From Extractable Collision Resistance to Succinct Non-Interactive Arguments of Knowledge, and Back Again

Nir Bitansky* Ran Canetti† Alessandro Chiesa‡ Eran Tromer*

August 17, 2011

Abstract

The existence of non-interactive succinct arguments (namely, non-interactive computationally-sound proof systems where the verifier's time complexity is only polylogarithmically related to the complexity of deciding the language) has been an intriguing question for the past two decades. The question has gained renewed importance in light of the recent interest in delegating computation to untrusted workers. Still, other than Micali's CS proofs in the Random Oracle Model, the only existing candidate construction is based on an elaborate assumption that is tailored to the specific proposal [Di Crescenzo and Lipmaa, CiE '08]. We modify and re-analyze that construction:

- We formulate a general and relatively mild notion of extractable collision-resistant hash functions (ECRHs), and show that if ECRHs exist then the modified construction is a non-interactive succinct argument (SNARG) for NP. Furthermore, we show that (a) this construction is a proof of knowledge, and (b) it remains secure against adaptively chosen instances. These two properties are arguably essential for using the construction as a delegation of computation scheme.
- We show that existence of SNARGs of knowledge (SNARKs) for NP implies existence of ECRHs, as well as extractable variants of some other cryptographic primitives. This provides further evidence ECRHs are necessary for the existence of SNARKs.
- Finally, we propose several quite different candidate ECRHs.

Similarly to other extractability (or "knowledge") assumptions, the assumption that ECRHs exist does not fit into the standard mold of cryptographic assumptions. Still, ECRH is a natural and basic primitive that may deserve investigation in of itself. Indeed, we demonstrate its power in obtaining a goal that is provably out of reach in more traditional methods [Gentry and Wichs, STOC '10].

*Tel Aviv University, {nirbitan, tromer}@tau.ac.il
†Boston University and Tel Aviv University, canetti@tau.ac.il
‡MIT, alexch@csail.mit.edu

Contents

1 Introduction

The notion of interactive proof systems [GMR89] is central to both modern cryptography and complexity theory. One extensively studied aspect of interactive proof systems is their expressibility; this study culminated with the celebrated result that $\mathsf{IP} = \mathsf{PSPACE}$ [Sha92]. Another aspect of such systems, which is the focus of this work, is that proofs for rather complex NP-statements can potentially be verified much faster than with traditional NP verification.

We know that if statistical soundness is required then any non-trivial savings would cause unlikely complexity-theoretic collapses (see, e.g., [BHZ87, GH98, GVW02, Wee05]). However, if we settle for proof systems with only computational soundness (also known as interactive arguments [BCC88]) then significant savings can be made. Indeed, using collision-resistant hash functions (CRHs), Kilian [Kil92] shows a four-message interactive argument for NP: The prover first uses Merkle hashing to bind itself to a polynomial-size PCP (Probabilistically Checkable Proof) oracle for the statement, and then locally opens the root of the Merkle tree to reveal the PCP verifier's queries. Then, for a security parameter k, the time to verify an instance y for which a valid witness can be checked in time t is bounded by $p(k, |y|, \log t)$, where p is a polynomial independent of the NP language. Following tradition, we call such argument systems *succinct*.

A natural application of succinct argument systems, which has become ever more relevant with the advent of cloud computing, is to delegation of computation: Here a client has some computational task (typically in P) and wishes to delegate the task to an untrusted worker, who responds with the result along with a proof that the result is correct. Indeed, using a succinct argument, the client would be able to verify the correctness of the result using resources that are significantly smaller than those necessary to perform the task from scratch. (We note that current delegation schemes, such as [KR06, GKR08, KR09, GGP10, CKV10], require either more than two messages or much work to be done by the verifier in a preprocessing stage.)

So what is the best possible round complexity for succinct argument systems? By applying the Fiat-Shamir paradigm [FS87] to Kilian's protocol, Micali showed the existence of a one-message succinct argument in the Random Oracle model [Mic00]. In the plain model, however, it is easy to see that one-message succinct arguments do not exist except for "quasi-trivial" languages (i.e., languages in $\mathsf{BPtime}(n^{\mathrm{polylog}n})$). A somewhat more relaxed notion of succinct non-interactive arguments is to first allow the verifier to send ahead of time a succinct string, which we call a *verifier-generated reference string (VGRS)*, that is independent of the statements to be proven later.

Can such non-interactive succinct arguments for NP exist in the plain model?
And if so, under what assumptions can we prove their existence?

Attempted solutions. To answer the above question, Aiello et al. [ABOR00] propose to avoid Kilian's hash-then-open paradigm, and instead use a polylogarithmic PIR (Private Information Retrieval) scheme to access the PCP oracle as a long database. The verifier's first message consists of the queries of the underlying PCP verifier, encrypted using the PIR chooser algorithm. The prover applies the PIR sender algorithm to the PCP oracle, and the verifier then runs the underlying PCP verifier on the values obtained from the PIR protocol. However, Dwork et al. [DLN+04] point out that this "PCP+PIR approach" is inherently problematic, because a cheating prover could "zigzag" and answer different queries according to different databases. (Natural extensions that try to force consistency by using multiple PIR instances run into trouble due to potential PIR malleability.)

Di Crescenzo and Lipmaa [DCL08] propose to address this problem by further requiring the prover to bind itself (in the clear) to a specific database using Merkle hashing as in Kilian's protocol. Intuitively, the prover should now be forced to answer according to a single PCP string. In a sense, this "PCP+MT+PIR approach" squashes Kilian's 4-message protocol down to 2 messages "under the PIR". However, while initially appealing, it is not a-priori clear how this intuition can be turned into a proof of security. Indeed, Di Crescenzo and Lipmaa only provide a security analysis under an assumption that essentially amounts to directly assuming that a convincing prover must use a single well-defined database. Mie [Mie08] also uses such a PCP+MT+PIR approach and provides a proof of security based on a concise knowledge assumption; however, his construction is not succinct, because the verifier's runtime is polynomially related to the time needed to verify the witness. Using different techniques, Groth [Gro10] considers a specific number-theoretic knowledge and, exploiting the homomorphic structure provided by bilinear groups, obtains one-message arguments; however, both the verifier-generated reference string and the verifier's run time are not succinct (as again they are polynomially related to the time to needed to verify the witness).

Recently, Gentry and Wichs [GW11] showed that *no non-interactive (adaptively-sound) succinct argument* can be proven sound via a black-box reduction to a falsifiable assumption [Nao03]. This holds even for *designated-verifier* protocols, where the verifier needs secret randomness in order to verify. Their result somewhat explains the difficulties encountered by previous constructions, and suggests that non-standard assumptions, such as that of [DCL08], may be inherent.

Our result in a nutshell: We revisit the PCP+MT+PIR approach of [DCL08] and show that it can be modified and based on a natural and relatively mild *extractability* assumption on the hash function in use (for which we suggest several candidates). Moreover, we show that the modified construction is in fact an *argument of knowledge against adaptive adversaries*, thereby enabling important applications. Our extractability assumption in fact turns out to be necessary, as it is easily implied by the proof of knowledge property of our (succinct) argument.

However, before going into our results and techniques in more detail, let us review the notions of adaptive arguments of knowledge and extractability.

Adaptive arguments of knowledge. When using succinct arguments for delegation, or in conjunction with other protocols, two enhancements to the "plain" (computational) soundness property of succinct arguments become important. First, soundness should be required to hold even when the claimed theorem is adaptively chosen by the adversary based on previously seen information (including the verifier-generated reference string). Second, not only are we interested in establishing that a witness for a claimed theorem *exists*, we also want that such a witness can be *extracted* from a convincing prover; that is, we require *proof of knowledge* (or rather, an *argument of knowledge*).

The ability to efficiently extract a witness for an adaptively-chosen theorem seems almost essential for making use of a delegation scheme when the untrusted worker is expected to contribute its own input, e.g. a short hash of a database, to the computation. Furthermore, when using arguments (either succinct or not) within other protocols (or other instances of the same protocol), proofs of knowledge become essential to the security analysis [BG08]. (Furthermore, if in addition one achieves public verifiability for succinct arguments, then suddenly *proof composition* becomes viable, a technique that has already been shown to enable many desirable cryptographic tasks [Val08, CT10, BSW11].)

Extractability assumptions. *Extractability assumptions* capture our belief that certain computational tasks can be achieved efficiently only by (essentially) going through specific intermediate stages and thereby obtaining, along the way, some specific intermediate values. This is captured by an assertion that, for any efficient algorithm that achieves the task, there exists a *knowledge extractor* algorithm that efficiently recovers the said intermediate values.

One instance of such assumptions is *extractable functions* (or rather *extractable primitives*). We say that $\mathcal{F}$ is *extractable* if, given a random $f \leftarrow \mathcal{F}$, it is infeasible to produce $y \in \mathsf{Image}(f)$, without actually "knowing" x such that $f(x) = y$. Namely, for any efficient $\mathcal{A}$ there is an efficient extractor $\mathcal{E}_{\mathcal{A}}$ such that, if f, if $\mathcal{A}(f) = f(x)$ for some x, then $\mathcal{E}_{\mathcal{A}}(f)$ almost always outputs x' such that $f(x') = f(x)$. For such a family to be interesting, $\mathcal{F}$ should also encapsulate some sort of hardness, e.g., one-wayness. Assuming that a certain function family is extractable does not typically fit the mold of efficiently falsifiable assumptions as in [Nao03]. In particular, the impossibility result of Gentry and Wichs [GW11] does not apply to such assumptions.

A number of different extractability assumptions exist in the literature, most of which are specific number theoretic assumptions (such as several variants of the *Knowledge of Exponent* assumption [Dam92]) and it is hard to gain assurance in their relative strengths. In an attempt to abstract out from such specific assumptions, Canetti and Dakdouk [CD09, Dak09] formulate general notions of extractability for one-way functions and other basic primitives. We follow this approach.

1.1 Our Results

(i) Extractable collision-resistant hash functions. We start by defining a natural strengthening for collision-resistant hash functions (CRHs): a function ensemble $\mathcal{H} = \{\mathcal{H}_k\}_k$ is an extractable CRH (ECRH) if (a) it is collision resistant in the standard sense, and (b) it is extractable in the sense sketched above. More precisely, extractability is defined as follows:

Definition 1. *A function ensemble $\mathcal{H} = \{\mathcal{H}_k\}_k$ mapping $\{0,1\}^{\ell(k)}$ to $\{0,1\}^k$ is* extractable *if for any poly-size adversary $\mathcal{A}$ there exists a poly-size extractor $\mathcal{E}_{\mathcal{A}}$ such that for large enough security parameter $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr_{h \leftarrow \mathcal{H}_k}\left[\begin{array}{c} y \leftarrow \mathcal{A}(h,z) \\ \exists\, x : h(x) = y \end{array} \wedge \begin{array}{c} x' \leftarrow \mathcal{E}(h,z) \\ h(x') \neq y \end{array}\right] \leq \mathrm{negl}(k) \ .$$

For collision-resistance and extractability to coexist, the image of almost any $h \in \mathcal{H}_k$ should be sparse in $\{0,1\}^k$ (i.e., with cardinality at most $2^{k-\omega(\log k)}$).[1] However, we do not make any verifiability requirements regarding this fact. That is, there need not be any efficient way to tell whether a given string in $\{0,1\}^k$ is in the range of a given $h \in \mathcal{H}_k$.

We also note that the above definition accounts for any (poly-size) auxiliary-input; for our main result we can actually settle for a relaxed definition that only considers a specific distribution over auxiliary inputs (see further discussion in Section 7).

(ii) From ECRHs to adaptive succinct arguments of knowledge, and back again. We modify the "PCP+MT+PIR" construction of [DCL08] and show that the modified construction can be proven secure based solely on the existence of ECRHs and polylogarithmic PIR. Additional features of the modified construction are: (a) The verifier's message can be generated offline independently of the theorem being proved, thus we refer to it as a *verifier-generated reference string* (VGRS); (b) The input can be chosen adaptively by the prover based on previous information, including the VGRS; (c) It is an (adaptive) argument of knowledge; (d) The running time of the verifier and prover, as well as the proof length are "universally succinct" and do not depend on the specific NP-relation at hand. We call arguments satisfying these properties (designated-verifier) *succinct non-interactive arguments of knowledge (SNARKs).* We show:

[1] Actually this condition is sufficient (and natural) but in certain cases it might not be necessary. See discussion in Section 7

Theorem 1 (informal)**.** *If there exist ECRHs and (appropriate) PIRs then there exist SNARKs for* NP.

A single VGRS suffices for only logarithmically many proofs; however, since the VGRS is succinct, the cost of occasionally resending a new one is limited.

Since SNARKs directly imply two-message delegation schemes where the verifier's response need not remain secret, our construction also yields such schemes under the preconditions of Theorem 1. We note that SNARKs for NP (which have "universal" succinctness), rather than (full-fledged) universal SNARKs, are sufficient for this application. For more details, see Sections 5 and 8.

For full adaptivity, we require that the PIR in use supports random queries with respect to an a-priori unknown database size; any FHE-based PIR (e.g., [BV11]) inherently has this feature. When an a-priori bound on the size of the statement is given (e.g., as in the case of delegation) the requirement can be removed altogether.

We complement Theorem 1 by showing that ECRHs are in fact *essential* for SNARKs:

Theorem 2 (informal)**.** *If there exist SNARKs and (standard) CRHs then there exist ECRHs.*

We also show that SNARKs can in fact be used to construct extractable variants of other cryptographic primitives. A naive strategy to obtain this may be to "just add a succinct proof of knowledge" to a cryptographic primitive. While this strategy does not work as such because the proof may leak secret information, we show that in many cases this can be overcome by combining SNARKs with (non-extractable) *leakage-resilient* primitives. Since CRHs and subexponentially-hard OWFs are leakage-resilient, we obtain:

Theorem 3 (informal)**.**

- *SNARKs and (standard) CRHs imply extractable one-way functions and extractable computationally hiding and binding commitments.*
- *SNARKs and (standard) subexponentially-hard one-way functions imply extractable one-way functions. Furthermore, if these functions are one-to-one, we can further construct perfectly-binding computationally-hiding extractable commitments.*

We believe that this approach merits further investigation. One question, for example, is whether extractable pseudorandom generators and extractable pseudorandom functions can be constructed from generic extractable primitives (as was asked and left open in [CD09]). Seemingly, our SNARK-based approach can be used to obtain the weaker variants of extractable pseudo-entropy generators and pseudo-entropy functions, by relying on previous results regarding leakage-resilience of PRGs [DK08, RTTV08, GW11] and leakage-resilient pseudo-entropy functions [BHK11].

(iii) Candidate ECRHs. We give several candidate ECRHs. The first one is based on a generalization of the knowledge of exponent assumption in large algebraic groups. The assumption, which we call t-Knowledge-of-Exponent Assumption, or t-KEA for short proceeds as follows. For any polynomial-size adversary, there exists a polynomial-size extractor such that, on input $g_1, \ldots, g_t$ and $g_1^\alpha, \ldots, g_t^\alpha$ where each g_i is a random generator (of an appropriately-sized random cyclic group) and α is a random exponent, if the adversary outputs (f, f') satisfying $f^\alpha = f'$, then the extractor finds a vector of "coefficients" $(x_1, \ldots, x_t)$ such that $f = \prod_{i\in[t]} g_i^{x_i}$. We note that this assumption is implied by the assumption used by Groth in [Gro10]; furthermore, since Groth proved his assumption to hold in the generic group model, our assumption is also secure there too, thereby giving evidence towards its being true. Informally, we have the following claim:

Theorem 4. *If t-KEA holds in a group where taking discrete logs is hard, then there exists an ECRH whose compression is proportional to t.*

The construction is quite straightforward: we define a function family parameterized by two vectors $(g_1, \ldots, g_t)$ and $(g_1^\alpha, \ldots, g_t^\alpha)$ that, input $(x_1, \ldots, x_t)$, outputs the two group elements $(\prod_{i\in[t]} g_i^{x_i}, \prod_{i\in[t]} g_i^{\alpha x_i})$. The knowledge property directly follows from t-KEA, while collision-resistance is ensured by the hardness of taking discrete logs.

The second candidate is based on the hardness of knapsack (subset sum) problem. Here the underlying assumption is essentially the following: for any poly-size adversary, there exists a polytime extractor, such that whenever the adversary, given a list of elements $l_1, \ldots, l_t$ that are taken from an appropriate distribution over a finite group, outputs a value $y = \sum_{i\in S} l_i$ for some subset $S \subseteq [t]$, the extractor outputs a subset S' such that $y' = \sum_{i\in S'} l_i$ and y are sufficiently close. We call this assumption a *Knowledge of Knapsack* assumption. Specifically, we propose the Knowledge of Knapsack problem where the l_i are noisy integer multiples of a secret real number. This is inspired by a cryptosystem of Regev [Reg03, Reg04], and is shown to be related to hardness of lattice problems.

Note that we cannot in general expect the extractor to output a subset S such that $y = \sum_{i\in S} l_i$ exactly, since the adversary can always output a value that's a slight perturbation of a known subset sum, without "knowing" the preimage of the perturbed value; in our candidates, the perturbed value is likely to be a subset sum as well.

Thus, we do not construct strict ECRHs out of this assumption. Instead, we construct a primitive that's a slight variant that still suffices for our purposes. This variant, called *blurry ECRHs*, says there exist a "proximity" relation $\approx$ on values in the range, and an extension of the hash to a larger domain D, fulfilling the following: (a) given $h \leftarrow \mathcal{H}$, it is hard to find x, x' such that $h(x) \approx h(x')$ holds, and (b) for any polytime adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ such that whenever $h(x) \leftarrow \mathcal{A}(f, z)$, we have that $x' \leftarrow \mathcal{E}(h, z)$ where $x' \in D$ and $h(x) \approx h(x')$.

The actual construction is simple: a description of h includes the group G and vectors $l_1, \ldots, l_t$. Then, $h_{G,l_1,\ldots,l_t}(S) = \Sigma_{i\in S} l_i$. (In fact, to guarantee superpolynomial sparseness of the image, we concatenate several instances of h in a single instance of the actual hash function.) Blurry collision resistance is proven following [Reg03, Reg04]. We show:

Theorem 5. *If Knowledge of Knapsack with the appropriate parameters holds then there exists blurry ECRHs. Furthermore, these blurry ECRHs suffice for obtaining SNARKs as in Theorem 1.*

We note that the notion of a blurry ECRH provides a somewhat different tradeoff between collision resistance and extractability. Specifically, the extractability requirements are somewhat relaxed, whereas the collision resistance properties are stronger than the standard ones (but still hold, for our candidate, based on the same standard hardness assumptions).

1.2 High-level proof strategy for Theorem 1

In this section we provide some high level intuition for the proof of our main technical result: showing that the existence of ECRHs and (appropriate) PIR schemes imply the existence of SNARKs.

The "PCP+MT+PIR approach", a recap. Recall from the introduction that the "PCP+MT+PIR approach" taken by [DCL08] is to "squash" Kilian's 4-message protocol into a 2-message protocol as follows. Instead of first obtaining from the prover a Merkle hash to a PCP oracle and only then asking the prover to locally open the queries requested by the PCP verifier, the verifier sends in advance a PIR-encrypted version of these queries. The prover on his side can then prepare the required PCP oracle, compute and send a Merkle hash of it, and answer the PIR queries according to a database that contains the (short) opening information to each of the bits of the PCP oracle.

[DCL08] base their proof of soundness on the assumption that any convincing prover $\mathcal{P}^*$ must essentially behave as an honest prover; namely the prover should have in mind a *full* PCP oracle π, which maps under the Merkle hash procedure to the claimed root, and such a proof π can be obtained by an efficient extractor $\mathcal{E}_{\mathcal{P}^*}$. [DCL08] then show that, if this is the case, the extracted string must contain valid opening information, for otherwise the extractor can be used to obtain collisions in the underlying hash or break the privacy of the PIR.[2]

The main challenges and our solutions. Recall that our goal is to obtain the stronger notion of *adaptive SNARGs of knowledge* (SNARKs), based on the more restricted assumption that ECRHs exist. At a very high-level, we wish to show that by building the Merkle tree using an ECRH rather than a mere CRH, we can *lift* the "local" extraction guarantee given by the ECRH to a "global" guarantee on the entire Merkle tree. Specifically, we wish to argue that whenever the prover manages to convince the verifier, we can utilize the (local) ECRH-extraction in order to obtain an "extracted PCP oracle" $\tilde{\pi}$ that will be "sufficiently satisfying" for extracting a witness.

We now describe the required modifications, the main challenges, and the way we overcome them towards the above goal. Full details are contained in Section 5, and the construction is summarized in Figure 1.

<u>Extracting a witness.</u> Being interested in SNARKs, we first have to instantiate the underlying PCP system with PCPs of knowledge, which allows for extracting a witness from any sufficiently-satisfying proof oracle. (See details for the requisite PCP system in Section 3.4.)

<u>Adaptivity.</u> In our setting, the prover is allowed to choose the claimed theorem *after* seeing the verifier's first message (or, rather, the verifier-generated reference string). In order to enable the (honest) verifier to do this, we PIR-encrypt the PCP verifier's coins rather than its actual queries (as the former are independent of the instance), and require the prover to prepare an appropriate database (containing all the possible answers for each random tape, rather than a proof oracle). To account for cases in which no a-priori bound on the time to verify the witness is given (and thus there is also no a-priori bound on the size of the corresponding PCP oracle), we require that the PIR supports random queries with respect to a-priori unknown database size. (Any FHE-based PIR, e.g., [BV11] inherently has this feature. Also, when an a-priori bound is given, e.g., in the setting of delegation of computation, this requirement can be removed.)

<u>From local to global extraction.</u> The main technical challenge lies with establishing a "global" knowledge feature (namely, a sufficiently satisfying proof $\tilde{\pi}$) from a very "local" one (namely, the fact that it is infeasible to produce images of the ECRH h without actually knowing a preimage). A natural attempt is to start from the root of the Merkle tree and "working back towards the leaves"; that is, extract a candidate proof $\tilde{\pi}$ by recursively applying the ECRH-extractor to extract the entire Merkle tree $\widetilde{\mathrm{MT}}$, where the leaves should correspond to $\tilde{\pi}$.

However, recursively composing ECRH-extractors already encounters a difficulty: each level of extraction incurs a polynomial blowup in computation size. Hence, (without making a very strong assumption on the amount of "blowup" incurred by the extractor,) we can only apply extraction a constant number of times. We address this problem by opting to use a "squashed" Merkle tree, where the fan-in of each node is polynomial rather than the standard binary tree. Consequently the depth of the tree becomes a constant (that depends on the specific language).

[2] We remark that, as originally formulated the assumption of [DCL08] seems to be false; indeed, a malicious prover can always start from a good PCP oracle π for a true statement and compute an "almost full" Merkle hash on π, skipping very few branches — so one should at least formulate an analogous but more plausible assumption by only requiring "sufficient consistency" with the claimed root.

A tougher issue is that when applying ECRH-extraction to the circuit that produces some intermediate node label ℓ, we are guaranteed that the extracted children map (under the hash) to ℓ only ℓ is indeed a proper image. Hence, the extracted tree might have some inconsistent branches (or rather "holes").[3] Nevertheless, we are indeed able to show (relying solely on ECRH-extraction) that the extracted leaves are sufficiently satisfying for witness-extraction.

Proof at high level. Given the foregoing discussion, we show the correctness of the extraction procedure in two steps:

- *Step 1: "local consistency".* We first show that whenever the verifier is convinced, the recursively extracted string $\tilde{\pi}$ satisfies the PCP verifier with respect to the specific PIR-encrypted coins. Otherwise, it is possible to find collisions within the ECRH h; indeed, if this was not the case then a collision finder could simulate the PIR-encryption on its own, invoke both the extraction procedure and the prover, and obtain two paths that map to the same root but must differ somewhere (as one is satisfying and the other is not).

- *Step 2: "from local to global consistency".* Next, using the privacy guarantees of the PIR scheme, we show that whenever we are able to extract a set of leaves that are satisfying with respect to the PIR-encrypted coins, then the same set of leaves must also be satisfying for almost all other coins and is hence sufficient for witness-extraction. Indeed, if this was not the case then we would be able to use the poly-size extraction circuit to break the semantic security of the PIR.

  The actual reduction to the PIR privacy is a bit more involved, since it requires the ability to test whether a candidate PCP oracle is "sufficiently satisfying" *without having to sample from the randomness of the PCP verifier superpolynomially many times.*

For further details we refer the reader to Sections 3.4 and 5.2.

What does succinctness mean? Our construction ensures that the communication complexity and the verifier's time complexity are bounded by a polynomial in the security parameter, the size of the instance, and the logarithm of the time it takes to verify a valid witness for the instance; this polynomial is *independent* of the specific NP language at hand, i.e., is "universal".

As for soundness, our main construction is not universal, in the sense that the verifier needs to know a constant c such that the verification time of an instance y does not exceed $|y|^c$. Fortunately, this very weak dependence on the specific NP language at hand (weak because it does not even depend on the Turing machine verifying witnesses) does not affect the application to delegation of computation, because the delegator *knows* c at delegation time, having in mind a specific polynomial-time task to delegate.

Nonetheless, we also show how, assuming the existence of exponentially-hard one-way functions, our main construction can be extended to be a *universal* SNARK, that is, a single protocol that can simultaneously work with all NP languages.

1.3 Discussion

We conclude the introduction with an attempt to briefly motivate the premise of this work. Our main contribution can be seen as providing additional understanding of the security of a construction that has frustrated researchers. Towards this goal we prove strong security properties of the scheme based on a new cryptographic primitive that, while not fitting into the mold of "standard cryptographic primitives or assumptions",

[3] This captures for example the behavior of the prover violating the [DCL08] assumption described above.

can be defined concisely and investigated separately. Furthermore, this primitive comes with a number of quite different candidate constructions. Looking beyond the context of this particular protocol, this work can be seen as another step towards understanding the nature of extractability assumptions and their power in cryptography.

1.4 Organization

In Section 2, we discuss more related work. In Section 3, we give basic definitions for the cryptographic primitives that we use (along with any non-standard properties that we may need). In Section 4, we give the definitions for SNARKs.

We then proceed to give details about each of our three main contributions, in each of the next three sections (Section 5, Section 6, and Section 7). Finally, in Section 8, we provide further discussion for how SNARGGoKs can be used in the setting of delegation of computation.

2 Other Related Work

Knowledge assumptions. A popular class of knowledge assumptions, which have been successfully used to solve a number of (at times notoriously open) cryptographic problems, is that of *Knowledge of Exponent* assumptions. These have the following flavor: if an efficient circuit, given the description of a finite group along with some other public information, computes a list of group elements that satisfies a certain algebraic relation, then there exists a knowledge extractor that outputs some related values that "explain" how the public information was put together to satisfy the relation. Most such assumptions have been proven secure against generic algorithms (see Nechaev [Nec94], Shoup [Sho97], and Dent [Den06]), thus offering some evidence for their truth. In the following we briefly survey prior works which, like ours, relied on Knowledge of Exponent assumptions.

Damgård [Dam92] first introduced a Knowledge of Exponent assumption to construct a CCA-secure encryption scheme. Later, Hada and Tanaka [HT98] showed how to use two Knowledge of Exponent assumptions to construct the first three-round zero-knowledge argument. Bellare and Palacio [BP04] then showed that one of the assumptions of [HT98] was likely to be false, and proposed a modified assumption, using which they constructed a three-round zero-knowledge argument.

More recently, Abe and Fehr [AF07] extended the assumption of [BP04] to construct the first perfect NIZK for NP with "full" adaptive soundness. Prabhakaran and Xue [PX09] constructed statistically-hiding sets for trapdoor DDH groups [DG06] using a new Knowledge of Exponent assumption. Gennaro et al. [GKR10] used another Knowledge of Exponent assumption (with an interactive flavor) to prove that a modified version of the Okamoto-Tanaka key-agreement protocol [OT89] satisfies perfect forward secrecy against fully active attackers.

In a different direction, Canetti and Dakdouk [CD08, CD09, Dak09] study *extractable functions*. Roughly, a function f is extractable if finding a value x in the image of f implies knowledge of a preimage of x. The motivation of Canetti and Dakdouk for introducing extractable functions is to capture the abstract essence of prior knowledge assumptions, and to formalize the "knowledge of query" property that is sometimes used in proofs in the Random Oracle Model. They also study which security reductions are "knowledge-preserving" (e.g., whether it possible to obtain extractable commitment schemes from extractable one-way functions).

Prior succinct arguments from knowledge assumptions. Recently, Groth [Gro10] introduced a family of Knowledge of Exponent assumptions, generalizing those of [AF07], and used them to construct extractable length-reducing commitments, as a building block for non-interactive perfect zero-knowledge arguments system for circuit satisfiability. These arguments have very succinct proofs (independent of the circuit size), though the public key is large: quadratic in the size of the circuit. Groth's assumption holds in the generic group model. Our Assumption 7.1 is implied by Groth's.

Mie [Mie08] observes that the PCP+MT+PIR approach works as long as the PIR scheme is *database aware* — essentially, a prover that is able to provide valid answers to PIR queries must "know" a database consistent with those answers. Mie then shows how to make the PIR scheme of Gentry and Ramzan [GR05] PIR-aware, based on Damgård's Knowledge of Exponent assumption [Dam92]; unfortunately, while the communication complexity is very low, the sender in [GR05] is inefficient relative to the database size (specifically, has to perform a number of computations on the order of the square-root of the database size).

Delegation of computation. An important application of succinct arguments is *delegation of computation* schemes, where one also cares about privacy, and not only soundness, guarantees. Specifically, a succinct argument can be usually combined in a trivial way with fully-homomorphic encryption [Gen09] (in order to ensure privacy) to obtain a delegation scheme with similar parameters.

Within the setting of delegation, however, where the same weak verifier may be asking a powerful prover to evaluate an expensive function on many different inputs, a weaker preprocessing approach may still be meaningful. And, indeed, in the preprocessing setting a number of prior works have already achieved constructions where the online stage is only two messages [GGP10, CKV10, AIK10]; note that all of these works do not rely on any knowledge assumption, and the reason is that the preprocessing model is much stronger, and thus the impossibility results of [GW11] do not apply.

However, even given that the preprocessing model is very strong, all of the mentioned works maintain soundness over many delegations only as long as the verifier's answers remain secret. (A notable exception is the work of Benabbas et al. [BGV11], though their constructions are not generic, and are only for specific functionalities such as polynomial functions.)

3 Preliminaries

3.1 Collision-Resistant Hashes

A CRH (*Collision-Resistant Hash*) is a function ensemble for which it is hard to find two inputs that map to the same output. Formally:

Definition 3.1. *A function ensemble $\mathcal{H}$ is a CRH if it is collision resistant in the following sense: for every poly-size adversary $\mathcal{A}$,*

$$\Pr_{h \leftarrow \mathcal{H}_k}\left[\begin{array}{c} x \neq y \\ h(x) = h(y) \end{array} : (x, y) \leftarrow A(h) \right] \leq \mathrm{negl}(k) \ .$$

We say that a function ensemble $\mathcal{H}$ is $(\ell(k), k)$-*compressing* if each $h \in \mathcal{H}_k$ maps strings of length $\ell(k)$ to strings of length $k < \ell(k)$.

3.2 Merkle Trees

Merkle tree (MT) hashing [Mer89] enables a party to use a CRH to compute a succinct commitment to a long string π and later to locally open to any bit of π (again in a succinct manner). Specifically, given a function $h\colon \{0,1\}^{\ell(k)} \to \{0,1\}^k$ randomly drawn from a CRH ensemble, the committer divides π into $|\pi|/\ell$ parts and evaluates h on each of these; the same operation is applied to the resulting string, and so on, until one reaches the single k-bit root. For $|\pi| = (\ell/k)^{d+1}$, this results in a tree of depth d, whose nodes are all the intermediate k-bit hash images. An opening to a leaf in π (or any bit within it) includes all the nodes and their siblings along the path from the root to the leaf and is of size ℓd. Typically, $\ell(k) = 2k$, resulting in a binary tree of depth $\log \pi$. In this work, we shall also be interested in "wide trees" with polynomial compression (rather than constant compression). Further details are given in Section 5.1 where we describe our main construction and its security analysis.

3.3 Private Information Retrieval

A single-server polylogarithmic PIR (*Private Information Retrieval*) scheme [CMS99] consists of algorithms $(\mathsf{PEnc}, \mathsf{PEval}, \mathsf{PDec})$ where:

- $\mathsf{PEnc}_R(1^k, i)$ outputs an encryption C_i of a DB query i, using randomness R,
- $\mathsf{PEval}(\mathsf{DB}, C_i)$ outputs a succinct blob e_i "containing" the answer $\mathsf{DB}[i]$, and
- $\mathsf{PDec}_R(e_i)$ decrypts the blob e_i to an answer $\mathsf{DB}[i]$.

Formally:

Definition 3.2. *A triple of algorithms* $(\mathsf{PEnc}, \mathsf{PEval}, \mathsf{PDec})$ *is a PIR if it has the following properties:*

1. **Correctness.** *For any database* DB*, any query* $i \in \{1, \ldots, |\mathsf{DB}|\}$*, and security parameter* $k \in \mathbb{N}$*,*
$$\Pr_R\left[\mathsf{PDec}_R(e_i) = \mathsf{DB}[i] : \begin{array}{c} C_i \leftarrow \mathsf{PEnc}_R(1^k, i) \\ e_i \leftarrow \mathsf{PEval}(\mathsf{DB}, C_i) \end{array}\right] = 1 \ ,$$
where $\mathsf{PEval}(\mathsf{DB}, C_i)$ *runs in* $\mathrm{poly}(k, |\mathsf{DB}|)$ *time.*

2. **Succinctness.** *The running time of both* $\mathsf{PEnc}_R(1^k, i)$ *and* $\mathsf{PEval}(\mathsf{DB}, C_i)$ *is bounded by*
$$\mathrm{poly}(k, \log |\mathsf{DB}|) \ .$$
In particular, the sizes of the two messages C_i *and* e_i *are also so bounded.*

3. **Semantic security.** *The query encryption is semantically secure, i.e., for any poly-size* $\mathcal{A}$*, all large enough security parameter* $k \in \mathbb{N}$ *and any two queries* $i, i' \in \{0,1\}^{\mathrm{poly}(k)}$*:*
$$\Pr\left[\mathcal{A}(\mathsf{PEnc}_R(1^k, i)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{PEnc}_R(1^k, i')) = 1\right] \leq \mathrm{negl}(k) \ .$$

PIR schemes with the above properties have been constructed under various hardness assumptions such as ΦHA [CMS99] or LWE [BV11].

A-priori unknown DB size. Because we are interested in an adaptive setting, we want the server to be able to specify the DB only after receiving the query. In such cases, the client might not be aware of the DB's size upon issuing its query, but will only be aware of some superpolynomial bound, e.g., $|\mathsf{DB}| = 2^\rho \leq 2^{\log^2 k}$ (where ρ is a-priori unknown). In this case we require that the PIR scheme allows the server to interpret an encrypted (long) query $r \in \{0,1\}^{\log^2 k}$ as its ρ-bit prefix $\hat{r} \in \{0,1\}^\rho$. In any FHE-based scheme, such as the one of [BV11] (which is in turn based on LWE), this extra property can be easily supported. Sometimes (as in the setting of delegation of computation), even if an adversary is adaptive, an a-priori bound on the database size is still available; whenever this is the case, then no additional properties are required of the PIR scheme.

3.4 Probabilistically Checkable Proofs of Knowledge

A verifier-efficient PCP (*Probabilistically Checkable Proof*) of knowledge for the universal relation $\mathcal{R}_{\mathcal{U}}$ is a triple of algorithms $(P_{\mathsf{pcp}}, V_{\mathsf{pcp}}, E_{\mathsf{pcp}})$, where P_{pcp} is the prover, V_{pcp} is the (randomized) verifier, and E_{pcp} is the knowledge extractor.

Given $(y, w) \in \mathcal{R}_{\mathcal{U}}$, $P_{\mathsf{pcp}}(y, w)$ generates a proof π of length $\mathrm{poly}(t)$ and runs in time $\mathrm{poly}(|y|, t)$. The verifier $V^{\pi}_{\mathsf{pcp}}(y, r)$ queries $O(1)$ locations in the proof π according to $\rho = O(\log t)$ coins, $r \in \{0,1\}^{\rho}$, and runs in time $\mathrm{poly}(|y|) = \mathrm{poly}(|M| + |x| + \log t)$. We require:

1. **Completeness.** For every $(y, w) = \big((M, x, t), w\big) \in \mathcal{R}_{\mathcal{U}}$, $\pi \leftarrow P_{\mathsf{pcp}}(y, w)$:

$$\Pr_{r \leftarrow \{0,1\}^{\rho(t)}} \left[V^{\pi}_{\mathsf{pcp}}(y, r) = 1\right] = 1 \ .$$

2. **Proof of knowledge (PoK).** There is a constant ϵ such that for any $y = (M, x, t)$ if

$$\Pr_{r \leftarrow \{0,1\}^{\rho(t)}} \left[V^{\pi}_{\mathsf{pcp}}(y, r) = 1\right] \geq 1 - \epsilon \ ,$$

   then $E_{\mathsf{pcp}}(y, \pi)$ outputs a witness w such that $(y, w) \in \mathcal{R}_{\mathcal{U}}$, and runs in time $\mathrm{poly}(|y|, t)$.

Note that proof of knowledge in particular implies that the soundness error is at most ϵ.

Soundness amplification and verifying "good" oracles. PCPs of knowledge as defined above can be based on the efficient-verifier PCPs of [BSS08, BSGH$^+$05]. (See [Val08] for a simple example of how a PCP of proximity can yield a PCP with a proof of knowledge.) Moreover, the latter PCPs' proofs are of quasi-linear length in t; for simplicity, we shall settle for a bound of t^2.

We shall typically apply the verifier V_{pcp} $q(k)$-times repeatedly to reduce the PoK threshold to $(1 - \epsilon)^q$, where k is the security parameter and $q(k) = \omega(\log k)$. Namely, extraction should succeed whenever $\Pr_{\mathbf{r}}\left[V^{\pi}_{\mathsf{pcp}}(y, \mathbf{r}) = 1\right] \geq (1 - \epsilon)^q$, where $\mathbf{r} = (r_i)_{i \in [q]}$ and $V^{\pi}_{\mathsf{pcp}}(y, \mathbf{r}) = \bigwedge_{i \in [q]} V^{\pi}_{\mathsf{pcp}}(y, r_i)$. We stress that checking this condition is done in time $\mathrm{poly}(t)$ by statistically estimating the acceptance probability of the *non-repeated* verifier.

4 SNARKs

4.1 The Universal Relation and NP Relations

The *universal relation* [BG08] is defined to be the set $\mathcal{R}_{\mathcal{U}}$ of instance-witness pairs (y, w), where $y = (M, x, t)$, $|w| \leq t$, and M is a Turing machine, such that M accepts (x, w) after at most t steps. While the witness w for each instance $y = (M, x, t)$ is of size at most t, there is *no a-priori* polynomial bounding t in terms of $|x|$.

Also, for any $c \in \mathbb{N}$, we denote by $\mathcal{R}_c$ the subset of $\mathcal{R}_{\mathcal{U}}$ consisting of those pairs $(y, w) = \big((M, x, t), w\big)$ for which $t \leq |x|^c$. (This is a "generalized" NP relation, where we do not insist on the same Turing machine accepting different instances, but only insist on a fixed polynomial bounding the running time in terms of the instance size.)

4.2 Succinct Non-Interactive Arguments

A SNARG (*Succinct Non-Interactive Argument*) consists of three algorithms $(\mathcal{P}, (\mathcal{G}_{\mathcal{V}}, \mathcal{V}))$. For a security parameter k, the verifier runs $\mathcal{G}_{\mathcal{V}}(1^k)$ to generate $(\mathsf{vgrs}, \mathsf{priv})$, where vgrs is a (public) verifier-generated reference string and priv are corresponding private verification coins; the honest prover $\mathcal{P}(y, w, \mathsf{vgrs})$ produces a proof Π for the statement $y = (M, x, t)$ given a witness w; then $\mathcal{V}(\mathsf{priv}, y, \Pi)$ verifies the validity of Π. In an *adaptive* SNARG, the prover may choose the statement *after* seeing the vgrs.

Definition 4.1. *The triple of algorithms $(\mathcal{P}, (\mathcal{G}_{\mathcal{V}}, \mathcal{V}))$ is a SNARG for the relation $\mathcal{R} \subseteq \mathcal{R}_{\mathcal{U}}$ if the following conditions are satisfied:*

1. **Completeness.** *For any $(y, w) \in \mathcal{R}$:*

$$\Pr\left[\mathcal{V}(\mathsf{priv}, y, \Pi) = 1 : \begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ \Pi \leftarrow \mathcal{P}(y, w, \mathsf{vgrs}) \end{array}\right] = 1 .$$

   In addition, $\mathcal{P}(y, w, \mathsf{vgrs})$ runs in time $\mathrm{poly}(k, |y|, t)$.

2. **Succinctness.** *The length of the proof Π that $\mathcal{P}(y, w, \mathsf{vgrs})$ outputs, as well as the running time of $\mathcal{V}(\mathsf{priv}, y, \Pi)$, are bounded by*

$$p(k + |y|) = p(k + |M| + |x| + \log t) \ ,$$

   where p is a universal polynomial that does not depend on $\mathcal{R}$. In addition, $\mathcal{G}_{\mathcal{V}}(1^k)$ runs in time $\mathrm{poly}(k)$; in particular, $(\mathsf{vgrs}, \mathsf{priv})$ are of length $\mathrm{poly}(k)$.

3. **Soundness.** *We will refer to several notions (of different strength):*

   - **Non-adaptive soundness.** *For any poly-size prover $\mathcal{P}^*$, all large enough $k \in \mathbb{N}$, and all $y \notin \mathcal{L}_{\mathcal{R}}$:*

$$\Pr\left[\mathcal{V}(\mathsf{priv}, y, \Pi) = 1 : \begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ \Pi \leftarrow \mathcal{P}^*(y, \mathsf{vgrs}) \end{array}\right] \leq \mathrm{negl}(k) .$$

   - **Adaptive soundness.** *For any poly-size prover $\mathcal{P}^*$ and all large enough $k \in \mathbb{N}$:*

$$\Pr\left[\mathcal{V}(\mathsf{priv}, y, \Pi) = 1 : \begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ (y, \Pi) \leftarrow \mathcal{P}^*(\mathsf{vgrs}) \\ y \notin \mathcal{L}_{\mathcal{R}} \end{array}\right] \leq \mathrm{negl}(k) .$$

   - **Adaptive proof of knowledge.** *For any poly-size prover $\mathcal{P}^*$ there exists a poly-size extractor $\mathcal{E}_{\mathcal{P}^*}$ such that for all large enough $k \in \mathbb{N}$ and all auxiliary inputs $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr\left[\begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ (y, \Pi) \leftarrow \mathcal{P}^*(z, \mathsf{vgrs}) \\ \mathcal{V}(\mathsf{priv}, y, \Pi) = 1 \end{array} \wedge \begin{array}{c} (y, w) \leftarrow \mathcal{E}_{\mathcal{P}^*}(z, \mathsf{vgrs}) \\ w \notin \mathcal{R}(y) \end{array}\right] \leq \mathrm{negl}(k) .$$

   We use SNARK as a shorthand for SNARG of knowledge.[4]

[4] One can also formulate weaker PoK notions; in this work we focus on the above strong notion.

Universal arguments vs. weaker notions A SNARG for the relation $\mathcal{R} = \mathcal{R}_{\mathcal{U}}$ is called a *universal argument*. A weaker notion that we will also consider is a SNARG for the relation $\mathcal{R} = \mathcal{R}_c$ for a constant $c \in \mathbb{N}$. In this case the verifier will act according to c and we will only require soundness (or PoK) w.r.t. $\mathcal{R}_c$. Nevertheless, we require (and achieve) *universal succinctness*, where a universal polynomial p, independent of c, upper bounds the length of every proof and the verification time.

Designated verifiers vs. public verification. In a *publicly-verifiable* SNARG the verifier does not require a private state priv. In this work, however, we concentrate on *designated verifier* SNARGs, where priv must remain secret for soundness to hold.

The verifier-generated reference string. A very desirable property is the ability to generate the verifier-generated reference string vgrs once and for all and then reuse it in polynomially-many proofs (potentially by different provers). In publicly verifiable SNARGs, this *multi-theorem soundness* is automatically guaranteed; in designated verifier SNARGs, however, multi-theorem soundness needs to be required explicitly as an additional property. Usually, this is achieved by ensuring that the verifier's response "leaks" only a negligible amount of information about priv. (Note that $O(\log k)$-theorem soundness always holds; the "non-trivial" case is for $\omega(\log k)$. Weaker solutions to support more theorems include assuming that the verifier's responses remain secret, or re-generating vgrs every logarithmically-many rejections, e.g., as in [KR06, Mie08, GKR08, KR09, GGP10, CKV10].)

The SNARK extractor $\mathcal{E}$. Above, we require that any poly-size family of circuits $\mathcal{P}^*$ has a specific poly-size family of extractors $\mathcal{E}_{\mathcal{P}^*}$; in particular, we allow the extractor to be of arbitrary poly-size and to be more non-uniform than $\mathcal{P}^*$. In addition, we require that for any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(n)}$ that the prover might get, the poly-size extractor manages to perform its witness-extraction task given the same auxiliary input z. The definition can be naturally relaxed to consider only specific distributions of auxiliary inputs according to the required application.[5]

One could consider stronger notions in which the extractor is a uniform machine that gets $\mathcal{P}^*$ as input, or even only gets black-box access to $\mathcal{P}^*$. (For the case of adaptive SNARK, the black-box notion cannot be achieved based on falsifiable assumptions [GW11].) In common security reductions, however, where the primitives (to be broken) are secure against arbitrary poly-size non-uniform adversaries, the non-uniform notion seems to suffice. In our case, going from a knowledge assumption to SNARKs, the notion of extraction will be preserved: if you start with uniform extraction you will get SNARK with uniform extraction.

5 From ECRHs to SNARKs

In this section we describe and analyze our construction of adaptive SNARKs for NP based on ECRHs. In Section 5.3 we discuss the universal features of our constructions, and the difficulties in extending it to a full-fledged universal argument; we propose a solution that can overcome the difficulties based on exponentially hard one-way functions.

Before we proceed, though, let us recall the definition of a ECRH:

Definition 5.1. *A $(\ell(k), k)$-compressing ECRH is a $(\ell(k), k)$-compressing CRH that is extractable. (See Definition 3.1 and Definition 1.)*

Our modified PCP+MT+PIR approach. We modify the PCP+MT+PIR approach of [DCL08] and show that the knowledge assumption of [DCL08] (which involves the entire PIR+MT structure) can be replaced

[5] In our setting, the restrictions on the auxiliary-input handled by the knowledge extractor will be related to the auxiliary-input the the underlying ECRH extractor can handle. See further discussion in Section 7.1.

by the simpler generic assumption that ECRHs exist. Furthermore, our modification enables us to improve the result from a two-message succinct argument with non-adaptive soundness to an adaptive SNARG of knowledge (SNARK) — this improvement is crucial for our main application which is delegation of computation. Specifically, we perform two modifications.

1) We instantiate the Merkle tree hash using an ECRH and, unlike the traditional construction where a $(2k, k)$-CRH is used, we use a polynomially-compressing (k^2, k)-ECRH; in particular, for k^{d+1}-long strings the resulting tree will be of depth d (rather than $d \log k$).[6] As we shall see later, doing so allows us to avoid superpolynomial blowup of the final knowledge extractor that will be built via composition of ECRH-extractors. The initial construction we present will be specialized for "generalized" NP-relations $\mathcal{R}_c$; after presenting and analyzing it, we propose an extension to the universal relation $\mathcal{R}_{\mathcal{U}}$ by further invoking a strong hardness assumption.

2) In order to ensure that the first message of the verifier does not depend on the theorem being proved, the database that we use does not consist of (authenticated) bits of π; rather, the r-th entry of the database corresponds to the authenticated answers to the queries of $V^{\pi}_{\mathsf{pcp}}(y, r)$ where y is chosen by the prover and, of course, the authentication is relative to a single string π (to avoid cheating provers claiming one value for a particular location of π in one entry of the database, and another value for the same location of π in another entry of the database). An additional requirement for this part is the use of a PIR scheme that can support databases where the exact size is a-priori unknown (and only a superpolynomial bound is known).

5.1 Construction Details

We start by providing a short description of our MT and then present the detailed construction of the protocol in Figure 1.

The Merkle tree. By padding when required, we assume without loss of generality that the compressed string π is of size k^{d+1} (where d is typically unknown to the verifier). A node in the MT of distance j from the root can be represented by a string $\mathbf{i} = i_1 i_2 \ldots i_j \in [k]^j$ containing the path traversal indices (and the root is represented by the empty string). We then label the nodes with k-bit strings according to π and the hash $h : \{0,1\}^{k^2} \to \{0,1\}^k$:

- The leaf associated with $\mathbf{i} = i_1 \ldots i_d \in [k]^d \cong [k^d]$ is labeled by the $\mathbf{i}$th k-bit block of π, denoted by $\ell_{\mathbf{i}}$ (here $\mathbf{i}$ is interpreted as number in $[k^d]$).
- An internal node associated with $\mathbf{i} = i_1 \ldots i_j \in [k]^j$ is labeled by $h(\ell_{\mathbf{i}1} \ell_{\mathbf{i}2} \ldots \ell_{\mathbf{i}k})$, denoted by $\ell_{\mathbf{i}}$.
- Thus, the label of the root is $h(\ell_1 \ell_2 \ldots \ell_k)$, which we denote by ℓ_ϵ.

The MT commitment is the pair (d, ℓ_ϵ). An opening $\mathsf{dcom}_{\mathbf{i}}$ to a leaf $\mathbf{i}$ consists of all the labels of all the nodes and their siblings along the corresponding path. To verify the opening information, evaluate the hash h from the leaves upwards. Specifically, for each node $\mathbf{i}' = \mathbf{i}j$ along the opening path labeled by $\ell_{\mathbf{i}'} = \ell_{\mathbf{i}j}$ and his siblings' labels $\ell_{\mathbf{i}1}, \ell_{\mathbf{i}2}, \ldots, \ell_{\mathbf{i}(j-1)}, \ell_{\mathbf{i}(j+1)}, \ldots, \ell_{\mathbf{i}k}$, verify that $h(\ell_{\mathbf{i}1}, \ldots, \ell_{\mathbf{i}k}) = \ell_{\mathbf{i}}$.

Theorem 5.1. *For any NP-relation $\mathcal{R}_c$, the construction in Figure 1 yields a SNARK that is secure against adaptive provers. Moreover, the construction is universally succinct: the proof's length and verifier's running-time are bounded by the same universal polynomials for all $\mathcal{R}_c \subset \mathcal{R}_{\mathcal{U}}$.*

[6] We note that any $(k^\epsilon, k^{\epsilon'})$-compressing ECRH would have sufficed (for any constants $\epsilon > \epsilon'$); for the sake of simplicity, we stick with (k^2, k)-compression.

Ingredients.

- A universal efficient-verifier PCP of knowledge $(P_{\mathsf{pcp}}, V_{\mathsf{pcp}}, E_{\mathsf{pcp}})$ for $\mathcal{R}_{\mathcal{U}}$; for $((M, x, t), w) \in \mathcal{R}_{\mathcal{U}}$, a proof π is s.t. $|\pi| \leq t^2$ and the non-repeated verifier uses $\rho = O(\log t)$ coins and $m = O(1)$ queries.
- A succinct PIR $(\mathsf{PEnc}, \mathsf{PEval}, \mathsf{PDec})$ that supports an a-priori unknown DB size. [a]
- An (k^2, k)-ECRH.

Setup $\mathcal{G}_{\mathcal{V}}(1^k)$**.**

- Generate private verification state:
  - Sample coins for $q = \omega(\log k)$ repetitions of V_{pcp}: $\mathbf{r} = (r_1, \ldots, r_q) \xleftarrow{U} \{0,1\}^{\log^2 k \times q}$.
  - Sample coins for encrypting q PIR-queries: $R \xleftarrow{U} \{0,1\}^{\mathrm{poly}(k)}$.
  - Sample an ECRH: $h \leftarrow \mathcal{H}_k$.
  - Set $\mathsf{priv} = (h, \mathbf{r}, R)$.
- Generate corresponding verifier-generated reference string:
  - Compute $C_{\mathbf{r}} = \mathsf{PEnc}_R(1^k, \mathbf{r})$.
  - Set $\mathsf{vgrs} = (h, C_{\mathbf{r}})$.

Proof generation by $\mathcal{P}$**.**

- Input: $1^k, \mathsf{vgrs}, (y, w) \in \mathcal{R}_c$ where $y = (M, x, t)$ and $t \leq |x|^c$.
- Proof generation:
  - Compute a PCP proof $\pi \leftarrow P_{\mathsf{pcp}}(y, w)$ of size $|\pi| = k^{d+1} \leq t^2$.
  - Compute an MT commitment for π: $\ell_\epsilon = \mathrm{MT}_h(\pi)$ of depth d.
  - Let $\rho = O(\log t) < \log^2 k$ be the amount of coins required by V_{pcp}. Compute a DB with 2^ρ entries; in each entry $\hat{r} \in \{0,1\}^\rho$ store the openings $\mathsf{dcom}_{\hat{r}}$ for all the locations of π that are queried by $V^{\pi}_{\mathsf{pcp}}(y, \hat{r})$. [b]
  - Compute $C_{\mathsf{dcom}_{\hat{\mathbf{r}}}} = \mathsf{PEval}(\mathsf{DB}, C_{\mathbf{r}})$. Here each coordinate of $\mathbf{r}$, $r_j \in \{0,1\}^{\log^2 k}$ is interpreted by the PIR as a shorter query $\hat{r}_j \in \{0,1\}^\rho$.
  - The proof is set to be $\Pi = (d, \ell_\epsilon, C_{\mathsf{dcom}_{\hat{\mathbf{r}}}})$.

Proof verification by $\mathcal{V}$**.**

- Input: $1^k, \mathsf{priv}, y, \Pi$, where $y = (M, x, t), \Pi = (d, \ell_\epsilon, C_{\mathsf{dcom}_{\hat{\mathbf{r}}}})$.
- Proof verification:
  - Verify[c] that $k^{d+1} \leq t^2 \leq |x|^{2c}$.
  - Decrypt PIR answers $\mathsf{dcom}_{\hat{\mathbf{r}}} = \mathsf{PDec}_R(C_{\mathsf{dcom}_{\hat{\mathbf{r}}}})$, and verify opened paths (vs h and ℓ_ϵ).
  - Let $\pi|_{\hat{\mathbf{r}}}$ be the opened values of π in the locations corresponding to $\hat{\mathbf{r}}$ (where again $\hat{r}$ is the interpretation of $r \in \{0,1\}^{\log^2 k}$ as $\hat{r} \in \{0,1\}^\rho$). Check whether $V^{\pi|_{\hat{\mathbf{r}}}}_{\mathsf{pcp}}(y, \hat{\mathbf{r}})$ accepts.
  - In case any of the above fail, reject.

[a] Recall that such PIRs can interpret a "long" query $r \in \{0,1\}^{\log^2 k}$ as a shorter query $\hat{r} \in \{0,1\}^\rho$, where $\hat{r}$ is the ρ-bit prefix of r (see Section 3.3).

[b] V_{pcp}'s queries might be adaptively according to previous answers; such behavior can be simulated by the prover.

[c] This is the single spot where the protocol depends on c. See further discussion in Section 5.3.

Figure 1: A SNARK for the relation $\mathcal{R}_c$.

The completeness of the construction follows directly from the completeness of the PCP and PIR. In Section 5.2, we give a security reduction establishing the PoK property (against adaptive provers). In Section 5.3, we discuss *universal succinctness* and possible extensions of our construction to a full-fledged universal argument.

5.2 Proof of Security

A high-level overview of the proof and main technical challenges are presented in Section 1.2. We now turn to the detailed proof, which concentrates on establishing and proving the validity of the knowledge extractor.

Proposition 5.1 (Adaptive SNARK). *For any poly-size $\mathcal{P}^*$ there exists a poly-size extractor $\mathcal{E}_{\mathcal{P}^*}$, such that for all large enough $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr_{h,\mathbf{r},R}\left[\begin{array}{c}(y,\Pi) \leftarrow \mathcal{P}^*(z,h,\mathsf{PEnc}_R(\mathbf{r}))\\ \mathcal{V}((1^k,h,R,\mathbf{r}),y,\Pi)=1\end{array} \wedge \begin{array}{c}(y,w) \leftarrow \mathcal{E}_{\mathcal{P}^*}(1^k,z,h,\mathsf{PEnc}_R(\mathbf{r}))\\ w \notin \mathcal{R}_c(y)\end{array}\right] \leq \mathrm{negl}(k)\,,$$

where h is an ECRH, $\mathbf{r}$ *are the PCP coins and R are the PIR coins.*

We start by describing how the extraction circuit is constructed and then prove that it satisfies Proposition 5.1. In order to simplify notation, we will address provers $\mathcal{P}^*$ that get as input only $(h, C_{\mathbf{r}})$, where $C_{\mathbf{r}} = \mathsf{PEnc}_R(\mathbf{r})$; the analysis can be extended to the case that $\mathcal{P}^*$ also gets additional auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$.

The extraction procedure. As explained above, extraction is done in two phases: first, we recursively extract along all the paths of the Merkle tree (MT); doing so results in a string (of leaf labels) $\tilde{\pi}$; then, we apply to $\tilde{\pi}$ the PCP witness-extractor E_{pcp}. As we shall see, $\tilde{\pi}$ will exceed the knowledge-threshold ϵ of the PCP and hence E_{pcp} will produce a valid witness.

We next describe the recursive extraction procedure of the of the ECRH-based MT. Given a poly-size prover $\mathcal{P}^*$, let d be such that $|\mathcal{P}^*_k| \leq k^d$. We derive $2cd$ circuit families of extractors, one for each potential level of the MT. $\mathcal{E}_1 := \mathcal{E}^{\mathcal{H}}_{\mathcal{P}^*}$ is the ECRH extractor for $\mathcal{P}^*$; like $\mathcal{P}^*$ it expects input $(h, C_{\mathbf{r}}) \in \{0,1\}^{\mathrm{poly}(k)}$ and returns a string $\tilde{\ell}_1, \ldots, \tilde{\ell}_k \in \{0,1\}^{n \times n}$ (which will be a preimage in case $\mathcal{P}^*$ produces a valid image). $\mathcal{E}'_1$ is a an augmented family of circuits that expects input $(h, C_{\mathbf{r}}, i_1)$, where $i_1 \in [k]$ and returns $\tilde{\ell}_{i_1}$ the i_1'th k-bit block of $\mathcal{E}_1(h, C_{\mathbf{r}})$. $\mathcal{E}_2 = \mathcal{E}^{\mathcal{H}}_{\mathcal{E}'_1}$ is then defined to be the extractor for $\mathcal{E}'_1$. In general $\mathcal{E}_{j+1} = \mathcal{E}^{\mathcal{H}}_{\mathcal{E}'_j}$ is the extractor for $\mathcal{E}'_j$ and it expects input $(h, C_{\mathbf{r}}, \mathbf{i})$, where $\mathbf{i} \in [k]^j$. For each $\mathbf{i} \in [k]^j$, $\mathcal{E}_{j+1}(h, C_{\mathbf{r}}, \mathbf{i})$ is meant to extract the labels $\tilde{\ell}_{\mathbf{i}1}, \ldots, \tilde{\ell}_{\mathbf{i}k}$. Recall, however, that the ECRH extractor $\mathcal{E}_{j+1}$ is only guaranteed to output a preimage in case the corresponding circuit $\mathcal{E}'_j$ outputs a true image. For simplicity, we assume that in case $\mathcal{E}'_j$ doesn't output a true image, $\mathcal{E}_{j+1}$ still outputs some string of length k^2 (without any guarantee on this string).

Overall, the witness extractor $\mathcal{E}_{\mathcal{P}^*}$ operates as follows. Given input $(1^k, h, C_{\mathbf{r}})$, it first invokes $\mathcal{P}^*(h, C_{\mathbf{r}})$ and obtains (y, Π); it obtains the depth $\tilde{d}$ from Π, and in case $\tilde{d} > 2cd$ it aborts. Otherwise, for each $\mathbf{i} \in [k]^{\tilde{d}-1}$, extract the labels $(\tilde{\ell}_{\mathbf{i}1}, \ldots, \tilde{\ell}_{\mathbf{i}k}) \leftarrow \mathcal{E}_{\tilde{d}}(h, C_{\mathbf{r}}, \mathbf{i})$. Let $\tilde{\pi}$ be the extracted PCP-proof given by the leaves; apply the PCP witness extractor $\tilde{w} \leftarrow E_{\mathsf{pcp}}(y, \tilde{\pi})$ and output $\tilde{w}$.

We now turn to prove that (except with negligible probability), whenever the verifier is convinced, the extractor $\mathcal{E}_{\mathcal{P}_*}$ outputs a valid witness. The proof is divided to two main claims as outlined above.

A reminder and some notation. Recall that prior to the prover's message, the randomness for the PCP is of the form $\mathbf{r} = (r_i)_{i \in [q]} \in \{0,1\}^{\log^2 \times q}$ (and $q = \omega(\log k)$ is some fixed function). Once the verifier receives (y,Π), where $y=(M,x,t), \Pi=(\tilde{d},\ell_\epsilon,C_{\mathsf{dcom}})$, it computes the amount of coins required $\rho = O(\log t) < \log^2 k$ and interprets each r_j as a shorter $\hat{r}_j \in \{0,1\}^\rho$. ($hr_j$ can be assumed ρ-bit prefix of r_j; in particular, when r_j is uniformly-random so is $\hat{r}_j$.) The corresponding PCP-proof π (or the extracted $\tilde{\pi}$) is of size $k^{\tilde{d}+1}$. We shall denote by $\tilde{\pi} = \mathcal{E}_{\tilde{d}}(h, \mathsf{PEnc}_R(\mathbf{r})) = \bigcup_{\mathbf{i}\in[k]^{\tilde{d}-1}} \mathcal{E}_{\tilde{d}}(h,\mathsf{PEnc}_R(\mathbf{r}),\mathbf{i})$ the extraction of the full set of leaves.

We now show based on collision resistance and ECRH-extraction that (almost) whenever the verifier is convinced, we extract a proof $\tilde{\pi}$ that locally satisfies the queries induced by the encrypted $\mathsf{PEnc}_R(\mathbf{r})$.

Claim 5.1 (Local consistency)**.** *Let $\mathcal{P}^*$ be a poly-size prover strategy, where $|\mathcal{P}^*_k| \le k^d$, and let $(\mathcal{E}_1,\ldots,\mathcal{E}_{2cd})$ be its ECRH extractors as defined above. Then for all large enough $k\in\mathbb{N}$:*

$$\Pr_{(h,R,\mathbf{r})\leftarrow\mathcal{G}_{\mathcal{V}}(1^k)}\left[\begin{array}{c}(y,\Pi)\leftarrow\mathcal{P}^*(h,\mathsf{PEnc}_R(\mathbf{r}))\\ y=(M,x,t),\Pi=(\tilde{d},\ell_\epsilon,C_{\mathsf{dcom}})\\ \mathcal{V}(1^k,(h,R,\mathbf{r}),y,\Pi)=1\end{array}\wedge\begin{array}{c}\tilde{\pi}\leftarrow\mathcal{E}_{\tilde{d}}(1^k,h,\mathsf{PEnc}_R(\mathbf{r}))\\ V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}})=0\end{array}\right]\le \mathrm{negl}(k)\,,$$

where $\hat{\mathbf{r}}\in\{0,1\}^{\rho\times q}$ is the interpretation of $\mathbf{r}\in\{0,1\}^{\log^2 k\times q}$ as (a vector of) shorter random strings (as detailed above.)

Proof. Let us say that $(h,R,\mathbf{r})$ are "bad" if they lead to the above event. Assume towards contradiction that for infinitely many $k\in\mathbb{N}$, there is a noticeable fraction $\epsilon(k)$ of bad tuples $(h,R,\mathbf{r})$. We show how to find collisions in $\mathcal{H}$. Given h, sample PIR-encryption coins R and coins $\mathbf{r}$ for the PCP verifier to simulate $\mathsf{PEnc}_R(\mathbf{r})$. Given that the resulting $(h,R,\mathbf{r})$ are bad, let us show how to produce a collision in h.

First, invoke $P^*(h,\mathsf{PEnc}_R(\mathbf{r}))$ to obtain (y,Π), where $y=(M,x,t),\Pi=(\tilde{d},\ell_\epsilon,C_{\mathsf{dcom}})$. Next, decrypt C_{dcom} (using R) and obtain a set S of $O(q)$ opened paths (each $r_j\in\mathbf{r}=(r_i)_{i\in[q]}$ induces a constant amount of queries). Each path corresponds to some leaf $\mathbf{i}\in[k]^{\tilde{d}}$ and contains $\tilde{d}$ k^2-bit strings $\mathbf{l}^{\mathbf{i}}_1,\ldots,\mathbf{l}^{\mathbf{i}}_{\tilde{d}}\in\{0,1\}^{k^2\times\tilde{d}}$; each string $\mathbf{l}^{\mathbf{i}}_j$ contains the label of the jth-node along the path and the labels of all its siblings.

Next, note that $\tilde{d}\le 2cd$. Indeed, if the verifier accepts then: $k^{\tilde{d}}\le|x|^{2c}$, and in our case $|x|\le|\mathcal{P}^*_k|;\le k^d$. Accordingly, we can utilize our extractors as follows: for each opened path in $\mathbf{i}\in S$, where $\mathbf{i}=i_1\ldots i_{\tilde{d}}\in[k]^{\tilde{d}}$, invoke:

$$\begin{array}{l}\mathcal{E}_1(h,\mathsf{PEnc}_R(\mathbf{r}))\\ \mathcal{E}_2(h,\mathsf{PEnc}_R(\mathbf{r}),i_1)\\ \qquad\qquad\vdots\\ \mathcal{E}_{\tilde{d}}(h,\mathsf{PEnc}_R(\mathbf{r}),i_1i_2\ldots i_{\tilde{d}-1})\end{array}$$

and obtain $\tilde{\mathbf{l}}^{\mathbf{i}}_1,\ldots,\tilde{\mathbf{l}}^{\mathbf{i}}_{\tilde{d}}\in\{0,1\}^{k^2\times\tilde{d}}$.

Let $\pi|_S=\left(\mathbf{l}^{\mathbf{i}}_{\tilde{d}}\right)_{\mathbf{i}\in S}$ be the leaves $\mathcal{P}^*$ opened to and let $\tilde{\pi}|_S=\left(\tilde{\mathbf{l}}^{\mathbf{i}}_{\tilde{d}}\right)_{\mathbf{i}\in S}$ be the extracted leaves. Since $(h,R,\mathbf{r})$ are bad, it holds that $V^{\pi|_S}_{\mathsf{pcp}}(x,\hat{\mathbf{r}})=1$ while $\mathcal{V}^{\tilde{\pi}|_S}_{\mathsf{pcp}}(x,\hat{\mathbf{r}})=0$; in particular, there exist some $\mathbf{i}\in S$ such that $\mathbf{l}^{\mathbf{i}}_{\tilde{d}}\ne\tilde{\mathbf{l}}^{\mathbf{i}}_{\tilde{d}}$. We focus from hereon on this specific $\mathbf{i}$. Let $j\in[\tilde{d}]$ be the smallest index such that $\mathbf{l}^{\mathbf{i}}_j\ne\tilde{\mathbf{l}}^{\mathbf{i}}_j$ (we just established that such an index exists); then it holds that $\mathbf{l}^{\mathbf{i}}_{j-1}=\tilde{\mathbf{l}}^{\mathbf{i}}_{j-1}$. Furthermore, since $(h,R,\mathbf{r})$ are bad $\mathcal{V}$ accepts; this in turn implies that h compresses $\mathbf{l}^{\mathbf{i}}_j$ to the i_{j-1}th block of $\mathbf{l}^{\mathbf{i}}_{j-1}=\tilde{\mathbf{l}}^{\mathbf{i}}_{j-1}$, which we will denote by ℓ. However, the latter is also the output of $\mathcal{E}'_{j-1}(h,\mathsf{PEnc}_R(\mathbf{r}),i_1\ldots i_{j-1})$, which in turn implies that $\mathcal{E}_j(h,\mathsf{PEnc}_R(\mathbf{r}),i_1\ldots i_{j-1})=\tilde{\mathbf{l}}^{\mathbf{i}}_j$ is also compressed by h to the same ℓ (except with negligible probability of extraction failure). It follows that $\mathbf{l}^{\mathbf{i}}_j\ne\tilde{\mathbf{l}}^{\mathbf{i}}_j$ form a collision within h. $\square$

The second step in the proof of Proposition 5.1, is to show that if the aforementioned extractor outputs a proof $\tilde{\pi}$ that convinces the PCP-verifier w.r.t the encrypted randomness, then the same proof $\tilde{\pi}$ must be globally satisfying (at least for witness extraction); otherwise, the poly-size extractor can be used to break the semantic PIR security.

Claim 5.2 (From local satisfaction to extraction)**.** *Let $\mathcal{P}^*$ be a poly-size prover and let $\mathcal{E}_{\mathcal{P}^*}$ be its poly-size extractor*[7]*. Then for all large enough $k \in \mathbb{N}$:*

$$\Pr_{(h,R,\mathbf{r})\leftarrow\mathcal{G}_{\mathcal{V}}(1^k)}\left[\begin{array}{c} V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}})=1 \\ E_{\mathsf{pcp}}(y,\tilde{\pi})\notin\mathcal{R}(y)\end{array} : \begin{array}{c}(y,\Pi)\leftarrow\mathcal{P}^*(h,\mathsf{PEnc}_R(\mathbf{r})) \\ y=(M,x,t),\Pi=(\tilde{d},\ell_\epsilon,C_{\mathsf{dcom}}) \\ \tilde{\pi}\leftarrow\mathcal{E}_{\mathcal{P}^*}(1^k,h,\mathsf{PEnc}_R(\mathbf{r}))\end{array}\right]\leq \operatorname{negl}(k)\,,$$

where $\hat{\mathbf{r}} \in \{0,1\}^{\rho\times q}$ is the interpretation of $\mathbf{r} \in \{0,1\}^{\log^2 k\times q}$ as (a vector of) shorter random strings (as detailed above.)

Proof. Assume towards contradiction that for infinitely many $k \in \mathbb{N}$ the above event occurs with noticeable probability; we show how to break the semantic security of PEnc. First note that whenever the event occurs, it holds that $\Pr_{\hat{\mathbf{r}}\overset{U}{\leftarrow}\{0,1\}^{\rho\times q}}[V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}})=1]\leq(1-\epsilon)^q$, where ϵ is the (constant) knowledge threshold of the PCPoK (see Section 3.4), and $q=\omega(\log k)$ is the number of repetitions. Consider now a CPA game, where the breaker hands its challenger two independent strings of PCP randomness, $(\mathbf{r}_0,\mathbf{r}_1)\in\{0,1\}^{\log^2 k\times 2}$, and gets back $\mathsf{PEnc}_R(\mathbf{r}_b)$, for a random $b\in\{0,1\}$. The breaker then samples a random h, and runs $\mathcal{P}^*(h,\mathsf{PEnc}_R(\mathbf{r}_b))$, $\mathcal{E}_{\mathcal{P}^*}(1^k,h,\mathsf{PEnc}_R(\mathbf{r}_b))$ to obtain an instance $y=(M,x,t)$ from the prover, and an extracted PCP proof $\tilde{\pi}$. It then computes the amount of coins required for V_{pcp}, $\rho=\rho(t)$, and computes accordingly $\hat{\mathbf{r}}_0,\hat{\mathbf{r}}_1$. Next, it checks whether $V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}}_0)\oplus V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}}_1)=1$ and $\Pr_{\hat{\mathbf{r}}\overset{U}{\leftarrow}\{0,1\}^{\rho\times q}}[V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}})=1]\leq(1-\epsilon)^q$ (recall that this can be done efficiently based on the non-repeated verifier, see Section 3.4). If so it answers the challenger with the single b', such that $V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}}_{b'})=1$. If either check fails, it answers with a random b'. By our assumption, with noticeable probability it holds that $\Pr_{\hat{\mathbf{r}}\overset{U}{\leftarrow}\{0,1\}^{\rho\times q}}[V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}})=1]\leq(1-\epsilon)^q$. Moreover, given that the latter occurs, it also holds that $V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}}_b)=1$ with noticeable probability and $V_{\mathsf{pcp}}^{\tilde{\pi}}(y,\hat{\mathbf{r}}_{1-b})=1$ with negligible probability at most $(1-\epsilon)^q$. The result follows. $\square$

Putting it all together. By Claim 5.1 we conclude that whenever the verifier accepts, $\mathcal{E}_{\mathcal{P}^*}$ almost always extracts a proof $\tilde{\pi}$ which locally satisfies the PCP verifier on the encrypted randomness. By Claim 5.2, we deduce that when ever this occurs, $\tilde{\pi}$ must satisfy sufficiently many queries for PCP witness-extraction. This completes the proof of Proposition 5.1 and Theorem 5.1.

Efficiency: "universal succinctness". For input $y=(M,x,t)$ (where $t<k^{\log k}$ for security parameter k), the proof $\Pi=(\ell_\epsilon,d,C_{\mathsf{dcom}_{\hat{\mathbf{r}}}})$ is essentially dominated by the PIR answers $C_{\mathsf{dcom}_{\hat{\mathbf{r}}}}$; this includes $q=\operatorname{polylog}(k)$ PIR answers for entries of size $\tilde{O}(k^2)$.[8] In the PIR scheme of [BV11] the size of each PIR-answer is bounded by $E\cdot k\cdot\operatorname{polylog}(k)+\log D$, where E is the size of an entry and D is the size of the entire DB. Hence, the overall length of the proof is bounded by a fixed polynomial $\tilde{O}(k^2)$, independently of $|x|,|w|,c$. The verifier's and prover's running time are bounded respectively by fixed universal polynomials $\operatorname{poly}(|y|,k)$, $\operatorname{poly}(k,t)$, again independently of any specific c.

[7] The claim actually holds for any circuit family $\mathcal{E}$, but we'll be interested in $\mathcal{P}^*$'s extractor

[8] Recall that $d=\log_k t<\log k$.

5.3 Extension to Universal Arguments

We now discuss the succinctness of our construction and the possibility of extending it to a full-fledged universal argument, namely an argument for the universal relation $\mathcal{R}_{\mathcal{U}}$ as defined in Section 3.

Indeed, Theorem 5.1 tells us that for every $c \in \mathbb{N}$ we obtain a specific protocol that is sound with respect to $\mathcal{R}_c$. However, we note that the dependence on c is only expressed in the verification first step, where $\mathcal{V}$ verifies that $k^{d+1} \leq |x|^{2c}$, while all other components are in fact meant to deal with the universal relation $\mathcal{R}_{\mathcal{U}}$. In particular, as we already mentioned, the running time of both the prover and verifier, as well as the proof-length, are universal and do *not* depend on c.

Towards a full-fledged universal argument. To obtain a full-fledged universal argument we might try to omit the above c-dependent size check. However, now we encounter the following difficulty: for the proof of knowledge to go through we must ensure that the number of recursive extractions is a-priori fixed to some constant $\tilde{d}$ (that may depend on the prover). In particular, we need to prevent the prover $\mathcal{P}^*$ from convincing the verifier of statements $y = (M, x, t)$ with $t > k^{\tilde{d}}$. The natural candidate for $\tilde{d}$ is typically related to the poly-size bound on the size of $\mathcal{P}^*$. Indeed, any prover that actually "writes down" a proof of size t should be of size at least t; intuitively, one could hope that being able to convince the verifier should essentially amount to writing down the proof and computing a Merkle hash of it. However, we have not been able to prove this. Instead, we propose the following addition to the protocol to make it a universal argument.

Proofs of work. For the relation $\mathcal{R}_c$, the above problem of $\mathcal{P}^*$ claiming an artificially large t can be avoided by ensuring that the size of a convincing proof t can only be as large as $|x|^c$, where $|x|$ is a lower-bound on the prover's size. More generally, to obtain a *universal argument*, we can omit the verifier's check (thus collapsing the family of protocols to a *single* protocol) and enhance the protocol of Figure 1 with a *proof of work* attesting that the prover has size at least t^ϵ for some constant $\epsilon > 0$. Concretely, if we are willing to make an additional (though admittedly quite strong) assumption we can obtain such proofs of work:

Theorem 5.2. *If there exist $2^{\epsilon n}$-hard one-way functions (where n is the input size), then, under the same assumptions as Theorem 5.1, we can modify the protocol in Figure 1 to obtain a universal argument.*

Proof sketch. Let $f\colon \{0,1\}^* \to \{0,1\}^*$ a $2^{\epsilon n}$-hard one-way function. Modify $\mathcal{G}_{\mathcal{V}}(1^k)$ to also output $z_1, \ldots, z_\ell$, with $\ell := \frac{\log^2 k}{\epsilon}$ and $z_i := f(s_i)$, where each s_i is drawn at random from $\{0,1\}^i$. Then, when claiming a proof Π for an instance $y = (M, x, t)$, the prover must also present s'_i such that $f(s'_i) = z_i$ where $i > \log t$; the verifier $\mathcal{V}$ can easily check that this is the case by evaluating f. (Note also that the honest prover will have to at most triple its running time when further requested to present this challenge.) Then, by the hardness of f, we know that if the prover has size k^d, then it must be that $k^d > 2^{\epsilon i} > t^\epsilon$, so that we conclude that $k^{d/\epsilon} > t$. Therefore, in the proof of security, we know that the claimed depth of the prover is a constant depending only on d and ϵ, and thus the same proof of security as that of Theorem 5.1 applies. $\square$

Admittedly, assuming exponentially-hard one-way functions is unsatisfying, and we hope to remove the assumption with further analysis; in the meantime, we would like to emphasize that his assumption as already been made, e.g., in natural proofs [RR94] or in works that improve PRG constructions [HHR06]. We also note that, in practice, the above strategy is also a viable solution, as the requisite property can be attained by standard heuristic constructions of hash functions (e.g., find a string whose image under SHA-256 has a given prefix) and block ciphers (e.g., find an AES key consistent with given plaintext-ciphertext examples).

6 From SNARKs to ECRHs (and More)

In this section we provide more details about Theorem 2 and Theorem 3. That is, we show that extractable collision-resistant hash functions (ECRHs) are in fact not only sufficient (together with polylog PIR) but also necessary for SNARKs (assuming standard CRHs). We then describe a general method for obtaining additional (non-interactive) extractable primitives. We start with a short recap of the basic notion of extractable primitives.

Recap: extractability and image verification. We say that a function ensemble $\mathcal{F} = \{\mathcal{F}_k\}_k$ is *extractable* if, given a random $f \leftarrow \mathcal{F}_k$, it is infeasible to produce $y \in \mathsf{Image}(f)$, without actually "knowing" $x \in \mathsf{Domain}(f)$ such that $f(x) = y$. This is formalized by the requirement that for any poly-size adversary $\mathcal{A}$ there is a poly-size extractor $\mathcal{E}_{\mathcal{A}}$ such that for any auxiliary input z and randomly chosen f: If $\mathcal{A}(z, f)$ outputs a proper image, $\mathcal{E}_{\mathcal{A}}(z, f)$ outputs a corresponding preimage. For such a family to be interesting, it is required that $\mathcal{F}$ also encapsulates some sort of hardness, e.g., one-wayness or collision-resistance; in particular, for the two features (of hardness and extractability) to coexist, $\mathsf{Image}(f)$ must be sparse in the (identifiable) range of the function.

As explained in Section 7, when considering extractable primitives it is usually required that one can perform *efficient image verification*; in this context, there are two notions that can be considered:

- *Public verification:* Given f and $y \in \mathsf{Range}(f)$, it should be possible to efficiently test whether $y \in \mathsf{Image}(f)$.
- *Private verification:* Together with the function f, $\mathcal{F}_k$ also generates a private verification state priv_f. Given f, priv_f and $y \in \mathsf{Range}(f)$, it should be possible to efficiently test whether $y \in \mathsf{Image}(f)$.

In addition, the weaker notion of *extractability with no efficient verification* might also be meaningful in certain scenarios. Indeed, for our main ECRH-based construction (presented in Section 5.1), this weak notion of extractability with no efficient verification suffices.

We now present the implications of SNARK to the existence of extractable primitives, starting with the necessity of ECRHs:

Proposition 6.1. *SNARKs and (standard) CRHs imply ECRHs. Moreover, the verifiability features of the SNARK carry over to the implied ECRH.*

Proof sketch. We show that designated-verifier SNARKs imply ECRHs with private verification. The proof can be easily extended to the case of public verifiability. Let $\mathcal{H}$ be an $(3k, k)$-compressing CRH. Let $(\mathcal{P}, (\mathcal{G}_{\mathcal{V}}, \mathcal{V}))$ be an (adaptive) SNARK such that, given security parameter $\hat{k}$, the length of any proof is bounded by $\hat{k}^c$.[9]

We define a $(3k, 2k)$-compressing ECRH, $\widetilde{\mathcal{H}} = \{\widetilde{\mathcal{H}}_k\}_k$. A function $\tilde{h}$ and private verification state $\mathsf{priv}_{\tilde{h}}$ are sampled by $\widetilde{\mathcal{H}}_k$ as follows:

1. Draw a function $h \leftarrow \mathcal{H}_k$,
2. Draw public and private parameters $(\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(k^{1/c})$, and
3. Set $\tilde{h} = (h, \mathsf{vgrs}), \mathsf{priv}_{\tilde{h}} = \mathsf{priv}$.

[9] More precisely, the length of any proof is bounded by $(\hat{k} + \log t)^c$, where t is the computation time; however, we only address statements where the computation is poly-time and in particular $\log t < \hat{k}$.

Then, for an input x and defining $y = h(x)$, we define $\tilde{h}(x) = (y, \Pi)$ where $\Pi = \mathcal{P}(\mathsf{vgrs}, \mathsf{thm}, x)$ is a proof of knowledge for the NP-statement thm = "there exists an $x \in \{0,1\}^{3k}$ such that $h(x) = y$".

The collision resistance of $\widetilde{\mathcal{H}}$ follows directly from that of $\mathcal{H}$, because any colliding pair (x, x') for $\widetilde{\mathcal{H}}$ is a colliding pair for $\mathcal{H}$. The extractability property of $\widetilde{\mathcal{H}}$ follows from the (adaptive) proof of knowledge of the SNARK $(\mathcal{P}, (\mathcal{G}_{\mathcal{V}}, \mathcal{V}))$; that is, for any image-computing poly-size adversary $\mathcal{A}$, the ECRH extractor is set to be the SNARK witness-extractor $\mathcal{E}_{\mathcal{A}}$. In addition, an image can be verified by invoking the SNARK verifier $\mathcal{V}$ with the private verification state priv, the proof Π and the corresponding statement. (We note that, for the proposition to go through, it is crucial for the SNARK to hold against adaptive provers; indeed, the adversary gets to choose on which inputs to compute the hash function, and these may very well depend on the public parameters.) □

We now can immediately deduce that SNARKs also imply *extractable one-way functions* (EOWFs) and *extractable computationally binding and hiding commitments* (ECOMs):

Corollary 6.1. *SNARKs and (standard) CRHs imply EOWFs and ECOMs. Moreover, the verifiability features of the SNARK carry over to the implied primitives.*

Proof sketch. First, note that any $(\ell(k), k)$-compressing ECRH is also a (keyed) EOWF, assuming that $\ell(k) > n + \omega(\log(k))$; indeed, it is a OWF since it is a CRH and independently of that it is also extractable (and verifiable).

Second, to get an extractable bit-commitment scheme, one can use the classic CRH plus *hardcore bit* construction [Blu81]. Specifically, the commitment scheme is keyed by a seed h for the ECRH and a commitment to a bit b is obtained by sampling $r, \hat{r} \stackrel{U}{\leftarrow} \{0,1\}^{\ell(k)}$ and computing

$$\mathsf{Eval}_{\mathsf{Com}}(h; b; r, \hat{r}) = h(r), \hat{r}, b \oplus \langle r, \hat{r} \rangle \ .$$

The fact that this is a computationally binding and hiding commitment holds for any CRH. Moreover, any adversary that computes a valid commitment $c = (y, \hat{r}, b)$ (under the random seed h) also computes a valid image y under h; hence, we can use the ECRH extractor to extract the commitment randomness r, such that $y = h(r)$ and $c = \mathsf{Eval}_{\mathsf{Com}}(h; b \oplus \langle r, \hat{r} \rangle; r, \hat{r})$. In addition, verifying a proper commitment is done by verifying that y is a proper image under h. □

From leakage-resilient primitives and SNARKs to extractable primitives. Naïvely, it seems that non-interactive adaptive proofs of knowledge offer a generic approach towards constructing extractable primitives: "simply add a non-interactive proof of pre-image knowledge" (which might seem to be applicable even without succinctness when compression is not required). However, this approach may actually compromise privacy as such proofs may leak too much information about the preimage. One may try to overcome this problem by applying *non-interactive zero-knowledge proofs of knowledge*, where the crs is generated as part of the seed. However, using this approach, privacy is also inherently compromised with respect to the party generating the seed for the extractable primitive, who has total freedom in choosing the crs; furthermore, this approach seems to only be applicable for public-coin SNARGs and not for designated-verifier SNARGs

We suggest an alternative natural way to overcome this problem, which is to consider stronger (non-extractable) primitives that are resilient to bounded amounts of leakage on the preimage. Then we can leverage the succinctness of SNARKs to claim that proving knowledge of a preimage does not leak too much and hence does not compromise the security of the primitive. Indeed, CRHs are in a sense optimally leakage-resilient OWFs; hence, the first part of Corollary 6.1 can be viewed as an application of this paradigm. Applying this paradigm to one-to-one subexponentially hard OWFs, yields:

Proposition 6.2. *Given any $2^{|x|^\epsilon}$-hard OWF $f : \{0,1\}^* \to \{0,1\}^*$ and SNARKs, there exist extractable OWFs (against poly-size adversaries). Moreover, the verifiability features of the SNARK carry over to the implied EOWF.*

Proof sketch. Let $f : \{0,1\}^* \to \{0,1\}^*$ be any $2^{|x|^\epsilon}$-hard OWF, where n is the size of the input. As in Proposition 5.1, we define an extractable function $\mathcal{F} = \{\mathcal{F}_k\}_k$. Let c be the constant such that any SNARK proof is bounded by $\hat{k}^c$ for security parameter $\hat{k}^c$. The functions generated by $\mathcal{F}_k$ are defined on the domain $\{0,1\}^{k^{2c/\epsilon}}$ and are indexed by a $\mathsf{vgrs} \leftarrow \mathcal{G}_{\mathcal{V}}(1^k)$. For $x \in \{0,1\}^{k^{2c/\epsilon}}$, $f_{\mathsf{vgrs}}(x) = (f(x), \Pi)$, where Π is a SNARK for the statement that "there exists an $x \in \{0,1\}^{k^{2c/\epsilon}}$ such that $f(x) = y$". As for ECRHs, extraction and verifiability follows directly from that of the SNARK. The fact that $\mathcal{F}$ is one-way follows from the fact that f is $2^{|x|^\epsilon}$-hard, and the proof Π is of length at most $|x|^{\epsilon/2}$. In particular, any poly-size adversary which inverts $\mathcal{F}$, can be transformed to a $2^{|x|^\epsilon}$-size adversary that inverts f by simply enumerating all the (short) proofs π. □

We remark that, given SNARK with proof size $\mathrm{polylog}(k)$, one can start from f that is only hard against quasi-poly-size adversaries. (As noted in Section 5.3 such SNARKs can be obtained as in Section 5.1, by making stronger assumptions regarding the ECRH and the PIR.) We also note that the above reduction essentially preserves the structure of the original OWF f; in particular, if f is one-to-one so is $\mathcal{F}$. We thus get:

Corollary 6.2. *Given any $2^{|x|^\epsilon}$-hard one-to-one OWF and SNARKs, there exist extractable commitments that are perfectly binding and computationally hiding (against poly-size adversaries).*

Proof sketch. Indeed, by Proposition 6.2 one-to-one EOWFs, which in turn imply perfectly-binding ECOMs, using the hardcore bit construction as in Corollary 6.1 instantiated with a one-to-one EOWF. (The fact that one-to-one EOWFs imply perfectly binding ECOMs was already noted in [CD09]). □

More extractable primitives based on SNARKs and leakage-resilience. We believe there is room to further investigate the above approach towards obtaining more powerful extractable primitives. In this context, one question that was raised by [CD09] is whether extractable *pseudorandom generators* and *pseudorandom functions* can be constructed from generic extractable primitives, e.g., EOWFs. (They show that the generic constructions of [HILL99] are not knowledge preserving.)

Our SNARK-based approach can seemingly be used to obtain two weaker variants; namely, extractable *pseudo-entropy generators* and *pseudo-entropy functions*. Specifically, the results of [DK08, RTTV08, GW11] imply that any strong enough PRG is inherently also leakage-resilient, in the sense that, even given leakage on the seed, its output still has high pseudo-entropy (specifically, *HILL entropy*). The results of Braverman et. al. [BHK11] show how to obtain the more general notion of leakage-resilient pseudo-entropy functions. We leave the investigation of these possibilities and their applicability for future work.

Non-verifiable extractable primitives from SNARK and perfectly-binding ECOMs. Perfectly-binding ECOMs (as given by Corollary 6.2) seem to provide, with the addition of SNARKs, a generic way of obtaining limited extractable primitives that do not admit efficient verification. Specifically, we transform a function $\mathcal{F}$ to an extractable $\widetilde{\mathcal{F}}$ as follows. The seed $\tilde{f}_{(\mathsf{vgrs},f,g)}$ generated by $\widetilde{\mathcal{F}}_k$ includes a $\mathsf{vgrs} \leftarrow \mathcal{G}_{\mathcal{V}}(1^k)$ for a SNARK, an $f \leftarrow \mathcal{F}_k$, and a seed for the perfectly binding ECOM $g \leftarrow \mathsf{Gen}_{\mathsf{Com}}(1^k)$. To apply the sampled function on x, sample extra randomness r for the commitment, and define $\tilde{f}_{(\mathsf{vgrs},f,g)}(x;r) = (f(x), \mathsf{Eval}_{\mathsf{Com}}(g;\Pi;r))$, where Π is a SNARG of knowledge of the pre-image x (w.r.t vgrs and $f(x)$). That is, add to $f(x)$ a perfectly-binding commitment to a proof of pre-image knowledge. This clearly prevents the problem of leakage on x induced by the proof Π. The fact that the commitment is perfectly-binding and extractable, together with the proof-of-knowledge of the SNARG, imply that $\widetilde{\mathcal{F}}$ is extractable. Indeed, any adversary that produces a valid image, also produces a valid perfectly-binding commitment to a valid proof; hence, using the extractor for the commitment, we obtain an adversary that outputs a valid proof of knowledge for the pre-image x, on which we can already use the SNARK witness extractor. A major caveat of this approach is that the resulting $\widetilde{\mathcal{F}}$ does not support efficient image-verification (the commitment is never opened, and the seed generator does not have any trapdoor on it). At this time we are not aware of applications for non-verifiable extractable primitives other than our non-verifiable ECRH-based SNARK construction. We leave this for future investigation.

7 ECRHs

Having characterized the tight relation between Extractable Collision Resistant Hash (ECRH) ensembles and SNARKs, this section will revisit the definition and construction of ECRH. We fist describe a candidate ECRH based on a generalized Knowledge of Exponent assumption. We will then introduce a "blurry" notion of ECRH (which suffices for our construction of SNARK), and a class of assumptions, *Knowledge of Knapsack*, which imply blurry ECHRs.

7.1 Defining ECRHs

Let us revisit the definition of ECRH from Section 1.1. As discussed, this definition captures the notion that for a given hash function ensemble $\mathcal{H}$, the only way an adversary $\mathcal{A}$ can sample elements in the image of the hash is by knowing a corresponding preimage (which an extractor $\mathcal{E}^{\mathcal{H}}_{\mathcal{A}}$ could in principle find):

Definition 7.1. *An efficiently-samplable function ensemble $\mathcal{H} = \{\mathcal{H}_k\}_k$ is an $(\ell(k), k)$-*compressing ECRH *if it is $(\ell(k), k)$-compressing, collision-resistant, and moreover* extractable*: for any poly-size adversary $\mathcal{A}$, there exists a poly-size extractor $\mathcal{E}^{\mathcal{H}}_{\mathcal{A}}$, such that for all large enough $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr_{h \leftarrow \mathcal{H}_k}\left[\begin{array}{c} y \leftarrow \mathcal{A}(h,z) \\ \exists\, x : h(x) = y \end{array} \wedge \begin{array}{c} x' \leftarrow \mathcal{E}^{\mathcal{H}}_{\mathcal{A}}(h,z) \\ h(x') \neq y \end{array}\right] \leq \mathrm{negl}(k) \ . \tag{1}$$

Image verification. In known applications of extractable primitives (e.g., 3-round zero knowledge [HT98, BP04, CD09]) an extra image-verifiability feature is required from the extractable primitive. Namely, given $y \in \{0,1\}^k$ and h, one should be able to efficiently test whether $y \in \mathsf{Image}(h)$. Here, there are two flavors to consider: (a) public verifiability, where to verify an image all that is required is the (public) seed h; and (b) private verifiability; that is, the seed h is generated together with private verification parameters priv, so that anyone in hold of priv may perform image verification. We emphasize that our main ECRH-based construction (presented in Section 5.1) *does not require any verifiability features*.

Necessity of sparseness. For $\mathcal{H}$ to be collision-resistant, it must also be one-way; namely, the image distribution $\mathcal{D}_h = \left\{h(x) : x \xleftarrow{U} \{0,1\}^{\ell(k)}\right\}$ should be hard to invert (except for negligible probability over h). In particular, $\mathcal{D}_h$ must be very far from the uniform distribution over $\{0,1\}^k$ (for almost all h). Indeed, suppose that the statistical distance between $\mathcal{D}_h$ and uniform is $1 - 1/\mathrm{poly}(k)$, and consider an adversary A that simply outputs range elements $y \in \{0,1\}^k$ uniformly at random, and any $\mathcal{E}^{\mathcal{A}}_{\mathcal{H}}$. In this case, there is no "knowledge" to extract from A, so $\mathcal{E}^{\mathcal{A}}_{\mathcal{H}}$ has to invert uniformly random elements of the range $\{0,1\}^k$. Thus, the success probability of $\mathcal{E}^{\mathcal{A}}_{\mathcal{H}}$ will differ by at most $1 - 1/\mathrm{poly}(k)$ from its success probability had the distribution been $\mathcal{D}_h$, which is negligible (by one-wayness); hence $\mathcal{E}_{\mathcal{H}}^{\mathcal{A}}$ will still fail with probability $1 - 1/\mathrm{poly}(k)$ often, thereby violating Equation (1).

A simple way to ensure that the image distribution $\mathcal{D}_h$ is indeed far from uniform is to make the support of $\mathcal{D}_h$ sparse. We will take this approach in all of the subsequent constructions, making sure that all $h(x)$ fall into a superpolynomially sparse subset of $\{0,1\}^k$: $\mathsf{Image}(h) < 2^{k-\omega(\log k)}$ (except for negligible probability over $h \leftarrow \mathcal{H}_k$).

Of course, this merely satisfies one necessary condition, and is a long way off from implying extractability. Still, this rules out one of the few generic attacks about which we can reason without venturing into the poorly-charted territory of non-blackbox extraction. Moreover, the sparseness (or more generally, statistical distance) requirement rules out many natural constructions; for example, traditional cryptographic CRH ensembles, and heuristic constructions such as the SHA family, have an image distribution $\mathcal{D}_h$ that is close to uniform (by design) and are thus not extractable.

On auxiliary input. The above definition requires that for any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(n)}$ that the prover might get, the poly-size extractor manages to perform its extraction task given the same auxiliary input z. This requirement seems rather strong considering the fact that z could potentially encode arbitrary circuits. For example, the auxiliary input z may encode a circuit that, given the random seed h as input, outputs $h(x)$ where $x = f_s(h)$ is the image of some hardwired pseudorandom function f_s. In this case, the extractor would essentially be required to reverse engineer the circuit, which seems to be a rather strong requirement (or even an impossible one, under certain obfuscation assumptions).

While for presentational purposes that above definition may be simple and convenient, for our main application (i.e. SNARKs) we can actually settle for a weaker definition that is restricted to a specific "benign distribution" on auxiliary inputs. Specifically, in our setting the extractor is required to handle an auxiliary input that includes (honestly-generated) PIR-encryptions of random strings. We note that in certain previous works (e.g. [AF07]), an extra auxiliary input z is seemingly not required (i.e. the extractor only gets the seed for the extractable primitive); however, these actually also inherently assume that the seed itself is "benign" (does not encode an obfuscated malicious circuit).

We also note that if one restricts the ECRHs to handle specific auxiliary-input distributions, then the resulting SNARK will naturally account for the same auxiliary-input distributions and vice-versa (i.e. the ECRHs implied by SNARKs account for the same auxiliary-input as the assumed SNARK).

7.2 ECRHs from t-Knowledge of Exponent

The *Knowledge of Exponent Assumption* (KEA) [Dam92] states that any adversary that, given a generator and a random group element (g, g^α), manages to produce $g^x, g^{\alpha x}$, must "know" the exponent x. The assumption was later extended [HT98, BP04], requiring that given $g^{r_1}, g^{r_1\alpha}, g^{r_2}, g^{r_2\alpha}$ it is infeasible to produce f, f^α without "knowing" x_1, x_2 such that $f = g^{x_1r_1}g^{x_2r_2} = g^{x_1r_1+x_2r_2}$. The t-KEA assumption is a natural extension to $t = \mathrm{poly}(k)$ pairs $g^{r_i}, g^{\alpha r_i}$.

Assumption 7.1 (t-KEA)**.** *There exists an efficiently-samplable ensemble $\mathcal{G} = \{\mathcal{G}_k\}$ where each $(\mathbb{G}, g) \in \mathcal{G}_k$ consists of a group of prime order p in $(2^{k-1}, 2^k)$ and a generator $g \in \mathbb{G}$, such that the following holds. For any poly-size adversary $\mathcal{A}$ there exists a poly-size extractor $\mathcal{E}_\mathcal{A}$ such that for all large enough $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr_{\substack{(\mathbb{G},g)\leftarrow\mathcal{G}_k \\ (\alpha,\mathbf{r})\xleftarrow{U}\mathbb{Z}_p\times\mathbb{Z}_p^t}}\left[\begin{array}{c} (f,f') \leftarrow \mathcal{A}(g^{\mathbf{r}}, g^{\alpha\mathbf{r}}, z) \\ f' = f^\alpha \end{array} \wedge \begin{array}{c} \mathbf{x} \leftarrow \mathcal{E}_\mathcal{A}(g^{\mathbf{r}}, g^{\alpha\mathbf{r}}, z) \\ g^{\langle \mathbf{x},\mathbf{r}\rangle} \neq f \end{array}\right] \leq \mathrm{negl}(k) \enspace ,$$

where $|\mathbb{G}| = p$, $\mathbf{r} = (r_1, \ldots, r_t)$, $g^{\mathbf{r}} = (g^{r_1}, \ldots, g^{r_t})$, $\mathbf{x} = (x_1, \ldots, x_t)$, and $\langle \cdot, \cdot \rangle$ denotes inner product.

A related, stronger assumption was made by Groth [Gro10]. There, instead of random $r_1, \ldots, r_t$, the exponents are powers of the same random element, i.e., $r_i = r^i$. Groth's assumption (and thus ours) holds in the generic group model (see [Gro10] for details).

A candidate ECRH from t-KEA. A $(k \cdot t(k), 2k)$-ECRH $\mathcal{H}$ can now be constructed in the natural way:

- To sample from $\mathcal{H}_k$: sample $(\mathbb{G}, g) \leftarrow \mathcal{G}_k$ and $(\alpha, \mathbf{r}) \overset{U}{\leftarrow} \mathbb{Z}_p \times \mathbb{Z}_p^t$, and output $h := (\mathbb{G}, g^{\mathbf{r}}, g^{\alpha\mathbf{r}})$.
- To compute $h(x_1, \ldots, x_t)$: output the pair $(g^{\langle \mathbf{r}, \mathbf{x} \rangle}, g^{\alpha \langle \mathbf{r}, \mathbf{x} \rangle}) = \left(\prod_{i \in [t]} g^{r_i x_i}, \prod_{i \in [t]} g^{\alpha r_i x_i} \right)$.

The extractability of $\mathcal{H}$ easily follows from the t-KEA assumption. We show that $\mathcal{H}$ is collision-resistant based on the hardness of computing discrete logarithms in $\mathcal{G}$.

Claim 7.1. *If $\mathcal{C}$ finds a collision within $\mathcal{H}$ w.p. ϵ, then we can compute discrete logarithms w.p. ϵ/t.*

Proof sketch. Given g^r, where $r \overset{U}{\leftarrow} \mathbb{Z}_p$, choose a random $i \in [t]$ and sample $\alpha, r_1, \ldots r_{i-1}, r_{i+1}, \ldots, r_t$. Denote $r_i = r$ and $\mathbf{r} = (r_1, \ldots, r_t)$. Feed $\mathcal{C}$ with $g^{\mathbf{r}}, g^{\alpha\mathbf{r}}$. By our initial assumption and the independent choice of i, $\mathcal{C}$ outputs $\mathbf{x}, \mathbf{x}'$ such that $x_i \neq x'_i$ and $g^{\langle \mathbf{x}, \mathbf{r} \rangle} = g^{\langle \mathbf{x}', \mathbf{r} \rangle}$, w.p. at least ϵ/t. It follows that $r_i = (x_i - x'_i)^{-1} \sum_{j \in [k] - \{i\}} (x_j - x'_j) r_j$. □

7.3 Blurry ECRHs

In Section 7.2 we presented a candidate ECRH based on a generalization of the Knowledge of Exponent assumption in large algebraic groups. In Section 7.4 we are going to introduce a class of knowledge assumptions with a "lattice flavor". We call this class of assumptions *Knowledge of Knapsack.*

We are not able to achieve the strict notion of ECRH from knowledge of knapsack assumptions. Instead, we obtain a "noisy", or *blurry*, notion of ECRH. (This might not be surprising, given that problems about lattices tend to involve statements about noise distributions, rather than about exact algebraic relations as in the case of t-KEA.) This section defines blurry ECRHs and argues why they suffice for our construction of SNARKs.

Definition 7.2. *An efficiently-samplable function ensemble $\mathcal{H} = \{\mathcal{H}_k\}_k$ is a* blurry $(\ell(k), k)$-compressing ECRH *if it is $(\ell(k), k)$-compressing, and for every h in the support of $\mathcal{H}_k$, there exist a "proximity relation" $\overset{h}{\approx}$ over pairs in $\{0,1\}^k \times \{0,1\}^k$, an "extended domain" $D_h \supseteq \{0,1\}^{\ell(k)}$ and an extension $\bar{h} : D_h \to \{0,1\}^k$ consistent with h (i.e., $\forall x \in \{0,1\}^{\ell(k)} : h(x) = \bar{h}(x)$), such that:*

1. *$\mathcal{H}$ is blurry-extractable in the following weakened sense:*

$$\Pr_{h \leftarrow \mathcal{H}_k} \left[\begin{array}{c} y \leftarrow \mathcal{A}(h, z) \\ \exists\, x \in \{0,1\}^{\ell(k)} = y : h(x) \end{array} \wedge \begin{array}{c} x' \leftarrow \mathcal{E}_{\mathcal{A}}^{\mathcal{H}}(h, z) \\ \neg \left(x' \in D_h \ \wedge\ \bar{h}(x') \overset{h}{\approx} y \right) \end{array} \right] < \mathrm{negl}(k) \ .$$

2. *$\mathcal{H}$ is* blurry-collision-resistant *in the following strengthened sense: for any poly-size adversary $\mathcal{A}$,*

$$\Pr_{h \leftarrow \mathcal{H}_k} \left[(x, x') \leftarrow A(h) \ \wedge\ x, x' \in D_h \ \wedge\ x \neq x' \ \wedge\ \bar{h}(x) \overset{h}{\approx} \bar{h}(x') \right] < \mathrm{negl}(k) \ .$$

The notions of blurry-extraction and blurry-collision-resistance are the same as standard extraction and collision-resistance in "strict" case, where $x \overset{h}{\approx} y$ is the identity relation and the domain is not extended ($D_h = \{0,1\}^{\ell(k)}$, $\bar{h} = h$). However, in general, blurry collision resistance is stronger than (standard) collision resistance, because even "near collisions" (i.e., $x \neq y$ such that $\bar{h}(x) \overset{h}{\approx} kh(y)$) must not be efficiently discoverable, not even over the extended domain D_h. Conversely, blurry extraction is weaker than (standard) extraction, since it suffices that the extractor finds a point mapping merely close the the adversary's output (i.e., finds x' such that $\bar{h}(x') \overset{h}{\approx} y$), and it suffices that the point is in the extended domain D_h. Thus, the notion of blurry ECRH captures another, somewhat more flexible tradeoff between the requirements of extractability and collision resistance. We will see that any point on this tradeoff (i.e., any choice of $\overset{h}{\approx}$, D_h and $\bar{h}$ fulfilling the conditions) suffices for the construction of SNARKs.

Why do blurry ECRHs suffice for SNARKs? We argue that the *same construction*, used in the proof of Theorem 1 to construct SNARKs from ECRHs, still obtains SNARKs even when the underlyin hash function is only *blurry* ECRHs.

First, observe that moving from ECRHs to blurry ECRHs only affects the "local consistency" step of our proof (as described in our high-level description in Section 1.2 and then formally as Claim 5.1). Indeed, in the proof based on ECRHs, the local-consistency step is where we employ collision-resistance to claim that the Merkle tree output by the extractor locally agrees with the opened paths (except with negligible probability).

The same argument holds even given only blurry collision resistance: Consider the tree output by the extractor. By the extraction guarantee, it must be that the hash image of every node label that appears in an opened path is "close" to the image of the corresponding node label in the extracted tree. By the blurry collision resistance, however, these two labels must in fact *be the same*; for if they were not, then we could utilize the prover and extractor to finding "blurry collisions". The rest of the proof of Theorem 1 remains unchanged.

7.4 Blurry ECRHs from Knowledge of Knapsack

We define a candidate blurry ECRH family, based on knowledge assumptions of the following form: Given a set of elements $l_1, \ldots, l_t$ in some group, the only way to compute a subset sum is (essentially) to pick a subset $S \subseteq [t]$ and output the subset sum $\sum_{i \in S} l_i$. As before, this is expressed by saying that for any adversary there exists an extractor such that whenever the adversary outputs a value y which happens to be a subset sum, the extractor "explains" this y by outputting a corresponding subset.

For convenience of exposition, we first define in a very general "Knowledge of Knapsack" template, where the set size t, the group, and the distribution of l_i are left as parameters, along with an amplification factor λ (saying how many such subset-sum instances are to be solved simultaneously).

Hashes from knapsacks. A *knapsack* is a tuple $K = (\mathbb{H}, l_1, \ldots, l_t)$, such that $\mathbb{H}$ is (the description of) an additive finite group and $l_1 \ \ldots, l_t \in \mathbb{H}$.

We construct hash function ensembles out of knapsack ensembles in a natural way. Given a size parameter $t = t(k)$, amplification parameter $\lambda = \lambda(k)$, and an ensemble of knapsacks $\mathcal{K} = \{\mathcal{K}_k\}_k$, we define the hash function ensemble $\mathcal{H}^{t,\lambda,\mathcal{K}} = \left\{\mathcal{H}_k^{t,\lambda,\mathcal{K}}\right\}_k$ as follows. For $K = (\mathbb{H}, l_1, \ldots, l_t) \leftarrow \mathcal{K}_k$, let $h^{t,K} : \{0,1\}^t \to \mathbb{H}$ be given by $h^{t,K}(\vec{s}) = \sum_{i:s_i=1} l_i$ represented in $\{0,1\}^{\lceil \log |\mathbb{H}| \rceil}$, where the summation is over $\mathbb{H}$. Then to sample $\mathcal{H}_k^{t,\lambda,\mathcal{K}}$, draw $K^1, \ldots, K^\lambda \leftarrow \mathcal{K}_k$ and output the hash function $h(x) = (h^{t,K^1}(x), \ldots, h^{t,K^\lambda}(x))$. (That is, $h()$ is the λ-wise repetition of $h^{t,K}()$.)

Knowledge of knapsack. The *Knowledge of Knapsack* assumption with respect to $(t, \lambda, \mathcal{K}, \overset{h}{\approx}, D_h)$ asserts that the the function ensemble $\mathcal{H}^{t,\lambda,\mathcal{K}}$ is blurry-extractable with respect to some proximity relation $\overset{h}{\approx}$, some extended domain $D_h \subseteq \mathbb{Z}^t$, and extended function $\bar{h} : D \to \mathbb{H}$ defined by taking a linear combinations with coefficients in D (rather than just subset sums). Explicitly:

Definition 7.3 (Knowledge of Knapsack)**.** *Let $t = t(k) \in \mathbb{N}$ (size parameter) and let $\lambda = \lambda(k) \in \mathbb{N}$ (amplification parameter). Let $\mathcal{K} = \{\mathcal{K}_k\}_k$ be an efficiently-samplable ensemble of knapsacks. For each h in the support of $\mathcal{K}_k$, let $\overset{h}{\approx}$ be a relation on the image of h and let D_h be an extended domain $D_h \subset \mathbb{Z}^t$ where $D_h \supseteq \{0, 1\}$.*

The Knowledge of Knapsack *assumption with respect to $(t, \lambda, \mathcal{K}, \overset{h}{\approx}, D_h)$ states the following: for any poly-size adversary $\mathcal{A}$ there exists a poly-size extractor $\mathcal{E}_\mathcal{A}$ which outputs subsets of $[t]$ such that for all large enough $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

$$\Pr_{(\mathbb{H}^j, l_1^j, \ldots, l_t^j) \leftarrow \mathcal{K}_k}\left[\begin{array}{c} (y^1, \ldots, y^\lambda) \leftarrow \mathcal{A}(K^1, \ldots, K^\lambda, z) \\ \exists\, \vec{x} \in \{0,1\}^t \ \ \forall j : y^j = \sum_i x_i j_i \end{array} \ \wedge \ \begin{array}{c} \vec{x}' \leftarrow \mathcal{E}_\mathcal{A}(K^1, \ldots, K^\lambda, z) \\ \neg\Big(\vec{x}' \in D_h \ \wedge \ \forall j : y^j \overset{h}{\approx} \sum_{i \in [t]} x_i' l_i^j\Big) \end{array}\right] \leq \mathrm{negl}(k)$$

where j ranges over $\{1, \ldots, \lambda\}$, the summations are in the group $\mathbb{H}$, and the multiplication mean adding an (integer number of) elements of $\mathbb{H}$.

Compression. If the groups in all the knapsacks in $\mathcal{K}$ are of size $s = s(k)$ then the function ensemble $\mathcal{H}^{t,\lambda,\mathcal{K}}$ compresses (λt)-bit strings to $(\lambda \log s)$-bit strings.

Discussion: Sparseness and amplification. As discussed in Section 7.1, we wish the candidate ECRH to be superpolynomially sparse. Sparseness grows exponentially with the amplification parameter λ: if each knapsack $K \leftarrow \mathcal{K}_k$ is ρ-sparse (i.e., $|\mathsf{Image}(h^{t,K})|/|\mathbb{H}| < \rho$), then with amplification λ we obtain the candidate ECRH $\mathcal{H}^{t,\lambda,\mathcal{K}}$ that is ρ^λ-sparse. Thus, as long as ρ is upper-bounded by some nontrivial constant, $\lambda > \omega(\log k)$ suffices to get superpolynomial sparseness. We will indeed use this below, in candidates where where the basic knapsacks $\mathcal{K}$ must be just polynomially sparse for the proof of (blurry) collision resistance to go through.

We now proceed to propose instantiations of the Knowledge of Knapsack approach.

7.4.1 Knowledge of Knapsack of Exponents

We first point out that the knowledge of knapsack template can be used to express also the knowledge of exponent assumptions, by considering subset-sums on pairs of the form (f, f^α). The result is similar to the t-KEA assumption (see Section 7.2), albeit with inferior parameters:

Assumption 7.2 (t-KKE)**.** *For $t = t(k) \in \mathbb{N}$, the t-KKE (Knowledge of Knapsack of Exponents) states that there exists an efficiently-samplable ensemble $\mathcal{G} = \{\mathcal{G}_k\}$ where each $(\mathbb{G}, g) \in \mathcal{G}_k$ consists of a multiplicative group of prime order p in $(2^{k-1}, 2^k)$ and a generator $g \in \mathbb{G}$, such that the Knowledge of Knapsack assumption with respect to $(t, 1, \mathcal{K}^\mathsf{E}, \equiv_\mathbb{H}, \{0,1\}^t)$ for the ensemble $\mathcal{K}^\mathsf{E} = \left\{\mathcal{K}_k^\mathsf{E}\right\}_k$ defined as follows (where $\equiv_\mathbb{H}$ is equivalence in the group $\mathbb{H}$ given below):*

To sample from $\mathcal{K}_k^\mathsf{E}$, draw $(\mathbb{G}, g) \leftarrow \mathcal{G}_k$, let $\mathbb{H} = \mathbb{G} \times \mathbb{G}$ considered as an additive group, draw $\alpha \leftarrow \mathbb{Z}_p$ and $\mathbf{r} \leftarrow \mathbb{Z}_p^t$, let $l_i = (g^{r_i}, g^{\alpha r_i}) \in \mathbb{H}$, and output $(\mathbb{H}, l_1, \ldots, l_t)$.

The hash function ensemble $\mathcal{H}^{t,1,\mathcal{K}^\mathsf{E}}$ is readily verified to be $(t(k), 2k)$-compressing, and collision-resistant under CDH. Note that its range is indeed sparse, as prescribed in Section 7.1: for $h \leftarrow \mathcal{H}^{t,1,\mathcal{K}^\mathsf{E}}$,

$|\mathsf{Image}(h)|/|\mathbb{H}| = |\mathbb{G}|/|\mathbb{G}\times\mathbb{G}| \approx 1/2^k$. Alas, we lost a factor of k in the compression compared to directly using t-KEA, since we hash t bits as opposed to t group elements as in t-KEA.

7.4.2 Knowledge of Knapsack of Noisy Multiples

Next we propose a new knowledge assumption based on the following problem: given noisy integer multiples of a secret real number, find a subset-sum of these multiples. The knowledge assumption says (roughly) that whenever an efficient adverary produces such a subset sum, it knows the corresponding subset. This however requires care: taken literally, the assumption is surely false, since any small integer is likely to be a subset sum for *some* subset (if the list of given multiples is long enough), so the adversary could simply output a random small integer. To cope with this and formulate a more plausible assumption, we use the notion of blurry ECRHs as defined above. This approach is inspired by a cryptosystem of Regev [Reg03, Reg04], where the public key is sampled from a similar distribution, and indeed our analysis of collision resistance and sparsity invokes Regev's.

Let $N \in \mathbb{Z}$, $\alpha \in \mathbb{R}$ and $\beta \in (0,1)$. We define the distribution $\mathsf{NM}_{\alpha,\sigma',N}$ of noisy multiples of α in the range $[0,\ldots,N-1)$, with relative noise of standard deviation σ', as follows. Draw an integer $x \overset{U}{\leftarrow} \{0,\ldots,\lfloor N/\alpha\rfloor\}$ and a noise fraction $y \leftarrow \mathcal{N}_{0,\sigma'^2}$ (the normal distribution with mean 0 and variance σ'^2). Output $\lfloor \alpha(x+y) \bmod N\rfloor$.

Assumption 7.3 ((t,σ)-KKNM). *For $t = t(k) > k \in \mathbb{N}$ and noise parameter $\sigma = \sigma(k) \in (0,1)$, the (t,σ)-KKNM (Knowledge of Knapsack of Noisy Multiples) states that the Knowledge of Knapsack assumption with respect to $(t, \mathcal{K}^{\mathsf{NM},t,\sigma}, \overset{h}{\approx}, D_h)$ holds for the following distribution of knapsack elements.*

The ensemble $\mathcal{K}^{\mathsf{NM},t,\sigma} = \left\{\mathcal{K}_k^{\mathsf{NM},t,\sigma}\right\}_k$ is sampled as follows. To sample from $\mathcal{K}_k^{\mathsf{NM},t,\sigma}$: let $N = 2^{8k^2}$, draw $h \overset{U}{\leftarrow} \left\{h \in [\,\sqrt{N}, 2\sqrt{N}\,) : |h - \lfloor h\rceil| < \frac{1}{16t}\right\}$ and draw σ' such that $\sigma'^2 \overset{U}{\leftarrow} [\,\sigma^2, 2\sigma^2\,)$. Let $\alpha = N/h$. Draw t values $l_1,\ldots,l_t \leftarrow \mathsf{NM}_{\alpha,\beta,N}$. Output $(\mathbb{Z}_N, l_1,\ldots,l_t)$.

For $h \leftarrow \mathcal{K}_k^{\mathsf{NM},t,\sigma}$, let $D_h = \left\{\vec{x} \in \mathbb{Z}^t : ||\vec{x}||_2 < t\log^2 t\right\}$, and let $\overset{h}{\approx}$ be s.t. for $y,y' \in \mathbb{Z}_N$, $y \overset{h}{\approx} y'$ if their distance in $\mathbb{Z}_N$ is at most $1/9\sqrt{N}$.

Relation to Regev's cryptosystem [Reg03, Reg04]. The above distributions are essentially the same as in Regev's cryptosystem, with minor changes for clarity in the present context. Explictly, the mapping is as follows. The distribution $Q_\beta = (\mathcal{N}_{0,\beta/2\pi} \bmod 1)$ from [Reg04, Section 2.1] is replaced by $\mathcal{N}_{0,\sigma'^2}$, for $\beta = 2\pi\sigma'^2$ (the statistical difference is negligible because σ' will be polynomially small). The distribution $\mathsf{NM}_{\alpha,\sigma',N}$ is a scaling up by N of $T_{h,\beta}$ as defined in [Reg04, above Definition 4.3], for $h = N/d$ (except for the above deviation, and a deviation due to the event $x+y > h$ which is also negligible in our setting). Thus, the distribution $(l_1,\ldots,l_t)$ sampled by $\mathcal{K}_{\mathsf{NM}}$ is negligibly close to that of public keys in [Reg04, Section 5] on parameters $n = k$, $m = t$, $\gamma(n) = \sqrt{2/\pi}\,/\,\sigma(k)$.

Collision resistance. We show that the hash function ensemble $\mathcal{H}^{\mathsf{KKNM}} = \mathcal{H}^{t,\lambda,\mathcal{K}^{\mathsf{NM},t,\sigma}}$ is blurry-collision-resistant for any $t = O(k^2)$, based on the $uSVP$ problem. We do so adapting results from [Reg03, Reg04]. It suffices to consider the case $\lambda = 1$ (no amplification), since $\lambda > 1$ follows by looking at just $j = 1$.

Claim 7.2. *The samples $l_1,\ldots,l_t$ drawn by $\mathcal{K}^{\mathsf{NM},t,\sigma}$ are pseudorandom (i.e., indistinguishable from t random integers in $\{0,\ldots N-1\}$), assuming hardness of $(\sqrt{2/\pi k}/\sigma(k))$-uSVP.*

Proof sketch. It suffices to show pseudorandomness for the distribution obtained by modifying $\mathcal{K}^{\mathsf{NM},t,\sigma}$ to sample $h \overset{U}{\leftarrow} [\,\sqrt{N}, 2\sqrt{N}\,)$ (for the same reason as in [Reg04, Lemma 5.4]). This pseudorandomness follows from [Reg04, Theorem 4.5] with $g(n) = \sqrt{2k/\pi}/\sigma(k)$. $\square$

Claim 7.3. *The function ensemble $\mathcal{H}^{\mathsf{KKNM}}$ is blurry-collision-resistant, with $\overset{h}{\approx}, D_h, \bar{h}$ defined as in Assumption 7.3, assuming hardness of $\tilde{O}(k^{3/2})$-uSVP, when $\sigma = \tilde{\Omega}(1/k)$ and $t = O(k^2)$.*

Proof sketch. By Claim 7.2, the hash functions drawn by $\mathcal{H}^{\mathsf{KKNM}}$ are indistinguishable from the the ensemble $\mathcal{U}$ of uniformly-random modular subset sums (as defined in [Reg04, Section 6]). It thus suffices to show that $\mathcal{U}$ is collision-resistent, since this implies finding collisions in $\mathcal{H}^{\mathsf{KKNM}}$ would distinguish it from $\mathcal{U}$. The ensemble $\mathcal{U}$ is indeed is collision-resistant assuming $\tilde{O}(k^{3/2})$-uSVP, by [Reg04, Theorem 6.5]. Moreover, the theorem still holds if in its statement, $\sum_{i=1}^{m} b_i a_i \equiv 0 (\mathrm{mod} N)$ is generalized to $\mathsf{frc}\left((\sum_{i=1}^{m} b_i a_i)/N\right) < 1/9\sqrt{N}$. Inside that theorem's proof, this implies $\mathsf{frc}\left((\sum_{i=1}^{m} b_i z_i)/N\right) < 1/8\sqrt{N}$ and thus, in the one-but-last displayed equation, $h \cdot \mathsf{frc}\left((\sum_{i=1}^{m} b_i z_i)/N\right) < h/9\sqrt{N} < 1/9$ so the last displayed equation still holds and the proof follows. Also note that Regev's bound $||b|| \le \sqrt{m}$ (in his notation) generalizes to $||b|| \le \tilde{O}(\sqrt{m})$. □

Relation to other lattice hardness assumptions. The collision-resistence is shown assuming hardness of the uSVP lattice problem. This can be generically translated to other (more common) lattice hardness assumptions following Lyubashevsky and Micciancio [LM09].

Sparseness and parameter choice. To make the extractability assumption plausible, we want the function's image to be superpolynomially sparse within its range, as discussed in Section 7.1. Consider first the distribution $\mathcal{H}^{\mathsf{KKNM}} = \mathcal{H}^{t,1,\mathcal{K}^{\mathsf{NM},t,\sigma}}$ (i.e., $\lambda = 1$, meaning no amplification). The image of h drawn from $\mathcal{H}^{\mathsf{KKNM}}$ becomes "wavy" (hence sparse) when the the noise (of magnitude $\sigma\alpha$) added to each multiple of α is sufficiently small, resulting in distinct peaks, so that any subset sum of t noisy multiples is still a noisy multiple:

Claim 7.4. *For $\sigma(k) = 1/16t \log^2 k$, the ensemble $\mathcal{K}^{\mathsf{NM},t,\sigma}$ is $\frac{1}{2}$-sparse:*

$$\Pr_{h \leftarrow \mathcal{H}_k^{t,1,\mathcal{K}^{\mathsf{NM},t,\sigma}}} \left[|\mathsf{Image}(h)|/N > 1/2 \right] < \mathrm{negl}(k)$$

Proof sketch. In terms of the corresponding Regev public key, this means decryption failure become impossible with all-except-negligible probability over the keys. For this, it clearly suffices that each of the t noisy multiples is at most $\alpha/16t$ away from a multiple of α, so that any sum of them will have accumulated noise at most $\alpha/16$ (plus another $\alpha/16$ term due to modular reductions, as in Regev's decryption lemma [Reg04, Lemma 5.2]). This indeed holds for $\sigma(k) = 1/16t \log^2 k$, by a tail bound on the noise terms $\alpha\mathcal{N}_{0,\sigma}$ followed by a union bound over the t samples. □

Thus, the image becomes somewhat sparse when $\sigma = \tilde{o}(1/t)$. However, superpolynomial sparseness would require superpolynomially small σ (and also a tighter distribution over h), for which the lattice hardness assumption of Claim 7.3 no longer holds. This is solved by amplification via repetition, i.e., choosing $\lambda > 1$. By setting $\sigma(k)$ to $\tilde{o}(1/t)$ and $\lambda = \omega(\log(k))$, we indeed obtain superpolynomial sparseness.

Another concern is that the adversary $\mathcal{A}$ may indeed compute and output a sum of the l_i, but one whose coefficients are small integers other than $\{0, 1\}$ (for example, output $y = 2l_1$). In this case, the result is still close to a noisy multiple of α and thus likely to be be in the image of h, but we cannot expect $\mathcal{E}$ to extract a subset-sum (with $\{0, 1\}$ coefficients) matching y. The extended domain D_h addresses this by allowing $\mathcal{E}$ to explain y via any vector using a linear combination whose coefficients have ℓ_2 norm at most $t \log^2 t$. Beyond this norm, the linear combination is unlikely to be in the image of h. It remains to observe that this extension preserves collision resistance.

Lastly, note that $k = n^2$ (or, indeed, any $k = n^{1+\epsilon}$) suffices for the SNARK construction.

7.4.3 Knowledge of Knapsack of Noisy Inner Products

Further ECRH candidates can be obtained from Knowledge of Knapsack problems on other lattice-based problems. In particular, the Learning with Error problem [Reg05] problem leads to a natural knapsack ensemble, sampled by drawing a random vector $\vec{s} \in \mathbb{Z}_p^n$ and then outputting a knapsack $K = (\mathbb{Z}_p^{n+1}, l_1, ..., l_t)$ where each l_i consists of a random vector $\vec{x} \xleftarrow{U} \mathbb{Z}_p^n$ along with the inner product $\vec{s} \cdot \vec{x} + \epsilon$ where ϵ is independently-drawn noise of small magnitude in $\mathbb{Z}_p$. For suitable parameters this ensemble is sparse, and blurry-collision-resistant following an approach similar to KKNM above: first show pseudorandomness assuming hardness of LWE [Reg05], and then rely on the collision-resistance of the uniform case (e.g., [MR07]).

In this case, amplification can be done more directly, by reusing the same $\vec{x}$ with multiple s_i instead of using the generic amplification of Definition 7.3.

8 Application: Delegation of Computation (Including Long Inputs)

Recall that, in a *two-message delegation scheme* (in the plain model): to delegate a T-time function F on input x, the delegator sends a message σ to the worker; the worker computes an answer (z, π) to send back to the delegator; the delegator outputs z if π is a convincing proof of the statement "$z = F(x)$". The delegator and worker time complexity are respectively bounded by $p(|F| + |x| + |F(x)| + \log T)$ and $p(|F| + |x| + |F(x)| + T)$, where p is a universal polynomial (not depending on the specific function being delegated).

(Throughout this section we ignore the requirement for input privacy because it can always be achieved by adding a semantically-secure fully-homomorphic encryption scheme.)

Folklore delegation from succinct arguments. There is a natural method to obtain a two-message delegation scheme from a designated-verifier non-interactive succinct argument for NP with *adaptive soundness*: the delegator sends the desired input x and function F to the worker, and asks him to prove that he evaluated the claimed output z correctly.

We remark that "for NP" above indicates that it is enough for there to exist a protocol specialized for each NP relation, but the succinctness requirement is still required to be universal, as captured by our Definition 4.2.

We also note that designated-verifier non-interactive succinct arguments for NP with adaptive soundness are of the "right" strength. For example, if we had *publicly-verifiable* non-interactive succinct arguments for NP with adaptive soundness, we would only gain public verifiability, which is rarely interesting once one adds fully-homomorphic encryption to further ensure input privacy. Thus, starting with designated-verifier non-interactive succinct arguments is usually "enough" for delegation of computation.

Furthermore, we would like to emphasize that the use of succinct arguments, in a sense, provides the "best" properties that one could hope for in a delegation scheme:

- There is no need for preprocessing and no need to assume that the verifier's answers remain secret. All existing work providing two-message *generic* delegation schemes are in the preprocessing setting and assume that the verifier's answers remain secret [KR06, Mie08, GKR08, KR09, GGP10, CKV10]. (Notable exceptions are the works of Benabbas et al. [BGV11] and Papamanthou et al. [PTT11], which however only deal with delegation of specific functionalities, such as polynomial functions or set operations.)

- The delegation scheme can also support "private inputs" of the worker: one can delegate functions $F(x, x')$ where x is supplied in the first message by the delegator, and x' is supplied by the worker. This extension is known as *non-interactive secure computation* [IKO$^+$11].

Our instantiation. Our main technical result, Theorem 1, provides an instantiation, based on a simple and generic knowledge assumption, of the designated-verifier non-interactive succinct argument for NP with adaptive soundness required for construction a two-message delegation scheme.

Corollary 8.1. *Assume that there exists extractable collision-resistant hash functions. Then there exists a two-message delegation scheme.*

We note that previous two-message arguments for NP [DCL08, Mie08] did not provide strong enough notions of succinctness or soundness to suffice for constructing delegation schemes.

Our specific instantiation also has additional "bonuses":

- Not only is the delegation sound, *but also has a proof of knowledge*. Therefore, $F(x, x')$ could involve cryptographic computations, which would still be meaningful because the delegator would know that a "good" input x' can be found in efficient time.

  For example, $F(x, x')$ could first verify whether the hash of a long x' is x and, if so, proceed to conduct an expensive computation; if the delegation where merely sound, the delegator would not be able to delegate such a computation, for such a x' certainly exists!

  Large inputs. As another example that is possible because of the proof of knowledge, we can enable the verifier to handle "large inputs". Indeed, if we are interested in evaluating many functions on a large input x, we could first in an offline stage compute a Merkle hash c_x for x and then communicate x to the worker; later, in order to be convinced of $z = F(x)$, we simply ask the worker to prove us the there is some $\tilde{x}$ such that the Merkle hash of $\tilde{x}$ is c_x and, moreover, $z = F(\tilde{x})$. By the collision resistance of the Merkle hash, we believe that indeed $z = F(x)$. Note that the Merkle hash of c_x can be computed in a streaming fashion, and x, which now "lies in the worker's untrusted memory", can be easily updated by letting the worker compute the updated x and prove that the new Merkle hash is a "good" one. In this way, we are able to give simple *two-message* constructions for both the tasks of *memory delegation* and *streaming delegation* [CTY10, CKLR11] — again getting the "best" that can hope for here.

  Of course, special cases of delegating "large datasets" such as [BGV11] and [PTT11] are also implied by our instantiation. (Though, in the case of [PTT11], we only get a designated-verifier variant.) While our construction is definitely not as practically efficient, it provides the only other construction that with two messages (i.e., is "non-interactive").

- Even if the construction from our Theorem 1 formally requires the argument to depend on a constant $c \in \mathbb{N}$ bounding the time to verify the theorem, the only real dependence is a simple verification by the verifier (i.e., checking that $t \leq |x|^c$), and thus in the setting of delegation of computation we obtain a *single* protocol because the delegator gets to choose c. Of course, despite the dependence on c, our construction still delivers the "universal succinctness" (as already remarked in Section 5.3, satisfying our Definition 4.2) required at the beginning of this section. (Indeed, if the polynomial bounding the verifier time complexity is allowed to depend on the function being delegated, then a trivial solution is to just let the verifier compute the function himself, as the function is assumed to be polynomial-time computable!)

- When our construction is instantiated with the quasilinear-time and quasilinear-length PCPs of Ben-Sasson et al. [BSS08, BSGH$^+$05] we get essentially optimal efficiency (up to polylogarithmic factors):
  - the delegator's first message requires time complexity $\mathrm{poly}(k, \log T)\tilde{O}(|F| + |x|)$;
  - the worker's computation requires time complexity $\mathrm{poly}(k)\tilde{O}(|F| + |x| + |F(x)| + T)$; and
  - the delegator's verification time requires time complexity $\mathrm{poly}(k, \log T)\tilde{O}(|F| + |x| + |F(x)|)$.

Are knowledge assumptions needed? If one insists on non-interactive secure computation for all of NP, then a knowledge assumption is in part justified: the impossibility result of Gentry and Wichs [GW11] implies that there is no proof of security via any black-box reduction to a falsifiable assumption. (Note that, adaptivity in the soundness is indeed needed, because the prover does get to choose the output, and thus the theorem that is being proved!)

However, non-interactive secure computation may still be possible without knowledge assumptions for "natural" NP languages. (Recall that the result of [GW11] involves constructing an "unnatural" NP language.) Moreover, for delegation schemes, where no input from the delegator is supported, it suffices to capture languages in P, because in such a case no witness is needed. In both of these cases, we are not aware of any evidence suggesting that a knowledge assumption may be needed.

Acknowledgements

Nir and Ran wish to thank Ben Riva and Omer Paneth for enlightening discussions in the early stages of this research. Eran wishes to thank Shai Halevi for early discussions about using extractable collision resistance as a solution approach, and Daniele Micciancio for a discussion of lattice-based Knowledge of Knapsacks assumptions.

References

[ABOR00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, ICALP '00, pages 463–474, 2000.

[AF07] Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In *TCC '07: Proceedings of the 4th Theory of Cryptography Conference on Theory of Cryptography*, pages 118–136, 2007. Online at `http://eprint.iacr.org/2006/423`.

[AIK10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: efficient verification via secure computation. In *ICALP '10: Proceedings of the 37th International Colloquium on Automata, Languages, and Programming*, pages 152–163, 2010.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BG08] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2008. Preliminary version appeared in CCC '02. Latest version available at `http://www.wisdom.weizmann.ac.il/~oded/PS/ua-rev3.ps`.

[BGV11] Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In *CRYPTO '11: Proceedings of the 31st Annual International Cryptology Conference on Advances in Cryptology*, pages ???–???, 2011. `http://eprint.iacr.org/2011/132`.

[BHK11] Mark Braverman, Avinatan Hassidim, and Yael Tauman Kalai. Leaky pseudo-entropy functions. In *ICS*, 2011.

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[Blu81] Manuel Blum. Coin flipping by telephone. In *CRYPTO '81: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 11–15, 1981.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *CRYPTO '04: Proceedings of the 24th Annual International Cryptology Conference on Advances in Cryptology*, pages 273–289, 2004.

[BSGH+05] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Short PCPs verifiable in polylogarithmic time. In *CCC '05: Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 120–134, 2005.

[BSS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM Journal on Computing*, 38(2):551–607, 2008.

[BSW11] Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: Homomorphic encryption for restricted computations. Cryptology ePrint Archive, Report 2011/311, 2011. Online at `http://eprint.iacr.org/`.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. FOCS, 2011. Available at `http://eprint.iacr.org/2011/344`.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In *ICALP '08: Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II*, pages 449–460, 2008.

[CD09] Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In *TCC '09: Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, pages 595–613, 2009.

[CKLR11] Kai-Min Chung, Yael Kalai, Feng-Hao Liu, and Ran Raz. Memory delegation. In *CRYPTO*, 2011.

[CKV10] Kai-Min Chung, Yael Kalai, and Salil Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO '10: Proceedings of the 30th Annual International Cryptology Conference on Advances in Cryptology*, pages 483–501, 2010. Full version online at `http://eprint.iacr.org/2010/241`.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT '99: Proceedings of the 18th Annual International Conference on Advances in Cryptology*, pages 402–414, 1999.

[CT10] Alessandro Chiesa and Eran Tromer. Proof-carrying data and hearsay arguments from signature cards. In *ICS '10: Proceedings of the 1st Symposium on Innovations in Computer Science*, pages 310–331, 2010.

[CTY10] Graham Cormode, Justin Thaler, and Ke Yi. Verifying computations with streaming interactive proofs. ECCC, 2010. Available at `http://eccc.hpi-web.de/report/2010/159/`.

[Dak09] Ronny Ramzi Dakdouk. *Theory and Application of Extractable Functions*. PhD thesis, Yale University, Computer Science Department, December 2009.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 445–456, 1992.

[DCL08] Giovanni Di Crescenzo and Helger Lipmaa. Succinct NP proofs from an extractability assumption. In *CiE '08: Logic and Theory of Algorithms, 4th Conference on Computability in Europe*, pages 175–185, 2008.

[Den06] Alexander W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006. Online at `http://eprint.iacr.org/2006/156`.

[DG06] Alexander Dent and Steven Galbraith. Hidden pairings and trapdoor DDH groups. In Florian Hess, Sebastian Pauli, and Michael Pohst, editors, *Algorithmic Number Theory*, volume 4076 of *Lecture Notes in Computer Science*, pages 436–451. Springer Berlin / Heidelberg, 2006.

[DK08] Stefan Dziembowski and Pietrzak Krzysztof. Leakage-resilient cryptography. In *FOCS*, pages 293–302, 2008.

[DLN+04] Cynthia Dwork, Michael Langberg, Moni Naor, Kobbi Nissim, and Omer Reingold. Succinct NP proofs and spooky interactions, December 2004. Available at `www.openu.ac.il/home/mikel/papers/spooky.ps`.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *CRYPTO '86: Proceedings of the 6th Annual International Cryptology Conference on Advances in Cryptology*, pages 186–194, 1987.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 169–178, 2009.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: outsourcing computation to untrusted workers. In *CRYPTO '10: Proceedings of the 30th Annual International Cryptology Conference on Advances in Cryptology*, pages 465–482, 2010. Online at `http://eprint.iacr.org/2009/547`.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for Muggles. In *STOC '08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 113–122, 2008.

[GKR10] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. In *ACNS '10: Proceedings of the 8th International Conference on Applied Cryptography and Network Security*, pages 309–328, 2010.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version appeared in STOC '85.

[GR05] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In *ICALP '05: In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 803–815, 2005.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT '10: Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security*, pages 321–340, 2010.

[GVW02] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1/2):1–53, 2002.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC '11: Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, 2011. Available at `http://eprint.iacr.org/2010/610`.

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold. Efficient pseudorandom generators from exponentially hard one-way functions. In *ICALP*, pages 228–239, 2006.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, pages 408–423, 1998.

[IKO$^+$11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In *EUROCRYPT '11: Proceedings of the 30th Annual International Conference on Advances in Cryptology*, pages 406–425, 2011.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC '92: Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 723–732, 1992.

[KR06] Yael Tauman Kalai and Ran Raz. Succinct non-interactive zero-knowledge proofs with preprocessing for LOGSNP. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 355–366, 2006.

[KR09] Yael Tauman Kalai and Ran Raz. Probabilistically checkable arguments. In *CRYPTO '09: Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, pages 143–159, 2009.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*, pages 577–594, 2009.

[Mer89] Ralph C. Merkle. A certified digital signature. In *CRYPTO '89: Proceedings of the 9th Annual International Cryptology Conference*, pages 218–238, 1989.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version appeared in FOCS '94.

[Mie08] Thilo Mie. Polylogarithmic two-round argument systems. *Journal of Mathematical Cryptology*, 2(4):343–363, 2008.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37:267–302, April 2007.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO '03: Proceedings of the 23rd Annual International Cryptology Conference on Advances in Cryptology*, pages 96–109, 2003.

[Nec94] Vassiliy Ilyich Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55:165–172, 1994.

[OT89] Eiji Okamoto and Kazue Tanaka. Key distribution system based on identification information. *Selected Areas in Communications, IEEE Journal on*, 7(4):481–485, May 1989.

[PTT11] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Optimal verification of operations on dynamic sets. In *CRYPTO*, 2011.

[PX09] Manoj Prabhakaran and Rui Xue. Statistically hiding sets. In *CT-RSA '09: Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, pages 100–116, 2009.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *STOC*, pages 407–416, 2003.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, 2004.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 84–93, 2005.

[RR94] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:204–213, 1994.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, pages 76–85, 2008.

[Sha92] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT '97: Proceedings of the 16th Annual International Conference on Advances in Cryptology*, pages 256–266, 1997.

[Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *TCC '08: Proceedings of the 5th Theory of Cryptography Conference on Theory of Cryptography*, pages 1–18, 2008.

[Wee05] Hoeteck Wee. On round-efficient argument systems. In *ICALP '05: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 140–152, 2005.