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

Nir Bitansky
Tel Aviv University
nirbitan@tau.ac.il

Ran Canetti
Boston University
Tel Aviv University
canetti@tau.ac.il

Alessandro Chiesa
MIT CSAIL
alexch@csail.mit.edu

Eran Tromer
Tel Aviv University
tromer@tau.ac.il

ABSTRACT

The existence of succinct non-interactive arguments for NP (i.e., non-interactive computationally-sound proofs where the verifier's work is essentially independent of the complexity of the NP nondeterministic verifier) has been an intriguing question for the past two decades. Other than CS proofs in the random oracle model [Micali, FOCS '94], the only existing candidate construction is based on an elaborate assumption that is tailored to a specific protocol [Di Crescenzo and Lipmaa, CiE '08].

We formulate a general and relatively natural notion of an *extractable collision-resistant hash function (ECRH)* and show that, if ECRHs exist, then a modified version of Di Crescenzo and Lipmaa's protocol is a succinct non-interactive argument for NP. Furthermore, the modified protocol is actually a succinct non-interactive *adaptive argument of knowledge (SNARK).* We then propose several candidate constructions for ECRHs and relaxations thereof.

We demonstrate the applicability of SNARKs to various forms of delegation of computation, to succinct non-interactive zero knowledge arguments, and to succinct two-party secure computation. Finally, we show that SNARKs essentially imply the existence of ECRHs, thus demonstrating the necessity of the assumption.

*This research was supported by the Check Point Institute for Information Security, by the Israeli Centers of Research Excellence (I-CORE) program (center No. 4/11), by the European Community's Seventh Framework Programme (FP7/2007-2013) grant 240258, by a European Union Marie Curie grant, and by a grant from the Israeli Science Foundation.

ITCS '12 Cambridge, Massachusetts USA

1. INTRODUCTION

For the Snark's a peculiar creature, that won't
Be caught in a commonplace way.
Do all that you know, and try all that you don't:
Not a chance must be wasted to-day!

The Hunting of the Snark,
Lewis Carroll

The notion of interactive proof systems [44] 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 $\mathsf{IP} = \mathsf{PSPACE}$ [66]. 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 by direct checking of an NP witness.

We know that if statistical soundness is required then any non-trivial savings would cause unlikely complexity-theoretic collapses (see, e.g., [13, 40, 41, 69]). However, if we settle for proof systems with only computational soundness (also known as interactive arguments [15]) then significant savings can be made. Indeed, using collision-resistant hash functions (CRHs), Kilian [51] shows a four-message interactive argument for NP: the prover first uses a Merkle hash tree to bind itself to a poly-size PCP (Probabilistically Checkable Proof) string for the statement, and then answers the PCP verifier's queries while demonstrating consistency with the Merkle tree. This way, membership of an instance y in an NP language L can be verified in time that is bounded by $p(k, |y|, \log t)$, where t is the time to evaluate the NP verification relation for L on input y, p is a fixed polynomial independent of L, and k is a security parameter that determines the soundness error. Following tradition, we call such argument systems *succinct*.

Can we have succinct argument systems which are *non-interactive*? Having posed and motivated this question, Micali [54] provides a one-message succinct non-interactive argument for NP, in the random oracle model, by applying the Fiat-Shamir paradigm [32] to Kilian's protocol. It is easy to see that, in the standard model, one-message succinct arguments do not exist except for "quasi-trivial" languages (i.e., languages in $\mathsf{BPtime}(n^{\mathrm{polylog}\, n})$). However, known impossibility results leave open the possibility of succinct non interactive arguments in a slightly more relaxed model, where the verifier (or a trusted entity) sends ahead of time a succinct string, that is independent of the statements to be proven later.

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

Attempted solutions. To answer the above question, Aiello et al. [2] 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. [30] 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 [28] propose to address this problem by further requiring the prover to bind itself (in the clear) to a specific database using a Merkle Tree (MT) 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 four-message protocol down to two 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 under some well defined properties of the Merkle tree hash. Indeed, to prove soundness of their protocol Di Crescenzo and Lipmaa use an assumption that is non-standard in two main ways: first, it is a "knowledge assumption," in the sense that any adversary that generates a value of a certain form is assumed to "know" a corresponding preimage (see more discussion on such assumptions below). Furthermore, their assumption is very specific and intimately tied to the actual hash, PIR, and PCP schemes in use, as well as the language under consideration. Two other non-interactive arguments for NP, based on more concise knowledge assumptions, are due to Mie [56] and Groth [45]. However, neither of these protocols is succinct. Specifically, in both protocols the verifier's runtime is polynomially related to the time needed to directly verify the NP witness.

Recently, Gentry and Wichs [38] showed that some of the difficulty is indeed inherent: no non-interactive succinct argument can be proven to be adaptively-sound via a black-box reduction to a falsifiable assumption (as defined in [57]). This holds even for *designated-verifier* protocols, where the verifier needs secret randomness in order to verify. This suggests that non-standard assumptions, such as the knowledge (extractability) assumptions described next, may be inherent.

Knowledge (extractability) assumptions. Knowledge (or 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. Such an assumption asserts that, for any efficient algorithm that achieves the task, there exists a *knowledge extractor* algorithm that efficiently recovers the said intermediate values.

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 [24]). It is indeed hard to gain assurance regarding their relative strengths. Abstracting from such specific assumptions, one can formulate general notions of extractability for one-way functions and other basic primitives (see [19, 23]). That is, say that a function family $\mathcal{F}$ 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 $f(x) = y$. This is expressed by saying that for any efficient adversary $\mathcal{A}$ there is an efficient extractor $\mathcal{E}_{\mathcal{A}}$ such that, if $\mathcal{A}(f) = f(x)$ for some x, then $\mathcal{E}_{\mathcal{A}}(f)$ almost always outputs x' such that $f(x') = f(x)$. Typically, for such a family to be interesting, $\mathcal{F}$ should also have some sort of hardness property, e.g., one-wayness. Assuming that a certain function family is extractable does not typically fit in he mold of efficiently falsifiable assumptions [57]. In particular, the impossibility result of Gentry and Wichs [38] does not apply to such assumptions.

Delegation of computation and adaptive arguments of knowledge. Before proceeding to describe our results, let us make a quick detour to describe a prominent applications of succinct non-interactive arguments. This application, 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. Current delegation schemes, such as [49, 42, 50, 34, 22], require either more than two messages or much work to be done by the verifier in a preprocessing stage. A succinct non-interactive argument system (SNARG) for NP could improve on these by minimizing both interaction and the verifier's computational effort.

However, the application to delegation schemes brings with it two additional security concerns that do not naturally come up in the simplistic model described above. 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). This is so since a cheating worker might choose a bogus result of the computation based on the delegators first message. 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*). Indeed, 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 to a computation — e.g., when the computation is done over a database that is stored by the worker.[1]

Another application where proofs of knowledge are crucial is *proof composition*, which is a technique that has already been shown to enable desirable cryptographic tasks [68, 21, 12].

1.1 Summary of Our Results

We revisit the PCP+MT+PIR approach of [28], and show that it can be modified to achieve soundness based on a general and relatively-natural extractability assumption about the hash function used to construct the Merkle tree. More precisely:

- We formulate a notion of *extractable collision-resistant hash functions* (ECRHs). We then show that if ECRHs exist then

[1] For example, the untrusted worker may store a long database z whose short Merkle hash $h = \mathrm{MT}(z)$ is known to the delegator; the delegator may then ask the worker to compute $F(z)$ for some function F. However, from the delegator's perspective, merely being convinced that "there exists $\tilde{z}$ such that $h = \mathrm{MT}(\tilde{z})$ and $F(\tilde{z}) = f$" is not enough. The delegator should also be convinced that the worker knows such a $\tilde{z}$, which implies due to collision resistance of MT that indeed $\tilde{z} = z$.

the modified construction is a SNARG for NP. Furthermore, we show that (a) this construction is a proof of knowledge, and (b) it remains secure against adaptively chosen instances. We call such systems *SNARGs of knowledge* (SNARKs).

- We describe some applications of SNARKs: (a) to delegation of computations (even with long delegator input and with worker input), (b) to constructing *zero-knowledge* SNARKs, and (c) for constructing succinct non-interactive secure computation (in the common reference string model).
- We propose candidate ECRH constructionss. One is based on a Knowledge of Exponent assumption and the hardness of taking discrete logs. Two others are based on knapsack (subset-sum) problems related to hard problems on lattices. Yet others can be based on unstructured hash functions such as SHA-2. (All constructions other than the first obtain slightly weaker notions than full fledged ECRH; however, we show that these weaker notions suffice.)
- We show that existence of SNARKs for NP implies existence of (the weaker variants of) ECRHs, as well as extractable variants of some other cryptographic primitives. This provides further evidence that ECRHs are necessary for the existence of SNARKs.

The rest of the introduction describes these results in more detail. This extended abstract does not contain the full details of all of our results; see the full version of this paper [10].

1.2 ECRHs, SNARKs, and Applications

(i) Extractable collision-resistant hash functions. We start by defining a natural strengthening of 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 following sense:

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 $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

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

We do not require that there is an efficient way to tell whether a given string in $\{0,1\}^k$ is in the image of a given $h \in \mathcal{H}_k$. We note that:

- For extractability and collision resistance (or one-wayness) to coexist, it should be hard to "obliviously sample images"; that is, the image of almost any $h \in \mathcal{H}_k$ should be sparse in $\{0,1\}^k$, i.e., with cardinality at most $2^{k-\omega(\log k)}$. (This remark is a bit over-simplified and not entirely accurate; see discussion in Section 6.1.)
- 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 6.1.

(ii) From ECRHs to adaptive succinct arguments of knowledge, and back again. We modify the "PCP+MT+PIR" construction of [28] 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 and thus we refer to this message 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) The construction is an (adaptive) argument of knowledge; (d) The running time of the verifier and the proof length are "universally succinct"; in particular, they 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:

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

We also note that a single VGRS in our construction suffices for only logarithmically many proofs; however, since the VGRS is succinct and easy to generate, the cost of occasionally resending a new one is limited.

Throughout, in order to obtain "full adaptivity" (i.e., there is no concrete upper bound on the size of theorems supported, though there may be an asymptotic one such as when considering a specific NP language) we require that the PIR in use supports random queries with respect to an a-priori unknown database size (and this is what we meant by "appropriate" in the theorem statement); any FHE-based PIR (e.g., [14]) 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 or secure computation) the requirement can be removed altogether. Section 1.4 holds a sketch of the proof. The full proof appears in Section 5.

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.*

More accurately, we show that SNARKs and CRHs imply a slightly relaxed notion of ECRHs that we call *proximity ECRHs*, and which is still sufficient for our construction of SNARKs. To simplify the exposition of our main results we defer the discussion of the details of this relaxation to Section 1.3.

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

THEOREM 3 (INFORMAL). *Assume SNARKs and (standard) CRHs exist. Then there exist extractable one-way functions and extractable computationally hiding and binding commitments. Alternatively, if there exist SNARKs and (standard) subexponentially-hard one-way functions then there exist extractable one-way functions. Furthermore, if these functions are one-to-one, then we can 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 [19]). 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 [31, 65, 38] and leakage-resilient pseudo-entropy functions [16].

(iii) Applications of SNARKs. As discussed earlier, SNARKs provide a means for non-interactive delegation of computation, which also extends to the cases where the delegator has a very long input or where the worker supplies his own input to the computation. An important property of SNARK-based delegation is that it does not require expensive preprocessing and (as a result) soundness can be maintained even when the prover learns the verifier's responses between subsequent delegation sessions because a fresh VGRS can simply be resent for each time.

In addition, SNARKs can be used to obtain zkSNARKs, that is, *zero-knowledge* succinct non-interactive arguments of knowledge in the common reference string (CRS) model. (In fact, we provide two constructions depending on whether the succinct argument is "on top or below" the NIZK.)

In an additional step, it is possible to obtain succinct non-interactive two party secure computation against malicious adversaries, where the amount of work done by the receiver is independent of the complexity of the evaluated function and the sender's input. To do so, start with a non-interactive two-party computation protocol that is secure against "almost" honest-but-curious adversaries, who are allowed to choose arbitrary randomness; such protocols are known to exist, e.g., based on fully-homomorphic encryption [36]. Then, to make the protocol resilient to malicious adversaries, let the sender attach to his message a zkSNARK that his computation was performed honestly. The succinct receiver can use a standard NIZK to prove knowledge of his inputs. As for the sender, if his input is short he can also prove knowledge using a standard NIZK; otherwise, he must rely on the zkSNARK proof of knowledge. In particular, in the latter case, we only get non-concurrent security (rather than UC security) as the simulation relies on the non-black-box SNARK extractor.

In summary, SNARKs can be used for a number of applications:

COROLLARY 1.1 (INFORMAL). *If there exist SNARKs, then:*

1. *There exist two-message delegation schemes where the verifier's response need not remain secret.*

   (Furthermore, there are such schemes that allow the worker to contribute it own input, as well as ones allowing to delegate memory and data streams.)

2. *In the CRS model, there exist zero-knowledge SNARKs.*

3. *In the CRS model, there exist succinct non-interactive secure computation schemes (with UC security for short sender input and non-concurrent security for long sender input).*

For more details on the aforementioned applications see the full version of this paper [10].

1.3 ECRH Candidate Constructions and Sufficient Relaxations

We propose a natural candidate ECRH based on a generalization of the knowledge of exponent assumption in large algebraic groups [24]. The assumption, which we call *t-Knowledge-of-Exponent Assumption* (or t-KEA for short) proceeds as follows. For any poly-size adversary, there exists a poly-size extractor such that, on input $g_1, \ldots, g_t, g_1^\alpha, \ldots, g_t^\alpha$ where each g_i is a random generator (of an appropriate group) and α is a random exponent: if the adversary outputs (f, f^α), then the extractor finds a vector of "coefficients" $(x_1, \ldots, x_t)$ such that $f = \prod_{i\in[t]} g_i^{x_i}$. This assumption can be viewed as a simplified version of the assumption used by Groth in [45] (the formal relation between the assumptions is discussed in Section 8.1). Similarly to Groth's assumption, t-KEA holds in the *generic group model*.

THEOREM 4 (INFORMAL). *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 straightforward: the function family is parameterized by $(g_1, \ldots, g_t, g_1^\alpha, \ldots, g_t^\alpha)$. Given input $(x_1, \ldots, x_t)$, the function outputs the two group elements $(\prod_{i\in[t]} g_i^{x_i}, \prod_{i\in[t]} g_i^{\alpha x_i})$. Extractability directly follows from t-KEA, while collision resistance is ensured by the hardness of taking discrete logs. See Section 8.1 for more details.

Next we proceed to propose ECRH candidates that are based on the subset sum problem in finite groups. Here, however, we are only able to construct candidates for somewhat weaker variants of ECRHs that are still sufficient for constructing SNARKs. While these variants are as not elegantly and concisely stated as the "vanilla"ECRH notion, they are still natural. Furthermore, we can show that these variants are necessary for SNARKs. We next proceed to formulate these weaker variants.

1.3.1 Proximity ECRH

We say that $\mathcal{H}$ defined on domain D is a *proximity ECRH* (PECRH) if (for any $h \in \mathcal{H}$) there exist a reflexive "proximity" relation $\overset{h}{\approx}$ on values in the range and an extension of the hash to a larger domain $D_h \supseteq D$ fulfilling the following: (a) **proximity collision resistance:** given $h \leftarrow \mathcal{H}$, it is hard to find $x, x' \in D_h$ such that $h(x) \overset{h}{\approx} h(x')$, and (b) **proximity extraction:** for any poly-time adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ such that, whenever $\mathcal{A}$ outputs $y \in h(D)$, $\mathcal{E}$ outputs $x \in D_h$ such that $h(x) \overset{h}{\approx} y$. (See Definition 6.2 for further details.)

Harder to find collisions, easier to extract. The notions of proximity extraction and proximity collision resistance are the same as standard extraction and collision resistance in the "strict" case, where $x \overset{h}{\approx} y$ is the equality relation and the domain is not extended ($D_h = \{0,1\}^{\ell(k)}$, $\bar{h} = h$).

However, in general, proximity collision resistance is stronger than (standard) collision resistance, because even "near collisions" (i.e., $x \neq y$ such that $\bar{h}(x) \overset{h}{\approx} \bar{h}(y)$) must not be efficiently discoverable, not even over the extended domain D_h. Conversely, proximity 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 $\bar{h}(x') \overset{h}{\approx} y$); moreover, it suffices that the point is in the extended domain D_h. Thus, the notion of PECRH captures another, somewhat more flexible tradeoff between the requirements of extractability and collision resistance.

We show 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:

THEOREM 5 (INFORMAL). *If there exist PECRHs and (appropriate) PIRs then there exist SNARKs for* NP.

Candidate PECRHs based on knapsack (subset sum) problems. A necessary property of ECRHs is that the image should be sparse;

knapsack-based CRHs, which typically rely on a proper algebraic structure, can often be tweaked to obtain this essential property. For example, in the t-KEA-based ECRH, we start from a standard knapsack hash $f = \prod_{i \in [t]} g_i^{x_i}$ and extend it to a "sparsified" knapsack hash (f, f^α) for a secret α. While for t-KEA this step is enough for plausibly assuming precise extractability (leading to a full fledged ECRH), for other knapsack-based CRHs this is not the case.

For example, let us consider the task of sparsifying modular subset-sum [62]. Here, the hash function is given by random coefficients $l_1, \ldots, l_t \in \mathbb{Z}_N$ and the hash of $x \in \{0,1\}^t$ is simply the corresponding modular subset-sum $\sum_{i:x_i=1} l_i \bmod N$. A standard way to sparsify the function is, instead of drawing random coefficients, drawing them from a distribution of noisy multiples of some secret integer. However, by doing so, we lose the "precise structure" of the problem. Hence, now we also have to deal with new "oblivious image-sampling attacks" that exploit the noisy structure. For example, slightly perturbing an honestly computed subset-sum is likely to "hit" another image of the function. This is where the relaxed notion of proximity extraction comes into play: it allows the extractor to output the preimage of the nearby (honest) image and, more generally, to thwart "perturbation attacks".

Sparsification of modular subset-sum in fact introduces additional problems. For instance, an attacker may take "small-norm" combinations of the coefficients that are not $0/1$ and still obtain an element in the image (e.g., if there are two even coefficients); to account for this, we need to further relax the notion of extraction by allowing the extractor to output a preimage in an extended domain, while ensuring that (proximity) collision resistance still holds for the extended domain too. Additionally, in some cases a direct naïve sparsification is not sufficient and we also need to consider amplified knapsacks.

The relaxations of extractability discussed above have to be matched by a corresponding strengthening of collision resistance following the definition of PECRH. Fortunately, this can still be done under standard hardness assumptions.

A similar approach can be taken in order to sparsify the modular matrix subset-sum CRH [3, 39], resulting in a a noisy inner-product knapsack hash based on the LWE assumption [64]. Overall, we propose three candidate for PECRHs:

THEOREM 6 (INFORMAL). *There exist PECRHs under any of the following assumptions:*

1. *A Knowledge of Knapsack of Exponent assumption (which in fact follows from t-KEA) and hardness of discrete logs.*
2. *A Knowledge of Knapsack of Noisy Multiples assumption and lattice assumptions.*
3. *A Knowledge of Knapsack of Noisy Inner Products assumption and learning with errors.*

1.3.2 Weak PECRHs[2]

Our second weakening is essentially orthogonal to the first one and relates to the condition that determines when the extractor has to "work". The ECRH and PECRH definitions required extraction whenever the adversary outputs a valid image; here the sparseness of the image appears to be key. In particular, unstructured CRHs where one can sample elements in the image obliviously of their preimage have no hope to be either ECRH or PECRH. However, for our purposes it seems sufficient to only require the extractor to "work" when the adversary outputs an image y together with extra encoding of a preimage that can be verified given proper trapdoor information; oblivious image-sampling, on its own, is no longer sufficient for failing the extractor.

More formally, a family $\mathcal{H}$ of functions is weakly extractable if for any efficient adversary $\mathcal{A}$ there exists an efficient extractor $\mathcal{E}_{\mathcal{A}}^{\mathcal{H}}$ such that for any auxiliary input z and efficient decoder $\mathcal{Y}$, the probability of the event that $\mathcal{A}$ outputs, given z and a random function h from the family, values y and e such that: (a) $\mathcal{E}_{\mathcal{A}}^{\mathcal{H}}$, given z and h, does not find a preimage of y, but (b) $\mathcal{Y}$, given e, does find a preimage of y, is negligible. See Definition 6.3 for further details.

We stress that the decoder circuit $\mathcal{Y}$ is allowed to arbitrarily depend on the auxiliary input z. This is $\mathcal{Y}$'s advantage over the extractor $\mathcal{E}_{\mathcal{A}}^{\mathcal{H}}$ (that must be efficient in z); in particular, the extractability condition is not trivially true. Another interpretation of the above definition is the following: The definition requires that there exists a single extractor $\mathcal{E}_{\mathcal{A}}^{\mathcal{H}}$ that does as well as any other efficient extractor (that may depend on z); namely, any given decoder circuit $\mathcal{Y}$ can be thought of as a candidate extractor and the extractor $\mathcal{E}_{A}^{H}$ has to succeed whenever $\mathcal{Y}$ does despite being efficient in the auxiliary input. In particular, whenever extraction is possible when given a trapdoor information related to the auxiliary input, it should also be possible without such trapdoor information. Indeed, the ability of the extractor to forgo the trapdoor is the key to a successful use of the above definition in our construction of SNARKs:

THEOREM 7 (INFORMAL). *If there exist weak PECRHs and (appropriate) PIRs then there exist SNARKs for* NP.

Unlike ECRHs and PECRHs, weak ECRHs and PECRHs may in principle not require sparsity of the image or algebraic structure. For example, it may be plausible to assume that (a properly keyed) SHA-2 is a weak ECRHs.

We remark that the notion of weak ECRH is somewhat reminiscent of the notion of *preimage awareness* considered by Dodis et al. [29]. Their notion, however, is set in an idealized model where the hash function can only be accessed as a black-box. When ported to such an ideal model, our definition can be viewed as a strengthening of the [29] definition to account for auxiliary input. Similarly to the [29] definition, the idealized version of our definition also holds in the (compressing) *random oracle model*. (We do not know if an idealized ECRH or PECRH can be constructed unconditionally in the random oracle model; however, assuming the existence of standard CRHs, so that a concise description for a CRH is available, it is possible to construct PECRHs in the random oracle model using CS proofs of knowledge.)

1.4 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 [28] is to "squash" Kilian's four-message protocol into a two-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.

[28] base their proof of soundness on the assumption that any convincing prover $\mathcal{P}^*$ must essentially behave as an honest prover;

[2]This further weakening was inspired by private communication with Ivan Damgård.

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 $\mathcal{E}_{\mathcal{P}^*}$. [28] 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.[3]

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 constructing 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 guarantee 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.

Extracting a witness. 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.)

Adaptivity. 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 setting of the coins, 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 no a-priori bound on the size of the corresponding PCP oracle is known), we require that the PIR supports random queries with respect to a-priori unknown database size. (Any FHE-based PIR, e.g., [14] 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.)

From local to global extraction. 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 binary as is usually the case. Consequently, the depth of the tree becomes $\log_k t$ where t is the claimed time-bound and this quantity is constant whenever t is polynomial in the security parameter k.

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 if ℓ is indeed a proper image. Hence, the extracted tree might have some inconsistent branches (or rather "holes").[4] Nevertheless, we indeed 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 coins that were PIR-encrypted. Otherwise, it is possible to find collisions within the ECRH h as follows. 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) and therefore obtain a collision.

- *Step 2: "from local to global consistency".* Next, using the privacy guarantees of the PIR scheme, we show that, whenever we one can extract a set of leaves that are satisfying with respect to the PIR-encrypted coins, the same set of leaves must also be satisfying for almost all other coins and is thus 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.

For further details we refer the reader to Section 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 poly-time task to delegate.

Nonetheless, we also show how, by 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.

1.5 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

[3] Note that, as originally formulated, the assumption of [28] 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.

[4] This captures for example the behavior of the prover violating the [28] assumption described above.

security properties of the scheme based on a new cryptographic primitive, ECRHs, that, while not fitting into the mold of "standard cryptographic primitives or assumptions", can be defined concisely and investigated separately.

Furthermore, we investigate a number of relaxations of ECRHs as well as a number of candidate constructions that have quite different underlying properties. Looking beyond the context of our particular protocol, this work can be seen as another step towards understanding the nature of extractability assumptions and their power in cryptography.

1.6 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. In Section 5, we give our main construction showing that ECRHs, along with appropriate PIRs, imply SNARKs. In Section 6, we discuss two relaxations of ECRHs, namely PECRHs and weak PECRHs, that still suffice for SNARKs. In Section 7, we explain how SNARKs imply proximity PECRHs, thus showing an almost tight relation between the existence of SNARKs and PECRHs. In Section 8, we propose candidate constructions for ECRHs and PECRHs.

2. OTHER RELATED WORK

Concurrent work. Two concurrent and independent works by Goldwasser et al. [43] and Damgård et al. [25] achieve results similar to some of ours.

Goldwasser et al. [43] consider a notion of ECRH similar to ours and construct adaptive SNARGs (but not SNARKs) for the purpose of delegation of computation (also by squashing Kilian's protocol). Their assumption differs from ours in that the extractor is required to output a vector of preimages whenever an adversary outputs a vector of images; while this assumption might seem slightly stronger than ours, the two are essentially equivalent because, just like our notion of ECRH, their notion is implied by SNARKs.

Damgård et al. [25] consider the problem of succinct non-interactive secure computation (as described in point (iii) Section 1.2), focusing on the case where the sender only has short input (while we also study the long input case). They also consider a notion of ECRH, which, unlike ours, requires seemingly stronger interactive extractability assumptions; for example, their notion does not seem to be implied by the existence of SNARKs.

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 [58], Shoup [67], and Dent [27]), 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 [24] first introduced a Knowledge of Exponent assumption to construct a CCA-secure encryption scheme. Later, Hada and Tanaka [46] showed how to use two Knowledge of Exponent assumptions to construct the first three-round zero-knowledge argument. Bellare and Palacio [6] then showed that one of the assumptions of [46] 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 [1] extended the assumption of [6] to construct the first perfect NIZK for NP with "full" adaptive soundness. Prabhakaran and Xue [60] constructed statistically-hiding sets for trapdoor DDH groups [26] using a new Knowledge of Exponent assumption. Gennaro et al. [35] used another Knowledge of Exponent assumption (with an interactive flavor) to prove that a modified version of the Okamoto-Tanaka key-agreement protocol [59] satisfies perfect forward secrecy against fully active attackers.

In a different direction, Canetti and Dakdouk [18, 19, 23] 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 (somewhat) succinct arguments from Knowledge of Exponent assumptions. Knowledge of Exponent (KE) assumptions have been used to obtain somewhat succinct arguments, in the sense the non-interactive proof is short, but the verifier's running time is long.

Recently, Groth [45] introduced a family of KE assumptions, generalizing those of [1], and used them to construct extractable length-reducing commitments, as a building block for short 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. For a comparison between our t-KEA assumption and Groth's assumptions see Section 8.1.

Mie [56] 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" their decrypted values, or, equivalently, must "know" a database consistent with those answers (by arbitrarily setting the rest of the database). Mie then shows how to make the PIR scheme of Gentry and Ramzan [37] PIR-aware, based on Damgård's Knowledge of Exponent assumption [24]; unfortunately, while the communication complexity is very low, the sender in [37] is inefficient relative to the database size. We note that PIR schemes with database awareness can be constructed directly from ECRHs (without going through the PCPs of a SNARK construction); moreover, if one is willing to use PCPs to obtain a SNARK, one would then be able to obtain various stronger notions of database awareness.

Delegation of computation. An important application of succinct arguments is *delegation of computation* schemes, where one usually 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 [36] (in order to ensure privacy) to obtain a delegation scheme with similar parameters.

Within the setting of delegation, however, where the same weak delegator may be asking a powerful untrusted worker to evaluate an expensive function on many different inputs, a weaker preprocessing approach may still be meaningful. In such a setting, the delegator performs a one-time function-specific expensive setup phase, followed by inexpensive input-specific delegations to amortize the initial expensive phase. Indeed, in the preprocessing setting a number of prior works have already achieved constructions where the online stage is only two messages [34, 22, 4]. These constructions do not allow for an untrusted worker to contribute his own input to the computation, namely they are "P-delegation schemes" rather than "NP-delegation schemes". Note that all of these works do not rely on any knowledge assumption; indeed, the impossibility results of [38] only apply for NP and not for P.

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. [9], though their constructions are not generic, and are only for specific functionalities such as polynomial functions.)

Goldwasser et al. [42] construct interactive proofs for log-space uniform NC where the verifier running time is quasi-linear. When combining [42] with the PIR-based squashing technique of Kalai and Raz [49], one can obtain a succinct two-message delegation scheme. Canetti et al. [20] introduce an alternative way of squashing [42], in the preprocessing setting; their scheme is of the public coin type and hence the verifier's answers need not remain secret (another bonus is that the preprocessing state is publicly verifiable and can thus be used by anyone).

3. PRELIMINARIES

In this section we give basic definitions for the cryptographic primitives that we use (along with any non-standard properties that we may need). Throughout, $\nu(k)$ is any negligible function in k.

3.1 Collision-Resistant Hashes

A *collision-resistant hash* (CRH) 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 $\mathcal{A}$,*

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

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 [53] 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 fan-in (relying on CRHs with polynomial 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 *private information retrieval* (PIR) scheme [17] consists of a triplet of algorithms $(\mathsf{PEnc},\mathsf{PEval},\mathsf{PDec})$ where:

- $\mathsf{PEnc}_R(1^k,i)$ outputs an encryption C_i of query i to a database DB 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* $k\in\mathbb{N}$*,*

$$\Pr_R\left[\mathsf{PDec}_R(e_i)=\mathsf{DB}[i] : \begin{array}{c} C_i\leftarrow\mathsf{PEnc}_R(1^k,i)\\ e_i\leftarrow\mathsf{PEval}(\mathsf{DB},C_i)\end{array}\right]=1 \enspace ,$$

   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*

$$\mathrm{poly}(k,\log|\mathsf{DB}|) \enspace .$$

   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 for multiple queries, i.e., for any poly-size $\mathcal{A}$, all large enough security parameter $k\in\mathbb{N}$ and any two tuples of queries* $\mathbf{i}=(i_1\cdots i_q),\mathbf{i}'=(i'_1\cdots i'_q)\in\{0,1\}^{\mathrm{poly}(k)}$*,*

$$\Pr\left[\mathcal{A}(\mathsf{PEnc}_R(1^k,\mathbf{i}))=1\right] - \Pr\left[\mathcal{A}(\mathsf{PEnc}_R(1^k,\mathbf{i}'))=1\right]\leq\nu(k) \enspace ,$$

   where $\mathsf{PEnc}_R(1^k,\mathbf{i})$ *the coordinate-wise encryption the tuple* $\mathbf{i}$*.*

PIR schemes with the above properties have been constructed under various hardness assumptions such as ΦHA [17] or LWE [14].

A-priori unknown DB size. In certain cases, 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 his 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 [14] (which is in turn based on LWE), this extra property can be easily supported. In other cases (as in 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 *probabilistically checkable proof* (PCP) 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
$$\Pr_{r \leftarrow \{0,1\}^{\rho(t)}} \left[V^{\pi}_{\mathsf{pcp}}(y, r) = 1\right] \geq 1 - \varepsilon \ ,$$
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.)

PCPs of knowledge as defined above can be based on the efficient-verifier PCPs of [8, 7]. (See [68] for a simple example of how a PCP of proximity can yield a PCP with a proof of knowledge.) Moreover, the latter PCPs' proof length is quasi-linear 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 $(1-\varepsilon)^q$, where k is the security parameter and $q(k) = \omega(\log k)$. Namely, extraction should succeed whenever $\Pr_{\mathbf{r}}\left[V^{\pi}_{\mathsf{pcp}}(y, \mathbf{r}) = 1\right] \geq (1-\varepsilon)^q$, where $\mathbf{r} = (r_i)_{i \in [q]}$ and $V^{\pi}_{\mathsf{pcp}}(y, \mathbf{r}) = \bigwedge_{i \in [q]} V^{\pi}_{\mathsf{pcp}}(y, r_i)$.

4. SNARKS

In this section we formally introduce the main cryptographic primitive studied in this paper — the SNARK.

4.1 The Universal Relation and NP Relations

The *universal relation* [5] 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 *succinct non-interactive argument* (SNARG) is a triple of 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. The SNARG is *adaptive* if the prover may choose the statement *after* seeing the vgrs, otherwise, it is *non-adaptive*.

DEFINITION 4.1. *A 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 $\mathcal{V}(\mathsf{priv}, y, \Pi)$, is 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.** *Depending on the notion of adaptivity:*

   - **Non-adaptive soundness.** *For all poly-size prover $\mathcal{P}^*$, large enough $k \in \mathbb{N}$, and $y \notin \mathcal{L}_{\mathcal{R}}$,*
$$\Pr\left[\mathcal{V}(\mathsf{priv}, y, \Pi) = 1 : \begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ \Pi \leftarrow \mathcal{P}^*(y, \mathsf{vgrs}) \end{array}\right] \leq \nu(k) \ .$$
   - **Adaptive soundness.** *For all poly-size prover $\mathcal{P}^*$ and large enough $k \in \mathbb{N}$,*
$$\Pr\left[\mathcal{V}(\mathsf{priv}, y, \Pi) = 1 : \begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ (y, \Pi) \leftarrow \mathcal{P}^*(\mathsf{vgrs}) \\ y \notin \mathcal{L}_{\mathcal{R}} \end{array}\right] \leq \nu(k) \ .$$

A *SNARG of knowledge*, or SNARK for short, is a SNARG where soundness is strengthened as follows:

DEFINITION 4.2. *A triple of algorithms $(\mathcal{P}, \mathcal{G}_{\mathcal{V}}, \mathcal{V})$ is a SNARK if it is a SNARG where adaptive soundness is replaced by the following stronger requirement:*

- **Adaptive proof of knowledge.**[5] *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 $z \in \{0,1\}^{\mathrm{poly}(k)}$,*
$$\Pr\left[\begin{array}{c} (\mathsf{vgrs}, \mathsf{priv}) \leftarrow \mathcal{G}_{\mathcal{V}}(1^k) \\ (y, \Pi) \leftarrow \mathcal{P}^*(z, \mathsf{vgrs}) \\ \mathcal{V}(\mathsf{priv}, y, \Pi) = 1 \end{array} \wedge \begin{array}{c} (y, w) \leftarrow \mathcal{E}_{\mathcal{P}^*}(z, \mathsf{vgrs}) \\ w \notin \mathcal{R}(y) \end{array}\right] \leq \nu(k) \ .$$

Universal arguments vs. weaker notions. A SNARG for the relation $\mathcal{R} = \mathcal{R}_{\mathcal{U}}$ is called a *universal argument*.[6] 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, soundness is only required to hold with respect to $\mathcal{R}_c$; in particular, the verifier algorithm may depend on 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.

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

[6] Barak and Goldreich [5] define universal arguments for $\mathcal{R}$ with a black-box "weak proof-of-knowledge" property; in contrast, our proof of knowledge property is not restricted to black-box extractors, and does not require the extractor to be an implicit representation of a witness.

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 whenever $\omega(\log k)$. 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 [49, 56, 42, 50, 34, 22].)

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 prover auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$, 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. (In our setting, the restrictions on the auxiliary-input handled by the knowledge extractor will be related to the auxiliary-input that the underlying ECRH extractor can handle. See further discussion in Section 6.1.)

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 SNARKs, this notion cannot be achieved based on black-box reductions to falsifiable assumptions [38].) 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. (Recall that an ECRH is a CRH as in Definition 3.1 that is extractable as in Definition 1).

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.

Our modified approach. We modify the PCP+MT+PIR approach of [28] and show that the knowledge assumption of [28] (which involves the entire PIR+MT structure) can be replaced 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 cryptographic applications. 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 $d \log k$).[7] 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 assuming the existence of exponentially-hard one-way functions.

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).

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

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 shallow 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 $\mathbf{i} = i_1 \cdots i_j \in [k]^j$ 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 $h : \{0,1\}^{k^2} \to \{0,1\}^k$ as follows:

- The leaf associated with $\mathbf{i} = i_1 \cdots i_d \in [k]^d \cong [k^d]$ 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 $\mathbf{i} = i_1 \cdots i_j \in [k]^j$ is labeled by $h(\ell_{\mathbf{i}1}\ell_{\mathbf{i}2} \cdots \ell_{\mathbf{i}k})$, 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 $h(\ell_{\mathbf{i}1}, \ldots, \ell_{\mathbf{i}k}) = \ell_\mathbf{i}$.

We shall prove the following theorem:

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 $\mathcal{R}_c \subseteq \mathcal{R}_\mathcal{U}$.*

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.

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 $O(1)$ queries.
- A succinct PIR $(\mathsf{PEnc}, \mathsf{PEval}, \mathsf{PDec})$ that supports an a-priori unknown database 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 $C_{\mathbf{r}} \leftarrow \mathsf{PEnc}_R(1^k, \mathbf{r})$.
  - 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 database 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 $C_{\mathsf{dcom}_{\hat{\mathbf{r}}}} \leftarrow \mathsf{PEval}(\mathsf{DB}, C_{\mathbf{r}})$. Here, each coordinate $r_j \in \{0,1\}^{\log^2 k}$ of $\mathbf{r}$ 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 (against 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] Such a PIR interprets a "long" query $r \in \{0,1\}^{\log^2 k}$ as a shorter one $\hat{r} \in \{0,1\}^\rho$, the ρ-bit prefix of r (see Section 3.3).
[b] V_{pcp}'s queries might be adaptive; such behavior can be simulated by the prover.
[c] This is the single place where the verification algorithm depends on c. See further discussion in Section 5.3.

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

5.2 Proof of Security

A high-level overview of the proof and main technical challenges are presented in Section 1.4. We now turn to the detailed proof, which concentrates on establishing and proving the validity of the knowledge extractor.

PROPOSITION 5.2 (ADAPTIVE PROOF OF KNOWLEDGE). *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 $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

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

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 5.2. 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 $z \in \{0,1\}^{\mathrm{poly}(k)}$. We note that, formally, a prover $\mathcal{P}^*$ is a family $\{\mathcal{P}^*_k\}$ of poly-size circuits, and so is its corresponding extractor $\mathcal{E}_{\mathcal{P}^*}$; for notational convenience, we omit the subscript k.

The extraction procedure. As discussed in Section 1.4, 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 now describe the recursive extraction procedure of the of the ECRH-based MT. Given a poly-size prover $\mathcal{P}^*$, let d be such that $|\mathcal{P}^*| \le k^d$. We derive $2cd$ circuit families of extractors, one for each potential level of the MT. Define $\mathcal{E}_1 := \mathcal{E}^{\mathcal{H}}_{\mathcal{P}^*}$ to be the ECRH extractor for $\mathcal{P}^*$; like $\mathcal{P}^*$, $\mathcal{E}_1$ also expects input $(h, C_{\mathbf{r}}) \in \{0,1\}^{\mathrm{poly}(k)}$ and returns a string $(\tilde{\ell}_1,\dots,\tilde{\ell}_k) \in \{0,1\}^{k\times k}$ (which will be a preimage in case $\mathcal{P}^*$ produces a valid image). We can interpret the string output by $\mathcal{E}_1$ as k elements in the range of the hash. Since the ECRH extraction guarantee only considers a single image, we define an augmented family of circuits $\mathcal{E}'_1$ that expects input $(h, C_{\mathbf{r}}, i_1)$, where $i_1 \in [k]$, and returns $\tilde{\ell}_{i_1}$, which is the i_1-th k-bit block of $\mathcal{E}_1(h, C_{\mathbf{r}})$.

Next, we define $\mathcal{E}_2 := \mathcal{E}^{\mathcal{H}}_{\mathcal{E}'_1}$ to be the extractor for $\mathcal{E}'_1$. In general, we define $\mathcal{E}_{j+1} := \mathcal{E}^{\mathcal{H}}_{\mathcal{E}'_j}$ to be the extractor for $\mathcal{E}'_j$, and $\mathcal{E}'_j$ expects an 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 $\tilde{\ell}_{\mathbf{i}1},\dots,\tilde{\ell}_{\mathbf{i}k}$.

Recall, however, that the ECRH extractor $\mathcal{E}_{j+1}$ is only guaranteed to output a preimage whenever 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 $(1^k, h, C_{\mathbf{r}})$, (a) first invoke $\mathcal{P}^*(h, C_{\mathbf{r}})$ and obtain (y, Π); (b) obtain the depth $\tilde{d}$ from Π, and abort if $\tilde{d} > 2cd$; (c) otherwise, for each $\mathbf{i} \in [k]^{\tilde{d}-1}$, extract the labels $(\tilde{\ell}_{\mathbf{i}1},\dots,\tilde{\ell}_{\mathbf{i}k}) \leftarrow \mathcal{E}_{\tilde{d}}(h, C_{\mathbf{r}}, \mathbf{i})$; (d) letting $\tilde{\pi}$ 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 into two main claims as outlined in Section 1.4.

A reminder and some notation. Recall that prior to the prover's message, the randomness for the PCP verifier is of the form $\mathbf{r} = (r_i)_{i\in[q]} \in \{0,1\}^{(\log^2 k)\times q}$ (and $q = \omega(\log k)$ is some fixed function). Once the verifier receives (y, Π), where $y = (M, x, t)$ and $\Pi = (\tilde{d}, \ell_\epsilon, C_{\mathsf{dcom}})$, he computes the amount of coins required $\rho = O(\log t) < \log^2 k$ and interprets each r_j as a shorter $\hat{r}_j \in \{0,1\}^\rho$. (Recall that $\hat{r}_j$ is the 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 $\tilde{\pi} = \mathcal{E}_{\tilde{d}}(h, \mathsf{PEnc}_R(\mathbf{r})) = \cup_{\mathbf{i}\in[k]^{\tilde{d}-1}} \mathcal{E}_{\tilde{d}}(h, \mathsf{PEnc}_R(\mathbf{r}), \mathbf{i})$ the extraction of the full set of leaves.

Using collision resistance and ECRH extraction, we show 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 5.3 (LOCAL CONSISTENCY). *Let $\mathcal{P}^*$ be a poly-size prover strategy, where $|\mathcal{P}| \le k^d$, and let $(\mathcal{E}_1,\dots,\mathcal{E}_{2cd})$ be its ECRH extractors as defined above. Then for all large enough $k \in \mathbb{N}$,*

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

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 a tuple $(h, R, \mathbf{r})$ is "bad" if it leads to the above event. Assume towards contradiction that for infinitely many $k \in \mathbb{N}$, there is a noticeable fraction $\varepsilon(k)$ of bad tuples $(h, R, \mathbf{r})$. We show how to find collisions in $\mathcal{H}$.

Given $h \leftarrow \mathcal{H}$, sample coins R for the PIR encryption and coins $\mathbf{r}$ for the PCP verifier to simulate $\mathsf{PEnc}_R(\mathbf{r})$. Given that the resulting $(h, R, \mathbf{r})$ is bad, let us show how to produce a collision in h.

First, invoke $\mathcal{P}^*(h, \mathsf{PEnc}_R(\mathbf{r}))$ to obtain (y, Π), where $y = (M, x, t)$ and $\Pi = (\tilde{d}, \ell_\epsilon, C_{\mathsf{dcom}})$. 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 $\mathbf{l}^{\mathbf{i}}_1,\dots,\mathbf{l}^{\mathbf{i}}_{\tilde{d}} \in \{0,1\}^{k^2\times\tilde{d}}$; each string $\mathbf{l}^{\mathbf{i}}_j$ contains the label of the j-th 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 $|x| \le |\mathcal{P}^*|; \le k^d$. Accordingly, we can use our extractors as follows: for each opened path in $\mathbf{i} \in S$, where $\mathbf{i} = i_1 \cdots i_{\tilde{d}} \in [k]^{\tilde{d}}$, invoke:

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

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

Let $\pi|_S = \left(\mathbf{l}^{\mathbf{i}}_{\tilde{d}}\right)_{\mathbf{i}\in S}$ be the leaves $\mathcal{P}^*$ opened 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 $\mathbf{l}^{\mathbf{i}}_{\tilde{d}} \neq \tilde{\mathbf{l}}^{\mathbf{i}}_{\tilde{d}}$. We focus from hereon on this specific $\mathbf{i}$. Let $j \in [\tilde{d}]$ be the smallest index such that $\mathbf{l}^{\mathbf{i}}_j \neq \tilde{\mathbf{l}}^{\mathbf{i}}_j$ (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 $\mathcal{E}'_{j-1}(h, \mathsf{PEnc}_R(\mathbf{r}), i_1 \cdots i_{j-1})$, which in turn implies that $\mathcal{E}_j(h, \mathsf{PEnc}_R(\mathbf{r}), i_1 \cdots i_{j-1}) = \tilde{\mathbf{l}}^{\mathbf{i}}_j$ is also compressed by h to the same ℓ^* (except when extraction fails, which occurs with negligible probability). It follows that $\mathbf{l}^{\mathbf{i}}_j \neq \tilde{\mathbf{l}}^{\mathbf{i}}_j$ form a collision in h. □

The second step in the proof of Proposition 5.2, is to show that if the aforementioned extractor outputs a proof $\tilde{\pi}$ that convinces the PCP verifier with respect to 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 security of the PIR.

CLAIM 5.4 (FROM LOCAL SATISFACTION TO EXTRACTION). *Let $\mathcal{P}^*$ be a poly-size prover and let $\mathcal{E}_{\mathcal{P}^*}$ be its poly-size extractor.*[8]*. Then for all large enough $k \in \mathbb{N}$,*

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

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 $\delta = \delta(k)$; we show how to break the semantic security of PEnc. First note that whenever the event occurs, it holds that $\Pr_{\hat{\mathbf{r}} \xleftarrow{U} \{0,1\}^{\rho\times q}}[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}) = 1] \leq (1-\varepsilon)^q$, where ε is the (constant) knowledge threshold of the PCP (see Section 3.4), and $q = \omega(\log k)$ is the number of repetitions.

To break semantic security, we consider the following CPA game, where a breaker $\mathcal{B}$ 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 $b \in \{0,1\}$. Now, $\mathcal{B}$ samples a random h, and runs $\mathcal{P}^*(h, \mathsf{PEnc}_R(\mathbf{r}_b))$ and $\mathcal{E}_{\mathcal{P}^*}(1^k, h, \mathsf{PEnc}_R(\mathbf{r}_b))$ to obtain an instance $y = (M,x,t)$ from the prover and an extracted PCP proof $\tilde{\pi}$ from the extractor. Then, $\mathcal{B}$ computes the amount of coins required for V_{pcp}, $\rho = \rho(t)$, and derives the corresponding ρ-prefixes $(\hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1)$ of $(\mathbf{r}_0, \mathbf{r}_1)$.

The breaker now runs the PCP extractor E_{pcp} on input $(y, \tilde{\pi})$ to obtain a string $\tilde{w}$ and verifies whether it is a valid witness for y (which can be done in $\mathrm{poly}(|x|) = \mathrm{poly}(k)$ time). In case the witness $\tilde{w}$ is valid or $V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_0) = V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_1)$, the breaker $\mathcal{B}$ outputs a random guess for the bit b. Otherwise, the breaker outputs the single b' such that $V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{b'}) = 1$.

We now analyze the success probability of $\mathcal{B}$. We define two events F and E over a random choice of $(h, R, \hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, b)$; note that any choice of $(h, R, \hat{\mathbf{r}}_0, \hat{\mathbf{r}}_1, b)$ induces a choice of $y = (M,x,t)$ and $\tilde{\pi}$. Define F to be the event that $t \leq |x|^c$ and $E_{\mathsf{pcp}}(y,\tilde{\pi})$ fails to output a valid witness $\tilde{w}$; next, define E to be the event that $V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_0) \neq V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_1)$.

First, since we have assumed by way of contradiction that the event in the statement of the claim occurs with probability δ, we know that

$$\Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_b) = 1 \mid F\right] = \frac{\delta}{\Pr[F]} \ .$$

Second, since E_{pcp} cannot extract a valid witness from $\tilde{\pi}$ and $\hat{\mathbf{r}}_{1-b}$ are random coins independent of $(\tilde{\pi}, y)$, we also know that

$$\Pr_{\hat{\mathbf{r}}_{1-b} \xleftarrow{U} \{0,1\}^{\rho\times q}}\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{1-b}) = 1 \mid F\right] \leq (1-\varepsilon)^q \ .$$

Combining these two facts, we deduce that

$$\begin{aligned} &\Pr\left[E \mid F\right] \\ &\geq \Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_b) = 1 \wedge V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{1-b}) = 0 \mid F\right] \\ &\geq \Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_b) = 1 \mid F\right] - \Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{1-b}) = 1 \mid F\right] \\ &\geq \Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_b) = 1 \mid F\right] - (1-\varepsilon)^q = \frac{\delta}{\Pr[F]} - \nu(k) \ , \end{aligned}$$

so that, in particular, we can also deduce that

$$\Pr\left[F \wedge E\right] \geq \delta - \nu(k) \ .$$

Therefore,

$$\begin{aligned} &\Pr\left[\mathcal{B} \text{ guesses } b \mid F \wedge E\right] \\ &\geq 1 - \Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{1-b}) = 1 \mid F \wedge E\right] \\ &= 1 - \frac{\Pr\left[V^{\tilde{\pi}}_{\mathsf{pcp}}(y,\hat{\mathbf{r}}_{1-b}) = 1 \wedge E \mid F\right]}{\Pr\left[E \mid F\right]} \\ &\geq 1 - \frac{(1-\varepsilon)^q}{\delta / \Pr\left[F\right]} \\ &\geq 1 - \nu(k) \ . \end{aligned}$$

We now deduce that the breaker $\mathcal{B}$ guesses b with a noticeable advantage; indeed,

$$\begin{aligned} &\Pr\left[\mathcal{B} \text{ guesses } b\right] \\ &= \Pr\left[F \wedge E\right] \Pr\left[\mathcal{B} \text{ guesses } b \mid F \wedge E\right] \\ &\quad + \left(1 - \Pr\left[F \wedge E\right]\right) \Pr\left[\mathcal{B} \text{ guesses } b \mid \bar{F} \vee \bar{E}\right] \\ &= \Pr\left[F \wedge E\right] \Pr\left[\mathcal{B} \text{ guesses } b \mid F \wedge E\right] \\ &\quad + \left(1 - \Pr\left[F \wedge E\right]\right) \cdot \frac{1}{2} \\ &= \frac{1}{2} + \Pr\left[F \wedge E\right] \left(\Pr\left[\mathcal{B} \text{ guesses } b \mid F \wedge E\right] - \frac{1}{2}\right) \\ &\geq \frac{1}{2} + (\delta - \nu(k)) \left(\frac{1}{2} - \nu(k)\right) \\ &\geq \frac{1}{2} + \frac{\delta - \nu(k)}{2} \ , \end{aligned}$$

thus completing the proof of the claim. □

Putting it all together. By Claim 5.3 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 5.4, we deduce that whenever this occurs, $\tilde{\pi}$ must satisfy sufficiently many queries for PCP witness-extraction. This completes the proof of Proposition 5.2 and thus of Theorem 5.1.

Efficiency: "universal succinctness". For input $y = (M,x,t)$ (where $t < k^{\log k}$ for a security parameter k), the proof $\Pi =$

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

$(\ell_\epsilon, d, C_{\mathsf{dcom}_{\tilde{\mathsf{r}}}})$ is essentially dominated by the PIR answers $C_{\mathsf{dcom}_{\tilde{\mathsf{r}}}}$; this includes $q = \mathrm{polylog}(k)$ PIR answers for entries of size $\tilde{O}(k^2)$.[9] In the PIR scheme of [14] the size of each PIR-answer is bounded by $E \cdot k \cdot \mathrm{polylog}(k) + \log D$, 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 $|x|$, $|w|$ or c. The verifier's and prover's running time are bounded respectively by fixed universal polynomials $\mathrm{poly}(|y|, k)$, $\mathrm{poly}(k, t)$, again independently of c.

Parameter scaling. In Kilian's original protocol, succinctness of the proof could be improved by making stronger hardness assumptions. For example, for security parameter k, if one is willing to assume collision-resistant hash functions with a $\mathrm{polylog}(k)$-long output, the proof length would be $\mathrm{polylog}(k)$, rather than $\mathrm{poly}(k)$. Unfortunately, in our construction we use a Merkle tree with $\mathrm{poly}(k)$ fan-in; therefore, we cannot afford the same scaling. Specifically, even if we assume that our hash and PIR scheme have polylogarithmic-size output, each node in the Merkle tree still has $\mathrm{poly}(k)$ siblings.[10] Nonetheless, scaling can be performed if we make a stronger extractability assumption, such as the interactive one of [25], because in such a case there is no need to consider Merkle trees with polynomial fan-in as binary Merkle trees suffice for the security reduction.

5.3 Extension to Universal Arguments

We now discuss the possibility of extending our construction to a full-fledged universal argument, namely an argument for the universal relation $\mathcal{R}_\mathcal{U}$ as defined in Section 4.1.

Indeed, Theorem 5.1 tells us that for every $c \in \mathbb{N}$ we obtain a specific protocol that is sound with respect to $\mathcal{R}_c$. The dependence on c, however, only appears in the first step of $\mathcal{V}$, where it is checked that $k^{d+1} \leq |x|^{2c}$. In particular, as we already discussed, 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 modification 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 $\varepsilon > 0$. Concretely, if we are willing to make an additional (though admittedly quite strong) assumption we can obtain such proofs of work:

THEOREM 5.5. *If there exist $2^{\varepsilon n}$-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 $2^{\varepsilon n}$-hard one-way function. Modify $\mathcal{G}_\mathcal{V}(1^k)$ to also output $z_1, \ldots, z_\ell$, with $\ell := \log^2 k$ 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 $i > \log t$.[11] The verifier $\mathcal{V}$ can easily check that this is the case by evaluating f. (Note also that the honest prover will have to pay at most an additive factor of $\tilde{O}(t)$ in 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 $k^d > 2^{\varepsilon i} > t^\varepsilon$, so that we conclude that $k^{d/\varepsilon} > t$. 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 has already been made, e.g., in natural proofs [61] or in works that improve PRG constructions [47].

6. ECRHS: A CLOSER LOOK

In this section we take a closer look at the notion of ECRH, and propose relaxations of this notion that still suffice for constructing SNARKs. These relaxations are crucial to expand our set of candidate constructions; for more details on the constructions, see Section 8.

6.1 ECRHs

We begin by discussing several technical aspects regarding the definition of ECRH. Recall that an ECRH is a collision-resistant function ensemble $\mathcal{H}$ that is extractable in the sense of Definition 1, which we reproduce:

DEFINITION 6.1. *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 $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

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

In other words, 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).

Image verification. In known applications of extractable primitives (e.g., 3-round zero knowledge [46, 6, 19]), an extra image-verifiability feature is required. 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

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

[10] We thank Kai-Min Chung and the anonymous referees of ITCS for pointing out the scaling problem.

[11] Note that in any case the verifier will reject any claim for t above the superpolynomial universal bound $2^{\log^2 k}$; hence, $\ell = \log^2 k$ challenges are sufficient for any poly-size prover.

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

$$\mathcal{I}_h = \left\{ h(x) : x \overset{U}{\leftarrow} \{0,1\}^{\ell(k)} \right\}$$

should be hard to invert (except with negligible probability over h). In particular, $\mathcal{I}_h$ must be very far from the uniform distribution over $\{0,1\}^k$ (for almost all h).

Indeed, suppose that the statistical distance between $\mathcal{I}_h$ and uniform is $1-1/\mathrm{poly}(k)$ and consider an adversary $\mathcal{A}$ 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 $\mathcal{A}$, 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 $\mathcal{I}_h$, which is negligible (by one-wayness); hence $\mathcal{E}^{\mathcal{A}}_{\mathcal{H}}$ will still fail with probability $1-1/\mathrm{poly}(k)$ often, thereby violating Equation (1).

A simple way to ensure that the image distribution $\mathcal{I}_h$ is indeed far from uniform is to make the support of $\mathcal{I}_h$ sparse. We will take this approach when constructing candidates, 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 with 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-black-box 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 $\mathcal{I}_h$ that is close to uniform (by design) and are thus not extractable.

On auxiliary input. The ECRH definition requires that for any adversary auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$, 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 (efficiently) 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 the 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 (C, I) consisting of (honestly-generated) PIR-encryptions C of random strings and a short path index I of length $O(\log k)$. We note that by using a PIR scheme where honestly generated ciphers are pseudo-random, we can actually restrict the "main" part C of the auxiliary input to be a random string (for example, the LWE-based PIR scheme of [14] has this property). As for the logarithmically-short auxiliary input I, one can show that it gives no additional power to the adversary. Specifically, an ECRH for auxiliary input distribution C is also an ECRH for auxiliary input (C, I) as long as $|I| = O(\log k)$. (Roughly, the extractor of an adversary $\mathcal{A}$ for auxiliary input (C, I) can be constructed from a polynomial number of extractors, obtained by considering the extractor of $\mathcal{A}_I(C) := \mathcal{A}(C, I)$ for every possible value of I; for the formal argument one has to adapt this intuition to circuit families.)

We note that in certain previous works (e.g. [1]), 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.

6.2 PECRHs

As discussed in Section 1.3.1, our first weakening of ECRH, namely proximity ECRH, captures a more flexible tradeoff between the requirements of extractability and collision resistance. Formally:

DEFINITION 6.2. *An efficiently-samplable ensemble of functions $\mathcal{H} = \{\mathcal{H}_k\}_k$ is an $(\ell(k), k)$-*compressing PECRH *if it is $(\ell(k), k)$-compressing and, for every h in the support of $\mathcal{H}_k$, there exist a reflexive "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 $\bar{h} : D_h \rightarrow \{0,1\}^k$ consistent with h (i.e., $\forall x \in \{0,1\}^{\ell(k)}$ it holds that $h(x) = \bar{h}(x)$), such that:*

1. *$\mathcal{H}$ is* proximity-extractable *in the following weakened sense: for any poly-size adversary $\mathcal{A}$ there exists a poly-size extractor $\mathcal{E}^{\mathcal{H}}_{\mathcal{A}}$ such that for large enough security parameter $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$:*

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

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

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

We now discuss why any point on the tradeoff (i.e., any choice of $\overset{h}{\approx}$, D_h and $\bar{h}$ fulfilling the conditions) suffices for the construction of SNARKs as claimed in Theorem 5 in Section 1.3.1.

PROOF SKETCH FOR THEOREM 5. We argue that the *same construction* the we use in the proof of Theorem 1 to construct SNARKs from ECRHs still suffices even when the underlying hash function is only a PECRH.

First, observe that moving from ECRHs to PECRHs only affects the "local consistency" step of our proof (as described in our high-level description in Section 1.4 and then formally as Claim 5.3). 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. By the proximity 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 proximity collision resistance, however, these two node labels must in fact *be the same*; for if they were not, then we could utilize the prover and extractor for finding "proximity collisions". The rest of the proof of Theorem 1 remains unchanged.

We emphasize that the proximity relation $\overset{h}{\approx}$ need not be efficiently computable for the above to hold. $\square$

6.3 Weak PECRHs

As discussed in Section 1.3.2, our second weakening of ECRH, namely weak PECRH, relaxes the condition that determines when the extractor has to work. Formally:

DEFINITION 6.3. *An efficiently-samplable function ensemble $\mathcal{H} = \{\mathcal{H}_k\}_k$ is a $(\ell(k), k)$*-compressing weak PECRH *if it satisfies Definition 6.2 with the following modified first condition:*

1. *$\mathcal{H}$ is* weak proximity-extractable *in the following sense: for any poly-size adversary $\mathcal{A}$ there exists a poly-size extractor $\mathcal{E}^{\mathcal{H}}_{\mathcal{A}}$ such that for large enough security parameter $k \in \mathbb{N}$ and any auxiliary input $z \in \{0,1\}^{\mathrm{poly}(k)}$ and poly-size decoder circuit $\mathcal{Y}$:*

$$\Pr_{h \leftarrow \mathcal{H}_k}\left[\begin{array}{c} (y,e) \leftarrow \mathcal{A}(h,z) \\ h(\mathcal{Y}(e)) = y \\ \wedge \quad \begin{array}{c} x' \leftarrow \mathcal{E}^{\mathcal{H}}_{\mathcal{A}}(h,z) \\ \neg\left(x' \in D_h \ \wedge\ \bar{h}(x') \overset{h}{\approx} y\right) \end{array} \end{array}\right] < \nu(k) \ .$$

We show that even weak PECRHs are enough for the construction of SNARKs as claimed in Theorem 7.

PROOF SKETCH OF THEOREM 7. We argue that the *same construction* that we use in the proof of Theorem 1 to construct SNARKs from ECRHs also obtains SNARKs even when the underlying hash function is only a weak PECRH. As was the case when moving from ECRHs to PECRHs, moving from PECRHs to weak PECRHs only affects the "local consistency" step of our proof (as described in our high-level description in Section 1.4 and then formally as Claim 5.3); specifically, we must still be able to guarantee local consistency even when the condition under which the extractor is guaranteed to work is weakened to the case where the adversary outputs an encoding of a preimage (as opposed to when the adversary merely outputs a value in the image).

In the construction, this is always the case, because preimages along the opened path are provided as encrypted authentication paths, which can be "decoded" by a decoder that knows encryption the secret key (i.e., the PIR private coins). Therefore, we are still able to show local consistency. $\square$

Unlike PECRHs, weak PECRHs may in principle not require sparsity of the image or special algebraic structure. While an attacker trying to fool a PECRH extractor only has to obliviously sample an image of the function, now, to fool a weak PECRH extractor, it needs to simultaneously obliviously sample an image of the function and an encoding of a corresponding preimage. This raises the following natural question: "is any CRH also a weak PECRH?" We believe this to be unlikely; indeed, the following example shows that (assuming the existence of one-way permutations) there exists a CRH that is not a weak PECRH when the proximity relation is forced to be equality. (To extend this to an actual counter-example, one would have to rule out all proximity relations.)

Let $\mathcal{H} = \{\mathcal{H}_k\}$ be any CRH mapping $\{0,1\}^{2k}$ to $\{0,1\}^k$ and P any one-way permutation mapping $\{0,1\}^k$ to itself. Consider the (contrived) new CRH $\mathcal{H}'$, mapping $\{0,1\}^{2k}$ to $\{0,1\}^{k+1}$, that is defined as follows. A seed $h' \in \mathcal{H}'_k$ corresponds to a seed $h \in \mathcal{H}_k$; each $(x_1||0^k) \in \{0,1\}^k \times \{0^k\}$ is mapped by h' to $0||P(x_1)$ and each $(x_1||x_2) \in \{0,1\}^k \times (\{0,1\}^k \setminus \{0^k\})$ is mapped by h' to $1||h(x_1||x_2)$.

Since P is one-to-one and the intersection of the image sets $h'(\{0,1\}^k \times \{0^k\})$ and $h'(\{0,1\}^k \times (\{0,1\}^k) \setminus \{0^k\}))$ is empty, any collision in h' implies a corresponding collision in h and, therefore, $\mathcal{H}'$ is also a CRH (that is, a proximity CRH relative to the equality proximity relation). However, $\mathcal{H}'$ is not weakly proximity extractable relative to the equality proximity relation; indeed, consider the auxiliary input distribution $z = P(x_1)$ for a random $x_1 \leftarrow \{0,1\}^k$, with a corresponding (correlated) decoder $\mathcal{Y}$ that always outputs $x_1||0^k$. In addition, consider a dummy adversary that given z, h simply outputs $0||z = 0||P(x_1) = h(x_1||0^k)$. Note that since the decoder always outputs a valid preimage, any extractor would have to do the same. However, any efficient extractor that manages to do so has to invert the one-way permutation P.

7. FROM SNARKS TO PECRHS (AND MORE)

In this section we provide more details about Theorem 2 and Theorem 3, which were only informally stated in the introductory discussion of Section 1.2. That is, we show that (proximity) extractable collision-resistant hash functions (PECRHs) are in fact not only sufficient (together with appropriate polylog PIRs) but also necessary for SNARKs (assuming standard CRHs). We then describe a general method for obtaining additional (non-interactive) extractable primitives.

Extractability and proximity extractability. 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. Typically, for such a family to be interesting, it is required that $\mathcal{F}$ also has some hardness property, e.g., one-wayness; in particular, for the two features (of hardness and extractability) to coexist, $\mathsf{Image}(f)$ must be sparse in the (recognizable) range of the function.

As explained (for ECRHs) in Section 1.3.1 and later in Section 6.1, the above notion of extraction can be relaxed to consider proximity extraction, according to which it is infeasible to produce $y \in \mathsf{Image}(f)$ without knowing an x such that $f(x)$ is proximate to y relative to some given reflexive proximity relation (e.g, relative hamming distance at most $1/2$). For such a notion of extraction to be useful, the corresponding hardness of f should be upgraded accordingly. For example, a *proximity extractable one-way function* $(f, \approx)$, where $\approx$ is some proximity relation, is such that, given $f(x)$ for a random x, it is infeasible to find an x' for which $f(x') \approx f(x)$ (and it is proximity extractable in the sense that we just described).

In Section 6 we discussed even further relaxations: in one, the extractor also had the freedom to output elements x from an extended domain $D_f \supseteq \mathsf{Domain}(f)$; in another, the extractor only had to work when the adversary manages to output not only an image but also an encoding of a corresponding preimage. In this section, however, we do not consider these further relaxations.

Note that, naturally, known cryptographic schemes (e.g., 3-round zero-knowledge) have relied on standard extraction rather than proximity extraction; however, to the best of our knowledge we can safely replace standard extraction in these schemes with proximity extraction (as we did for the purpose of constructing SNARKs).

Verification and proximity verification. Another extraction-related issue is *image verification*; here, there are two notions that can be considered:

- *Public verification:* Given f and $y \in \mathsf{Range}(f)$, one can 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 $y \in \mathsf{Range}(f)$, one can efficiently test whether $y \in \mathsf{Image}(f)$.

In addition, when considering proximity extractability, we can consider a corresponding notion of *proximity verifiability*, where the verifier should only check whether $y \in_{\approx} \mathsf{Image}(f)$, namely there is some element $y' \in \mathsf{Image}(f)$ for which $y \approx y$. Again we note that for the known applications of (verifiable) extractable primitives, proximity verification is sufficient.

Also note that the weaker notion of *extractability with no efficient verification* might also be meaningful in certain scenarios. Indeed, for our main ECRH-based construction of SNARKs (presented in Section 5.1), this weak notion of extractability with no efficient verification suffices, and in fact ultimately allows us to deduce, through the SNARK construction, many extractable primitives with efficient private (proximity) verification.

7.1 From SNARKs to PECRHs

We now present the implications of SNARKs to the existence of extractable primitives, starting with the necessity of PECRHs:

PROPOSITION 7.1. *If there exist SNARKs and (standard) CRHs, then there exist proximity verifiable PECRHs.*

PROOF SKETCH. We show that designated-verifier SNARKs imply PECRHs with private proximity verification. The proof can be easily extended to the case of public verifiability (where publicly-verifiable SNARKs imply PECRHs with public proximity verification). Let $\mathcal{H}$ be a (tk, k)-compressing CRH, where $t = t(k) > 1$ is a compression parameter. 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 $\hat{k}^c$.[12] We define a $(tk, 2k)$-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 $\tilde{h} = (h, \mathsf{vgrs}), \mathsf{priv}_{\tilde{h}} = \mathsf{priv}$.

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 $x \in \{0,1\}^{tk}$ such that $h(x) = y$".

We now show that the above is a PECRH with respect to the relation $\approx$, where $(y, \Pi) \approx (y', \Pi')$ if and only if $y = y'$.

The proximity collision resistance of $\widetilde{\mathcal{H}}$ follows directly from the collision resistance of $\mathcal{H}$, because any proximity-colliding pair (x, x') for $\widetilde{\mathcal{H}}$ is a colliding pair for $\mathcal{H}$. The proximity 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 proximity-verified (with respect to $\approx$) 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. □

[12] 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 $\log t < \hat{k}$.

Why PECRH and not ECRH? At first glance, it is not clear why the above does not imply an ("exact") ECRH rather than a PECRH. The reason lies in the fact that the extractor is only guaranteed to output one of many possible witnesses (preimages). In particular, given an honest image $(h(x), \Pi_x)$ (corresponding to some preimage x), the extractor may output x' such that $h(x') = h(x)$ but applying the honest prover to x' (or, more precisely, the NP-statement proving knowledge of x') results in a proof $\Pi_{x'} \neq \Pi_x$.

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

COROLLARY 7.2. *If there exist SNARKs and (standard) CRHs, then there exist PEOWFs and PECOMs. Moreover, the verifiability features of the SNARK carry over to the implied primitives.*

PROOF SKETCH. First, note that any $(\ell(k), k)$-compressing PECRH is also a (keyed) PEOWF (with respect to the same image proximity relation). Indeed, it is a proximity OWF since it is a proximity CRH and independently of that it is also proximity extractable (and verifiable).

Second, to get an extractable bit-commitment scheme, one can use the classic CRH plus hardcore bit construction of Blum [11]. Specifically, the commitment scheme is keyed by a seed h for the PECRH and a commitment to a bit b is obtained by sampling $r, \hat{r} \xleftarrow{U} \{0,1\}^{\ell(k)}$ 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 (note that any proximity CRH is in particular a 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 PECRH extractor to extract the commitment randomness r, such that $y \approx h(r)$ and $c \approx \mathsf{Eval}_{\mathsf{Com}}(h; b \oplus \langle r, \hat{r} \rangle; r, \hat{r})$, where $c = (y, \hat{r}, b) \approx c' = (y', \hat{r}', b')$ if and only if $y \approx y, \hat{r} = \hat{r}', b = b'$.

In addition, verifying a proximity commitment is done by verifying that y is proximate to an image under h. □

7.2 From Leakage-Resilient Primitives and SNARKs to Extractable Primitives

Given the results in the previous section, naïvely, it seems that non-interactive adaptive arguments of knowledge offer a generic approach towards constructing extractable primitives: "simply add a non-interactive proof of preimage knowledge" (which might seem to be applicable even without succinctness when compression is not needed). However, this approach may actually compromise privacy because the attached proofs may leak too much information about the preimage.

One may try to overcome this problem by using non-interactive *zero-knowledge* proofs of knowledge; however, this can only be done in the common reference string model, which typically trivializes the notion of extractable functions. (Nonetheless, in later sections, we will still discuss zero-knowledge SNARKs and some of their meaningful applications in the CRS model.)

In this section, we consider a different approach towards overcoming the problem of proof-induced preimage leakage: we suggest 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 7.2 can be viewed as an application of this paradigm. Moreover, in this approach, there is no need to assume a trusted third party to set up a common reference string (as would be the case if we were to use zero-knowledge techniques).

Applying this paradigm to one-to-one subexponentially hard OWFs, yields:

PROPOSITION 7.3. *Given any $2^{|x|^\varepsilon}$-hard OWF $f : \{0,1\}^* \to \{0,1\}^*$ and SNARKs, there exist extractable PEOWFs (against poly-size adversaries). Moreover, the verifiability features of the SNARK carry over to the implied PEOWF.*

PROOF SKETCH. Let $f : \{0,1\}^* \to \{0,1\}^*$ be any $2^{|x|^\varepsilon}$-hard OWF, where n is the size of the input. As in Proposition 5.2, 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 $\{0,1\}^{k^{2c/\varepsilon}}$ and are indexed by a $\mathsf{vgrs} \leftarrow \mathcal{G}_\mathcal{V}(1^k)$. For $x \in \{0,1\}^{k^{2c/\varepsilon}}$, $f_{\mathsf{vgrs}}(x) = (f(x), \Pi)$, where Π is a SNARK for the statement that "there exists an $x \in \{0,1\}^{k^{2c/\varepsilon}}$ such that $f(x) = y$". As for PECRHs, we define the proximity relation to be $(y, \Pi) \approx (y', \Pi')$ if and only if $y = y'$. As in the proof of Proposition 7.1, proximity extraction and verifiability follow directly from the extraction and verifiability of the SNARK. We claim that $\mathcal{F}$ is one-way with respect to $\approx$; namely, given the image (y, Π) of a random x in the domain, it is infeasible to come up with an x' such that $f(x') = y$. This follows from the fact that f is $2^{|x|^\varepsilon}$-hard, and the proof Π is of length at most $|x|^{\varepsilon/2}$. In particular, any poly-size adversary which proximity-inverts $\mathcal{F}$, can be transformed to a $2^{|x|^\varepsilon}$-size adversary that inverts f by simply enumerating all the (short) proofs π. □

Note 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 our SNARK security proof does not scale in this way, but given stronger extractability assumptions it would.) 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 $\mathcal{F}$. Moreover, in such a case in the NP-language corresponding to f, any theorem claiming that $y = f(x)$ is a proper image has a single witness x. In this case we would get (exact) EOWF rather than PEOWF. We thus get:

COROLLARY 7.4. *Given any $2^{|x|^\varepsilon}$-hard one-to-one OWF and SNARKs, there exist (exactly) extractable commitments that are perfectly binding and computationally hiding (against poly-size adversaries).*

PROOF SKETCH. Indeed, the EOWFs given by Proposition 7.3 would now be one-to-one, which in turn imply perfectly-binding ECOMs, using the hardcore bit construction as in Corollary 7.2 instantiated with a one-to-one EOWF. (The fact that one-to-one EOWFs imply perfectly binding ECOMs was already noted in [19].) □

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 [19] 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 [48] 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 [31, 65, 38] imply that any strong enough PRG is inherently also leakage-resilient, in the sense that, even given leakage on the seed, the PRG's output still has high pseudo-entropy (specifically, *HILL entropy*). The results of Braverman et. al. [16] 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. Perfectly-binding ECOMs (as given by Corollary 7.4) provide a generic way of obtaining limited extractable primitives that do not admit efficient verification (and if compression is needed SNARKs can be used on top). Specifically, one can transform a function $\mathcal{F}$ to an extractable $\widetilde{\mathcal{F}}$ as follows. The seed $\tilde{f}_{(f,g)}$ generated by $\widetilde{\mathcal{F}}_k$ includes $f \leftarrow \mathcal{F}_k$ 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 $\tilde{f}_{(f,g)}(x; r) = (f(x), \mathsf{Eval}_{\mathsf{Com}}(g; x; r))$. That is, add to $f(x)$ a perfectly-binding commitment to a preimage. The hiding property of the commitment clearly prevents the problem of leakage on x. The fact that the commitment is perfectly-binding and extractable implies that $\widetilde{\mathcal{F}}$ is also extractable. Indeed, any adversary that produces a valid image, also produces a valid perfectly-binding commitment to a valid pre-image; hence, using the extractor for the commitment, we obtain a valid preimage. A major caveat of this approach is that the resulting $\widetilde{\mathcal{F}}$ does not support efficient image-verification; indeed, 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 an investigation of other possible applications of non-verifiable extractable primitives for future work.

8. CANDIDATE ECRH AND PECRH CONSTRUCTIONS

In this section we discuss:

- a candidate construction for an ECRH, based on a Knowledge of Exponent assumption and the hardness of discrete logs; and
- a generic technique for obtaining candidate constructions for PECRHs, which we instantiate in three different ways.

As already discussed, the relaxation of ECRHs to PECRHs is crucial for (a) obtaining more candidate constructions, and (b) arguing the necessity of PECRHs to the construction of SNARKs.

8.1 ECRHs from t-Knowledge of Exponent

Recall that ECRHs are formally discussed in Definition 6.1. The *Knowledge of Exponent assumption* (KEA) [24] states that any adversary that, given a generator g and a random group element g^α, manages to produce $g^x, g^{\alpha x}$, must "know" the exponent x. The assumption was later extended in [46, 6], by 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 $f = g^{x_1 r_1} g^{x_2 r_2} = g^{x_1 r_1 + x_2 r_2}$. The t-KEA assumption is a natural extension to $t = \text{poly}(k)$ pairs $g^{r_i}, g^{\alpha r_i}$.

ASSUMPTION 8.1 (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 $z \in \{0,1\}^{\text{poly}(k)}$,*

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

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 assumption was made by Groth [45]; there, instead of random $r_1, \ldots, r_t$, the exponents are powers of the same random element, i.e., $r_i = r^i$. (As formalized in [45], the assumption does not account for auxiliary inputs, but it could naturally be strengthened to do so.)

Our assumption can be viewed as a simplified version of Groth's assumption; in particular, we could use Groth's assumption directly to get ECRHs. Furthermore, Groth's assumption is formally stated in bilinear groups, while in our setting bilinearity is not necessary. When considered in (non bilinear) groups where t-DDH is assumed to hold, the two assumptions are actually equivalent.[13] Therefore, as Groth shows that his assumption holds in the generic group model (independently of the bilinear structure) and as t-DDH is also known to hold in this model, our assumption holds in the generic group model as well.

A candidate ECRH from t-KEA. A $(k \cdot t(k), 2k)$-compressing 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 $(\alpha, \mathbf{r}) \xleftarrow{U} \mathbb{Z}_p \times \mathbb{Z}_p^t$, and output $h := (\mathbb{G}, g^\mathbf{r}, g^{\alpha\mathbf{r}})$.
- To compute $h(x_1, \ldots, x_t)$: output the pair $(g^{\langle \mathbf{r}, \mathbf{x} \rangle}, g^{\langle \alpha\mathbf{r}, \mathbf{x} \rangle}) = \left(\prod_{i\in[t]} g^{r_i x_i}, \prod_{i\in[t]} g^{\alpha r_i x_i}\right)$.

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 8.2. *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 $r \xleftarrow{U} \mathbb{Z}_p$, 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 $r_i = (x_i - x'_i)^{-1} \sum\limits_{j\in[k]-\{i\}} (x_j - x'_j) r_j$. $\square$

[13] t-DDH asserts that, over suitable groups, tuples of the form $g^x, g^{x^2}, \ldots, g^{x^t}$ are indistinguishable from random tuples.

8.2 PECRHs from Knowledge of Knapsack

In Section 8.1 we presented a candidate ECRH based on a generalization of the Knowledge of Exponent assumption in large algebraic groups. We are now going to introduce a class of knowledge assumptions with a "lattice flavor", which we call *Knowledge of Knapsack*, to construct candidates for the weaker notion of a proximity ECRH (PECRH). Recall that PECRHs are formally discussed in Definition 6.2.

Indeed, we are not able to achieve the strict notion of ECRH from "lattice-flavor" Knowledge of Knapsack assumptions; instead, we only obtain the "noisy" notion of ECRH that we have formalized as a PECRH (which yet is still sufficient, and essentially necessary, for constructing SNARKs, as discussed in Section 6.2). 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.

At high level, we define a candidate PECRH 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 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 $(t, \lambda, \mathcal{K}, \stackrel{h}{\approx}, D_h)$ asserts that the function ensemble $\mathcal{H}^{t,\lambda,\mathcal{K}}$ is proximity-extractable with respect to some proximity relation $\stackrel{h}{\approx}$, some extended domain $D_h \subseteq \mathbb{Z}^t$, and extended function $\bar{h} : D_h \to \mathbb{H}$ defined by taking a linear combinations with coefficients in D_h (rather than just subset sums). Explicitly:

DEFINITION 8.3 (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 $\stackrel{h}{\approx}$ be a relation on the image of h and let D_h be an extended domain $D_h \subseteq \mathbb{Z}^t$ where $D_h \supseteq \{0,1\}$.*

The Knowledge of Knapsack *assumption with respect to $(t, \lambda, \mathcal{K}, \stackrel{h}{\approx}, D_h)$ 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 $z \in$*

$\{0,1\}^{\mathrm{poly}(k)}$,

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

where j ranges over $\{1,\ldots,\lambda\}$, the summations are in the group $\mathbb{H}$, and the multiplications 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 6.1, we wish the candidate PECRH (just like a 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 PECRH $\mathcal{H}^{t,\lambda,\mathcal{K}}$ that is ρ^λ-sparse. Thus, for example, 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 the basic knapsacks $\mathcal{K}$ must be just polynomially sparse for the proof of (proximity) collision resistance to go through.

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

8.2.1 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 8.1), albeit with inferior parameters:

ASSUMPTION 8.4 (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 $(t,1,\mathcal{K}^{\mathsf{E}},\equiv_{\mathbb{H}},\{0,1\}^t)$ holds for the ensemble $\mathcal{K}^{\mathsf{E}}=\{\mathcal{K}_k^{\mathsf{E}}\}_k$ 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 assuming the hardness of taking discrete logs. Note that its range is indeed sparse, as prescribed in Section 6.1: 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.

8.2.2 Knowledge of Knapsack of Noisy Multiples

Next, we propose a new knowledge assumption based on the following problem: given noisy integer multiples $L=(l_i,\ldots,l_t)$ in $\mathbb{Z}_N$ of a secret real number α (of magnitude about $\sqrt{N}$), find a subset-sum of these multiples.[14] The knowledge assumption says (roughly) that whenever an efficient adversary produces such a subset sum, it knows the corresponding subset. This however requires care, since, taken literally, the assumption is clearly false. To motivate our definition, we describe several attempted attacks, and how the definition avoids them.

- *Perturbation attack.* Any small integer is close to a multiple of α (i.e., 0), and is thus likely to be a sum of *some* subset of L (when L is long enough, as it is in our setting). Thus, the adversary $\mathcal{A}$ could simply output a random small integer and thereby challenge the extractor $\mathcal{E}$ to find a corresponding subset. We let the extractor avoid this difficult task by using the notion of PECRHs defined above, with the proximity relation $\overset{h}{\approx}$ chosen so that the extractor only needs to output a subset that sums to *approximately* the adversary's output (in the above example, the extractor can output the empty set).
- *Integer-coefficients attack.* An adversary $\mathcal{A}$ could pick an integer combination of L with coefficients that are small but not all 0 and 1. Even though this is not a valid computation of a sum over a subset of L, the result y is still close to a multiple of the secret real number, and thus, as above, is likely to be a subset sum of for *some* subset, so the extractor $\mathcal{E}$ must "explain" y. We aid $\mathcal{E}$ by enlarging the extended domain D_h to allow small integer coefficients, so that the (non-blackbox) extractor may output the coefficients used by the adversary.
- *Fractional-coefficients attack.* An adversary $\mathcal{A}$ could pick a fractional combination of elements of L. For example, $l_1/2$ will be close to a multiple of α whenever l_1 happens to be close to an even multiple of α (that is, with probability half). However, we amplify our knapsack to consider λ instances concurrently (each consisting of noisy multiples L of some different α), so the extractor is challenged only in the exponentially-unlikely event that all λ instances have l_1 that is close to an even multiple.

Comparison to t-KKE. The above complications arise due to the addition of noise to l_i in the generation of the knapsack instances (otherwise α would be found computing the greatest common divisor on L, easily leading to collisions). Thus the collection of resulting subset-sums constitutes a train of "hills" (each clustered around a multiple of α), which an adversary can traverse by the aforementioned attacks. Conversely, in t-KKE from Section 8.2.1, the underlying discrete log problem does not require injection of noise, hence the subset sums constitute a set of distinct "well-spaced" points in $\mathbb{G}\times\mathbb{G}$ and so (one may hope) the adversary can navigate the structure of the image only by algebraic operations that the extractor can unravel.

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

ASSUMPTION 8.5 ((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.*

[14] Our construction is inspired by a cryptosystem of Regev [62, 63], where the public key is sampled from a similar distribution, and indeed our analysis of collision resistance and sparsity invokes Regev's. This is elaborated below.

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 $\mathcal{K}_k^{\mathsf{NM},t,\sigma}$ do the following: let $N = 2^{8k^2}$, draw $h \xleftarrow{U} \left\{h \in [\sqrt{N}, 2\sqrt{N}) : |h - \lfloor h \rceil| < \frac{1}{16t}\right\}$ and draw $\bar{\sigma}$ such that $\bar{\sigma}^2 \xleftarrow{U} [\sigma^2, 2\sigma^2)$. Let $\alpha = N/h$. Draw t values $l_1, \ldots, l_t \leftarrow \mathsf{NM}_{\alpha,\bar{\sigma},N}$. 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 $\sqrt{N}/9$.

Relation to Regev's cryptosystem [62, 63]. The above distributions are essentially the same as in Regev's cryptosystem, with minor changes for clarity in the present context. Explicitly, the mapping is as follows. The distribution $Q_\beta = (\mathcal{N}_{0,\beta/2\pi} \bmod 1)$ from [63, Section 2.1] is replaced by $\mathcal{N}_{0,\bar{\sigma}^2}$, for $\beta = 2\pi\bar{\sigma}^2$ (the statistical difference between the two is negligible because $\bar{\sigma}$ will be polynomially small). The distribution $\mathsf{NM}_{\alpha,\bar{\sigma},N}$ is a scaling up by N of $T_{h,\beta}$ as defined in [63, 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 [63, 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 proximity-collision-resistant for any $t = O(k^2)$ and suitable λ and σ, assuming on the hardness of the Unique Shortest Vector Problem (uSVP) in lattices. Recall that $f(\mu)$-uSVP is the computational problem of finding a shortest vector in a lattice of dimension μ given that the shortest vector is at least $f(\mu)$ times shorter than any other (non-parallel) lattice vector (see [63, 52].

CLAIM 8.6. *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 the interval $\{0, \ldots, N-1\}$), assuming hardness of $(\sqrt{2/\pi}\mu/\sigma(\mu))$-uSVP.*

PROOF SKETCH. It suffices to show pseudorandomness for the distribution obtained by modifying $\mathcal{K}^{\mathsf{NM},t,\sigma}$ to sample $h \xleftarrow{U} [\sqrt{N}, 2\sqrt{N})$ (for the same reason as in [63, Lemma 5.4]). This pseudorandomness follows from [63, Theorem 4.5] with $g(n) = \sqrt{2\mu/\pi}/\sigma(\mu)$. $\square$

CLAIM 8.7. *The function ensemble $\mathcal{H}^{\mathsf{KKNM}}$ is proximity-collision-resistant, with $\overset{h}{\approx}, D_h, \bar{h}$ defined as in Assumption 8.5, for $t = O(k^2)$, assuming hardness of $\tilde{O}\left(\max\left(\mu^{3/2}, \sqrt{\mu}/\sigma(\mu)\right)\right)$-uSVP.*

PROOF SKETCH. By Claim 8.6, the hash functions drawn by $\mathcal{H}^{\mathsf{KKNM}}$ are indistinguishable from the ensemble $\mathcal{U}$ of uniformly-random modular subset sums (as defined in [63, Section 6]), assuming $\tilde{O}(\sqrt{\mu}/\sigma(\mu))$-uSVP. It thus suffices to show that $\mathcal{U}$ is proximity-collision-resistant, since this implies finding collisions in $\mathcal{H}^{\mathsf{KKNM}}$ would distinguish it from $\mathcal{U}$. The ensemble $\mathcal{U}$ is collision-resistant assuming $\tilde{O}(\mu^{3/2})$-uSVP, by [63, Theorem 6.5]. Moreover, the proximity relation $\overset{h}{\approx}$ is accommodated by noting that the theorem still holds if in its statement, $\sum_{i=1}^{m} b_i a_i \equiv 0 \pmod{N}$ 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. The extended domain D_h and induced $\bar{h}$ are accommodated by noting that Regev's bound $||b|| \leq \sqrt{m}$ (in his notation) generalizes to $||b|| \leq \tilde{O}(\sqrt{m})$. $\square$

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 6.1. 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 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 8.8. *For $\sigma(k) = 1/16t \log^2 k$, the ensemble $\mathcal{K}^{\mathsf{NM},t,\sigma}$ is $\frac{1}{2}$-sparse:*

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

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 [63, 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. $\square$

Thus, the image becomes somewhat sparse when $\sigma = \tilde{o}(1/t)$. However, superpolynomial sparseness would require making σ superpolynomially small (and likewise a tighter distribution over h), in which case Claim 8.7 would require assuming hardness of $\mu^{\omega(1)}$-uSVP; this assumption is unmerited in light of the excellent heuristic performance of LLL-type lattice reduction algorithms on lattices with large gaps (e.g., [33] conjecture, from experimental evidence, that 1.02^μ-uSVP is easy). Instead, we can set $\sigma = \tilde{\Omega}(1/k^2)$ so that Claim 8.7 needs to assume merely hardness of $\tilde{O}(\mu^{3/2})$-uSVP, and then amplify via repetition, by choosing sufficiently large λ. In particular, by setting $\sigma(k) = \tilde{o}(1/t)$, $\lambda = \omega(\log(k))$ and $t = O(k^2)$, we indeed obtain superpolynomial sparseness.

Regarding the aforementioned integer-coefficient attack, note that the extended domain D_h allows $\mathcal{E}$ to explain y via any vector using a linear combination whose coefficients have ℓ_2 norm at most $t \log^2 t$, since beyond this norm, the linear combination is unlikely to be in the image of h.

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

Relation to other lattice hardness assumptions. The collision resistance 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 [52].

8.2.3 Knowledge of Knapsack of Noisy Inner Products

Further PECRH candidates can be obtained from Knowledge of Knapsack problems on other lattice-based problems. In particular, the Learning with Error problem [64] 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 $\vec{s} \cdot \vec{x} + \varepsilon$ where ε is independently-drawn noise of small magnitude in $\mathbb{Z}_p$. For suitable parameters this ensemble is sparse, and proximity-collision-resistant following an approach similar to KKNM above: first show pseudorandomness assuming hardness of LWE [64], and then rely on the collision resistance of the uniform case (e.g., [3, 39, 55]).

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 8.3.

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.

9. REFERENCES

[1] M. Abe and S. Fehr. Perfect NIZK with adaptive soundness. In *Proceedings of the 4th Theory of Cryptography Conference*, pages 118–136, 2007.

[2] W. Aiello, S. N. Bhatt, R. Ostrovsky, and S. 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*, pages 463–474, 2000.

[3] M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pages 99–108, 1996.

[4] B. Applebaum, Y. Ishai, and E. Kushilevitz. From secrecy to soundness: efficient verification via secure computation. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming*, pages 152–163, 2010.

[5] B. Barak and O. Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2008. Preliminary version appeared in CCC '02.

[6] M. Bellare and A. Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Proceedings of the 24th Annual International Cryptology Conference*, pages 273–289, 2004.

[7] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Short PCPs verifiable in polylogarithmic time. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 120–134, 2005.

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

[9] S. Benabbas, R. Gennaro, and Y. Vahlis. Verifiable delegation of computation over large datasets. In *Proceedings of the 31st Annual International Cryptology Conference*, pages 111–131, 2011.

[10] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. Cryptology ePrint Archive, Report 2011/443, 2011.

[11] M. Blum. Coin flipping by telephone. In *Proceedings of the 18th Annual International Cryptology Conference*, pages 11–15, 1981.

[12] D. Boneh, G. Segev, and B. Waters. Targeted malleability: Homomorphic encryption for restricted computations. Cryptology ePrint Archive, Report 2011/311, 2011.

[13] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[14] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science*, 2011.

[15] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[16] M. Braverman, A. Hassidim, and Y. T. Kalai. Leaky pseudo-entropy functions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science*, pages 353–366, 2011.

[17] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 402–414, 1999.

[18] R. Canetti and R. R. Dakdouk. Extractable perfectly one-way functions. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pages 449–460, 2008.

[19] R. Canetti and R. R. Dakdouk. Towards a theory of extractable functions. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 595–613, 2009.

[20] R. Canetti, B. Riva, and G. N. Rothblum. Two 1-round protocols for delegation of computation. Cryptology ePrint Archive, Report 2011/518, 2011.

[21] A. Chiesa and E. Tromer. Proof-carrying data and hearsay arguments from signature cards. In *Proceedings of the 1st Symposium on Innovations in Computer Science*, pages 310–331, 2010.

[22] K.-M. Chung, Y. Kalai, and S. Vadhan. Improved delegation of computation using fully homomorphic encryption. In *Proceedings of the 30th Annual International Cryptology Conference*, pages 483–501, 2010.

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

[24] I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Proceedings of the 11th Annual International Cryptology Conference*, pages 445–456, 1992.

[25] I. Damgård, S. Faust, and C. Hazay. Secure two-party computation with low communication. Cryptology ePrint Archive, Report 2011/508, 2011.

[26] A. Dent and S. Galbraith. Hidden pairings and trapdoor DDH groups. In F. Hess, S. Pauli, and M. Pohst, editors, *Algorithmic Number Theory*, volume 4076 of *Lecture Notes in Computer Science*, pages 436–451. 2006.

[27] A. W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006.

[28] G. Di Crescenzo and H. Lipmaa. Succinct NP proofs from an extractability assumption. In *Proceedings of the 4th Conference on Computability in Europe*, pages 175–185, 2008.

[29] Y. Dodis, T. Ristenpart, and T. Shrimpton. Salvaging merkle-damgård for practical applications. In *Proceedings of the 28th Annual International Conference on the Theory and*

Applications of Cryptographic Techniques, pages 371–388, 2009.

[30] C. Dwork, M. Langberg, M. Naor, K. Nissim, and O. Reingold. Succinct NP proofs and spooky interactions, December 2004. Available at `www.openu.ac.il/home/mikel/papers/spooky.ps`.

[31] S. Dziembowski and P. Krzysztof. Leakage-resilient cryptography. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 293–302, 2008.

[32] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings of the 6th Annual International Cryptology Conference*, pages 186–194, 1987.

[33] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 31–51, 2008.

[34] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: outsourcing computation to untrusted workers. In *Proceedings of the 30th Annual International Cryptology Conference*, pages 465–482, 2010.

[35] R. Gennaro, H. Krawczyk, and T. Rabin. Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. In *Proceedings of the 8th International Conference on Applied Cryptography and Network Security*, pages 309–328, 2010.

[36] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 169–178, 2009.

[37] C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 803–815, 2005.

[38] C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, 2011.

[39] O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. Technical report, 1996. ECCC TR95-042.

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

[41] O. Goldreich, S. Vadhan, and A. Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1/2):1–53, 2002.

[42] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: Interactive proofs for Muggles. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 113–122, 2008.

[43] S. Goldwasser, H. Lin, and A. Rubinstein. Delegation of computation without rejection problem from designated verifier CS-proofs. Cryptology ePrint Archive, Report 2011/456, 2011.

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

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

[46] S. Hada and T. Tanaka. On the existence of 3-round zero-knowledge protocols. In *Proceedings of the 18th Annual International Cryptology Conference*, pages 408–423, 1998.

[47] I. Haitner, D. Harnik, and O. Reingold. Efficient pseudorandom generators from exponentially hard one-way functions. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 228–239, 2006.

[48] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[49] Y. T. Kalai and R. Raz. Succinct non-interactive zero-knowledge proofs with preprocessing for LOGSNP. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 355–366, 2006.

[50] Y. T. Kalai and R. Raz. Probabilistically checkable arguments. In *Proceedings of the 29th Annual International Cryptology Conference*, pages 143–159, 2009.

[51] J. Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 723–732, 1992.

[52] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Proceedings of the 29th Annual International Cryptology Conference*, pages 577–594, 2009.

[53] R. C. Merkle. A certified digital signature. In *Proceedings of the 9th Annual International Cryptology Conference*, pages 218–238, 1989.

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

[55] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37:267–302, April 2007.

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

[57] M. Naor. On cryptographic assumptions and challenges. In *Proceedings of the 23rd Annual International Cryptology Conference*, pages 96–109, 2003.

[58] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55:165–172, 1994.

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

[60] M. Prabhakaran and R. Xue. Statistically hiding sets. In *Proceedings of the The Cryptographers' Track at the RSA Conference 2009*, pages 100–116, 2009.

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

[62] O. Regev. New lattice based cryptographic constructions. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 407–416, 2003.

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

[64] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 84–93,

2005.

[65] O. Reingold, L. Trevisan, M. Tulsiani, and S. P. Vadhan. Dense subsets of pseudorandom sets. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 76–85, 2008.

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

[67] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 256–266, 1997.

[68] P. Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Proceedings of the 5th Theory of Cryptography Conference*, pages 1–18, 2008.

[69] H. Wee. On round-efficient argument systems. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 140–152, 2005.