit channel in the following sense. We will show that the success probability of Eve in changing the message c to a message c' is related by a constant factor to the edit distance between these two messages; since this distance must be greater than $e_A\lambda_c$ for Bob to accept at the end, Eve will fail if $a_A\lambda_c$ is high enough.

Indeed, consider what Eve can do. She can not deliver a message from Alice to Bob (this corresponds to deleting a bit from the c), but then she would have to reply to Alice's challenge contained in that message on her own to avoid detection; the probability of guessing a correct response to such a challenge is at most $2^{-\tau+1}$. She can also try to change a message from Alice from conveying a "0" bit to conveying a "1" bit, but that would require coming up with a response to Bob's challenge, which again can happen with probability at most $2^{-\tau+1}$. If Eve attempts to do these things multiple times, the probabilities are (almost) multiply, because of the freshness of random extractor seeds, which ensures (almost) independence She can also "insert" bits by replying to Bob's challenges on her own or change a "1" to a "0", but since the number of 1s and the total length are fixed, she will have to pay elsewhere with deletions and changes of "0" to "1".

We translate this intuition into a proof in the next section.

5. PROOF OF MAIN THEOREM

In this section, we prove our main theorem. Namely we show that the new authentication protocol we presented in Section 4 is a secure authentication scheme with $\Theta(\lambda_m)$ entropy loss, where λ_m is the length of the message being authenticated. Some proof details are omitted for lack of space and are presented in the full version [4].

We prove this in two broad steps. First we show that the authentication protocol gives Eve an edit distance channel in the following sense – Eve can modify several bits of the message that Alice authenticates to Bob. However, irrespective of what algorithm Eve uses or what modifications she does, the edit distance between the two messages can be bounded in terms of the security parameter.

Technical Challenges.

As mentioned before, the security of [19, 12] rely on showing that for Eve to modify the message that Alice sent, she would have to respond to at least one fresh random challenge on her own. They prove this using induction on the length of the message received by Bob so far and show that at any stage either Eve has responded to a random challenge on her own or the string received by Bob is not non-trivially different from what Alice sent.

We, on the other hand, cannot use such an inductive proof on the length of the message for the following reason. The statement we want to make is not about whether Eve responded to a random challenge on her own or not. Instead, we would want to keep track of how many random challenges Eve responded to **and** precisely study the effects of responding to these challenges on the edit distance of the messages. Since the entire protocol is just an interleaving of challenges and responses sent back and forth, categorizing points in the protocol where Eve responded to a fresh random challenge on her own becomes a delicate task. In fact, it turns out that viewing the protocol in terms of the message received by Bob (or Alice) as was done in [19, 12] does not capture all the information needed to categorize the points where Eve had to respond to a random challenge.

Instead, we need to use a new proof technique in which we view the entire protocol from Eve's perspective. In fact, we represent Eve's view of any valid run of the authentication protocol by a string E. This string will allow us to capture all the information including the order in which Eve interacted with the honest parties. This turns out be crucial in categorizing points in the protocol in which Eve had to respond to a random challenge. Once we do this, we use combinatorial arguments to relate the number of random challenges that Eve responded to on her own to the edit distance between the messages of the honest parties. Finally, we compute the probability with which Eve can respond to all the fresh random challenges.

Organization.

In Section 5.1 we will introduce the notation for the string representation of the protocol. Using that we will characterize precisely (in Section 5.1.1), the points in the protocol (that actually correspond to literals in the string E) where Eve must respond to a fresh random challenge. We call these points as costly literals. Now, if we could compute the probability with which Eve can respond to every fresh random challenge, and relate the number of costly literals

in E to the edit distance between the two messages m and m', then we will be done. We do precisely this. In Section 5.2, we present the details of relating the edit distance between m and m' to the number of costly literals in the string E. Finally, we compute the probability with which Eve can respond to fresh random challenge (that correspond to the costly literals in E) in Section 5.3.

Notation.

Let $\mathbb{A}$ be an alphabet. Let $a \in \mathbb{A}$ be a literal from the alphabet. When we write a^*, we mean all strings of the form $\underbrace{aa\cdots a}_{i}$, where $i \geq 0$ is an integer. When $i \geq 1$, we write a^+. Let x_1 and x_2 denote any two strings. We write $x_1||x_2$ to denote the concatenation of the two strings.

5.1 String representation of the authentication protocol

We now present our proof ideas in more detail. As mentioned before, we view the entire protocol as it takes place from Eve's perspective. In any round i, Eve's interaction with Alice will consist of Eve sending a challenge y_i and response $(\mathsf{Ext}(w; x_{i-1}))$ to the challenge issued by Alice in the previous round (x_{i-1}). Alice then sends Eve a challenge x_i and a response to the challenge received by her in the previous round (i.e., $\mathsf{Ext}(w; y_{i-1})$). We will call this two-message interaction that starts with a message from Eve to Alice and then from Alice to Eve as a *roundtrip* between Eve and Alice. To be more concise, we will use the literal a to denote this *roundtrip* that takes place between Eve and Alice. Let us now consider Eve's interaction with Bob. In any round i, Eve's interaction with Bob starts with Eve sending a challenge x_i and response $(\mathsf{Ext}(w; y_{i-1}))$ to the challenge issued by Bob in the previous round (y_{i-1}). Bob then sends Eve a challenge y_i and a response to the challenge received by her in the previous round (i.e., $\mathsf{Ext}(w; x_{i-1})$). We denote this *roundtrip* between Eve and Bob by b.

The notation a and b will be important for our proof. Note that we do not index a or b by the round i. This is because that information will largely be irrelevant to us. However, in any roundtrip with Alice, Alice is authenticating some bit (either 0 or 1) and Bob is receiving some bit (either 0 or 1). This allows us to have two different literals, a_0 and a_1 depending on which bit Alice is authenticating in that roundtrip. Likewise, we will also use b_0 and b_1 to denote Eve's roundtrips with Bob. If we do not subscript the a or b literals, it means that the claim holds irrespective of the bit being authenticated.

In the first interaction between Eve and Bob, he receives just a challenge from Eve and no response to any challenge. So we will distinguish the first interaction between Eve and Bob (from subsequent interactions) by denoting it as b_c. Likewise, in the last interaction between Eve and Alice, Alice receives **only** a response from Eve and no challenge (to which Alice would have had to respond to). So we will denote this last interaction between Eve and Alice by a_r.

Using this notation, we can write out the entire protocol from Eve's point of view by simply creating a string (call it E) denoting Eve's actions. As an example, if Eve is passive and Alice is authenticating $m = m_1, m_2, \cdots, m_{\lambda_m}$ to Bob, then $\mathrm{E} = b_c, a_{m_1}, b_{m_1}, a_{m_2}, b_{m_2}, \cdots, a_{m_{\lambda_m}}, b_{m_{\lambda_m}}, a_r$. Note that since Eve might be active, E will not necessarily take the structure as above. Instead E can be any interleaving of a and b variables.

Let $\alpha(\mathrm{E})$ be a function that outputs the subscripts of the a literals (other than a_r) read out in order. Then it is easy to see that $\alpha(\mathrm{E})$ represents the message authenticated by Alice. Likewise if $\beta(\mathrm{E})$ is a function that outputs the subscripts of the b literals (other than b_c) read out in order, then $\beta(\mathrm{E})$ is nothing but the message received by Bob. Observe that if Eve is passive, E will take the structure described above and $\alpha(\mathrm{E}) = \beta(\mathrm{E})$ (which is consistent with the fact that Alice's and Bob's messages are equal).

(Note that the literals a_r, b_c do not form a part of the messages $\alpha(\mathrm{E}), \beta(\mathrm{E})$. The literal b_c is defined for semantic purposes. The literal a_r, on the other hand, will be needed to make the edit distance bounds go through.)

Recall that we would like to use the string representation E, to categorize those points in the protocol where Eve would have to respond to a fresh random challenge. This brings us to the notion of *costly literals*.

DEFINITION 5. *A literal in E is marked as costly, if in the real run of the protocol (that E represents) Eve would have to respond to a fresh random challenge on her own in the roundtrip corresponding to the costly literal.*

5.1.1 Characterizing costly literals in string E

In this section, we precisely characterize the literals in E that are marked costly. We use the following lemma, the proof of which can be found in the full version of this paper [4].

LEMMA 4. *The following statements about costly literals are true:*

1. *Any $a-$literal is marked as costly if it is not immediately preceded by a $b-$literal in E.*
2. *A b_1 literal is a costly literal if it is not immediately preceded by a $a-$literal.*
3. *In addition, a b_1-literal is also marked as costly literal if all the $a-$ literals contained between this b_1-literal and the previous $b-$literal occurring in E are all a_0-literals. In other words, if the sequence looks like this $b^* a_0^+ \underline{b_1}$ then the underlined b_1 literal is marked as costly.*

5.2 Bounding edit distance in terms of the number of costly literals

We wish to show that a high edit distance between $\alpha(\mathrm{E})$ and $\beta(\mathrm{E})$ cannot be achieved without a lot of costly literals. In other words, we wish to construct a method to convert $\alpha(\mathrm{E})$ into $\beta(\mathrm{E})$ such that the total number of edit operations (insertions and deletions) performed is bounded by a constant times the number of costly literals. We will do this in two steps:

1. First, we will present an algorithm that converts the string E into a new string E$'$ with the property that Alice and Bob receive the exact same messages in E$'$ i.e., $\alpha(\mathrm{E}') = \beta(\mathrm{E}')$. Moreover this message is nothing but the message received by Bob in E i.e., $\beta(\mathrm{E}') = \beta(\mathrm{E})$. We will do this in such a manner that the total number of edit operations performed (in order to convert E to E$'$) will be at most $4\mathsf{L}$, where L is the number of costly literals in E.

2. Next, it will become apparent from the above algorithm description, that the exact edit operations used to convert E to E′ can also be used to convert α(E) into β(E) directly. (In that sense, the intermediary step of converting E into E′ is only for clarity of exposition; it does not change the edit distance in any way since the underlying edit operations are the same.)

The rest of this section is devoted to formally stating and proving the relation between the edit distance and the number of costly literals in E. For clarity of exposition, we state our theorems in the case where the string to be authenticated is balanced (i.e., the number of 1s and 0s are equal). The proof trivially extends to the case when this may not be true, but Bob knows the number of 1s in the string (as is the case with the edit distance code $\mathsf{Edit}(\cdot)$ used). Let the number of costly literals in E be at most L. Then the following theorem states that the edit distance between the message authenticated by Alice and received by Bob is at most $4\mathsf{L}$.

THEOREM 3. *Let E be a string consisting of a_0, b_0, a_1, b_1 literals (as well as the special literals a_r, b_c) as defined above. Let it hold that the number of times each literal appears in the string E is the same (excluding the special literals which appear just once). Let the number of costly literals in E be at most L. Let E' be the string as defined above. Then the edit distance between E and E' is at most $4\mathsf{L}$.*

PROOF. As mentioned before, we will first focus on converting E that has at most L costly literals into E′ such that $\alpha(\mathrm{E}') = \beta(\mathrm{E}') = \beta(\mathrm{E})$ using at most $4\mathsf{L}$ edit operations. It will be evident after this, that these edit operations can be used to convert α(E) into β(E). We first present a sketch of the proof.

1. We first scan E and mark what we call the *edit* literals in E. These edit literals are marked such that the edit distance between E and E′ is precisely the number of edit literals in E.
2. Now, we need to show that if the number of costly literals is L, then the number of edit literals is at most $4\mathsf{L}$. It will be easier to prove the above statement by categorizing edit literals into (disjoint sets of) *good edit literals* and *bad edit literals*. Our proof will be in three steps:
   (a) Using the notation, $\#bad$ to denote the number of bad edit literals and $\#edit$ to denote the number of edit literals, we first show that $\#edit \leq 2\times\#bad$ (Lemma 5).
   (b) Next, letting $\#costly$ to denote the number of costly literals, we will show that $\#bad \leq 2\times\#costly = 2\mathsf{L}$ (Lemma 6).
   (c) Finally, combining the above two lemmas, we get $\#edit \leq 4\times \#costly = 4\mathsf{L}$, which is the required statement.

□

We now proceed to give the details of the above proof. We begin by defining *edit* literals. Next, we show how to mark literals as good edit literals and bad edit literals.

DEFINITION 6. *Edit Literals: An edit literal is any a literal that needs to be deleted from E or any b literal that needs to be inserted into E to obtain E′. (When we say that a b literal is inserted, we actually mean that the "b literal" is inserted as a corresponding "a literal". If a literal b_p, $p \in \{0,1\}$ is marked as an edit literal then an a_p literal is inserted in the appropriate position.)*

Clearly the edit distance between E and E′ is precisely the number of edit literals in E. As we describe next, to mark the edit literals in E, we first split E into disjoint substrings and mark out the edit literals in each substring.

MARKING BAD/GOOD EDIT LITERALS IN E. To do this, let E be written as the concatenation of k strings; i.e., $\mathrm{E} = \mathrm{E}_1||\mathrm{E}_2||\cdots||\mathrm{E}_k$. Each E_i consists of a continuous sequence of one or more a literals followed by one or more b literals (except for the first and last E_is. E_1 might not have any a literals and E_k might not have any b literals.) As examples, consider $\mathrm{E} = \underbrace{a_1, a_1, a_1, b_c, b_1}_{\mathrm{E}_1}, a_0, \cdots, b_0, \underbrace{a_1, a_r, b_0, b_0}_{\mathrm{E}_k}$ and $\mathrm{E} = \underbrace{b_c}_{\mathrm{E}_1}, \underbrace{a_1, a_1, a_1, b_0, b_1}_{\mathrm{E}_2}, a_0, \cdots, b_0, \underbrace{a_1, a_0, a_r}_{\mathrm{E}_k}$. In the former, all substrings including E_1 and E_k are of the form a^+b^+. In the latter, E_1 is of the form a^*b^+ and E_k is of the form a^+b^*.

We now describe the algorithm to mark literals in each substring E_i as (good/bad) edit literals.

Algorithm MarkEdits

1. If E_i has the form $a^*a_pb_qb^*$ (where $p, q \in \{0,1\}$), call the last a and the first b literal as pivots and proceed as follows:
   (a) if $p = q$, mark every literal in the E_i other than the pivots as a bad edit literal.
   (b) if $p = 1, q = 0$, mark every literal in the E_i other than the $a-$pivot as a bad edit literal. In addition, mark the $a-$pivot as a good edit literal.
   (c) if $p = 0, q = 1$: If E_i is of the form $a^*a_1a^*a_0b_1b^*$, make the a literal subscripted by 1 as the pivot (instead of the original $a-$pivot) and mark all literals other than the pivots as bad edit literals. (There may be many different a_1 literals. Which specific one we chose as the pivot is immaterial.)
   (d) if $p = 0, q = 1$: If E_i is not of the form $a^*a_1a^*a_0b_1b^*$ (i.e., E_i is of the form $a_0^*a_0b_1b^*$), mark every literal in the E_i other than the $a-$pivot as a bad edit literal. In addition, mark the $a-$pivot as a good edit literal.
2. If E_i has the form a^+ mark all but one of the a literals as bad.
3. If E_i has the form b^+ then the first b must be b_c. Mark all but the b_c literal as bad.

We use the following lemmas, the proofs of which can be found in the full version [4].

LEMMA 5. $\#edit \leq 2 \times \#bad$.

LEMMA 6. $\#bad \leq 2 \times \#costly$.

PROOF. To prove this lemma, we will categorize our bad edit literals into bad_{b_0}, bad_{b_1}, and bad_a disjoint sets of literals (we mostly will not distinguish between the different kinds of bad_a literals.) To show that $\#bad \leq 2 \times \#costly$, we need to show that $bad_{b_0} + bad_{b_1} + bad_a \leq 2 \times \#costly$. We will prove this in two steps. First we will show that $\#bad_{b_1} + \#bad_a \leq \#costly$ (Lemma 7). Next we will show that $\#bad_{b_0} \leq \#bad_{b_1} + \#bad_a$ (Lemma 8). □

LEMMA 7. $\#bad_{b_1} + \#bad_a \leq \#costly$.

LEMMA 8. $\#bad_{b_0} \leq \#bad_{b_1} + \#bad_a$.

Theorem 3 states if E has at most L costly literals, then the edit distance between E and E′ is at most 4L. We will use the contra-positive to this theorem which states that Eve cannot create an E that has just L costly literals such that the edit distance between E and E′ is $> 4\mathsf{L}$.

As we mentioned earlier, the process of converting E to E′ using just edit operations is isomorphic to the process of converting α(E) and β(E). Therefore the edit distance between E and E′ is precisely the edit distance between α(E) and β(E), which are nothing but the messages sent by Alice call it (m_A)and received by Bob(m_B) respectively.

We use this and the fact that the number costly literals correspond exactly to the number of fresh random challenges that Eve responded to on her own to get the following corollary to Theorem 3.

COROLLARY 2. *Let Alice and Bob execute protocol Auth in the presence of an active adversary Eve. Let m_A denote the message sent by Alice and let m_B denote the message received by Bob. Let the edit distance between m_A and m_B be at least* 4L. *Then Eve must have responded to at least* L *fresh random challenges on her own.*

5.3 Relating the number of fresh random challenge responses to the probability of Eve's success

So far we have shown that the edit distance between the message authenticated by Alice and received by Bob gives us a lower bound on the number of fresh random challenges (or extractor seeds) that Eve needs to respond to (with extractor outputs) on her own. In this section, our goal will be to get an upper bound on the probability with which she can succeed in responding to all of those random challenges.

It follows from Lemma 3 that the output of the extractor looks unpredictable to an adversary who has access to some information about the secret W (as long as the seed is a fresh random seed chosen independently of E). This lemma was used in [12] to bound the probability with which Eve succeeds in responding to a *single* fresh random challenge on her own. In this section our goal will be to extend this argument to the case where Eve will have to respond to multiple random challenges. We would expect that if Eve succeeds in responding to a single random challenge with probability at most $2^{-(\lambda_r - 1)}$ then the probability that Eve responds to μ fresh random challenges (chosen independently) would be at most $2^{-\mu(\lambda_r - 1)}$. Unfortunately, for reasons explained after the proof of the next lemma, that's not quite the case, but we can get something close: we show that the probability that Eve responds to μ fresh random challenges on her own is $2^{-\mu(\lambda_r - 1)} + 2^{-\kappa - 1}$, if the average min-entropy of w at the last extraction is at least $t + \kappa + 1$ (where, recall, t is the entropy needed for the extractor to work).

Consider a run of Protocol Auth where Eve receives (not necessarily at the same time and separated by other protocol message) μ random challenges $x_1, \cdots, x_\mu$, to which Eve needs to respond (with a guess for $\mathsf{Ext}(w, x_i)$ of length λ_r) on her own. Let $S_i = 1$ if Eve's response to the ith challenge is correct, and 0 otherwise. For simplicity of notation, assume that the protocol goes on even if $S_i = 0$ (in reality, Alice or Bob will abort). Let Tr_i denote all the information that Eve has about w just before she responds with $\mathsf{Ext}(w, x_i)$. We will assume that Tr_i includes $S_1, \ldots, S_{i-1}$, but not x_i. (Note that it follows from the authentication protocol that x_i is independent of Tr_i.)

We prove the following lemma (the proof can be found in the full version).

LEMMA 9. *Assume $\widetilde{\mathbf{H}}_\infty(w \mid \mathsf{Tr}_\mu) \geq \kappa + 1 + t$. Then,* $\Pr$[*Eve successfully responds to μ fresh random challenges*] $\leq 2^{-\mu(\lambda_r - 1)} + 2^{-\kappa - 1}$ *(where the probability is taken over the w and randomness of the challenges.)*

5.4 Putting it all together

In this section, we combine our theorems to give the proof of our main result (Theorem 2).

PROOF. In protocol $\mathsf{NewAuth}(w, m)$, Alice first converts the message m to a codeword $c = \mathsf{Edit}(\cdot)$. Next, Alice executes protocol $\mathsf{Auth}(w, c)$ and authenticates the bits of c to Bob. We have λ_m, the length of message m, to be $4(\kappa + 1)$ and hence $\lambda_c = \frac{4(\kappa+1)}{\rho}$.

The correctness of the protocol follows from the correctness of $\mathsf{Auth}(w, c)$ and the distance property of the code $\mathsf{Edit}(\cdot)$.

We consider an adversary Eve that succeeds in the security game of the interactive authentication protocol. Now, in order to succeed in the security game, Eve must make Bob accept a message $m' \neq m$. We first note that if $m' \neq m$, then by the property of the edit distance code $\mathsf{Edit}(\cdot)$, $\mathsf{EditDis}(c', c) > e_A \lambda_c$, where $c = \mathsf{Edit}(m), c' = \mathsf{Edit}(m')$. So, now if Eve were to make Bob accept a message $m' \neq m$, she must make Bob accept a message $c' \neq c$ in $\mathsf{Auth}(w, c)$, where the edit distance between c' and c is greater than $e_A \lambda_c = \frac{4e_A(\kappa+1)}{\rho}$. Now, by Corollary 2, this means that Eve must respond to more than $\frac{e_A(\kappa+1)}{\rho}$ fresh random challenges. Next, by Lemma 9, the probability of Eve responding to $\frac{e_A(\kappa+1)}{\rho}$ fresh random challenges is at most $p = 2^{-\frac{e_A(\kappa+1)}{\rho}(\tau - 1)} + 2^{-(\kappa+1)}$. Now, we have $\tau = \frac{\rho}{e_A} + 1$ and hence we get $p < 2^{-\kappa}$.

To calculate the entropy loss, first note that we authenticate a message c of length $\frac{4(\kappa+1)}{\rho}$. While authenticating a 0 bit, Eve gets to see 1 extractor response and while authenticating a 1 bit, Eve gets to see 2 extractor responses. Since the length of each extractor response is τ, we get that the entropy loss is at most $2 \times \frac{4(\kappa+1)}{\rho} \times \tau = \frac{8\tau(\kappa+1)}{\rho}$, which proves the theorem.

□

As mentioned before, the above authentication protocol can be used to construct a privacy amplification protocol. Since ρ, e_A and τ are constants, we obtain $\Theta(\kappa)$ entropy loss in all the protocols.

6. CONCLUSIONS

We have presented a protocol that allows two parties sharing a low entropy secret to extract a shared key of optimal length – if the shared secret has entropy m, then the length of the extracted key is $m - \Theta(\kappa)$ where κ is the security parameter. We obtain our result through a somewhat unexpected application of edit distance codes. While our protocol has optimal entropy loss, it has a round complexity of $\Theta(\kappa)$. On the other hand, Dodis and Wichs [7] showed nonconstructively that there exists a protocol with both optimal entropy loss and optimal round complexity (2 rounds). An interesting open problem would be to bring down the round complexity of a protocol with optimal entropy loss (such as ours) to 2 with a polynomial-time protocol.

7. ACKNOWLEDGMENTS

We thank Alexandr Andoni, Yevgeniy Dodis, and Madhu Sudan for helpful discussions. We also thank the anonymous reviewers of STOC 2010 for their detailed comments.

8. REFERENCES

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

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

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

[4] N. Chandran, B. Kanukurthi, R. Ostrovsky, and L. Reyzin. Privacy amplification with asymptotically optimal entropy loss. Cryptology ePrint Archive, 2010. http://eprint.iacr.org/2010/.

[5] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In C. Dwork, editor, *Advances in Cryptology—CRYPTO 2006*, volume 4117 of *LNCS*, pages 232–250. Springer-Verlag, 20–24 Aug. 2006.

[6] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. arXiv:cs/0602007.

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

[8] V. Guruswami, C. Umans, and S. P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In *IEEE Conference on Computational Complexity*, pages 96–108, 2007.

[9] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Extracting correlations. In *50th Annual Symposium on Foundations of Computer Science*, Atlanta, Georgia, Oct. 2009. IEEE.

[10] B. Kanukurthi and L. Reyzin. An improved robust fuzzy extractor. In R. Ostrovsky, R. D. Prisco, and I. Visconti, editors, *Sixth Conference on Security and Cryptography in Networks SCN '08*, volume 5229 of *LNCS*, pages 206–223, Sept. 2008.

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

[12] B. Kanukurthi and L. Reyzin. Key agreement from close secrets over unsecured channels. In A. Joux, editor, *Advances in Cryptology—EUROCRYPT 2009*, volume 5479 of *LNCS*. Springer, 2009.

[13] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–710, Feb. 1966.

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

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

[16] 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.

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

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

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

[20] R. Renner and S. Wolf. The exact price for unconditionally secure asymmetric cryptography. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 109–125. Springer-Verlag, 2004.

[21] L. J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, 1999.

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