How to Extract Useful Randomness from Unreliable Sources

Divesh Aggarwal[3], Maciej Obremski[3], João Ribeiro[2], Luisa Siniscalchi[1], and Ivan Visconti[4]

[1] Concordium Blockchain Research Center, Aarhus University, Aarhus, Denmark
lsiniscalchi@cs.au.dk
[2] Imperial College London, London, United Kingdom
j.lourenco-ribeiro17@imperial.ac.uk
[3] National University of Singapore, Singapore, Singapore
divesh@comp.nus.edu.sg, obremski.math@gmail.com
[4] University of Salerno, Fisciano, Italy
visconti@unisa.it

Abstract. For more than 30 years, cryptographers have been looking for public sources of uniform randomness in order to use them as a set-up to run appealing cryptographic protocols without relying on trusted third parties. Unfortunately, nowadays it is fair to assess that assuming the existence of physical phenomena producing public uniform randomness is far from reality.

It is known that uniform randomness cannot be extracted from a single weak source. A well-studied way to overcome this is to consider several independent weak sources. However, this means we must trust the various sampling processes of weak randomness from physical processes.

Motivated by the above state of affairs, this work considers a set-up where players can access multiple *potential* sources of weak randomness, several of which may be jointly corrupted by a computationally unbounded adversary. We introduce *SHELA* (Somewhere Honest Entropic Look Ahead) sources to model this situation.

We show that there is no hope of extracting uniform randomness from a *SHELA* source. Instead, we focus on the task of *Somewhere-Extraction* (i.e., outputting several candidate strings, some of which are uniformly distributed – yet we do not know which). We give explicit constructions of *Somewhere-Extractors* for *SHELA* sources with good parameters.

Then, we present applications of the above somewhere-extractor where the public uniform randomness can be replaced by the output of such extraction from corruptible sources, greatly outperforming trivial solutions. The output of somewhere-extraction is also useful in other settings, such as a suitable source of random coins for many randomized algorithms.

In another front, we comprehensively study the problem of *Somewhere-Extraction* from a *weak* source, resulting in a series of bounds. Our bounds highlight the fact that, in most regimes of parameters (including those relevant for applications), *SHELA* sources significantly outperform *weak* sources of comparable parameters both when it comes to the process of *Somewhere-Extraction*, and in the task of amplification of success probability in randomized algorithms. Moreover, the low quality

of somewhere-extraction from weak sources excludes its use in various efficient applications.

1 Introduction

Perfect (i.e., uniform) public randomness is an extremely valuable resource in computer science, and in cryptography in particular. For example, it can be used to create a Common Reference String (CRS) drawn from an uniform distribution, which is a widely used set-up for cryptographic protocols. However, the randomness that we can obtain from physical phenomena (such as solar radiation, temperature readings, and electricity fluctuations) is far from perfect (in particular when public randomness sources are taken into account). Such phenomena belong to the family of *weak* randomness sources [20]. These are sources that carry some min-entropy, but are still very far from uniformly distributed. As a result, in most applications a so-called randomness extractor must be applied to the weak sources in order to extract (close to) uniformly distributed bits. A basic result about randomness extraction dictates that deterministic extraction from one weak source is not possible. Nevertheless, deterministic extraction *is* possible if one has access to at least two independent weak sources.

Sampling from several independent physical weak sources presents serious security issues. For example, if different phenomena are being publicly measured (to ensure some kind of independence), then different instrumentation and potentially different entities must be involved in the sampling process. Not only that, but sampling may also be compromised by instrument failures. Going back to our CRS example, if we want to generate CRS from such sources, then we are assuming that every instrument and entity that took part in sampling the weak sources is trusted. This is not a desireable situation, and indeed it was previously noticed that generating a uniformly distributed CRS from such weak sources is complicated [15]. A natural question follows: *Which forms of common public set-up can we achieve (or, more generally, what kind of randomness can we extract) if some of the sources are maliciously corrupted, but some of them remain honest?*

Intuitively, this scenario leads us to define a structured weak source in an adversarial setting where a sample from the source is divided into multiple subparts, that we call *blocks*. One may imagine that each block corresponds to a different sampling process as per the previous paragraph. In this setting there is an ordered sequence of samplings from the sub-sources and some of them are controlled by the adversary. More specifically, the adversary can decide the positions of the honest blocks since it can decide which sampling processes to corrupt. Honest blocks correspond to (correct) samples from independent weak sources (these sources are known to the adversary but are not controlled by the adversary). Given a sequence of blocks the sampling proceeds by obtaining blocks in chronological order. As a result, if the i-th block is to be corrupted, then the adversary is allowed to fix it to any value based on the (already determined) values from the first through $(i-1)$-th blocks.

We will call such source a "t-out-of-ℓ" Somewhere Honest Entropic Look-Ahead (SHELA) source, where ℓ indicates the total number of blocks, out of which t must be honest. We consider only the case $t \geq 2$, since the case $t = 1$ essentially reduces to the setting with a single weak source. Moreover, we assume without loss of generality[5] that each block has length n, and the honest blocks have min-entropy at least k for some decent parameter k. Observe that corrupted blocks are heavily correlated with previous honest blocks, and may even have zero min-entropy. Moreover, we allow the number of honest blocks t to be any function of ℓ, as long as $t \geq 2$.

There is a second real-world scenario that can be naturally modelled as a SHELA source. Some blockchains can be considered as sequences of blocks generated in chronological order, some of which contain high min-entropy strings. For instance, such strings could be the new wallet's identifier used to cash a reward when a new block is added to the chain, financial data containing some min-entropy [21], or a random nonce added for some security reasons. It is well-known [40,59] that in a sequence of blocks of the blockchain there will be a fraction ν of them added by honest players. Moreover, we could assume that when a new block is added to the blockchain by an honest player, such a block (sometimes) contains high min-entropy strings that are independent of the previous ones already in the blockchain (we notice that a similar assumption has already been used in [66]). Therefore, if we consider ℓ consecutive blocks and for each of them we consider the part of the block that, in case the block is honest, could contain an independent weak source with decent min-entropy, we obtain a public SHELA source[6].

1.1 Our Contributions

Our main goal in this paper is to study SHELA sources and what kind of applications their availability enables.

The first natural question that arises when encountering SHELA sources is the following: *Are we able to extract independent and (close to) uniformly distributed bits from it?* We will prove in this work that the answer to this question is negative. Given this, we shift our focus from standard randomness extraction, and instead we investigate the possibility of constructing a deterministic *somewhere-extractor* $\mathsf{SomeExt}$ for SHELA sources. Intuitively, the somewhere-extractor $\mathsf{SomeExt}$ takes as input a SHELA source and outputs a distribution that is close (in statistical distance) to a convex combination of so-called "T-out-of-L" Somewhere-Random (SR) sources. SR sources are composed of L blocks, T of which (at fixed, unknown

[5] Given blocks of different sizes, one can always fill out the shorter blocks with zeros, similarly given blocks of different min-entropy we can assume k to be the minimum of min-entriopies of honest blocks.

[6] In this example we are assuming that when using a blockchain as a SHELA source, the adversary of the sampling procedure from a SHELA source has no control over the choices of the honest blocks posted permanently in the blockchain (i.e., the adversary does not decide which honest block is selected and remains permanently in the blockchain out of multiple candidates).

positions) are jointly independent and uniformly distributed. We call a convex combination of SR-sources a convSR-source for short.

It turns out that convSR sources are an extremely useful type of randomness. For example, armed with our somewhere-extractor, we show how to build non-interactive (and thus accepted by any receiver) commitments from one-way functions and non-interactive (and thus publicly verifiable) witness indistinguishable proofs from generic complexity assumptions[7] when both players (a sender and a receiver, or a prover and a verifier, respectively) have access to a public SHELA source. Remarkably, convSR-sources are also important intermediate objects used in the construction of multi-source and non-malleable extractors for weak sources (we discuss this in more detail later).

Parameters of the somewhere-extractor for SHELA sources. The computational complexity and security of our applications of convSR-sources will heavily depend on various parameters of the convSR-source: the number of total blocks L, the number of "good" (i.e., independent and uniformly distributed) blocks T, and the length m of each block. In turn, these depend on the parameters of the underlying SHELA source and the quality of the somewhere-extractor.

Ideally, we want our somewhere-extractor SomeExt to extract a convSR source with low error, small number of total blocks, and large block length from a SHELA source. More precisely, the error ε of SomeExt should satisfy $\varepsilon = 2^{-\Omega(n)}$, where n is the block length of the SHELA source, the total number L of blocks of the convSR source should be at most $O(\ell)$, where ℓ denotes the total number of blocks in the SHELA source, and the length m of each output block should satisfy $m = \Omega(n)$. We will comment later that these parameters ensure that the output of SomeExt can be used in our applications without compromising security, while ensuring that the efficiency and reliability of the application in question remain good enough.

Moreover, we do not want to assume that honest blocks in the SHELA source must have significant amounts of min-entropy for extraction to be successful. Instead, we aim to extract such high-quality convSR-sources from SHELA sources whose honest blocks have *arbitrary* constant min-entropy rate. In other words, we allow the min-entropy k of each honest n-bit block to satisfy $k = \delta n$ for an arbitrarily small constant $\delta > 0$.

A very first naive approach to designing a somewhere-extractor (that we will denote by NaiveSomeExt) is to apply a c-source extractor, for $c \geq 2$, to every subset of c blocks of a SHELA source. This immediately leads to a convSR-source. However, the total number of output blocks satisfies $L = \Theta(\ell^c)$ for $c \geq 2$, where ℓ denotes the total number of blocks of the SHELA source. This leads to a much worse efficiency blow-up for applications than what we aim to obtain, as detailed earlier. Another problem of the naive construction is that, if we wish to minimize the blowup of L with respect to ℓ by setting $c = 2$, we run into problems of

[7] We will show how to start from any public-coin 2-round WI proof system in the standard model which in turn means any non-interactive zero-knowledge proof system in the common random string model [34].

explicitness. In fact, known explicit constructions of 2-source extractors require sources with high min-entropy to achieve exponentially small error [12,46,18]. We also note that, besides leading to worse efficiency, using a c-source extractor for $c > 2$ requires assuming that there are at least $c > 2$ honest blocks in the SHELA source, which might not be reasonable in some scenarios.

In this work, we design a non-trivial somewhere-extractor $\mathsf{SomeExt}$ that achieves our ideal goals put forth above. We begin by looking at the setting where the min-entropy rate k/n of honest blocks in the SHELA source is a large enough constant. In this case, if $X \in \{0,1\}^{n \cdot \ell}$ is a t-out-of-ℓ SHELA source with honest block min-entropy $k = \delta n$, then $\mathsf{SomeExt}(X)$ is ε-close to a T-out-of-L convSR-source $Y \in \{0,1\}^{m \cdot L}$ with $T = t - 1$, $L = \ell - 1$, $\varepsilon = 2^{-\Omega(n)}$, and output block length $m = \Omega(n)$. The only thing missing is that, as previously discussed, we wish to extract with similar parameters from SHELA sources whose honest blocks have arbitrarily small constant min-entropy rate (i.e., $k = \delta n$ for arbitrarily small constant $\delta > 0$). Notably, using a modified construction, we are able to transfer these ideal parameters to the "arbitrary constant min-entropy rate" setting. The only difference is that now $L = O(\ell)$.

Somewhere-extraction of SHELA source vs. weak source. We have already established that we can deterministically extract high-quality convSR-sources from SHELA sources. However, an attentive reader might notice that deterministic somewhere extraction is also possible from *weak* sources. In fact, any strong seeded (k, ε)-extractor with seed length d yields a somewhere-extractor with error ε, $L = 2^d$ total output blocks, and $T = 1$ uniform blocks for weak sources with min-entropy at least k by considering a block for each possible fixing of the seed. This naive construction of a convSR-source is actually crucial in many constructions of multi-source extractors (we expand on this later in this section). However, it has strong limitations. In particular, even if we use an optimal strong seeded extractor, seed length lower bounds [61] imply that

$$L = \Omega\left(\frac{1}{\varepsilon^2}\right). \tag{1}$$

This means that if we require $\varepsilon = 2^{-\Omega(n)}$, then $L = 2^{\Omega(n)}$, which precludes any efficient cryptographic application of the resulting convSR-source.

Given the above shortcoming, one might wonder whether significantly better somewhere-extractors exist for weak sources. We dedicate part of our paper to the study of this problem. It turns out that the answer to this question is largely negative. In particular, a disperser-based lower bound shows that, similarly to the naive construction above, *every* somewhere-extractor for weak sources with error $\varepsilon = 2^{-\Omega(n)}$ and output block length $m = \Omega(n)$ must have $L = 2^{\Omega(n)}$ total output blocks.

In our work, we derive a set of lower bounds that complement each other and succeed in showing that somewhere-extractors for weak sources must perform significantly worse than the analogous objects for SHELA sources over various regimes of parameters. We are particularly interested in lower bounds on the total

number of blocks of the output convSR-source, as this dictates the computational complexity blow-up suffered by a protocol when using this source. In the end, we put forth the conjecture that the above lower bound (1) actually holds for *every* somewhere-extractor (regardless of the output block length m), and we make some progress towards proving it.

Randomized algorithms and amplification of success probability using SHELA source vs. weak source. We remark that convSR-sources are well-suited for simulation of randomized algorithms whose outputs can be efficiently checked for correctness (e.g., searching for witnesses for the membership of some string in an NP language, or approximation algorithms for NP languages). In fact, one can simply run the algorithm using each block as its randomness. As a result, one obtains a few candidate solutions, and can efficiently check if at least one of them is correct. The success probability of the algorithm is thus amplified by the number of good (i.e., uniform) blocks.

It is well-known and easy to see that, in the procedure above, we do not need good blocks to be exactly uniformly distributed. Indeed, it is enough to rely on the weaker guarantee that good blocks are sufficiently close to uniform in statistical distance, say, $1/\mathrm{poly}(n)$-close, where n is some soundness parameter. We call this weaker family of sources *somewhere-amplifiable* (SA) sources, and denote the class of convex combinations of SA-sources as convSA-sources.

While weak sources can be used to efficiently produce convSA-sources, we show that this comes at a heavy price: Roughly speaking, if one wants to generate enough, and long enough, good blocks for appropriate and efficient success probability amplification, then the weak source needs to have very high min-entropy. Therefore, in many reasonable regimes of parameters, one is unable to extract suitable convSA-sources from weak sources, while one can extract high-quality convSR-sources (a stronger notion) from SHELA sources in those regimes. We refer to Section 6 for a more detailed discussion.

We conclude from the two discussions above that there is a fundamental separation between somewhere-extraction from SHELA and weak sources. Indeed, we are able to efficiently extract convSR-sources with much higher quality from a SHELA source than what we can obtain from a weak source.

Non-interactive witness indistinguishable proofs assuming public-coin ZAPs and relying on public SHELA sources. In a proof system, a prover proves to a verifier the veracity of some statement $x \in \mathcal{L}$ (where $\mathcal{L}$ is an NP-language). A soundness property guarantees that it is unlikely that an honest verifier accepts the proof of a false statement. When a proof system is non-interactive any verifier is able to check the validity of the proof. Non-interative proofs are therefore publicly verifiable and they are very appealing since the prover computes the proof once, while still it can be useful in many different cases (i.e., with many different verifiers). Non-interactiveness is usually trivial since a prover could just send a witness proving membership in the language. The interesting case consists of offering some form of privacy for the secret (i.e., the witness) of the prover. We will in particular consider witness indistinguishability

[36] that requires that the proof hides which witness has been used by the prover out of multiple witnesses. A special category of interactive proof systems is called "public coin" and refers to the role of the verifier that sends random strings only as messages. When there is only one message played by the verifier then a 2-round witness indistinguishable proof system is referred as ZAP[34]. The round of the verifier can be recycled among any polynomial number of proofs givens by provers. Since public-coin ZAPs exist, a natural question is whether the verifier can just be replaced by a sample from a high min-entropy source, therefore obtaining a non-interactive WI proof under the same computational assumptions of ZAPs and relying on the existence of SHELA sources. The answer is unfortunately negative. Indeed, consider the ZAP of [34]. The message of the prover consists of computing some non-interactive zero-knowledge (NIZK) proofs in the common random string model. In general, NIZK proofs (e.g., [36]) are not sound when the common random string is replaced by the output of high min-entropy sources. In turn, when trying to make a generic public-coin ZAP relying on a high min-entropy source non-interactive, soundness could be lost. Moreover, the issue with soundness remains also in case of parallel repetition since for some high min-entropy sources an accepting proof of a false statement can be produced with probability 1.

On the positive side, equipped with our constructive results about obtaining a convSR-source from a SHELA source, we show that assuming a public SHELA source, non-interactive witness indistinguishable proofs exist by just using a parallel repetition of any public-coin ZAP[8].

Finally, we notice that Goyal and Goyal [41] construct a non-interactive zero-knowledge argument of knowledge relying on any proof-of-stake (PoS) blockchain. The construction of [41] requires the existence of non-interactive witness-indistinguishable proof systems. If the proof-of-stake blockchain can be used to implement a SHELA source (as discussed previously), then by plugging our non-interactive witness-indistinguishable proof system in the construction of [41] we obtain a a non-interactive zero-knowledge argument of knowledge with improved complexity assumptions using specific PoS blockchains.

Non-interactive commitments from one-way functions and SHELA sources. In a commitment scheme, sender and receiver interact in a commitment phase so that the (even malicious) sender can later on show only one message consistent with such interaction, while the (even malicious) receiver has no specific advantage in detecting the message committed by the sender. The security property for the receiver is called "binding" while the security for the sender is called "hiding".

Non-interactive commitments guarantee that the sender has to work only once to produce a commitment of a message, while this commitment can be used to convince any receiver about the committed message. We focus on statistically

[8] Notice that we are considering generic weak sources and it is unknown whether such distributions can all be efficiently simulatable. Consequently we cannot obtain a non-interactive zero knowledge proof.

binding commitments where, except with negligible probability, there is a unique message that is consistent with the transcript of the commitment phase, regardless of the computational power of the (even malicious) sender. A commitment scheme is "public coin" if the receiver sends only random strings.

Public-coin statistically binding commitment schemes in two rounds exist under the minimal assumption of the existence of any one-way function [56]. A natural question is whether, given any public-coin 2-round commitment scheme from one-way functions, the receiver can just be replaced by a sample from a high min-entropy source, therefore obtaining a non-interactive commitment scheme relying on the existence of SHELA sources[9]. We show that the answer is in general negative, by providing a variation of the construction of [56] where the binding property breaks down when the first round is sampled from a specific SHELA source. Moreover, parallel repetitions do not help to obtain binding. The construction of [56] can become non-interactive using any SHELA source, however in this last case there is a price to pay in communication complexity since the size of the resulting non-interactive commitment scheme is equal to the size of the SHELA source X.

The real good news come from using our tool: a convSR-source extracted from a SHELA source (without adding any computational assumption). Indeed, in this case we can get a non-interactive statistically binding commitment scheme just by running a parallel repetition of any public-coin 2-round statistically binding commitment scheme. When applied to the scheme of [56], we can get better communication complexity compared to the previously described approach that consists of using a SHELA source directly. Indeed, consider a 2-round statistically binding commitment scheme where the first round of the receiver (in the commitment phase) consists of λ bits, and let us assume that in each high min-entropy honest block of a 2-out-of-ℓ SHELA there are k bits of min-entropy, where $k \gg \lambda$. If $Y = \mathsf{SomeExt}(X) \in \{0,1\}^{m \cdot L}$ for $L = \ell - 1$ and we set $m = \lambda$ (by truncation), then $|Y| = m \cdot L \ll n \cdot \ell = |X|$. Therefore, with the parameters discussed above, if we instantiate the scheme of [56] using X directly, the resulting non-interactive commitment scheme has significantly worse communication complexity than the one built from the convSR-source.

Additional contributions. In the full version of this work [1], we also consider somewhere-extraction from an *online* variant of SHELA sources.

1.2 Related Work

Applications of convSR-sources in pseudorandomness. We would like to point out that the convSR-sources are also very useful in a context different than those already presented. Indeed, convSR-sources are key intermediate objects in several constructions of multi-source and non-malleable randomness extractors for weak sources. A central approach in such constructions is to reduce the task of extracting a uniform string from independent weak sources to that of extracting

[9] We recall that obviously a SHELA source is also a high min-entropy source.

such a string from one or more independent convSR-sources potentially satisfying a few additional properties, sometimes coupled with additional independent weak sources or small uniform seeds.

The connection between multi-source extraction and convSR-sources has been known since they were first defined [67]. convSR-sources have also been used in early constructions of seeded extractors [55].

Barak et al. [2] and Raz [64] showed how to convert two independent weak sources into an convSR-source with few blocks. This reduction was then used directly to obtain 3- and 4-source extractors with constant error. Such an approach has also proved useful in the construction of dispersers [2,3].

To obtain extractors for a constant number of sources with lower error and min-entropy requirement $n^{\Omega(1)}$, Rao [63] transforms independent input sources into several independent *aligned* convSR-sources, i.e., there is at least one position at which all convSR-sources have a uniform block. If the number of blocks in each convSR-source is not too large, then an iterative procedure succeeds in extracting a uniform string from such independent aligned convSR-sources with small error. Li [48] also used a similar approach with aligned convSR-sources to obtain better 3-source extractors.

An important step in many recent constructions of 2- and 3-source extractors [52,50,53,18,51,7] consists in generating convSR-sources with many "good" blocks (i.e., blocks close to uniform) which additionally satisfy a notion of w-wise independence for an appropriate parameter w: Every set of w good blocks is also close to jointly uniformly distributed. convSR-sources are also used in other recent constructions of multi-source extractors [23,22].

The usefulness of convSR-sources extends to more recent notions of randomness extraction. In fact, convSR-sources have been used in the construction of seedless non-malleable extractors [17] for weak sources, which are closely connected to non-malleable codes.

The ubiquity of convSR-sources (generated from weak sources) in extractor constructions provides one more compelling reason for our study of lower bounds for deterministic somewhere-extraction from weak sources.

Finally, we should mention that, because of the close connection between convSR-sources and randomness extraction from general weak sources, several works other than those already mentioned have focused directly on designing randomness extractors for the restricted class of convSR-sources [73,32,31,33,29]. Such extractors are usually called *mergers*.

Deterministic randomness extraction from restricted classes of sources. Our work is also related to the fundamental and well-studied problem of deterministic randomness extraction. Given the impossibility of deterministic extraction from general weak sources, the following natural question arises: *Under which conditions is deterministic randomness extraction possible from imperfect sources of randomness?*

Several works (some even predating the definition of weak sources [20]) have studied this question from various perspectives. Some works have considered deterministic randomness extraction from streams of bits generated i.i.d. with unknown

bias [57,35], or according to a Markov chain [11]. In a parallel line of research, settings where some input bits may be (adversarially or not) fixed, while the remaining ones are random, have also been considered [19,69,8,54,27,39,45,62,24,60]. Other classes of sources considered in the context of deterministic extraction include sources with efficient sampling procedures [68,25] or sampled in small space [44], sources defined over subspaces [38,13,62,26,72,49,14,51], sources determined by zero sets of polynomials [30,47], sources sampled by Turing machines [70] or small circuits [71], and sets of independent weak sources (already discussed in this section). Some works have constructed such extractors for subclasses of Santha-Vazirani sources [5,4], which are known not to admit deterministic extraction in general. We note that Bentov, Gabizon, and Zuckerman [9] studied deterministic randomness extraction from the blockchain of Bitcoin, which has some connections to our model. However, their focus is on standard deterministic extraction, instead of somewhere-extraction. They show that standard deterministic extraction is impossible against an adversary with an unbounded budget, and then study the same problem against a "budget-constrained" adversary.

Although we are not dealing with standard randomness extraction like most of the works above, we present a result of a similar flavor: The restricted (and practically motivated) class of SHELA sources allows for deterministic *somewhere*-extraction with much better parameters than the class of weak sources.

Randomness extraction from adversarial sources. Subsequently to the announcement of our work, the problem of extracting randomness from adversarial sources (of which SHELA sources are an example) has received significant attention.

Chattopadhyay, Goodman, Goyal, and Li [16] study randomness extraction from an adversarial source model similar to SHELA sources. However, there are important distinctions between the two models, which we discuss next. In both cases, a source can be divided into blocks, some of which are independently generated and contain appropriate min-entropy, while other blocks are adversarially controlled. However, in SHELA sources the adversarial block is allowed to depend arbitrarily on all previous blocks (but *not* on subsequent blocks), while in [16] is only allowed to depend on at most d other arbitrary blocks for a small "locality parameter" d. Deterministic randomness extraction turns out to be possible in the adversarial model from [16], while it is impossible in the SHELA model and we instead study deterministic *somewhere*-extraction and its applications. Based on this, the results in these two models are incomparable.

Dodis, Vaikuntanathan, and Wichs [28] study seeded randomness extraction from so-called *extractor-dependent* sources. This adversarial model differs significantly from SHELA sources. At a very high level, a source is sampled by an adversary that is first allowed to query the extractor on different inputs with the same seed, with the condition that the source contains enough min-entropy and other sensible constraints to make the problem non-trivial. Extractor-dependent sources aim to capture scenarios where a random seed may be re-used several times.

1.3 Technical Overview on Deterministic Somewhere-Extraction from SHELA and Weak Sources

Impossibility of deterministic extraction from SHELA sources. We show that if at most a γ-fraction of the ℓ blocks in a SHELA source are honest, where $\gamma \in [0, 1)$ is an *arbitrary* constant, and ℓ is a large enough constant depending on γ, then deterministic randomness extraction is impossible from this class of SHELA sources. Notably, this impossibility result holds even if we allow the honest blocks to be *uniformly distributed*, instead of only requiring them to have enough min-entropy.

This result is obtained by reducing the problem of deterministic extraction from SHELA sources to the problem of deterministic extraction from so-called *resettable* sources, introduced in [9]. In the same work, the latter problem has been shown to be closely related to deterministic extraction from Santha-Vazirani (SV) sources [65], which is widely known to be an impossible task. For more details we refer to [1].

Constructions of somewhere-extractors for SHELA sources. Our constructions of somewhere-extractors for SHELA sources are mainly based on the following trick, which we illustrate for a SHELA source with three blocks B_1, B_2, B_3, two of which are honest. If we applied the naive somewhere-extractor previously discussed with a 2-source extractor, we would obtain a convSR-source with three rows. Recall that one of our main goals is to reduce the total number of blocks in the resulting convSR-source as much as possible due to efficiency concerns. With this in mind, instead of applying the naive somewhere-extractor, we can notice that there are two cases:

- B_3 is honest. Then, B_3 and (B_1, B_2) are two independent weak sources. This means we can extract randomness from the two sources (B_1, B_2) and B_3;
- B_3 is not honest. Then, B_1 and B_2 are honest, and hence are independent weak sources. In this case, we can extract randomness from the two sources B_1 and B_2.

For the sake of this example, let Ext_1 and Ext_2 be two-source extractors, and compute $\mathsf{Ext}_1((B_1, B_2), B_3)$ and $\mathsf{Ext}_2(B_1, B_2)$.[10] The key observation, stemming from the two cases above, is that we are guaranteed that at least one of the two outputs is close to uniformly distributed. As a result, we obtain a convSR-source with two rows instead of three.

As already mentioned, we design explicit somewhere-extractors in two main settings. Our first, simpler, somewhere-extractor can be applied whenever the underlying SHELA source has $t \geq 2$ honest n-bit blocks with min-entropy $k = (1-\gamma)n$ for a small enough constant $\gamma > 0$. The construction is a generalization of

[10] In reality, we are able to use strong seeded extractors (for which we know much better explicit constructions) in place of two-source extractors. This is due to the disproportion in the size of the sources. In fact, the size of one of the sources given to the extractor grows linearly with the total number of blocks.

the reasoning we presented for three blocks above. It proceeds by iteratively using a strong seeded extractor to extract randomness from ever-growing sequences of blocks (using another block as a seed). A bit more precisely, if $X \in \{0,1\}^{n \cdot \ell}$ is a SHELA source and $X = (B_1, B_2, \ldots, B_\ell)$, then for every $i = 2, 3, \ldots, \ell$ we consider

$$B'_i = \mathsf{Ext}_i((B_1, \ldots, B_{i-1}), B_i), \tag{2}$$

where $(B_1, \ldots, B_{i-1})$ acts as the input weak source, B_i acts as the seed, and Ext_i is an appropriate strong seeded extractor. Then, we set $\mathsf{SomeExt}(X) = (B'_2, \ldots, B'_\ell)$. The first problem we run into is that in usual applications of seeded extractors, the seed is uniformly distributed. This is not the case here, since, even if B_i is an honest block, it is only guaranteed to have min-entropy $(1-\gamma)n$. However, it is not hard to show, using the strongness of the extractor, that using a source with high min-entropy as the seed is sufficient. Another issue we encounter is that we are reutilizing many SHELA blocks when computing output blocks via (2). This appears to be at odds with the requirement that good output blocks should be close (in statistical distance) to independent and uniformly distributed. A careful conditioning argument, again exploiting the strongness of the extractor, shows that independence and uniformity are actually attained with small error. In fact, whenever B_i is honest and there is an honest block in $(B_1, \ldots, B_{i-1})$, we succeed in generating (with small error) a new good block of the output convSR-source. Instantiating this construction with the nearly-optimal GUV strong seeded extractor [43] and assuming the SHELA source $X \in \{0,1\}^{n \cdot \ell}$ has t honest blocks, we output a distribution $Y \in \{0,1\}^{m \cdot L}$ that is $(t \cdot 2^{-\Omega(n)})$-close to a T-out-of-L convSR-source with $m = \Omega(n)$. Moreover, from the discussion above it follows that $L = \ell - 1$ and $T = t - 1$.

In the second setting, we consider deterministic somewhere-extractors for SHELA sources with honest blocks having *arbitrary* constant min-entropy rate k/n. In other words, we allow the min-entropy requirement k of honest blocks to satisfy $k = \delta n$ for arbitrarily small $\delta > 0$. Notably, in this significantly harder setting we are able to obtain essentially the same parameters as the somewhere-extractor for the high min-entropy setting detailed above. In fact, all parameters remain unchanged, except that now we cannot guarantee that $L = \ell - 1$, and instead have the (still highly desireable) relationship $L = O(\ell)$. The main barrier towards making the previous construction work in this setting is that if honest blocks do not have high min-entropy, they can no longer be used as seeds for strong seeded extractors. This issue is surpassed by using the somewhere-condenser for weak sources from [64,2]. Intuitively, a somewhere-condenser is to a randomness condenser as a deterministic somewhere-extractor is to an extractor. On input a weak source with low min-entropy, the somewhere-condenser $\mathsf{SomeCond}$ outputs (with small error) a constant number of (sufficiently long) blocks with the guarantee that at least one block has very high min-entropy rate. Because the focus is not on extraction of *perfect* randomness, somewhere-condensers for weak sources are allowed to have much better parameters than somewhere-extractors for the same class of sources. We modify the construction for honest blocks with high min-entropy above by adding a first step of somewhere-condensation for

each block of the input SHELA source. We show that our somewhere-extractors designed for SHELA sources can also be applied to *online* SHELA sources as is to extract convSR-sources (for full definitions and discussion please see [1]).

Lower bounds for deterministic somewhere-extraction from weak sources. We consider the natural problem of understanding the performance of somewhere-extractors for weak sources, and derive a set of lower bounds which show that, particularly for parameters relevant to cryptographic applications, *every* somewhere-extractor (regardless of efficiency) for weak sources must have significantly worse parameters than the somewhere-extractors we obtain for the class of SHELA sources. As previously discussed, these negative results for weak sources are strong enough that they preclude the use of convSR-sources generated from weak sources in efficient cryptographic protocols.

Suppose $\mathsf{SomeExt} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ is a somewhere-extractor for $(\tilde{n}, k)$-sources[11]. We begin by noting that a simple reasoning analogous to the proof of impossibility of deterministic extraction from weak sources immediately shows that $L = \Omega(\tilde{n} - k)$. Our first non-trivial lower bound is obtained by relating a somewhere-extractor to a *disperser* (for weak sources). Roughly speaking, a disperser is a fundamental pseudorandom object that transforms a weak source and a short uniform seed into an output distribution that hits every appropriately large subset of the output space with non-zero probability. Optimal seed length lower bounds are known for dispersers [61]. We show that if $\mathsf{SomeExt} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ is a somewhere-extractor for $(\tilde{n}, k)$-sources with error ε, then the function $G : \{0,1\}^{\tilde{n}} \times [L] \to \{0,1\}^m$ given by

$$G(x, i) = \mathsf{SomeExt}(X)_i$$

is a disperser with seed length $\log L$ and error ε. This immediately leads to a lower bound on the number L of output blocks of SomeExt (excluding a minor technicality that does not affect the quality of the lower bound),

$$L = \Omega\left(\frac{\tilde{n} - k}{\max(\varepsilon, 2^{-m})}\right). \tag{3}$$

This means, as discussed in more detail in Section 5, weak sources behave exponentially worse than comparable SHELA sources for somewhere-extraction in the linear output block length regime.

Note that the two lower bounds in the previous paragraph do not give anything when $k \approx \tilde{n}$ and m is small. This naturally leads us to consider lower bounds for L in an extreme 1-bit block setting with $k = \tilde{n} - 1$ and $m = 1$. Although we do not obtain a lower bound for extraction of convSR-sources in this extreme regime, we are able to prove a non-trivial lower bound that scales with the error for the harder, but related, task of extracting an SR-source from a weak source (*not* a convex combination of SR-sources as before). Note that, in particular, the naive

[11] The set of $(\tilde{n}, k)$-sources consists of all weak sources over $\{0,1\}^{\tilde{n}}$ with min-entropy at least k. We use $\tilde{n}$ to avoid confusion with the block length of SHELA sources.

somewhere-extractor obtained by enumerating the seed of a strong extractor satisfies this property. To be precise, we show that in this setting we must have

$$L = \Omega\left(\log\left(\frac{1}{\max(\varepsilon, 2^{-k})}\right)\right). \tag{4}$$

The lower bound in (4) is obtained by an adaptive version of the basic argument for the impossibility of deterministic extraction from weak sources. Given a candidate function $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^L$, our goal is to show the existence of a weak source $X^\star$ with enough min-entropy such that *every* bit $F(X^\star)_i$ is sufficiently biased. We begin by setting $X_0^\star$ to be uniformly distributed over $\{0,1\}^{\tilde{n}}$, and analyze its performance w.r.t. F. If $F_i(X_0^\star)$ is the first bit close to uniform, we remove an appropriate set of elements from the support of $X_0^\star$ to obtain $X_1^\star$ such that $F_i(X^\star)$ biased enough. Then, we repeat the reasoning with the new source $X_1^\star$ and so on, until every bit is biased[12]. Then, L must be large enough to ensure the outcome $X^\star$ of this process has too small support (and hence does not satisfy the min-entropy requirement of F), which yields the lower bound.

With these bounds in mind, it is natural to consider whether arguments that yield lower bounds of this type on the seed length of extractors, more precisely the granularity argument of Nisan and Zuckerman [58, Theorem 3] and the techniques due to Radhakrishnan and Ta-Shma [61, Section 2.2], could be extended to the setting of somewhere-extraction. Unfortunately, such arguments crucially rely on the ability of picking a seed at random: There, one is only worried about showing that the bias is large enough *on average*, while we must show that the bias is large enough *for every choice of the seed*[13].

1.4 Technical Overview on Non-Interactive Proof Systems and Commitments from Public SHELA Sources

Non-interactive (publicly verifiable) witness indistinguishable proof system. We will now describe how to construct a non-interactive (and therefore publicly verifiable) Witness Indistinguishable (WI) proof system Π_{pv} from a public SHELA source X and starting with the existence of a public-coin ZAP Π. Π_{pv} works as follows: The prover of Π_{pv} receives X and runs the somewhere-extractor $\mathsf{SomeExt}$ on X to obtain $(R_1, \ldots, R_L)$. Then, the prover on input the witness w for the statement x computes a second-round π_i from Π using R_i for $i = 1, \ldots, L$. The verifier of Π_{pv}, having access to X, also computes $(R_1, \ldots, R_L) = \mathsf{SomeExt}(X)$, and accepts the proof only if all pairs (R_i, π_i) are accepting by the verifier of Π w.r.t. the statement x. Observe that WI of Π is preserved under parallel composition and holds even when the first round of Π is chosen by a malicious verifier. Therefore, Π_{pv} also enjoys the WI property. The soundness of Π_{pv} is based on the observation that T blocks of $(R_1, \ldots, R_L)$

[12] When biasing the next coordinates, we have to be careful not to 'spoil' biases of previous coordinates. This results in the log factor in the bound.

[13] By *seed* we mean i in $F_i(X^\star)$.

are negligibly close to a uniform distribution over $\{0,1\}^m$. Denote them by $R_{I_1}, \ldots, R_{I_T}$. Then, the soundness of Π ensures that a malicious prover could not cheat when the second round of Π is computed w.r.t. $R_{I_1}, \ldots, R_{I_T}$.

As a result, using known constructions of public-coin ZAPs, we are able to construct a non-interactive WI proof system from trapdoor permutations that requires as a set-up a SHELA source only. Notice that a SHELA source is a CRS that can be corrupted (in a natural, structured manner) by an unbounded adversary. Still, we assume that the adversarial verifier can run only in polynomial time to distinguish the witness, even though he does not have such restriction when affecting the sample from the public SHELA source. Previous constructions of non-interactive WI proof systems either require a common random string as set-up, or were based on specific number-theoretic hardness assumptions in bilinear groups [42,37], or on indistinguishability obfuscation and one-way permutations [10].

From another point of view, one can see our result as a Non-Interactive (NI) WI proof system where the soundness and the WI property hold even when the set-up phase is partially generated by the adversary. We note that the work of [6] investigates if soundness and WI of a NIWI proof system hold even when the adversary takes complete control of the set-up phase. They achieve a positive result relying on some specific number-theoretic assumption in bilinear groups. Instead, our NIWI proof system can be instantiated from trapdoor permutations and the adversary has only a partial control over the set-up.

Notice that [15] studies cryptographic protocols with simulatable security by considering a simulatable CRS drawn from a high min-entropy distribution. In this work we do not assume that public sources of randomness are simulatable and we do not investigate simulatable security. Our CRS is not a generic min-entropy string but instead corresponds to a structured min-entropy source that is partially controlled by an unbounded adversary.

Given the above construction of a non-interactive WI proof system Π_{pv}, one could argue that a convSA-source suffices for constructing Π_{pv}. Recall that a convSA-source is a convex combination of T-out-of-L SA-sources, which consist of L blocks, T of which are independent and $\frac{1}{\mathrm{poly}(n)}$-close to uniform in statistical distance, where n is some relevant security parameter. This is because the soundness of the protocol can be amplified by using the T "good" blocks, which correspond to independent parallel repetitions of the underlying protocol Π.

In order to adequately compare the performance of the protocol under convSA-extraction from weak sources and convSR-extraction from SHELA sources, we compare a t-out-of-ℓ SHELA source $X \in \{0,1\}^{n \cdot \ell}$ with honest blocks having linear min-entropy k' with an arbitrary weak $(\tilde{n} = n \cdot \ell, k = k' \cdot t)$-source $\tilde{X}$. We are able to show that convSR-sources extracted from X are much better suited for applications than convSA-sources generated from $\tilde{X}$ in two aspects:

1. **Efficiency:** The efficiency of Π_{pv} depends on L. It is not hard to see that every convSA-source extractor for weak sources $\tilde{X}$ must have $\Omega(\tilde{n}) = \Omega(n \cdot \ell)$ total output blocks (even if we only require constant error). On the other hand, we can extract convSR-sources from X with only $O(\ell)$ blocks.

2. **Security:** Let us assume that Π requires a first round of $m = \Omega(k')$ bits. Then, we show that every *efficient, low-error* convSA-source extractor for weak sources outputs at most $T = O(k/m) = O(k' \cdot t/m)$ *good* blocks of length m. As a result, if t is constant, it follows that such an extractor only outputs $T = O(1)$ good blocks. This is not enough to successfully amplify the soundness of the protocol. Finally, we note that if we build our Π_{pv} starting from a convSR-source extracted from a t-out-of-ℓ SHELA source with constant t, the analysis of soundness described in this subsection holds, and therefore Π_{pv} is sound.

Improving the efficiency of [66]. We note that the work of [66] constructs a publicly verifiable proof system from any blockchain under some assumptions on the min-entropy of honestly generated blocks. Notably, under the same assumptions the blockchain can be used to implement also a SHELA source. In [66], the authors construct a publicly verifiable proof system by applying the naive somewhere-extractor $\mathsf{NaiveSomeExt}$ (that we discussed earlier) to extract a convSR-source from the blockchain. Therefore our somewhere-extractor $\mathsf{SomeExt}$ (instead of $\mathsf{NaiveSomeExt}$) could be used in their work to immediately improve the efficiency of their proof system. More details are provided in [1].

Non-interactive statistically binding commitments. We introduce now a construction of non-interactive statistically binding commitments from a public SHELA source relying on one-way functions. This is achieved by making use of any two-round public-coin commitment scheme Π_{com} from one-way functions.

First of all, we remark that one can not simply replace the first round of Π_{com} with a sample from a source with linear min-entropy (say, min-entropy $0.5n$). Indeed, start from Π_{com} and consider a scheme Π'_{com} where: a) the random string played as first round of Π_{com} must be twice in length, and b) the sender ignores the first half of the first round and continues as in Π_{com} using the second half. It is straightforward to see that Π'_{com} is a 2-round public-coin statistically binding commitment scheme from any one-way functions. If we replace the first round of Π'_{com} with the output of a high min-entropy source we might have that the entire min-entropy is in the first half of the first round and is therefore wasted completely. The malicious sender could therefore violate binding since it would end up running Π_{com} on input a first round with zero min-entropy! Obviously, in this case parallel repetition does not help.

We now proceed to describe how our scheme Π_{compv} works starting with any 2-round public-coin statistically binding commitment scheme (including the above Π'_{com}). Moreover, Π_{compv} can be run with efficient parameters because of the use of $\mathsf{SomeExt}$.

Our commitment scheme Π_{compv} works as follows: First, the sender runs the somewhere-extractor $\mathsf{SomeExt}$ on the public SHELA source X, obtaining $\mathsf{SomeExt}(X) = (R_1, \ldots, R_L)$. Then, the sender on input the message m and R_i (used as the receiver's first round) computes a commitment com_i and the opening information dec_i using the sender of Π_{com}, for $i = 1, \ldots, L$. In the opening phase, the receiver on input $\mathsf{dec}_1, \ldots, \mathsf{dec}_L$ having access to X computes

$(R_1, \ldots, R_L) = \mathsf{SomeExt}(X)$, and outputs the message m only if it holds that for all $i = 1, \ldots, L$ the message committed in com_i is m. Hiding of our scheme holds from the observation that hiding is preserved under parallel composition and when the first round of Π_{com} is chosen by a malicious receiver. The binding of Π_{compv} is based on the observation that at least T blocks $R_{I_1}, \ldots, R_{I_T}$ are negligibly close to a uniform distribution over $\{0,1\}^m$. This implies that there are at least T commitments computed w.r.t. a good block R_{I_j} that is statistically close to a first round sent by a receiver of Π_{com}. Therefore, from the statistically binding of Π_{com} it follows that a malicious sender could not cheat when the commitment is computed w.r.t. $R_{I_1}, \ldots, R_{I_T}$.

1.5 Open Questions

We present some interesting directions for future research:

- Prove (or disprove) Conjecture 12.
- Given any SHELA or convSR source, we can define its *rate* as number of good[14] blocks divided by total number of blocks. Our constructions from Section 4 transform SHELA sources with rate t/ℓ into convSR-sources with rate $\frac{t-1}{\ell-1} \leq \frac{t}{\ell}$. We conjecture that the rate of the output convSR-source cannot be larger than t/ℓ.
- Find good bounds on the number of output blocks of convSA-source extractors for weak sources.

1.6 Organization of the Paper

We introduce relevant notation and definitions in Section 2. SHELA sources are defined in Section 3, and deterministic somewhere-extractors are presented in Section 4. Lower bounds for somewhere-extraction are studied in Section 5, and the limits of SA-source extraction are considered in Section 6. Detailed arguments, along with standard definitions and lemmas, have been deferred to the full version [1].

2 Preliminaries and Definitions

2.1 Notation

Sets are usually denoted by calligraphic letters such as $\mathcal{S}$ and $\mathcal{I}$. Random variables are usually denoted by uppercase letters such as X, Y, and Z. We may identify a random variable X with its distribution. The support of a distribution X is denoted by $\mathsf{supp}(X)$. We denote the uniform distribution over $\{0,1\}^m$ by U_m. We may write $X \sim Y$ to denote that X has the same distribution as Y. All logarithms log are taken to base 2. The Shannon entropy of a distribution X is denoted by $H(X)$, and we denote the binary entropy function by h. The notation $\mathrm{poly}(n)$ denotes an arbitrary polynomial in n. We denote a negligible function of a parameter n by $\mathsf{negl}(n)$.

[14] For a SHELA source, a good blocks correspond to honest blocks, while they correspond to jointly uniform blocks in convSR-sources.

2.2 Somewhere-Random Sources and Somewhere-Extractors

In this section, we define SR- and convSR-sources, along with the notion of a deterministic somewhere-extractor and a basic result.

Definition 1 (Somewhere-random source) *A distribution $X = (X_1, \ldots, X_L)$ over $\{0,1\}^{m \cdot L}$ is said to be a (T, L, m)-somewhere-random source, SR-source in short, if there exist indices $i_1 < i_2 < \cdots < i_T$ such that the tuple $(X_{i_1}, X_{i_2}, \ldots, X_{i_T})$ is uniformly distributed over $\{0,1\}^{m \cdot T}$. We denote the set of all (T, L, m)-somewhere-random sources by $\mathsf{SR}_{T,L,m}$, and the set of all convex combinations of sources in $\mathsf{SR}_{T,L,m}$ by $\mathsf{convSR}_{T,L,m}$.*

Definition 2 (Somewhere-extractor) *Given a set of sources $\mathcal{F}$ over $\{0,1\}^{\tilde{n}}$, a function $\mathsf{SomeExt} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ is said to be a (T, L, ε)-somewhere-extractor for $\mathcal{F}$ if for every $X \in \mathcal{F}$ there exists $Y \in \mathsf{convSR}_{T,L,m}$ such that*

$$\mathsf{SomeExt}(X) \approx_\varepsilon Y.$$

A simple construction shows that strong (k, ε)-extractors imply the existence of deterministic somewhere-extractors for the class of general (n, k)-sources with the same error ε.

Lemma 3 *Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a strong (k, ε)-extractor, and set $\{0,1\}^d = \{s_1, s_2, \ldots, s_{2^d}\}$. Given $x \in \{0,1\}^n$, define $\mathsf{SomeExt}(x) : \{0,1\}^n \to \{0,1\}^{m \cdot 2^d}$ as*

$$\mathsf{SomeExt}(x) = (\mathsf{Ext}(x, s_1), \mathsf{Ext}(x, s_2), \ldots, \mathsf{Ext}(x, s_{2^d})).$$

Then, $\mathsf{SomeExt}$ is a $(1, 2^d, \varepsilon)$-somewhere-extractor for the class of (n, k)-sources.

The construction from Lemma 3 actually guarantees that a very large fraction of blocks of $Y = \mathsf{SomeExt}(X)$ will be close to uniform over $\{0,1\}^m$, provided X is an (n, k)-source. However, there is no guarantee that any pair of blocks (Y_{i_1}, Y_{i_2}) will be close to uniformly distributed over $\{0,1\}^{2m}$, as we cannot ensure that such blocks are close to being independent. Therefore, we only know that Y is ε-close to a $(1, 2^d, m)$-somewhere-random source.

2.3 Somewhere-Condensers

In this section, we introduce somewhere-condensers and related notions.

Definition 4 (Somewhere-entropic source) *A distribution $X = (X_1, \ldots, X_L)$ over $\{0,1\}^{m \cdot L}$ is said to be a (T, L, m, k)-somewhere-entropic source if there exist indices $i_1 < i_2 < \cdots < i_T$ such that the random variables $X_{i_1}, X_{i_2}, \ldots, X_{i_T}$ are independently distributed and satisfy $\mathbf{H}_\infty(X_{i_j}) \geq k$ for all j. We denote the set of all (T, L, n, k)-somewhere-entropic sources by $\mathsf{SE}_{T,L,n,k}$, and the set of all convex combinations of sources in $\mathsf{SE}_{T,L,n,k}$ by $\mathsf{convSE}_{T,L,n,k}$.*

Definition 5 (Somewhere-condenser) *A function* $\mathsf{SomeCond} : \{0,1\}^n \to \{0,1\}^{m \cdot \ell}$ *is said to be a* (k, k', L, ε)-somewhere condenser *if for every* (n,k)-*source* X *there exists* $Y \in \mathsf{convSE}_{1,L,m,k'}$ *such that*

$$\mathsf{SomeCond}(X) \approx_\varepsilon Y.$$

There exist explicit constructions of somewhere-condensers with a constant number of output blocks, linear output block length, and exponentially small error for arbitrarily low linear min-entropy.

Lemma 6 ([64]) *For all constants* $\delta, \delta' > 0$ *there exist constants* $b, \beta, \rho > 0$ *such that for large enough* n *there exists an explicit* (k, k', b, ε)-*somewhere condenser* $\mathsf{SomeCond} : \{0,1\}^n \to \{0,1\}^{m \cdot b}$ *with* $k = \delta n$, $m = \beta n$, $k' = (1-\delta')m$, *and* $\varepsilon = 2^{-\rho m}$.

Remark 1. The version of Lemma 6 presented in [64] is specialized for $\delta' = \delta$. However, inspection of [64, Lemmas 4.2 and 4.3] shows that the construction works for any constant $\delta' > 0$, as long as we allow the constants ℓ, β, ρ to depend simultaneously on δ and δ'. This observation is similar to the remark in [2] after Theorem 5.2.

3 SHELA Sources

In this section, we give a formal definition of Somewhere Honest Entropic Look Ahead (SHELA) sources, and present explicit constructions of somewhere-extractors with good parameters for this class of sources.

Definition 7 (SHELA source) *A distribution* $X \in \{0,1\}^{n \cdot \ell}$ *is said to be an* (n,k,t,ℓ)-SHELA source *if there exist random variables* $1 \le I_1 < I_2 < \cdots < I_t \le \ell$ *with arbitrary joint distribution,* t *independent* (n,k)-*sources* $Z_1, Z_2, \ldots, Z_t$, *and a (possibly randomized) adversary* $\mathcal{A}$ *such that* X *is generated as follows:*

1. *Sample* $(i_1, i_2, \ldots, i_t) \leftarrow (I_1, I_2, \ldots, I_t)$;
2. *For each* $j \in [t]$, *set* $B_{i_j} \leftarrow Z_j$;
3. *For each* $i \in [\ell] \setminus \{i_1, \ldots, i_t\}$, $\mathcal{A}$ *sets* $B_i = \mathcal{A}(B_1, \ldots, B_{i-1}, i_1, \ldots, i_t)$;
4. *Set* $X = (B_1, B_2, \ldots, B_\ell)$.

We denote the set of all such SHELA sources by $\mathsf{SHELA}_{n,k,t,\ell}$.

A precise definition of online SHELA sources discussed in Section 1, along with associated notions and results on deterministic somewhere-extraction, can be found in [1].

4 Deterministic Somewhere-Extractors for SHELA Sources

In this section, we construct deterministic somewhere-extractors for regular SHELA sources.

4.1 Honest Blocks with High Min-Entropy

In this section, we consider the case where each honest block in a SHELA source has min-entropy $(1-\gamma)n$ for some sufficiently small constant $\beta > 0$. The following result states that an explicit somewhere-extractor with exponentially small error and linear output block length exists for such SHELA sources. Notably, it is also the case that if the number of honest input blocks is t and the total number of input blocks is ℓ, then the number of uniform output blocks is $T = t - 1$ and the number of total output blocks is $L = \ell - 1$.

Theorem 8 *There exists a small enough constant $\gamma > 0$ such that for n large enough and $2 \leq t \leq \ell \leq \mathrm{poly}(n)$ there exists an explicit $(t-1, \ell-1, \varepsilon')$-somewhere extractor $\mathsf{SomeExt} : \{0,1\}^{n\cdot\ell} \to \{0,1\}^{m\cdot(\ell-1)}$ for $\mathsf{SHELA}_{n,k',t,\ell}$ with $k' = (1-\gamma)n$, $m = \frac{(1-7\gamma)n}{3}$, and $\varepsilon' = 2(t-1)\cdot 2^{-\gamma n}$.*

The construction we use to prove Theorem 8 makes use of the following objects: For $i \in \{2, \ldots, \ell\}$, let $\mathsf{Ext}_i : \{0,1\}^{n\cdot(i-1)} \times \{0,1\}^n \to \{0,1\}^m$ be an average-case strong seeded (k, ε)-extractor with $k = 2k'/3$, $k' = (1-\gamma)n$, $m = \frac{(1-7\gamma)n}{3}$ and $\varepsilon = 2^{-2\gamma n}$ for a small enough constant $\gamma > 0$. These can be obtained by using the explicit GUV extractor [43] with appropriate parameters. The instantiation is detailed in [1]. We are now ready to describe our construction of the somewhere-extractor $\mathsf{SomeExt} : \{0,1\}^{n\cdot\ell} \to \{0,1\}^{m\cdot(\ell-1)}$ for $X \in \mathsf{SHELA}_{n,k,t,\ell}$. First, write $X = (B_1, B_2, \ldots, B_\ell)$. Then, the output $\mathsf{SomeExt}(X)$ can be written as $\mathsf{SomeExt}(X) = (B'_2, B'_3, \ldots, B'_\ell)$, where each B'_i is obtained as

$$B'_i = \mathsf{Ext}_i((B_1, B_2, \ldots, B_{i-1}), B_i) \in \{0,1\}^m. \tag{5}$$

4.2 Honest Blocks with Low Linear Min-Entropy

In this section, we construct somewhere-extractors for SHELA sources that have honest blocks with min-entropy δn for some arbitrarily small constant $\delta > 0$. We show that there is an explicit somewhere-extractor for such SHELA sources with exponentially small error and linear output block length. M oreover, if the number of input honest and total blocks are t and ℓ, respectively, then the number of output uniform and total blocks are $T = t - 1$ and $L = O(\ell)$, respectively.

Theorem 9 *For every constant $\delta > 0$ there exist constants $a_1, a_2, a_3 > 0$ such that for n large enough and all $2 \leq t \leq \ell \leq \mathrm{poly}(n)$ there exists an explicit (T, L, ε')-somewhere extractor $\mathsf{SomeExt} : \{0,1\}^{n\cdot\ell} \to \{0,1\}^{m\cdot L}$ for $\mathsf{SHELA}_{n,k',t,\ell}$ with $k' = \delta n$, $m = a_1 \cdot n$, $\varepsilon' = 2(t-1)2^{-a_2\cdot n}$, $T = t - 1$, and $L = a_3 \cdot \ell$.*

We now turn to a precise description of our construction. Fix a constant $\delta \in (0,1)$ and consider the $(\delta n, (1-\gamma)n', b, 2^{-\rho n'})$-somewhere-condenser $\mathsf{SomeCond} : \{0,1\}^n \to \{0,1\}^{b\cdot n'}$ from Lemma 6, where $\gamma > 0$ is a small constant to be determined, $n' \geq \beta n$, and b, β, and ρ depend only on δ and γ. For each $i = 2, \ldots, \ell$, consider also the average-case strong (k, ε)-extractor

$$\mathsf{Ext}_i : \{0,1\}^{b\cdot n'(i-1)} \times \{0,1\}^{n'} \to \{0,1\}^m$$

with $\varepsilon = 2^{-2\gamma n'}$, $k = \frac{2(1-3\gamma)n'}{3}$, and $m = \frac{(1-3\gamma)n'}{3}$. These extractors can be instantiated using the strong GUV extractor [43] with appropriate parameters.

We are now ready to define $\mathsf{SomeExt}(X)$ for $X = (B_1, \ldots, B_\ell) \in \mathsf{SHELA}_{n,k',t,\ell}$. We write

$$\mathsf{SomeCond}(B_i) = (B_{i1}, \ldots, B_{ib}) \in \{0,1\}^{n' \cdot b}.$$

Then, we have

$$\mathsf{SomeExt}(X) = (B'_{ij})_{i \in [\ell], j \in [b]} \in \{0,1\}^{m \cdot L}$$

for B'_{ij} defined as

$$B'_{ij} = \mathsf{Ext}_i((B_{i'j'})_{i'<i, j' \in [b]}, B_{ij}) \in \{0,1\}^m. \tag{6}$$

5 Lower Bounds for Deterministic Somewhere-Extraction from Weak Sources

In this section, we study lower bounds for somewhere-extractors that work for the general class of weak $(\tilde{n}, k)$-sources (we use $\tilde{n}$ to avoid confusion with the block length n of a SHELA source). Here, we are mostly interested in lower bounds on the number of output blocks generated by such somewhere-extractors with respect to the length $\tilde{n}$ of a source, the length m of an output block, and the error ε of the somewhere-extractor.

The only known construction of a somewhere-extractor for general $(\tilde{n}, k)$-sources described in Lemma 3 requires 2^d blocks, where d is the seed length of the underlying strong extractor/non-malleable extractor. As stated in [1], it holds that $d \geq \log(\tilde{n} - k) + 2\log(1/\varepsilon) + O(1)$ for every extractor, and so the somewhere-random source output by the somewhere-extractor from Lemma 3 has

$$L = \Omega\left(\frac{\tilde{n} - k}{\varepsilon^2}\right)$$

blocks. We remark that a probabilistic argument with a random function yields somewhere-extraction with the same number of output blocks.

The discussion in the previous paragraph leads to the following natural questions: *Is it possible to do better than Lemma 3 for $(\tilde{n}, k)$-sources? In particular, is it possible to obtain a number of output blocks comparable to that obtained from SHELA sources?*

We present some results that aim to answer this question in several parameter regimes. The first result comes from the observation that the basic argument for impossibility of deterministic extraction yields a non-trivial lower bound on the number of output blocks whenever the min-entropy requirement k is not very large.

Theorem 10 *Suppose $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ is a $(1, L, \varepsilon)$-somewhere extractor for $(\tilde{n}, k)$-sources with $\varepsilon \leq 1 - 2^{-c}$ for some $1 \leq c \leq m$ (i.e., ε is not trivial). Then, it holds that*

$$L \geq \frac{\tilde{n} - k}{c}.$$

The lower bound from Theorem 10 is already enough to yield a separation between somewhere-extraction of SHELA and comparable $(\tilde{n}, k)$-sources whenever the min-entropy requirement k is not extremely large. Consider a SHELA source with constant entropy rate and ℓ blocks, each of length $n = \tilde{n}/\ell$ (so that the total length of the source is $\tilde{n}$). The constructions from Theorems 8 and 9 applied to the SHELA source lead to convSR-sources with $L = O(\ell)$ blocks with small error and large output block length if honest blocks have some constant entropy rate. In particular, L *does not depend directly on the input block length* n. On the other hand, the lower bound from Theorem 10 forces that $L = \Omega(\tilde{n} - k) = \Omega(n \cdot \ell)$ for convSR-sources extracted from $(\tilde{n}, k)$-sources, even with error $\varepsilon = 1/2$ (assuming $k/\tilde{n}$ is constant).

The second result is a disperser-based lower bound on the number of output blocks L. This bound is considerably stronger than the one in Theorem 10 whenever the output block length m is not very small and the error ε is small.

Theorem 11 *Suppose* $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ *is a* $(1, L, \varepsilon)$*-somewhere extractor for* $(\tilde{n}, k)$*-sources with* $\varepsilon \leq 1/2$ *and* $L \leq \frac{(1-\max(\varepsilon, 2^{-m}))2^m}{2}$*. Then, it holds that*

$$L = \Omega\left(\frac{\tilde{n} - k}{\max(\varepsilon, 2^{-m})}\right).$$

Referring again to the comparison between SHELA and weak $(\tilde{n}, k)$-sources above, if we want to extract a 1-out-of-L convSR-source with block length $\Omega(n)$ from the weak source with error $2^{-\Omega(n)}$, as is possible for the relevant SHELA source, then Theorem 11 forces that $L = \tilde{n} \cdot 2^{\Omega(n)} = \ell \cdot n2^{\Omega(n)}$. On the other hand, the convSR-source we extract from the relevant t-out-of-ℓ SHELA source only has $O(\ell)$ blocks.

While Theorems 10 and 11 imply strong separation between SHELA and weak sources for any conceivable application, they do not yield useful lower bounds for some regimes of parameters. For example, in the easiest setting for somewhere-extraction, when the min-entropy requirement k is very large (say, $k = \tilde{n} - 1$) and the output block length is very small (say, $m = 1$), both theorems only give a trivial $\Omega(1)$ lower bound on L, *even when* ε *is exponentially small in* $\tilde{n}$. On the other hand, the number of output blocks in the somewhere-extractor obtained from Lemma 3 instantiated with an optimal strong extractor scales as $1/\varepsilon^2$ even when $k = \tilde{n} - 1$ and $m = 1$. We believe it is not possible to improve significantly on the basic construction from Lemma 3, and so we put forth the following conjecture.

Conjecture 12 *Suppose* $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ *is a* (T, L, ε)*-somewhere extractor for* $(\tilde{n}, k)$*-sources. Then, there exists a constant* $c > 0$ *such that if* $\varepsilon \leq c$*, we have*

$$L = \Omega\left(\frac{\tilde{n} - k}{\varepsilon^2}\right). \tag{7}$$

We do not prove Conjecture 12 and leave it as an interesting open problem. Nevertheless, we prove a weaker lower bound on L in a similar spirit to (7) under

a stronger property than somewhere-extraction, which is still satisfied by the construction from Lemma 3. This result can be regarded both as a first step towards a full proof of Conjecture 12, and a non-trivial lower bound on L (under this stronger property) that scales with ε and holds even when k is large and m is small. Before we state our result, we must first define the alternative notion of somewhere-extraction. Observe that the construction of F from Lemma 3 actually ensures that for every $(\tilde{n}, k)$-source X it holds that $F(X)$ is ε-close to an element of $\mathsf{SR}_{T,L,m}$, instead of only a convex combination of such elements. We call a function that satisfies this for all $(\tilde{n}, k)$-sources a *strong* (T, L, ε, k)-somewhere extractor.

We may think of a strong $(1, L, \varepsilon, k)$-somewhere-extractor $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^L$ as a family of L functions $F_1, \ldots, F_L$ such that for every $(\tilde{n}, k)$-source X, there is F_i such that $F_i(X) \approx_\varepsilon U_1$. Therefore, in order to show such a function F is not a strong somewhere-extractor, we must show the existence of an $(\tilde{n}, k)$-source X that is "bad" for all F_i's, in the sense that $F_i(X) \not\approx_\varepsilon U_1$ for every i. As previously discussed, existing techniques used in proving lower bounds for extractors cannot be applied to obtain similar lower bounds for strong somewhere-extractors. We use a fundamentally different technique to prove the following lower bound on L for strong somewhere-extractors.

Theorem 13 *Suppose $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ is a strong $(1, L, \varepsilon, k)$-somewhere extractor for $k \leq \tilde{n} - 1$. Then, there exists an absolute constant $c > 0$ such that if $\varepsilon < c$, we have*

$$L = \Omega\left(\log\left(\frac{1}{\max(\varepsilon, 2^{-k})}\right)\right). \tag{8}$$

6 Bounds for Somewhere-Amplifiable-Source Extraction from Weak Sources

The lower bounds obtained in Section 5 show that convSR-sources extracted from SHELA sources are much better (in terms of number of blocks with respect to desired extraction error) than convSR-sources extracted from weak sources. This has direct consequences in the time complexity blowup incurred when using convSR-sources in several applications, as discussed in Section 1. However, as discussed in that same section, it is possible in some scenarios to use a weaker object than convSR-sources, which we call *somewhere-amplifiable sources*, where the good independent blocks are not required to be exactly uniformly distributed. A precise definition follows.

Definition 14 (Somewhere-amplifiable source) *We say $Y = (Y_1, \ldots, Y_L)$ over $\{0,1\}^{m \cdot L}$ is a (T, L, ε)*-somewhere-amplifiable source *if there exist distinct indices $i_1, \ldots, i_T$ such that $Y_{i_1}, \ldots, Y_{i_T}$ are independent and $Y_{i_j} \approx_\varepsilon U_m$ for all $j = 1, \ldots, T$. The set of all such SA sources is denoted by $\mathsf{SA}_{T,L,\varepsilon}$, and the set of all convex combinations of sources in $\mathsf{SA}_{T,L,\varepsilon}$ is denoted by $\mathsf{convSA}_{T,L,\varepsilon}$.*

Since the error required from each good block in a convSA-source is not that small (in fact, it can even be constant), one may hope to transform weak sources

into convSA-sources whose number of blocks is much closer to that of convSR-sources obtained from SHELA sources, and which have blocks long enough to be used in the applications already discussed in Section 1 and later in Section 7. To this end, we define *somewhere-amplifiable source extractors* (convSA-source extractors).

Definition 15 (Somewhere-amplifiable source extractor) *A function* $\mathsf{SomeExt} : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ *is said to be a* $(T, L, k, \varepsilon_1, \varepsilon_2)$-somewhere-amplifiable extractor *if for every* $(\tilde{n}, k)$*-source* X *there exists* $Y \in \mathsf{convSA}_{T,L,\varepsilon_2}$ *such that*

$$\mathsf{SomeExt}(X) \approx_{\varepsilon_1} Y.$$

We begin by noting that Theorem 10 also applies to convSA-source extractors for weak sources. This shows that every such extractor (even with constant error) must have $L = \Omega(\tilde{n} - k)$. As discussed in Section 1, this already provides an efficiency separation between convSA-source extraction from weak sources and convSR-source extraction from SHELA sources.

The main result we prove in this section is a different type of separation between convSA-source extraction from weak sources and convSR-source extraction from SHELA sources. Roughly speaking, we show that if we want to extract a convSA-source with many good blocks (necessary to obtain good final error) from an $(\tilde{n}, k)$-source, then either the resulting convSA-source has too many blocks to allow for efficient construction of the publicly verifiable protocols, or the length of each block is very small, and so they may not be usable in some protocols. This is discussed for the particular case of our publicly verifiable proof system in Section 1.4. A precise statement follows.

Theorem 16 *Suppose* $F : \{0,1\}^{\tilde{n}} \to \{0,1\}^{m \cdot L}$ *is a* $(T, L, k, \varepsilon_1, \varepsilon_2)$*-somewhere-amplifiable extractor for* $\varepsilon_1 = \mathsf{negl}(\tilde{n})$*, and* $\varepsilon_2 \leq c_2$ *for some arbitrary constant* $c_2 \leq 1 - 2^{-m}$ *(so that* ε_1 *is useful for applications and* ε_2 *is non-trivial). Then, either the number of blocks* L *is superpolynomial in* $\tilde{n}$ *(and hence amplification is inefficient), or we have* $m = O(k/T)$*.*

Some comments are due about Theorem 16. First, Theorem 16 provides a strong separation between convSA-source extraction from weak sources and convSR-source extraction from SHELA sources, as already evidenced in Section 1.4. Consider a SHELA source with ℓ blocks of length n, $\ell = \mathrm{poly}(n)$, $t = 2$ of which are honest with arbitrary linear min-entropy. Then, Theorem 9 shows we can efficiently extract (to within error $2^{-\Omega(\mathrm{poly}(n))}$) a convSR-source with $\mathrm{poly}(n)$ number of blocks each of length $\Omega(n)$ and at least one good block from the SHELA source. Such SHELA source can be compared with an arbitrary weak $(\tilde{n} = n \cdot \ell, k = O(n))$-source. In this case, Theorem 16 shows that if we want to obtain a T-out-of-L convSA-source with block length $\Omega(n)$ from the weak source, then T must be constant. This precludes many applications of the resulting convSA-source as discussed in Section 1. Finally, note that Theorem 16 also applies to the extraction of convSR-sources with several uniform blocks from weak sources.

7 Non-Interactive Protocols from Public SHELA Sources

7.1 CRS Generation Through a SHELA Sample

The definitions of proof systems and commitment schemes in the plain model and in the CRS model are standard and can be found in [1].

Such definitions assume the existence of an efficient CRS generation procedure $\mathcal{G}$ that, however, will instead be realized in our protocols through a sample from a public SHELA source. Our constructions will convert 2-round public-coin protocols into non-interactive protocols by using a SHELA source and the somewhere-extrator to replace the first round. Therefore, following the notation in the CRS model, when running $\mathcal{G}$ on input 1^m to generate a sufficiently long CRS, we assume that the CRS is generated through a sample $\sigma \leftarrow \mathsf{SHELA}_{n,k,t,\ell}$ from a SHELA source such that when running $\mathsf{SomeExt}(\sigma)$ and obtaining blocks $R_1, \ldots, R_L$ we have that the size of each R_i is equal to the size of the first round of the 2-round public-coin protocol. We recall that $\mathcal{G}$ is not supposed to be efficient and neither simulatable. Moreover, this procedure allows an unbounded adversary to partially control the sampling process. We obviously require that the output of $\mathcal{G}$ be available to all players. In our protocols, some adversaries are restricted to run in polynomial-time only, but still can affect the outcome of the SHELA sample without such restriction.

Non-Interactive WI proof system $\Pi_{\mathsf{pv}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pv}}, \mathcal{V}_{\mathsf{pv}})$

CRS Generation: $\mathcal{G}$ on input 1^m outputs $\sigma \leftarrow \mathsf{SHELA}_{n,k,t,\ell}$.

Prover Procedure: $\mathcal{P}_{\mathsf{pv}}$. Input: instance x, witness w s.t. $(x, w) \in \mathcal{R}$ and $\sigma \in \mathsf{SHELA}_{n,k,t,\ell}$.

1. Run $\mathsf{SomeExt}(\sigma)$ obtaining $R_1, \ldots R_L$.
2. For $i = 1, \ldots, L$: Run $\pi_i \leftarrow \mathcal{P}(1^m, x, w, R_i)$.
3. Set $\pi = (\pi_1, \ldots, \pi_L)$, output π.

Verifier Procedure: $\mathcal{V}_{\mathsf{pv}}$. Input: instance x and $\sigma \in \mathsf{SHELA}_{n,k,t,\ell}$.

1. Run $\mathsf{SomeExt}(\sigma)$ obtaining $R_1, \ldots, R_L$.
2. If $\mathcal{V}(x, w, R_i, \pi_i) = 1\ \forall i = 1, \ldots, L$ accept, otherwise reject.

Fig. 1. Non-Interactive WI Proof System $\Pi_{\mathsf{pv}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pv}}, \mathcal{V}_{\mathsf{pv}})$.

7.2 Non-Interactive WI Proof System Π_{pv}

Here we present our construction of NIWI proof system from SHELA sources assuming public-coin ZAPs. In order to describe our proof system $\Pi_{\mathsf{pv}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pv}}, \mathcal{V}_{\mathsf{pv}})$ for the NP-language $\mathcal{L}$, we will make use of the following tools: 1) A somewhere extractor $\mathsf{SomeExt} : \{0,1\}^{n \cdot \ell} \to \{0,1\}^{m \cdot L}$ defined in Section 4[15]. 2) A 2-round public-coin WI proof system $\Pi = (\mathcal{P}, \mathcal{V})$. Our Non-Interactive WI proof system

[15] With high min-entropy we set $L = \ell - 1$, while with low min-entropy we set $L = O(\ell)$.

$\Pi_{\mathsf{pv}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pv}}, \mathcal{V}_{\mathsf{pv}})$ with a CRS generated through a sample from a SHELA source is described in Figure 1.

Theorem 17 *Assuming the existence of public SHELA sources, if public-coin ZAPs exist, then Π_{pv} is a non-interactive proof system for all NP-languages.*

We stress that our protocol can be instantiated using doubly enhanced trapdoor permutations. The proof can be found in [1].

7.3 Non-Interactive Commitment Scheme Π_{pvcom}

Here we present our construction of non-interactive statistically binding commitment scheme from SHELA sources assuming 2-round public-coin statistically binding commitments. In order to describe our commitment scheme $\Pi_{\mathsf{pvcom}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pvcom}}, \mathcal{V}_{\mathsf{pvcom}})$ for the message space M, we will make use of the following tools: 1) a somewhere extractor $\mathsf{SomeExt} : \{0,1\}^{n \cdot \ell} \to \{0,1\}^{m \cdot L}$ defined in Section 4[16]; 2) a 2-round public-coin statistically binding commitment scheme $\Pi_{\mathsf{com}} = (\mathcal{S}, \mathcal{R})$. Our Non-Interactive Commitment Scheme $\Pi_{\mathsf{pvcom}} = (\mathcal{G}, \mathcal{P}_{\mathsf{pvcom}}, \mathcal{V}_{\mathsf{pvcom}})$ using a public SHELA source is described in in Figure 2.

NON-INTERACTIVE COMMITMENT SCHEME $\Pi_{\mathsf{pvcom}} = (\mathcal{G}, \mathcal{S}_{\mathsf{pvcom}}, \mathcal{R}_{\mathsf{pvcom}})$

CRS GENERATION: $\mathcal{G}$ on input 1^m outputs $\sigma \leftarrow \mathsf{SHELA}_{n,k,t,\ell}$.

SENDER PROCEDURE: $\mathcal{S}_{\mathsf{pvcom}}$. Input: message msg and $\sigma \in \mathsf{SHELA}_{n,k,t,\ell}$.

1. Run $\mathsf{SomeExt}(\sigma)$ obtaining $R_1, \ldots R_L$.
2. For $i = 1, \ldots, L$: Run $\mathsf{com}_i, \mathsf{dec}_i \leftarrow \mathcal{S}(1^m, \mathsf{msg}, R_i)$.
3. Set $\mathsf{com} = (\mathsf{com}_1, \ldots, \mathsf{com}_L)$, $\mathsf{dec} = (\mathsf{dec}_1, \ldots, \mathsf{dec}_L)$ and output com.

RECEIVER PROCEDURE: $\mathcal{R}_{\mathsf{pvcom}}$. Input: commitment com, decommitment $\mathsf{dec}, \mathsf{msg}$ and $\sigma \in \mathsf{SHELA}_{n,k,t,\ell}$.

1. Run $\mathsf{SomeExt}(\sigma)$ obtaining $R_1, \ldots, R_L$.
2. If $\mathcal{R}(\mathsf{msg}, \mathsf{com}_i, R_i, \mathsf{dec}_i) = 1 \ \forall i = 1, \ldots, L$ outputs msg, otherwise reject.

Fig. 2. Non-Interactive Commitment Scheme from OWFs $\Pi_{\mathsf{pvcom}} = (\mathcal{G}, \mathcal{S}_{\mathsf{pvcom}}, \mathcal{R}_{\mathsf{pvcom}})$.

Theorem 18 *Assuming the existence of public SHELA sources, if 2-round public-coin statistically binding commitment schemes exist then Π_{pvcom} is a non-interactive commitment scheme.*

We stress that our protocol can be instantiated through a black-box use of any one-way function.

[16] We set L precisely as specified in the previous footnote.

Acknowledgments. DA and MO were funded by the Singapore Ministry of Education and the National Research Foundation under grant R-710-000-012-135. Part of this work was done while MO was visiting the University of Warsaw (visit supported by TEAM/2016-1/4 grant from the Foundation for Polish Science). Part of this work was done while JR was visiting the Centre for Quantum Technologies, National University of Singapore. Part of this work was done while LS was at the University of Salerno and visiting the Centre for Quantum Technologies, National University of Singapore. LS and IV were supported in part by the European Union's Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE) and in part by "GNCS - INdAM".

References

1. Aggarwal, D., Obremski, M., Ribeiro, J., Siniscalchi, L., Visconti, I.: How to extract useful randomness from unreliable sources. Cryptology ePrint Archive, Report 2019/1156 (2019), https://eprint.iacr.org/2019/1156
2. Barak, B., Kindler, G., Shaltiel, R., Sudakov, B., Wigderson, A.: Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. J. ACM **57**(4) (May 2010). https://doi.org/10.1145/1734213.1734214
3. Barak, B., Rao, A., Shaltiel, R., Wigderson, A.: 2-source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. Annals of Mathematics **176**(3), 1483–1543 (2012)
4. Beigi, S., Bogdanov, A., Etesami, O., Guo, S.: Optimal Deterministic Extractors for Generalized Santha-Vazirani Sources. In: APPROX/RANDOM 2018. LIPIcs, vol. 116, pp. 30:1–30:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2018.30
5. Beigi, S., Etesami, O., Gohari, A.: Deterministic randomness extraction from generalized and distributed Santha–Vazirani sources. SIAM J. Computing **46**(1), 1–36 (2017). https://doi.org/10.1137/15M1027206
6. Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: Security in the face of parameter subversion. In: ASIACRYPT 2016. pp. 777–804. Springer, Berlin, Heidelberg (2016)
7. Ben-Aroya, A., Chattopadhyay, E., Doron, D., Li, X., Ta-Shma, A.: A new approach for constructing low-error, two-source extractors. In: CCC 2018. pp. 3:1–3:19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Germany (2018)
8. Bennett, C.H., Brassard, G., Robert, J.M.: How to reduce your enemy's information (extended abstract). In: CRYPTO 1985. pp. 468–476. Springer, Berlin, Heidelberg (1986)
9. Bentov, I., Gabizon, A., Zuckerman, D.: Bitcoin beacon. arXiv e-prints arXiv:1605.04559 (May 2016)
10. Bitansky, N., Paneth, O.: Point obfuscation and 3-round zero-knowledge. In: TCC 2012. pp. 190–208. Springer, Berlin, Heidelberg (2012)
11. Blum, M.: Independent unbiased coin flips from a correlated biased source—a finite state Markov chain. Combinatorica **6**(2), 97–108 (Jun 1986)
12. Bourgain, J.: More on the sum-product phenomenon in prime fields and its applications. International Journal of Number Theory **01**(01), 1–32 (2005)

13. Bourgain, J.: On the construction of affine extractors. GAFA Geometric And Functional Analysis **17**(1), 33–57 (Apr 2007)
14. Bourgain, J., Dvir, Z., Leeman, E.: Affine extractors over large fields with exponential error. computational complexity **25**(4), 921–931 (Dec 2016)
15. Canetti, R., Pass, R., Shelat, A.: Cryptography from sunspots: How to use an imperfect reference string. In: FOCS 2007. pp. 249–259 (Oct 2007). https://doi.org/10.1109/FOCS.2007.70
16. Chattopadhyay, E., Goodman, J., Goyal, V., Li, X.: Extractors for adversarial sources via extremal hypergraphs. Cryptology ePrint Archive, Report 2019/1450 (2019), https://eprint.iacr.org/2019/1450, to appear in STOC 2020.
17. Chattopadhyay, E., Goyal, V., Li, X.: Non-malleable extractors and codes, with their many tampered extensions. In: STOC 2016. p. 285–298. ACM, New York, NY, USA (2016). https://doi.org/10.1145/2897518.2897547
18. Chattopadhyay, E., Zuckerman, D.: Explicit two-source extractors and resilient functions. Annals of Mathematics **189**(3), 653–705 (2019)
19. Chor, B., Goldreich, O., Hasted, J., Freidmann, J., Rudich, S., Smolensky, R.: The bit extraction problem or t-resilient functions. In: FOCS 1985. pp. 396–407 (Oct 1985). https://doi.org/10.1109/SFCS.1985.55
20. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM J. Computing **17**(2), 230–261 (1988). https://doi.org/10.1137/0217015
21. Clark, J., Hengartner, U.: On the use of financial data as a random beacon. In: EVT/WOTE 2010. pp. 1–8. USENIX Association, Berkeley, CA, USA (2010)
22. Cohen, G., Schulman, L.J.: Extractors for near logarithmic min-entropy. In: FOCS 2016. pp. 178–187 (Oct 2016). https://doi.org/10.1109/FOCS.2016.27
23. Cohen, G.: Local correlation breakers and applications to three-source extractors and mergers. SIAM J. Computing **45**(4), 1297–1338 (2016). https://doi.org/10.1137/15M1029837
24. Cohen, G., Shinkar, I.: Zero-fixing extractors for sub-logarithmic entropy. In: ICALP 2015. pp. 343–354. Springer, Berlin, Heidelberg (2015)
25. De, A., Watson, T.: Extractors and lower bounds for locally samplable sources. In: APPROX/RANDOM 2011. pp. 483–494. Springer, Berlin, Heidelberg (2011)
26. DeVos, M., Gabizon, A.: Simple affine extractors using dimension expansion. In: CCC 2010. p. 50–57. IEEE Computer Society, USA (2010). https://doi.org/10.1109/CCC.2010.14
27. Dodis, Y.: New imperfect random source with applications to coin-flipping. In: ICALP 2001. pp. 297–309. Springer, Berlin, Heidelberg (2001)
28. Dodis, Y., Vaikuntanathan, V., Wichs, D.: Extracting randomness from extractor-dependent sources. Cryptology ePrint Archive, Report 2019/1339 (2019), https://eprint.iacr.org/2019/1339
29. Dvir, Z., Kopparty, S., Saraf, S., Sudan, M.: Extensions to the method of multiplicities, with applications to Kakeya sets and mergers. SIAM J. Computing **42**(6), 2305–2328 (2013)
30. Dvir, Z., Gabizon, A., Wigderson, A.: Extractors and rank extractors for polynomial sources. Computational Complexity **18**(1), 1–58 (Apr 2009)
31. Dvir, Z., Raz, R.: Analyzing linear mergers. Random Structures & Algorithms **32**(3), 334–345 (2008)
32. Dvir, Z., Shpilka, A.: An improved analysis of linear mergers. computational complexity **16**(1), 34–59 (May 2007)
33. Dvir, Z., Wigderson, A.: Kakeya sets, new mergers, and old extractors. SIAM J. Computing **40**(3), 778–792 (2011). https://doi.org/10.1137/090748731

34. Dwork, C., Naor, M.: Zaps and their applications. In: FOCS 2000. pp. 283–293 (Nov 2000). https://doi.org/10.1109/SFCS.2000.892117
35. Elias, P.: The efficient construction of an unbiased random sequence. Ann. Math. Statist. **43**(3), 865–870 (06 1972)
36. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM J. Comput. **29**(1), 1–28 (1999)
37. Fuchsbauer, G., Orrù, M.: Non-interactive zaps of knowledge. In: ACNS 2018. pp. 44–62. Springer, Cham (2018)
38. Gabizon, A., Raz, R.: Deterministic extractors for affine sources over large fields. In: FOCS 2005. pp. 407–416 (Oct 2005). https://doi.org/10.1109/SFCS.2005.31
39. Gabizon, A., Raz, R., Shaltiel, R.: Deterministic extractors for bit-fixing sources by obtaining an independent seed. SIAM J. Computing **36**(4), 1072–1094 (2006)
40. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: EUROCRYPT 2015. pp. 281–310. Springer, Berlin, Heidelberg (2015)
41. Goyal, R., Goyal, V.: Overcoming cryptographic impossibility results using blockchains. In: TCC 2017. pp. 529–561. Springer, Cham (2017)
42. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: CRYPTO 2006. pp. 97–111. Springer, Berlin, Heidelberg (2006)
43. Guruswami, V., Umans, C., Vadhan, S.: Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. J. ACM **56**(4), 20:1–20:34 (Jul 2009)
44. Kamp, J., Rao, A., Vadhan, S., Zuckerman, D.: Deterministic extractors for small-space sources. Journal of Computer and System Sciences **77**(1), 191 – 220 (2011)
45. Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure–resilient cryptography. SIAM J. Computing **36**(5), 1231–1247 (2007)
46. Lewko, M.: An explicit two-source extractor with min-entropy rate near 4/9. Mathematika **65**(4), 950–957 (2019)
47. Li, F., Zuckerman, D.: Improved extractors for recognizable and algebraic sources. Electronic Colloquium on Computational Complexity (ECCC) **25**, 110 (2018)
48. Li, X.: Improved constructions of three source extractors. In: CCC 2011. pp. 126–136 (June 2011). https://doi.org/10.1109/CCC.2011.26
49. Li, X.: A new approach to affine extractors and dispersers. In: CCC 2011. pp. 137–147 (June 2011). https://doi.org/10.1109/CCC.2011.27
50. Li, X.: Extractors for a constant number of independent sources with polylogarithmic min-entropy. In: FOCS 2013. pp. 100–109 (Oct 2013). https://doi.org/10.1109/FOCS.2013.19
51. Li, X.: Improved two-source extractors, and affine extractors for polylogarithmic entropy. In: FOCS 2016. pp. 168–177 (Oct 2016). https://doi.org/10.1109/FOCS.2016.26
52. Li, X.: New independent source extractors with exponential improvement. In: STOC 2013. pp. 783–792. ACM, New York, NY, USA (June 2013)
53. Li, X.: Three-source extractors for polylogarithmic min-entropy. In: FOCS 2015. pp. 863–882 (Oct 2015). https://doi.org/10.1109/FOCS.2015.58
54. Lichtenstein, D., Linial, N., Saks, M.: Some extremal problems arising from discrete control processes. Combinatorica **9**(3), 269–287 (Sep 1989)
55. Lu, C.J., Reingold, O., Vadhan, S., Wigderson, A.: Extractors: Optimal up to constant factors. In: STOC 2003. pp. 602–611. ACM, New York, NY, USA (2003)
56. Naor, M.: Bit commitment using pseudorandomness. J. Cryptology **4**(2), 151–158 (1991)

57. von Neumann, J.: Various techniques used in connection with random digits. In: Monte Carlo Method, National Bureau of Standards Applied Mathematics Series, vol. 12, chap. 13, pp. 36–38. US Government Printing Office, Washington, DC (1951)
58. Nisan, N., Zuckerman, D.: Randomness is linear in space. Journal of Computer and System Sciences **52**(1), 43 – 52 (1996)
59. Pass, R., Seeman, L., Shelat, A.: Analysis of the blockchain protocol in asynchronous networks. In: EUROCRYPT 2017. pp. 643–673. Springer, Cham (2017)
60. Pudlak, P., Rodl, V.: Extractors for small zero-fixing sources. arXiv e-prints arXiv:1904.07949 (April 2019)
61. Radhakrishnan, J., Ta-Shma, A.: Bounds for dispersers, extractors, and depth-two superconcentrators. SIAM Journal on Discrete Mathematics **13**(1), 2–24 (2000)
62. Rao, A.: Extractors for low-weight affine sources. In: CCC 2009. pp. 95–101 (July 2009). https://doi.org/10.1109/CCC.2009.36
63. Rao, A.: Extractors for a constant number of polynomially small min-entropy independent sources. SIAM Journal on Computing **39**(1), 168–194 (2009)
64. Raz, R.: Extractors with weak random seeds. In: STOC 2005. p. 11–20. ACM, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060593
65. Santha, M., Vazirani, U.V.: Generating quasi-random sequences from slightly-random sources. In: FOCS 1984. pp. 434–440 (Oct 1984). https://doi.org/10.1109/SFCS.1984.715945
66. Scafuro, A., Siniscalchi, L., Visconti, I.: Publicly verifiable proofs from blockchains. In: PKC 2019. pp. 374–401. Springer, Cham (2019)
67. Ta-Shma, A.: On extracting randomness from weak random sources (extended abstract). In: STOC 1996. pp. 276–285. ACM, New York, NY, USA (1996)
68. Trevisan, L., Vadhan, S.: Extracting randomness from samplable distributions. In: FOCS 2000. pp. 32–42. IEEE Computer Society, Washington, DC, USA (2000)
69. Vazirani, U.V.: Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources. In: STOC 1985. pp. 366–378. ACM, New York, NY, USA (1985)
70. Viola, E.: Extractors for Turing-machine sources. In: APPROX/RANDOM 2012. pp. 663–671. Springer, Berlin, Heidelberg (2012)
71. Viola, E.: Extractors for circuit sources. SIAM Journal on Computing **43**(2), 655–672 (2014)
72. Yehudayoff, A.: Affine extractors over prime fields. Combinatorica **31**(2), 245 (Aug 2011)
73. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: STOC 2006. pp. 681–690. ACM, New York, NY, USA (2006)