Amplifying Privacy in Privacy Amplification

Divesh Aggarwal[1], Yevgeniy Dodis[1⋆], Zahra Jafargholi[2⋆⋆], Eric Miles[2], and Leonid Reyzin[3⋆⋆⋆]

[1] New York University
[2] Northeastern University
[3] Boston University

Abstract. We study the classical problem of privacy amplification, where two parties Alice and Bob share a weak secret X of min-entropy k, and wish to agree on secret key R of length m over a public communication channel completely controlled by a computationally unbounded attacker Eve.

Despite being extensively studied in the literature, the problem of designing "optimal" efficient privacy amplification protocols is still open, because there are several optimization goals. The first of them is (1) minimizing the *entropy loss* $L = k - m$. Other important considerations include (2) minimizing the number of communication rounds, (3) maintaining security even after the secret key is used (this is called *post-application robustness*), and (4) ensuring that the protocol P does not leak some "useful information" about the source X (this is called *source privacy*). Additionally, when dealing with a very long source X, as happens in the so-called Bounded Retrieval Model (BRM), extracting as long a key as possible is no longer the goal. Instead, the goals are (5) to touch as little of X as possible (for efficiency), and (6) to be able to run the protocol many times on the same X, extracting multiple secure keys.

Achieving goals (1)-(4) (or (2)-(6) in BRM) simultaneously has remained open. In this work we improve upon the current state-of-the-art, by designing a variety of new privacy amplification protocols, thereby achieving the following goals for the first time:

- 4-round (resp. 2-round) *source-private* protocol with *optimal entropy loss* $L = O(\lambda)$, whenever $k = \Omega(\lambda^2)$ (resp. $k > \frac{n}{2}(1-\alpha)$ for some universal constant $\alpha > 0$).
- 3-round *post-application-robust* protocols with *optimal entropy loss* $L = O(\lambda)$, whenever $k = \Omega(\lambda^2)$ or $k > \frac{n}{2}(1-\alpha)$ (the latter is also *source-private*).
- The first BRM protocol capable of extracting the optimal number $\Theta(k/\lambda)$ of session keys, improving upon the previously best bound $\Theta(k/\lambda^2)$. (Additionally, our BRM protocol is post-application-robust, takes 2 rounds, and can be made source-private by increasing the number of rounds to 4.)

⋆ Supported by NSF CNS Grants 1314568, 1319051, 1065288, 1017471, and Faculty Awards from Google and VMware.
⋆⋆ Supported by NSF grants CCF-0845003 and CCF-1319206.
⋆⋆⋆ Supported by NSF grants 1012798 and 1012910

1 Introduction

We study the classical problem of *privacy amplification* [3,22,2,23] (PA), in which two parties, Alice and Bob, share a weak secret X (of length n bits and min-entropy $k < n$) and wish to agree on a close-to-uniform secret key R of length m bits. We consider the active-adversary case, in which the communication channel between Alice and Bob can be not only observed, but also fully controlled, by a computationally unbounded attacker Eve. The most natural quantity to optimize here is the *entropy loss* $L = k - m$ (for a given security level $\varepsilon = 2^{-\lambda}$), but several other parameters (described below) are important as well.

Aside from being clean and elegant, this problem arises in a number of applications, such as biometric authentication, leakage-resilient cryptography, and quantum cryptography. Additionally, the mathematical tools used to solve this problem (such as randomness extractors [24]) have found many other applications in other areas of cryptography and complexity theory. Not surprisingly, PA has been extensively studied in the literature, as we survey below.

In the easier "passive adversary" setting (in which Eve can observe, but not modify), PA can be solved by applying a (strong) *randomness extractor* [24], which uses a uniformly random nonsecret seed S to extract nearly uniform secret randomness from the weak secret X. A randomness extractor accomplishes passive-adversary PA in one message: Alice sends the seed S to Bob, and both parties compute the extracted key $R = \mathsf{Ext}(X; S)$. Moreover, it is known that the optimal entropy loss of randomness extractors is $L = \Theta(\log(1/\varepsilon))$ [25], and this bound can be easily achieved (e.g. using the Leftover Hash Lemma [16]).

Active Eve Setting: Number of Rounds vs. Entropy Loss. The situation is more complex in the "active Eve" setting. Existing one-message solutions [23,9] work for min-entropy $k > n/2$ and require large entropy loss $L > n - k$. It was shown by [13,14] that $k > n/2$ is necessary, and that the large entropy loss of $n - k$ is likely necessary, as well. Thus, we turn to protocols of two or more rounds.

Two rounds were shown to be sufficient by [14], who proved, nonconstructively, the existence of two-round PA protocols with optimal entropy loss $L = \Theta(\log(1/\varepsilon))$ for any k. (This was done using a strengthening of extractors, called *non-malleable extractors*, whose existence was shown in [14].) Constructively, no such protocols are known, and all known constructive results sacrifice either the number of rounds, or the entropy loss, or the minimum entropy requirement. A protocol of [19, Theorem

1.9] (building on [27,17,6]) sacrifices the number of rounds: it achieves $L = O(\log(1/\varepsilon))$, but only in $O(1 + \log(1/\varepsilon)/\sqrt{k})$ rounds. The protocol of [19, Theorem 1.6] (building on [14]) sacrifices the minimum entropy requirement: it achieves $L = O(\log(1/\varepsilon))$ in two rounds, but only when $k = \Omega(\log^2(1/\varepsilon))$. Protocols of [10,7,18,20] make an incomparable minimum entropy requirement: they also achieve $L = O(\log(1/\varepsilon))$ in two rounds, but require that $k > n/2$ (with the exception of [20], who slightly relaxed it to $k > \frac{n}{2}(1-\alpha)$ for some tiny but positive constant α). These protocols also built the first constructive non-malleable extractors when $k > n/2$. The result of [19, Theorem 1.8] (building on [10,18]) further relaxes the entropy requirement to $k > \delta n$ for any constant $\delta > 0$. It also achieves $L = O(\log(1/\varepsilon))$ in two rounds, but the constant hidden in the O-notation is $g(\delta) = 2^{(1/\delta)^c}$ for some astronomical (and not even exactly known) constant c.[4] More generally, since some of the protocols mentioned above hide relatively large (or, as in the last example, even astronomical) constant factors, simpler protocols (such as [14] or [17]) may outperform asymptotically optimal ones for many realistic settings of parameters.

To summarize, the landscape of existing PA protocols is rather complex, even if we consider only the tradeoff between the min-entropy, the entropy loss, and the number of rounds. The situation becomes even more complex, if one adds additional highly desirable properties: *source privacy*, *post-application robustness*, and *local computability*. We consider those next.

Source Privacy. Intuitively, this property demands that the transcript of the protocol (even together with the derived key R!) does not reveal any "useful information" about the source X; or, equivalently (as shown by [12]), that the transcript does not reveal any information at all about the *distribution* of X (beyond a lower bound k on its min-entropy). For the case of passive Eve, source privacy was considered by Dodis and Smith [12], who showed that randomness extractors are indeed source-private. For active Eve, the only work that considered this notion is the elegant paper [4], which constructed a 4-round private protocol with entropy loss $L = O(\log^2(1/\varepsilon))$. Thus, unlike for PA protocols without source privacy,

(A) *no source-private PA protocol is known which achieves either optimal entropy loss* $L = O(\log(1/\varepsilon))$*, or fewer than four rounds.*

[4] The value c depends on some existential results in additive combinatorics. However, it appears safe to conclude that it is astronomical, which translates into "triply astronomical" $g(\delta) = 2^{(1/\delta)^c}$, even for $\delta = 0.49$.

Post-Application Robustness. Informally, the basic authenticity notion of PA protocols, called *pre-application robustness* by [9], simply states that Eve cannot force Alice and Bob to agree on different keys $R_A \neq R_B$. While easy to define, this property is likely insufficient for most applications of PA protocols, because in any two-party protocol, one party (say, Bob) has to finish before the other party. In this case, Bob is not sure if Alice ever received his last message, and must somehow decide to use his derived key R_B. In doing so, he might leak some partial information about R_B (possibly all of it!), and Eve might now use this partial (or full) information to modify the last message that Bob originally sent to Alice. Motivated by these considerations, [9] defined a strong property called *post-application robustness*, which (intuitively) requires that Eve cannot modify Bob's last message and cause Alice to output $R_A \neq R_B$, even if given Bob's key R_B.

The only protocols known to achieve post-application robustness are in [9,14,10]. Of those, only the protocol of [10] achieves asymptotically optimally entropy loss: for entropy $k > \delta n$, it achieves entropy loss $O((1/\delta)^c \log(1/\varepsilon))$ in $O((1/\delta)^c)$ rounds for some astronomical constant c mentioned in Footnote 4. Most protocols in [27,9,14,6,10,7,18,20,19] are proven only for pre-application robustness (some works simply ignored the distinction). In particular,

(B) *no post-application robust, constant-round protocol with optimal entropy loss is known (with the exception of protocol of [10] using astronomical constants mentioned above).*

Local Computability and Reusability. Local computability is of interest when the length and the min-entropy of the source X is much larger than the desired number of extracted bits m. In such a case, it is desirable to compute the output without having to read all of the source. This property is traditionally associated with the Bounded Retrieval Model (BRM) [15,8], where the random source X is made *intentionally huge*, so that X still has a lot of entropy k even after the attacker ("virus") managed to download a big fraction of X over time. For historical reasons, we will also use the term "BRM", but point out that local computability seems natural in any scenario where $k \gg m$, and not just the BRM application.

The right way to think about entropy loss in such a scenario is not via the formula $L = k - m$, because entropy from X is not "lost": much entropy remains in X even after the protocol execution, because most of X is not even accessed. In fact, the PA protocol may be run multiple times on

the same X, to obtain multiple keys, until the entropy of X is exhausted. Specifically focusing on $m = \Theta(\log(1/\varepsilon))$ (so that the extracted key can be used to achieve ε security), "optimal" reusability means the ability to extract $\Theta(k/\log(1/\varepsilon))$ keys (assuming the entropy rate of X is constant).

In the passive adversary case, optimal reusability is achievable with locally computable randomness extractors [21,28]. In the active adversary case, however, the story is again more complicated. The only prior work to consider local computability in this setting is the work of [14]. Reusability has not been explicitly considered before, but it is easy to see that the locally computable protocol of [14] allows the extraction of $\Theta(k/\log^2(1/\varepsilon))$ keys. Thus,

(C) *no prior locally computable protocol achieves optimal reusability.*

1.1 Our Results

In this work, we solve open problems (A), (B), and (C), by designing several new techniques for building PA protocols. Many of our techniques are *general transformations* that convert a given protocol P into a "better" protocol P'. Given a wide variety of incomparable existing PA protocols (surveyed above), this modular approach will often allow us to obtain several improved protocols in "one go".

Two Methods of Adding Source Privacy. Our first method (Section 3.2) maintains the number of rounds at 2, at the expense of using a strengthening of non-malleable extractors [14] (which we call *adaptive non-malleable extractors*) to derive a one-time pad to mask the "non-private" message which should be sent in the second round. (Given that we already use non-malleable extractors however, we might as well combine our protocol with the non-private protocol of [14] based on non-malleable extractors with similar parameters; this is what we do to keep things simple.) Our second method (Section 3.3), inspired by the specific protocol of [4], turns certain 2-round non-private protocols into 4-round private protocols, using standard extractors and XOR-universal hash functions. (The concrete protocol of [4] implicitly applied a very particular variant of our transformation to the two-round protocol of [14], but we get improved results using "newer" protocol [19].) In particular, either one of these transformations will provide (with different tradeoffs) a positive answer to Open Question (A). For completeness, we also observe (Section 3.1) that the 1-round PA protocols of [9] are already source-private.

Pre- to Post-Application Robustness. We make a very simple transformation which converts pre-application robust protocols to post-application robust protocols, at the cost of one extra round, but with almost no increase in the entropy loss. Although very simple, it immediately gives a variety of answers to Open Question (B) (and can also be combined with our first transformation, since it preserves source privacy).

Overall, by applying our transformations above to different protocols and in various orders, we get several improvements to existing protocols, summarized in Table 1 (which includes various solutions to Questions (A), (B), and more).

Result	Entropy	Rounds	Entropy Loss		Source Privacy
			Pre-app	Post-app	
[14] (non-expl.)	$k = \Omega(\log(1/\varepsilon))$	2	$\Theta(\log(1/\varepsilon))$	$\Theta(\log(1/\varepsilon))$	NO
This work (non-expl.)	$k = \Omega(\log(1/\varepsilon))$	2	$\Theta(\log(1/\varepsilon))$	$\Theta(\log(1/\varepsilon))$	**YES**
[9]	$k > \frac{n}{2}$	1	$n - k - \Theta(\log(1/\varepsilon))$	$\frac{n}{2} + \Theta(\log(1/\varepsilon))$	**YES**[5]
[19]	$k = \Omega(\log^2(1/\varepsilon))$	2	$\Theta(\log(1/\varepsilon))$	$\Theta(\log^2(1/\varepsilon))$	NO
This work	$k = \Omega(\log^2(1/\varepsilon))$	3	$\Theta(\log(1/\varepsilon))$	$\Theta(\log(1/\varepsilon))$	NO
[4]	$k = \Omega(\log^2(1/\varepsilon))$	4	$\Theta(\log^2(1/\varepsilon))$	$\Theta(\log^2(1/\varepsilon))$	YES
This work	$k = \Omega(\log^2(1/\varepsilon))$	4	$\Theta(\log(1/\varepsilon))$	$\Theta(\log^2(1/\varepsilon))$	YES
This work	$k = \Omega(\log^2(1/\varepsilon))$	5	$\Theta(\log(1/\varepsilon))$	$\Theta(\log(1/\varepsilon))$	YES
[20]	$k > \frac{n}{2}(1-\alpha)$	2	$\Theta(\log(1/\varepsilon))$	$\frac{n}{2}(1-\alpha) + \Theta(\log(1/\varepsilon))$	NO
This work	$k > \frac{n}{2}(1-\alpha)$	2	$\Theta(\log(1/\varepsilon))$	$\frac{n}{2}(1-\alpha) + \Theta(\log(1/\varepsilon))$	**YES**
This work	$k > \frac{n}{2}(1-\alpha)$	3	$\Theta(\log(1/\varepsilon))$	$\Theta(\log(1/\varepsilon))$	**YES**

Table 1. Our improvement (also marked in **RED**) over prior PA protocols.

Achieving Local Computability and Optimal Reusability. While only the work of [14] explicitly considered local computability, it is reasonable to ask if other existing protocols can be modified to be locally computable and reusable. To achieve optimal reusability, we focus on protocols with optimal entropy loss, because they have the property that the protocol transcript reduces the entropy of X by $O(\log(1/\varepsilon))$, leaving *residual* entropy of X high. They can be modified to extract a short key of length $\Theta(\log(1/\varepsilon))$, which will give optimal reusability.

To achieve local computability, extractors used within a protocol can be replaced with locally computable extractors. Indeed, the protocol of [6] seems to amenable to such modification. However, it is not constant-round. Most other constant-round protocols with optimal entropy loss [10,7,18,20] use non-malleable extractors, and this approach fails, because no locally

[5] We observe in this paper that this protocol is private.

computable (even non-constructive!) instantiations of non-malleable extractors are known.

However, we observe that the 2-round, optimal entropy loss protocol of [19, Theorem 1.6] does not use non-malleable extractors. Moreover, by making all extractors in that protocol locally computable, we get a locally computable, 2-round protocol. However, the security analysis of [19] uses a very delicate and interdependent setting of various parameters for the security proof to go through. Hence, it is not immediately clear if this intricate proof will go though if one uses locally computable extractors. Instead, we will develop a different, *modular* analysis underlying the key ideas of [19], which will give us a rigorous 2-round solution to open problem (C), as well as have other benefits we describe shortly. Specifically, we show a general transformation that turns certain (post-application) secure 2-round protocols into 2-round protocols *with optimal entropy loss* $L = O(\log(1/\varepsilon))$ *and residual min-entropy* $k' = k - O(\log(1/\varepsilon))$ (Section 5). The transformation uses two-source extractor of [26] to compress the second message of the protocol to only $O(\log(1/\varepsilon))$ bits. By applying this transformation to the original (non-BRM) protocol of [14], we get a protocol very similar to the protocol of [19], but with a much more modular and easier-to-follow security analysis. On the other hand, by using the locally computable protocol of [14] instead (see Section 6), we get a 2-round locally computable protocol with optimal residual entropy (and, thus, reusability), solving open problem (C).[6] Furthermore, we can add source privacy by using our 2-to-4-round transformation mentioned earlier, which can be done via local computation as well.

These results are summarized in Table 2.

Improving Entropy Loss of Post-Application Robust Protocols. As another advantage of our modular approach, we note that the transformation described in the previous paragraph is interesting not only in the context of local computability. It also allows one to turn *post-application* robust 2-round protocols with *sub-optimal* entropy loss L into 2-round *pre-application* robust protocols with *optimal* entropy loss, which then (using our pre-application to post-application transformation described above) can be turned into 3-round *post-application* robust protocols with *optimal* entropy loss. Namely, we can obtain optimal entropy

[6] Interestingly, the main limitation of the non-BRM protocol of [19] — high min-entropy requirement $k = \Omega((\log(1/\varepsilon))^2)$ — is not an issue in the BRM model. Thus, we can view our result as finding a "practical application scenario" for the very elegant communication reduction technique developed by [19].

loss at the expense of one extra round. (For the BRM setting, no extra round is needed, as we only extract "short" keys of length $O(\log(1/\varepsilon))$.)

Result	Rounds	Residual Min-entropy	# Keys Extracted	Source Privacy
[14]	2	$k - \Theta(\log^2(1/\varepsilon))$	$\Theta(k/\log^2(1/\varepsilon))$	NO
This work	2	$k - \Theta(\log(1/\varepsilon))$	$\Theta(k/\log(1/\varepsilon))$	NO
This work	4	$k - \Theta(\log(1/\varepsilon))$	$\Theta(k/\log(1/\varepsilon))$	**YES**

Table 2. Protocols in the Bounded Retrieval Model; each extracts $\Theta(\log(1/\varepsilon))$ bits per key, is post-application robust, and requires $k = \Omega(\log^2(1/\varepsilon))$. Entries in **RED** mark our improvements.

2 Preliminaries

For a set S, we let U_S denote the uniform distribution over S. For an integer $m \in \mathbb{N}$, we let U_m denote the uniform distribution over $\{0,1\}^m$, the bit-strings of length m. For a distribution or random variable X we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set S, we write $s \leftarrow S$ as shorthand for $s \leftarrow U_S$.

Entropy and Statistical Distance. The *min-entropy* of a random variable X is defined as $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. We say that X is an (n,k)-*source* if $X \in \{0,1\}^n$ and $\mathbf{H}_\infty(X) \geqslant k$. For $X \in \{0,1\}^n$, we define the *entropy rate* of X to be $\mathbf{H}_\infty(X)/n$. We also define *average (aka conditional) min-entropy* of a random variable X conditioned on another random variable Z as

$$\begin{aligned}\mathbf{H}_\infty(X|Z) &\stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z}\left[\ \max_x \Pr[X = x|Z = z]\ \right]\right)\\ &= -\log\left(\mathbb{E}_{z\leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right)\ ,\end{aligned}$$

where $\mathbb{E}_{z\leftarrow Z}$ denotes the expected value over $z \leftarrow Z$.
The *statistical distance* between two random variables W and Z distributed over some set S is

$$\Delta(W,Z) \stackrel{def}{=} \max_{T\subseteq S}(|W(T) - Z(T)|) = \frac{1}{2}\sum_{s\in S}|W(s) - Z(s)|.$$

Note that $\Delta(W,Z) = \max_D(\Pr[D(W) = 1] - \Pr[D(Z) = 1])$, where D is a probabilistic function. We say W is ε-close to Z, denoted $W \approx_\varepsilon Z$, if $\Delta(W,Z) \leq \varepsilon$. We write $\Delta(W,Z|Y)$ as shorthand for $\Delta((W,Y),(Z,Y))$.

We introduce some cryptographic primitives needed for our constructions.

EXTRACTORS. An extractor [24] can be used to extract uniform randomness out of a weakly-random value which is only assumed to have sufficient min-entropy. Our definition follows that of [11], which is defined in terms of conditional min-entropy.

Definition 1 (Extractors). *An efficient function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is an (average-case, strong)* (k,ε)-extractor*, if for all* X, Z *such that* X *is distributed over* $\{0,1\}^n$ *and* $\mathbf{H}_\infty(X|Z) \geq k$*, we get*

$$\Delta(\ (Z, Y, \mathsf{Ext}(X;Y))\ ,\ (Z, Y, U_m)\ ) \leqslant \varepsilon$$

where $Y \equiv U_d$ *denotes the coins of* Ext *(called the* seed*). The value* $L = k - m$ *is called the* entropy loss *of* Ext*, and the value* d *is called the* seed length *of* Ext*.*

MESSAGE AUTHENTICATION CODES. One-time message authentication codes (MACs) use a shared random key to authenticate a message in the information-theoretic setting.

Definition 2 (One-time MACs). *A function family* $\{\mathsf{MAC}_R : \{0,1\}^d \to \{0,1\}^v\}$ *is an* ε-secure one-time MAC *for messages of length* d *with tags of length* v *if for any* $w \in \{0,1\}^d$ *and any function (adversary)* $A : \{0,1\}^v \to \{0,1\}^d \times \{0,1\}^v$*,*

$$\Pr_R[\mathsf{MAC}_R(W') = T' \wedge W' \neq w \mid (W', T') = A(\mathsf{MAC}_R(w))] \leq \epsilon,$$

where R *is the uniform distribution over the key space* $\{0,1\}^\ell$*.*

XOR-UNIVERSAL HASH FUNCTIONS. We recall the definition of XOR-universal-hashing [5].

Definition 3 (ρ-XOR-Universal Hashing). *A family* $\mathcal{H}$ *of (deterministic) functions* $h : \{0,1\}^u \to \{0,1\}^v$ *is a called* ρ-XOR-universal hash family*, if for any* $x_1 \neq x_2 \in \{0,1\}^u$ *and any* $a \in \{0,1\}^v$ *we have* $\Pr_{h \leftarrow \mathcal{H}}[h(x_1) \oplus h(x_2) = a] \leq \rho$*. When* $\rho = 1/2^v$*, we say that* $\mathcal{H}$ *is (perfectly)* XOR-universal*. The value* $\log|\mathcal{H}|$ *is called the seed length of* $\mathcal{H}$*.*

2.1 Privacy Amplification

We define a privacy amplification protocol (P_A, P_B), executed by two parties Alice and Bob sharing a secret $X \in \{0,1\}^n$, in the presence of an active, computationally unbounded adversary Eve, who might have some partial information E about X satisfying $\mathbf{H}_\infty(X|E) \geqslant k$. Informally, this means that whenever a party (Alice or Bob) does not reject, the key R output by this party is random and statistically independent of Eve's view. Moreover, if both parties do not reject, they must output the same keys $R_A = R_B$ with overwhelming probability. The formal definition is given below.

Definition 4. *An interactive protocol (P_A, P_B), executed by Alice and Bob on a communication channel fully controlled by an active adversary Eve, is a* (k, m, ϵ)-privacy amplification protocol *if it satisfies the following properties whenever $\mathbf{H}_\infty(X|E) \geq k$:*

1. Correctness. *If Eve is passive, then* $\Pr[R_A = R_B \wedge \ R_A \neq \perp \wedge R_B \neq \perp] = 1$.

2. Robustness. *We start by defining the notion of* pre-application *robustness, which states that even if Eve is active,* $\Pr[R_A \neq R_B \wedge \ R_A \neq \perp \wedge R_B \neq \perp] \leqslant \epsilon$. *The stronger notion of* post-application *robustness is defined similarly, except Eve is additionally given the key R_A the moment she completed the left execution (P_A, P_E), and the key R_B the moment she completed the right execution (P_E, P_B). For example, if Eve completed the left execution before the right execution, she may try to use R_A to force Bob to output a different key $R_B \notin \{R_A, \perp\}$, and vice versa.*

3. Extraction. *Given a string $r \in \{0,1\}^m \cup \{\perp\}$, let* $\mathsf{purify}(r)$ *be $\perp$ if $r = \perp$, and otherwise replace $r \neq \perp$ by a fresh m-bit random string U_m:* $\mathsf{purify}(r) \leftarrow U_m$. *Letting E' denote Eve's view of the protocol, we require that*

$$\Delta(R_A, \mathsf{purify}(R_A) \mid E') \leq \epsilon \quad \text{and} \quad \Delta(R_B, \mathsf{purify}(R_B) \mid E') \leq \epsilon$$

Namely, whenever a party does not reject, its key looks like a fresh random string to Eve.

The quantity $k - m$ is called the entropy loss *and the quantity* $\log(1/\epsilon)$ *is called the* security parameter *of the protocol.*

Source Privacy. Following Bouman and Fehr [4], we now add the source privacy requirement for privacy amplification protocols. To define this property, we let $\mathsf{FullOutput}(X, E)$ denote the tuple (E', R_A, R_B),

where Alice and Bob share a secret X and output keys R_A and R_B, respectively, and Eve starts with initial side information E and ends with final view E' at the end of the protocol.

Definition 5 (Source Privacy). *An interactive protocol (P_A, P_B), executed by Alice and Bob on a communication channel fully controlled by an active adversary Eve, is* (k, ε)-private*, if for any two distributions (X_0, E) and (X_1, E), where $\mathbf{H}_\infty(X_0|E) \geq k$ and $\mathbf{H}_\infty(X_1|E) \geqslant k$, we have*

$$\Delta(\mathsf{FullOutput}(X_0, E), \mathsf{FullOutput}(X_1, E)) \leq \varepsilon$$

Our definition is stronger than the definition of [4], who only required that the final transcript E' does not reveal any information about X.

3 New Private Protocols

3.1 One Round Private Protocol

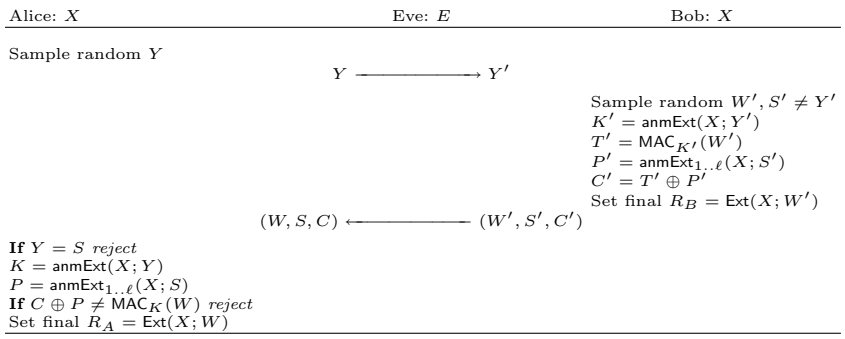
Dodis et al [9] gave a construction of robust extractors using which they gave one-round (k, m, ε)-secure privacy amplification protocols for $k > n/2 + O(\log(1/\varepsilon))$. We argue the source privacy of their protocols in the full version[1], and thus get the following result.

Theorem 1. *For $k > n/2$, there is an explicit polynomial-time, one-round $(k, 2\varepsilon + 2^{-n/2})$-private, (k, m, ε)-secure privacy amplification protocol with pre-application robustness and entropy loss $k - m = n - k + O(\log(1/\varepsilon))$. We get post-application robustness at the cost of increasing the entropy loss to $n/2 + O(\log(1/\varepsilon))$.*

3.2 Two Round Private Protocol with Optimal Entropy Loss

In this section, we give a two round protocol that achieves optimal entropy loss $O(\log(1/\varepsilon))$ for pre-application robustness. For post-application robustness, the entropy loss is about $n/2$, but we show how to improve it to $O(\log(1/\varepsilon))$ in Section 4 at the cost of 1 additional round.

Our Two Round Private Protocol. Our protocol (Protocol 1) makes the protocol of [14] private, using the same idea as [4]: we apply a one-time pad P' to the tag sent by Bob in the second round, T', where the pad P' is derived from X. We make use of an adaptive non-malleable extractor, where the adversary $\mathcal{A}$ is allowed to see Y, Z, and additionally either $\mathsf{anmExt}(X; Y)$ or $R \equiv U_m$ before producing the modified



Protocol 1: New 2-round Source-Private Protocol for $\mathbf{H}_\infty(X|E) > n/2$

seed Y', and still $\mathsf{anmExt}(X; Y)$ should be statistically close to R given $\mathsf{anmExt}(X; Y'), Y, Z$.

Using this, our protocol achieves the following result.

Theorem 2. *Let $2^{-n/4} < \varepsilon < 1/n$, and $\varepsilon' = \varepsilon/7$. Given a (τ, ε')-adaptive non-malleable extractor, for $k > \tau + \Theta(\log(1/\varepsilon))$ and output length $\Theta(\log(1/\varepsilon))$, there exists an explicit polynomial-time, two-round (k, ε)-private, (k, m, ε)-secure privacy amplification protocol with pre-application robustness and entropy loss $O(\log(1/\varepsilon))$. Furthermore, we get post-application robustness with entropy loss to $\tau + O(\log(1/\varepsilon))$.*

We can instantiate the above result using our construction (resp. existential proof) of adaptive non-malleable extractors to obtain the following results. The details can be found in the full version [1].

Corollary 1. *There exists a universal constant $\alpha > 0$, such that for $k > n/2(1-\alpha)$, there exists an explicit polynomial-time, two-round (k, ε)-private, (k, m, ε)-secure privacy amplification protocol with pre-application robustness and entropy loss $O(\log(1/\varepsilon))$. We get post-application robustness at the cost of increasing the entropy loss to $n/2(1-\alpha) + O(\log(1/\varepsilon))$.*

Corollary 2. *For $k = \Omega(\log(1/\varepsilon))$, there exists a two-round (k, ε)-private, (k, m, ε)-secure privacy amplification protocol with post-application robustness and entropy loss $k - m = O(\log(1/\varepsilon))$.*

3.3 Privacy using Extractors and XOR-Universal Hashing

In this section, we use a ρ-XOR universal hash function family to construct a 4-round protocol for private privacy amplification, given any 2

round privacy amplification protocol of the form Protocol 2, where the string sent in the first round is sampled independent of X. We note that all known 2 round protocols in the literature are of this generic form.

Alice: X	Eve: E	Bob: X
Sample random Y		Sample random W'
	$Y \longrightarrow Y'$	
		$K' = f_1(X, Y')$ $T' = f_2(K', W')$ Set final $R_B = g(X, W')$
	$(W, T) \longleftarrow (W', T')$	
$K = f_1(X, Y)$ **If** $T \neq f_2(K, W)$ *reject* Set final $R_A = g(X, W)$		

Protocol 2: A Generic 2-round Privacy Amplification Protocol

Let $\ell = \log(1/\varepsilon)$. Let $\mathcal{H}$ be a ε-XOR universal family of hash functions from $\{0,1\}^{|T|}$ to $\{0,1\}^{2\ell}$, and let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \mapsto \{0,1\}^{2\ell}$ be a $(k - 2\ell - 2|K| - |R_B|, \varepsilon)$ extractor. Using these, our protocol is depicted as Protocol 3.

Alice: X	Eve: E	Bob: X
Sample random Y, h		Sample random W', S'
	$Y \longrightarrow Y'$	
	$W, S \longleftarrow W', S'$	
	$h \longrightarrow h'$	
		$K' = f_1(X, Y')$ $T' = f_2(K', W')$ $C' = h'(T') \oplus \mathsf{Ext}(X; S')$ Set final $R_B = g(X, W')$
	$C \longleftarrow C'$	
$K = f_1(X, Y)$ $T = f_2(K, W)$ **If** $C \neq h(T) \oplus \mathsf{Ext}(X; S)$ *reject* Set final $R_A = g(X, W)$		

Protocol 3: A Generic 4-round Private Privacy Amplification Protocol

Theorem 3. *Let Protocol 2 be a 2-round $(k - u, m, \varepsilon)$-secure privacy amplification protocol with pre- (resp. post-) application robustness for $k - |T| - 2|K| - |R_B| \geqslant 2\ell$. Then Protocol 3 is a 4-round $(k, m, O(\sqrt{\varepsilon}))$-secure $(k, O(\sqrt{\varepsilon}))$-private privacy amplification protocol with pre- (resp. post-) application robustness.*

For a proof, refer to the full version. We apply this generic transformation to Li's recent 2 round (k, ε)-secure privacy amplification protocol for $k = \Omega(\log^2(1/\varepsilon))$, that achieves entropy loss $O(\log(1/\varepsilon))$ for pre-application robustness, and $O(\log^2(1/\varepsilon))$ for post-application robustness [19]. We get the following result.

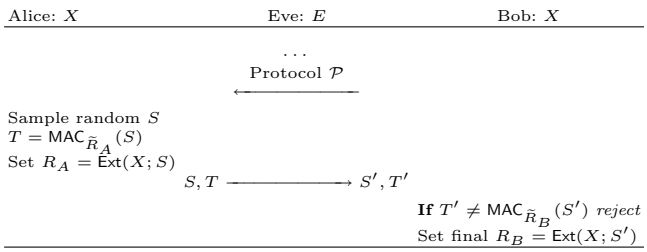
Corollary 3. *For $k = \Omega(\log^2(1/\varepsilon))$, there exists an explicit polynomial-time, 4-round (k, ε)-private, (k, m, ε)-secure privacy amplification protocol with pre-application robustness and entropy loss $L = k - m = O(\log(1/\varepsilon))$. We get post-application robustness with entropy loss $O(\log^2(1/\varepsilon))$.*

In Section 4, we will see how to get a 5-round private privacy amplification protocol with post-application robustness and entropy loss $O(\log(1/\varepsilon))$.

4 From Pre-application to Post-application Robustness

In this section, we show a generic transformation from a t-round privacy amplification protocol $\mathcal{P}$ that achieves pre-application robustness to a $(t+1)$-round protocol $\mathcal{P}'$ that achieves post-application robustness. The transformation can be described as follows.

Let $\ell = \log(1/\varepsilon)$. Without loss of generality, assume that the last message in $\mathcal{P}$ was sent from Bob to Alice. Let $\widetilde{R}_A, \widetilde{R}_B$ denote the first u bits of the keys computed by Alice and Bob, respectively (Set $\widetilde{R}_A = \perp$ if Alice rejects, and $\widetilde{R}_B = \perp$ if Bob rejects). We need a $(k - O(\ell), \varepsilon)$-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and an ε-secure one-time MAC for d-bit messages, whose key length is u. Using these, the $(t+1)$-round protocol is depicted as Protocol 4.



Protocol 4: $(t+1)$-round Privacy Amplification Protocol $\mathcal{P}'$ with post-application robustness.

Theorem 4. *If Protocol $\mathcal{P}$ is (k, m, ε)-secure privacy amplification protocol with pre-application robustness and residual entropy $k - O(\log(1/\varepsilon))$, then Protocol $\mathcal{P}'$ is a $(k, m - O(\log(1/\varepsilon)), O(\varepsilon))$ secure privacy amplification protocol with post-application robustness. Additionally, if $\mathcal{P}$ is (k, ε) private, then $\mathcal{P}'$ is $(k, O(\varepsilon))$ private.*

For a proof of this theorem, refer to the full version [1].

Using this result, we can get optimal entropy loss for post-application robustness for several protocols as described in the full version [1].

5 Increasing Residual Entropy

We now consider the task of preserving as much entropy as possible in the weak source X, which is a natural goal and has implications in the Bounded Retrieval Model (see section 6). Formally, the *residual entropy* of an interactive protocol is $\min_{E'}(\mathbf{H}_\infty(X \mid E'))$ where E' is the adversary's view after the protocol. We refer to $\mathbf{H}_\infty(X \mid E) - \min_{E'}(\mathbf{H}_\infty(X \mid E'))$ as the loss in residual entropy. Our main result is the following transformation achieving loss in residual entropy $O(\log(1/\varepsilon)$, i.e. linear in the security parameter, which is optimal up to constant factors.

Theorem 5. *Assume that there is a 2-round (k, m, ε)-secure privacy amplification protocol with post-application robustness in which the first message is independent of the (n, k)-source X and we have $\log n = O(\log(1/\varepsilon))$, $\varepsilon \geq 2^{-m/C}$, and $k \geq C \log(1/\varepsilon)$ for sufficiently large C.*

Then there is a 2-round (k', m', ε')-secure privacy amplification protocol with residual entropy $\geq k' - O(\log(1/\varepsilon'))$ provided that $k' \geq k + C' \log(1/\varepsilon)$ and $\varepsilon' \geq \varepsilon^{1/C'}$ for sufficiently large C', and $m' = k' - O(\log(1/\varepsilon'))$ for pre-application robustness or $m' = k' - k - O(\log(1/\varepsilon'))$ for post-application robustness.

To achieve the transformation of Theorem 5, we need the following notion of a *receipt* protocol, which is essentially a 2-round message authentication protocol in which the party who speaks first chooses the message. Such protocols can be defined as follows.

Definition 6. *A* (k, ℓ, ε)-receipt protocol (for messages of length d) *is a function* $\mathsf{Receipt} : \{0,1\}^d \times \{0,1\}^r \times \{0,1\}^n \to \{0,1\}^\ell$ *that satisfies the following: for $Y \equiv U_r$, every $\mu \in \{0,1\}^d$, every X such that $\mathbf{H}_\infty(X|E) \geq k$, and every $\mu' \neq \mu, Y'$ chosen by an adversary given μ, Y, E,*

$$\mathbf{H}_\infty(\mathsf{Receipt}(\mu, Y, X) \mid Y, \mathsf{Receipt}(\mu', Y', X)) \geq \log(1/\varepsilon).$$

The main ingredient in proving Theorem 5 is the following, the proof of which is deferred to the full version [1].

Theorem 6. *Assume that there exists a polynomial-time (k, ℓ, ε)-receipt protocol for d-bit messages such that Alice communicates $\leq \ell$ bits and $2^{-C\ell} \leq \varepsilon \leq 1/(C\ell)$ for sufficiently large C.*

Then for any $r \leq \log(1/\varepsilon)/100$, there exists a polynomial-time $(k, r, 2^{-\Omega(r)})$-receipt protocol for d-bit messages where Alice communicates $O(\ell)$ bits.

Finally, we obtain the following corollary by instantiating Theorem 5 using the 2-round privacy amplification protocol with post-application robustness due to Dodis and Wichs [14, Cor. 4].

Corollary 4. *For $k = \Omega(\log^2(1/\varepsilon))$, there exists an explicit polynomial-time 2-round (k, m, ε)-secure privacy amplification protocol with post-application robustness that achieves $m = \Omega(\log(1/\varepsilon))$ and residual entropy $k - O(\log(1/\varepsilon))$.*

6 Applications to the Bounded Retrieval Model

In the Bounded Retrieval Model (BRM) [8,15], Alice and Bob share an (intentionally) very large secret key X. The idea is that the size of X makes it infeasible for an attacker Eve to learn the entire string, even if she has infiltrated either Alice or Bob's storage device, because of limits on the amount of data that can be transmitted out of the device. Thus as in previous sections we assume that Eve has some adversarially chosen side information E about X, but that $k := \mathbf{H}_\infty(X|E)$ is not too small. Specifically here we think of $k = \alpha n$ for some constant $0 < \alpha < 1$.

Since reading the entire string X would be prohibitively inefficient, any function used by Alice or Bob that takes X as input must only read a small number of positions, i.e. it must be *locally computable*. Dodis and Wichs observe [14, Sec. 5] that their privacy amplification protocol has the property that each function taking X as input is a standard extractor. These can be replaced with the constructions of locally computable extractors due to Vadhan [28], and thus the protocol works in the BRM.

One downside of the [14] protocol is that the second message (which depends on X) has length $\Omega(\log^2(1/\varepsilon))$, and thus the loss in residual entropy is $\Omega(\log^2(1/\varepsilon)) = \Omega(m^2)$. It would be more desirable to have loss in residual entropy $O(m)$, as then Alice and Bob could derive a total of $\Omega(k/m)$ secret keys, as opposed to only $O(k/m^2)$ keys.

Corollary 4 shows that the loss in residual entropy can be reduced to $O(m)$. This protocol remains locally computable and thus applicable to

the BRM, because still every function that takes X as input is a standard extractor and can be replaced by a locally computable extractor. In summary, we have the following.

Theorem 7. *For $k = \Omega(\log^2(1/\varepsilon))$, there exists an explicit polynomial-time 2-round $(k, m = \Omega(\log(1/\varepsilon)), \varepsilon)$-secure privacy amplification protocol in the BRM with post-application robustness and residual entropy $k - O(\log(1/\varepsilon))$, thus allowing a total of $\Omega(k/m)$ keys to be derived.*

By relaxing the number of rounds to four, we can obtain a BRM protocol that additionally has *source privacy* by instead plugging the [14, Cor. 4] protocol into the transformation of Theorem 3.

Theorem 8. *For $k = \Omega(\log^2(1/\varepsilon))$, there exists an explicit polynomial-time 4-round $(k, m = \Omega(\log(1/\varepsilon)), \varepsilon)$-secure (k, ε)-private privacy amplification protocol in the BRM with post-application robustness and residual entropy $k - O(\log(1/\varepsilon))$, thus allowing a total of $\Omega(k/m)$ keys to be derived.*

References

1. D. Aggarwal, Y. Dodis, Z. Jafargholi, E. Miles, and L. Reyzin. Amplifying privacy in privacy amplification. Cryptology ePrint Archive, Report 2013/723, 2014.
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. C. H. Bennett, G. Brassard, and J. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, Apr. 1988.
4. N. J. Bouman and S. Fehr. Secure authentication from a weak key, without leaking information. In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 246–265. Springer, 2011.
5. L. Carter and M. N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.
6. N. Chandran, B. Kanukurthi, R. Ostrovsky, and L. Reyzin. Privacy amplification with asymptotically optimal entropy loss. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, 2010.
7. G. Cohen, R. Raz, and G. Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *IEEE Conference on Computational Complexity*, pages 298–308. IEEE, 2012.
8. G. D. Crescenzo, R. J. Lipton, and S. Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.
9. Y. Dodis, B. Kanukurthi, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. *IEEE Transactions on Information Theory*, 58(9):6207–6222, 2012.
10. Y. Dodis, X. Li, T. D. Wooley, and D. Zuckerman. Privacy amplification and non-malleable extractors via character sums. In R. Ostrovsky, editor, *FOCS*, pages 668–677. IEEE, 2011.

11. 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*, 38:97–139, 2008.
12. Y. Dodis and A. Smith. Entropic security and the encryption of high entropy messages. In *TCC*, pages 556 – 577, 2005.
13. Y. Dodis and J. Spencer. On the (non)universality of the one-time pad. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, page 376. IEEE Computer Society, 2002.
14. Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In M. Mitzenmacher, editor, *STOC*, pages 601–610. ACM, 2009.
15. S. Dziembowski. Intrusion-resilience via the bounded-storage model. In *TCC*, pages 207–224, 2006.
16. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
17. B. Kanukurthi and L. Reyzin. Key agreement from close secrets over unsecured channels. In *EUROCRYPT*, pages 206–223, 2009.
18. X. Li. Design extractors, non-malleable condensers and privacy amplification. In H. J. Karloff and T. Pitassi, editors, *STOC*, pages 837–854. ACM, 2012.
19. X. Li. Non-malleable condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. *CoRR*, abs/1211.0651, 2012.
20. X. Li. Non-malleable extractors, two-source extractors and privacy amplification. In *FOCS*, pages 688–697. IEEE Computer Society, 2012.
21. C.-J. Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *J. Cryptology*, 17(1):27–42, 2004.
22. U. M. Maurer. Protocols for secret key agreement by public discussion based on common information. In E. F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 461–470. Springer, 1992.
23. U. M. Maurer and S. Wolf. Privacy amplification secure against active adversaries. In *CRYPTO '97*, 1997.
24. N. Nisan and D. Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.
25. J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000.
26. R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.
27. R. Renner and S. Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In *CRYPTO*, pages 78–95, 2003.
28. S. P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.