Succinct Non-Interactive Arguments via Linear Interactive Proofs*

Nir Bitansky[1]**, Alessandro Chiesa[2], Yuval Ishai[3]***, Omer Paneth[4]†, and Rafail Ostrovsky[5]‡

[1] Tel Aviv University
[2] MIT
[3] Technion
[4] Boston University
[5] UCLA

Abstract. *Succinct non-interactive arguments* (SNARGs) enable verifying NP statements with lower complexity than required for classical NP verification. Traditionally, the focus has been on minimizing the length of such arguments; nowadays researches have focused also on minimizing verification time, by drawing motivation from the problem of delegating computation.

A common relaxation is a *preprocessing* SNARG, which allows the verifier to conduct an expensive offline phase that is independent of the statement to be proven later. Recent constructions of preprocessing SNARGs have achieved attractive features: they are publicly-verifiable, proofs consist of only $O(1)$ encrypted (or encoded) field elements, and verification is via arithmetic circuits of size linear in the NP statement. Additionally, these constructions seem to have "escaped the hegemony" of probabilistically-checkable proofs (PCPs) as a basic building block of succinct arguments.

---

* The full version of this paper can be found on ePrint [BCI+12].

** This research was done while visiting Boston University and IBM T. J. Watson Research Center. Supported by the Check Point Institute for Information Security, an ISF grant 20006317, and the Fulbright program.

*** Supported by the European Research Council as part of the ERC project CaC (grant 259426), ISF grant 1361/10, and BSF grant 2008411. Research done in part while visiting UCLA and IBM T. J. Watson Research Center.

† Supported by an NSF grant 1218461.

‡ Department of Computer Science and Department of Mathematics, UCLA. Email: rafail@cs.ucla.edu. Research supported in part by NSF grants CNS-0830803; CCF-0916574; IIS-1065276; CCF-1016540; CNS-1118126; CNS-1136174; US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is also based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

We present a general methodology for the construction of preprocessing SNARGs, as well as resulting concrete efficiency improvements. Our contribution is three-fold:

(1) We introduce and study a natural extension of the interactive proof model that considers *algebraically-bounded* provers; this new setting is analogous to the common study of algebraically-bounded "adversaries" in other fields, such as pseudorandomness and randomness extraction. More concretely, in this work we focus on linear (or affine) provers, and provide several constructions of (succinct two-message) *linear-interactive proofs* (LIPs) for NP. Our constructions are based on general transformations applied to both *linear* PCPs (LPCPs) and traditional "unstructured" PCPs.

(2) We give conceptually simple cryptographic transformations from LIPs to preprocessing SNARGs, whose security can be based on different forms of *linear targeted malleability* (implied by previous knowledge assumptions). Our transformations convert arbitrary (two-message) LIPs into designated-verifier SNARGs, and LIPs with degree-bounded verifiers into publicly-verifiable SNARGs. We also extend our methodology to obtain *zero-knowledge* LIPs and SNARGs. Our techniques yield SNARGs *of knowledge* and thus can benefit from known recursive composition and bootstrapping techniques.

(3) Following this methodology, we exhibit several constructions achieving new efficiency features, such as "single-ciphertext preprocessing SNARGs" and improved succinctness-soundness tradeoffs. We also offer a new perspective on existing constructions of preprocessing SNARGs, revealing a direct connection of these to LPCPs and LIPs.

1 Introduction

Interactive proofs [GMR89] are central to modern cryptography and complexity theory. One extensively studied aspect of interactive proofs is their expressibility, culminating with the result $\mathsf{IP} = \mathsf{PSPACE}$ [Sha92]. Another aspect, which is the focus of this work, is that proofs for NP statements can potentially be *much shorter* than an NP witness and be *verified much faster* than the time required for checking the NP witness.

1.1 Background

Succinct interactive arguments. In interactive proofs for NP with statistical soundness, significant savings in communication (let alone verification time) are unlikely [BHZ87, GH98, GVW02, Wee05]. If we settle for proof systems with *computational* soundness, known as *argument systems* [BCC88], then significant savings can be made. Using collision-resistant hashes (CRHs) and probabilistically-checkable proofs (PCPs) [BFLS91], Kilian [Kil92] showed a four-message interactive argument for NP where, to prove membership of an instance x in a given NP language L with NP machine M_L, communication and verification

time are bounded by $\text{poly}(\lambda + |M_L| + |x| + \log t)$, and the prover's running time is $\text{poly}(\lambda + |M_L| + |x| + t)$. Here, t is the classical NP verification time of M_L for the instance x, λ is a security parameter, and poly is a *universal* polynomial (independent of λ, M_L, x, and t). We call such argument systems *succinct*.

Proof of knowledge. A natural strengthening of computational soundness is (computational) *proof of knowledge*: it requires that, whenever the verifier is convinced by an efficient prover, not only can we conclude that a valid witness for the theorem *exists*, but also that such a witness can be *extracted efficiently* from the prover. This property is satisfied by most proof system constructions, including the aforementioned one of Kilian [BG08], and is useful in many applications of succinct arguments.

Removing interaction. Kilian's protocol requires four messages. A challenge, which is of both theoretical and practical interest, is the construction of *non-interactive* succinct arguments. As a first step in this direction, Micali [Mic00] showed how to construct publicly-verifiable *one-message* succinct non-interactive arguments for NP, in the random oracle model, by applying the Fiat-Shamir heuristic [FS87] to Kilian's protocol. In the plain model, one-message solutions are impossible for hard-enough languages (against non-uniform provers), so one usually considers the weaker goal of two-message succinct arguments where the verifier message is generated *independently* of the statement to be proven. Following [GW11], we call such arguments *SNARGs*. More precisely, a SNARG for a language L is a triple of algorithms (G, P, V) where: the generator G, given the security parameter λ, samples a *reference string* σ and a corresponding *verification state* τ (G can be thought to be run during an offline phase, by the verifier, or by someone the verifier trusts); the (honest) prover $P(\sigma, x, w)$ produces a proof π for the statement "$x \in L$" given a witness w; then, $V(\tau, x, \pi)$ verifies the validity of π. Soundness should hold even if x is chosen depending on σ.

Gentry and Wichs [GW11] showed that no SNARG can be proven secure via a black-box reduction to a falsifiable assumption [Nao03]; this may justify using non-standard assumptions to construct SNARGs. (Note that [GW11] rule out SNARGs only for (hard-enough) NP languages. For the weaker goal of verifying deterministic polynomial-time computations in various models, there are beautiful constructions relying on standard assumptions, such as [GKR08, KR09, AIK10, CKV10, GGP10, BGV11, CRR11, CTY11, CMT12, FG12]. We focus on verifying *nondeterministic* polynomial-time computations.)

Extending earlier works [ABOR00, DLN+04, Mie08, DCL08], several works showed how to remove interaction in Kilian's PCP-based protocol and obtain SNARGs of knowledge (*SNARKs*) using *extractable collision-resistant hashing* [BCCT12a, DFH12, GLR11], or construct MIP-based SNARKs using fully-homomorphic encryption *with an extractable homomorphism property* [BC12].

The preprocessing model. A notion that is weaker than a SNARK is that of a *preprocessing* SNARK: here, the verifier is allowed to conduct an *expensive*

offline phase. More precisely, the generator G takes as an additional input a time bound T, may run in time $\text{poly}(\lambda + T)$ (rather than $\text{poly}(\lambda + \log T)$), and generates σ and τ that can be used, respectively, to prove and verify correctness of computations of length at most T. Bitansky et al. [BCCT12b] showed that SNARKs can always be "algorithmically improved"; in particular, preprocessing SNARKs imply ones *without* preprocessing. (The result of [BCCT12b] crucially relies on the fast verification time and the adaptive proof-of-knowledge property of the SNARK.) Thus, "preprocessing can always be removed" at the expense of only a $\text{poly}(\lambda)$-loss in verification efficiency.

1.2 Motivation

The typical approach to construct succinct arguments (or, more generally, other forms of proof systems with nontrivial efficiency properties) conforms with the following methodology: first, give an information-theoretic construction, using some form of probabilistic checking to verify computations, in a model that enforces certain restrictions on provers (e.g., the PCP model [Kil92, Mic00, BG08, DCL08, BCCT12a, DFH12, GLR11] or other models of probabilistic checking [IKO07, KR08, SBW11, SMBW12, SVP+12, BC12, SBV+12]); next, use cryptographic tools to compile the information-theoretic construction into an argument system (where there are no restrictions on the prover other than it being an efficient algorithm).

Existing constructions of preprocessing SNARKs seem to diverge from this methodology, while at the same time offering several attractive features: such as public verification, proofs consisting of only $O(1)$ encrypted (or encoded) field elements, and verification via arithmetic circuits that are linear in the statement.

Groth [Gro10] and Lipmaa [Lip12] (who builds on Groth's approach) introduced clever techniques for constructing preprocessing SNARKs by leveraging knowledge-of-exponent assumptions [Dam92, HT98, BP04] in bilinear groups. At high level, Groth considered a simple reduction from circuit satisfaction problems to an algebraic satisfaction problem of quadratic equations, and then constructed a set of specific cryptographic tools to succinctly check satisfiability of this problem. Gennaro et al. [GGPR12] made a first step to better separate the "information-theoretic ingredient" from the "cryptographic ingredient" in preprocessing SNARKs. They formulated a new type of algebraic satisfaction problems, called *Quadratic Span Programs* (QSPs), which are expressive enough to allow for much simpler, and more efficient, cryptographic checking, essentially under the same assumptions used by Groth. In particular, they invested significant effort in obtaining an efficient reduction from circuit satisfiability to QSPs.

Comparing the latter to the probabilistic-checking-based approach described above, we note that a reduction to an algebraic satisfaction problem is a typical first step, because such satisfaction problems tend to be more amenable to probabilistic checking. As explained above, cryptographic tools are then usually invoked to enforce the relevant probabilistic-checking model (e.g., the PCP one). The aforementioned works [Gro10, Lip12, GGPR12], on the other hand,

seem to somehow skip the probabilistic-checking step, and directly construct specific cryptographic tools for checking satisfiability of the algebraic problem itself. While this discrepancy may not be a problem per se, we believe that understanding it and formulating a clear methodology for the construction of preprocessing SNARKs are problems of great interest. Furthermore, a clear methodology may lead not only to a deeper conceptual understanding, but also to concrete improvements to different features of SNARKs (e.g., communication complexity, verifier complexity, prover complexity, and so on). Thus, we ask:

Is there a general methodology for the construction of preprocessing SNARKs? Which improvements can it lead to?

1.3 Our Results

We present a general methodology for the construction of preprocessing SNARKs, as well as resulting concrete improvements. Our contribution is three-fold:

- We introduce a natural extension of the interactive proof model that considers *algebraically-bounded* provers. Concretely, we focus on *linear interactive proofs* (LIPs), where both honest and malicious provers are restricted to computing linear (or affine) functions of messages they receive over some finite field or ring. We then provide several (unconditional) constructions of succinct two-message LIPs for NP, obtained by applying simple and general transformations to two variants of PCPs.
- We give cryptographic transformations from (succinct two-message) LIPs to preprocessing SNARKs, based on different forms of *linear targeted malleability*, which can be instantiated based on existing knowledge assumptions. Our transformation is very intuitive: to force a prover to "act linearly" on the verifier message, simply encrypt (or encode) each field or ring element in the verifier message with an encryption scheme that only allows linear homomorphism.
- Following this methodology, we obtain several constructions that exhibit new efficiency features. These include "single-ciphertext preprocessing SNARKs" and improved succinctness-soundness tradeoffs. We also offer a new perspective on existing constructions of preprocessing SNARKs: namely, although existing constructions do not explicitly invoke PCPs, they can be reinterpreted as using *linear PCPs*, i.e., PCPs in which proof oracles (even malicious ones) are restricted to be a linear functions.[6]

We now discuss our results further, starting in Section 1.3 with the information-theoretic constructions of LIPs, followed in Section 1.3 by the cryptographic transformations to preprocessing SNARKs, and concluding in Section 1.3 with the new features we are able to obtain.

[6] A stronger notion of linear PCP has been used in other works [IKO07, SBW11, SMBW12, SVP+12, SBV+12] to obtain arguments for NP with nontrivial efficiency properties.

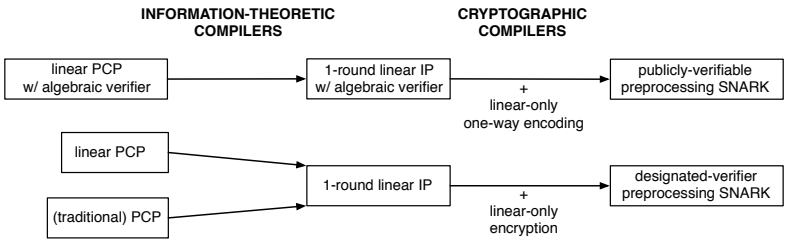

Fig. 1: High-level summary of our transformations.

Linear interactive proofs. The LIP model modifies the traditional interactive proofs model in a way analogous to the way the common study of algebraically-bounded "adversaries" modifies other settings, such as pseudorandomness [NN90, BV07] and randomness extraction [GR05, DGW09]. In the LIP model both honest and malicious provers are restricted to apply linear (or affine) functions over a finite field $\mathbb{F}$ to messages they receive from the verifier. (The notion can be naturally generalized to apply over rings.) The choice of these linear functions can depend on auxiliary input to the prover (e.g., a witness), but not on the verifier's messages.

With the goal of non-interactive succinct verification in mind, we restrict our attention to (input-oblivious) *two-message* LIPs for boolean circuit satisfiability problems with the following template. To verify the relation $\mathcal{R}_C = \{(x, w) : C(x, w) = 1\}$ where C is a boolean circuit, the LIP verifier V_{LIP} sends to the LIP prover P_{LIP} a message $\mathbf{q}$ that is a vector of field elements, depending on C but not on x; V_{LIP} may also output a verification state $\mathbf{u}$. The LIP prover $P_{\mathsf{LIP}}(x, w)$ applies to $\mathbf{q}$ an affine transformation $\Pi = (\Pi', \boldsymbol{b})$, resulting in *only a constant number* of field elements. The prover's message $\mathbf{a} = \Pi' \cdot \mathbf{q} + \boldsymbol{b}$ can then be quickly verified (e.g., with $O(|x|)$ field operations) by V_{LIP}, and the soundness error is at most $O(1/|\mathbb{F}|)$. From here on, we shall use the term LIP to refer to LIPs that adhere to the above template.

LIP complexity measures. Our constructions provide different tradeoffs among several complexity measures of an LIP, which ultimately affect the features of the resulting preprocessing SNARKs. The two most basic complexity measures are the number of field elements sent by the verifier and the number of those sent by the prover. An additional measure that we consider in this work is the *algebraic complexity* of the verifier (when viewed as an $\mathbb{F}$-arithmetic circuit). Specifically, splitting the verifier into a query algorithm Q_{LIP} and a decision algorithm D_{LIP}, we say that it has *degree* (d_Q, d_D) if Q_{LIP} can be computed by a vector of multivariate polynomials of total degree d_Q each in the verifier's randomness, and D_{LIP} by a vector of multivariate polynomials of total degree d_D each in the LIP answers $\mathbf{a}$ and the verification state $\mathbf{u}$. Finally, of course, the running times of the query algorithm, decision algorithm, and prover algorithm are all complexity measures of interest.

As mentioned above, our LIP constructions are obtained by applying general transformations to two types of PCPs. We now describe each of these transformations and the features they achieve. Some of the parameters of the resulting constructions are summarized in Table 1.

LIPs from linear PCPs. A *linear PCP* (LPCP) of length m is an oracle computing a linear function $\boldsymbol{\pi}\colon \mathbb{F}^m \to \mathbb{F}$; namely, the answer to each oracle query $\boldsymbol{q}_i \in \mathbb{F}^m$ is $a_i = \langle \boldsymbol{\pi}, \boldsymbol{q}_i \rangle$. Note that, unlike in an LIP where different affine functions, given by a matrix Π and shift $\boldsymbol{b}$, are applied to a message $\mathbf{q}$, in an LPCP there is one linear function $\boldsymbol{\pi}$, which is applied to different queries. (An LPCP with a *single* query can be viewed as a special case of an LIP.) This difference prevents a direct use of an LPCP as an LIP.

Our first transformation converts any (multi-query) LPCP into an LIP with closely related parameters. Concretely, we transform any k-query LPCP of length m over $\mathbb{F}$ into an LIP with verifier message in $\mathbb{F}^{(k+1)m}$, prover message in $\mathbb{F}^{k+1}$, and the same soundness error up to an additive term of $1/|\mathbb{F}|$. The transformation preserves the key properties of the LPCP, including the algebraic complexity of the verifier. Our transformation is quite natural: the verifier sends $\mathbf{q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{k+1})$ where $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_k$ are the LPCP queries and $\boldsymbol{q}_{k+1} = \alpha_1 \boldsymbol{q}_1 + \ldots + \alpha_k \boldsymbol{q}_k$ is a random linear combination of these. The (honest) prover responds with $a_i = \langle \boldsymbol{\pi}, \boldsymbol{q}_i \rangle$, for $i = 1, \ldots, k+1$. To prevent a malicious prover from using inconsistent choices for $\boldsymbol{\pi}$, the verifier checks that $a_{k+1} = \alpha_1 a_1 + \ldots + \alpha_k a_k$.

By relying on two different LPCP instantiations, we obtain two corresponding LIP constructions:

- A variant of the Hadamard-based PCP of Arora et al. [ALM+98] (ALMSS), extended to work over an arbitrary finite field $\mathbb{F}$, yields a very simple LPCP with three queries. After applying our transformation, for a circuit C of size s and input length n, the resulting LIP for $\mathcal{R}_C$ has verifier message in $\mathbb{F}^{O(s^2)}$, prover message in $\mathbb{F}^4$, and soundness error $O(1/|\mathbb{F}|)$. When viewed as $\mathbb{F}$-arithmetic circuits, the prover P_{LIP} and query algorithm Q_{LIP} are both of size $O(s^2)$, and the decision algorithm is of size $O(n)$. Furthermore, the degree of $(Q_{\mathsf{LIP}}, D_{\mathsf{LIP}})$ is $(2, 2)$.
- A (strong) quadratic span program (QSP), as defined by Gennaro et al. [GGPR12], directly yields a corresponding LPCP with three queries. For a circuit C of size s and input length n, the resulting LIP for $\mathcal{R}_C$ has verifier message in $\mathbb{F}^{O(s)}$, prover message in $\mathbb{F}^4$, and soundness error $O(s/|\mathbb{F}|)$. When viewed as $\mathbb{F}$-arithmetic circuits, the prover P_{LIP} is of size $\widetilde{O}(s)$, the query algorithm Q_{LIP} is of size $O(s)$, and the decision algorithm is of size $O(n)$. The degree of $(Q_{\mathsf{LIP}}, D_{\mathsf{LIP}})$ is $(O(s), 2)$.

A notable feature of the LIPs obtained above is the very low "online complexity" of verification: in both cases, the decision algorithm is an arithmetic circuit of size $O(n)$. Moreover, all the efficiency features mentioned above apply not only to satisfiability of boolean circuits C, but also to satisfiability of $\mathbb{F}$-arithmetic circuits.

In both the above constructions, the circuit to be verified is first represented as an appropriate algebraic satisfaction problem, and then probabilistic checking machinery is invoked. In the first case, the problem is a system of quadratic equations over $\mathbb{F}$, and, in the second case, it is a (strong) quadratic span program (QSP) over $\mathbb{F}$. These algebraic problems are the very same problems underlying [Gro10, Lip12] and [GGPR12].

As explained earlier, [GGPR12] invested much effort to show an efficient reduction from circuit satisfiability problems to QSPs. Our work does *not* subsume nor simplify the reduction to QSPs of [GGPR12], but instead reveals a simple LPCP to check a QSP, and this LPCP can be plugged into our general transformations. Reducing circuit satisfiability to a system of quadratic equations over $\mathbb{F}$ is much simpler, but generating proofs for the resulting problem is quadratically more expensive. (Concretely, both [Gro10] and [Lip12] require $O(s^2)$ computation already in the preprocessing phase).

LIPs from traditional PCPs. Our second transformation relies on traditional "unstructured" PCPs. These PCPs are typically more difficult to construct than LPCPs; however, our second transformation has the advantage of requiring the prover to send only a *single* field element. Concretely, our transformation converts a traditional k-query PCP into a 1-query LPCP, over a sufficiently large field. Here the PCP oracle is represented via its truth table, which is assumed to be a binary string of polynomial size (unlike the LPCPs mentioned above, whose truth tables have size that is exponential in the circuit size). The transformation converts any k-query PCP of proof length m and soundness error ε into an LIP, with soundness error $O(\varepsilon)$ over a field of size $2^{O(k)}/\varepsilon$, in which the verifier sends m field elements and receives only a single field element in return. The high-level idea is to use a sparse linear combination of the PCP entries to pack the k answer bits into a single field element. The choice of this linear combination uses additional random noise to ensure that the prover's coefficients are restricted to binary values, and uses easy instances of subset-sum to enable an efficient decoding of the k answer bits.

Taking time complexity to an extreme, we can apply this transformation to the PCPs of Ben-Sasson et al. [BSCGT12] and get LIPs where the prover and verifier complexity are both optimal up to polylog(s) factors, but where the prover sends a single element in a field of size $|\mathbb{F}| = 2^{\lambda \cdot \text{polylog}(s)}$. Taking succinctness to an extreme, we can apply our transformation to PCPs with soundness error $2^{-\lambda}$ and $O(\lambda)$ queries, obtaining an LIP with similar soundness error in which the prover sends a single element in a field of size $|\mathbb{F}| = 2^{\lambda \cdot O(1)}$. For instance, using the query-efficient PCPs of Håstad and Khot [HK05], the field size is only $|\mathbb{F}| = 2^{\lambda \cdot (3+o(1))}$.[7] (Jumping ahead, this means that a field element can be encrypted using a *single*, *normal-size* ciphertext of homomorphic encryption schemes such as Paillier or Elgamal even when $\lambda = 100$.) On the down side, the

[7] In the case of [HK05], we do not obtain an input-oblivious LIP, because the queries in their PCP depend on the input; while it is plausible to conjecture that the queries can be made input-oblivious, we did not check that.

degrees of the LIP verifiers obtained via this transformation are high; we give evidence that this is inherent when starting from "unstructured" PCPs.

starting point of LIP construction	# field elements in verifier message	# field elements in prover message	algebraic properties of verifier	field size for $2^{-\lambda}$ knowledge error
Hadamard PCP	$O(s^2)$	4	$(d_Q, d_D) = (2, 2)$	$2^{\lambda} \cdot O(1)$
QSPs of [GGPR12]	$O(s)$	4	$(d_Q, d_D) = (O(s), 2)$	$2^{\lambda} \cdot O(s)$
PCPs of [BSCGT12]	$\widetilde{O}(s)$	1	none	$2^{\lambda \cdot \mathrm{polylog}(s)}$
PCPs of [HK05]	$\mathrm{poly}(s)$	1	none	$2^{\lambda \cdot (3+o(1))}$

Table 1: Summary of our LIP constructions.

Honest-verifier zero-knowledge LIPs. We also show how to make the above LIPs zero-knowledge against honest verifiers (HVZK). Looking ahead, using HVZK LIPs in our cryptographic transformations results in preprocessing SNARKs that are zero-knowledge (against malicious verifiers in the CRS model).

For the Hadamard-based LIP, an HVZK variant can be obtained directly with essentially no additional cost. More generally, we show how to transform *any* LPCP where the decision algorithm is of low degree to an HVZK LPCP with the same parameters up to constant factors;this HVZK LPCP can then be plugged into our first transformation to obtain an HVZK LIP. Both of the LPCP constructions mentioned earlier satisfy the requisite degree constraints.

For the second transformation, which applies to traditional PCPs (whose verifiers, as discussed above, must have high degree and thus cannot benefit from our general HVZK transformation), we show that if the PCP is HVZK (see [DFK+92] for efficient constructions), then so is the resulting LIP; in particular, the HVZK LIP answer still consists of a *single* field element.

Proof of knowledge. In each of the above transformations, we ensure not only soundness for the LIP, but also a proof of knowledge property. Namely, it is possible to efficiently extract from a convincing affine function Π a witness for the underlying statement. The proof of knowledge property is then preserved in the subsequent cryptographic compilations, ultimately allowing to establish the proof of knowledge property for the preprocessing SNARK. As discussed in Section 1.1, proof of knowledge is a very desirable property for preprocessing SNARKs; for instance, it enables to remove the preprocessing phase, as well as to improve the complexity of the prover and verifier, via the result of [BCCT12b].

Preprocessing SNARKs from LIPs. We explain how to use cryptographic tools to transform an LIP into a corresponding preprocessing SNARK. At high level, the challenge is to ensure that an arbitrary (yet computationally-bounded) prover behaves as if it was a linear (or affine) function. The idea, which also

implicitly appears in previous constructions, is to use an encryption scheme with targeted malleability [BSW12] for the class of affine functions: namely, an encryption scheme that "only allows affine homomorphic operations" on an encrypted plaintext (and these operations are independent of the underlying plaintexts). Intuitively, the verifier would simply encrypt each field element in the LIP message $\mathbf{q}$, send the resulting ciphertexts to the prover, and have the prover homomorphically evaluate the LIP affine function on the ciphertexts; targeted malleability ensures that malicious provers can only invoke (malicious) affine strategies.

We concretize the above approach in several ways, depending on the properties of the LIP and the exact flavor of targeted malleability; different choices will induce different properties for the resulting preprocessing SNARK. In particular, we identify natural sufficient properties that enable an LIP to be compiled into a *publicly-verifiable* SNARK. We also discuss possible instantiations of the cryptographic tools, based on existing knowledge assumptions. (Recall that, in light of the negative result of [GW11], the use of nonstandard cryptographic assumptions seems to be justified.)

Designated-verifier preprocessing SNARKs from arbitrary LIPs. First, we show that *any* LIP can be compiled into a corresponding *designated-verifier* preprocessing SNARK with similar parameters. (Recall that "designated verifier" means that the verifier needs to maintain a secret verification state.) To do so, we rely on what we call *linear-only* encryption: an additively homomorphic encryption that is (a) semantically-secure, and (b) linear-only. The linear-only property essentially says that, given a public key pk and ciphertexts $\mathsf{Enc}_{\mathsf{pk}}(a_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(a_m)$, it is infeasible to compute a new ciphertext c' in the image of $\mathsf{Enc}_{\mathsf{pk}}$, except by "knowing" $\beta, \alpha_1, \ldots, \alpha_m$ such that $c' \in \mathsf{Enc}_{\mathsf{pk}}(\beta + \sum_{i=1}^{m} \alpha_i a_i)$. Formally, the property is captured by guaranteeing that, whenever $A(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(a_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(a_m))$ produces valid ciphertexts $(c'_1, \ldots, c'_k)$, an efficient extractor E (non-uniformly depending on A) can extract a corresponding affine function Π "explaining" the ciphertexts. As a candidate for such an encryption scheme, we propose variants of Paillier encryption [Pai99] (as also considered in [GGPR12]) and of Elgamal encryption [EG85] (in those cases where the plaintext is guaranteed to belong to a polynomial-size set, so that decryption can be done efficiently). These variants are "sparsified" versions of their standard counterparts; concretely, a ciphertext does not only include $\mathsf{Enc}_{\mathsf{pk}}(a)$, but also $\mathsf{Enc}_{\mathsf{pk}}(\alpha \cdot a)$, for a secret field element α. (This "sparsification" follows a pattern found in many constructions conjectured to satisfy "knowledge-of-exponent" assumptions.) As for Paillier encryption, we have to consider LIPs over the ring $\mathbb{Z}_{\mathfrak{pq}}$ (instead of a finite field $\mathbb{F}$); essentially, the same results also hold in this setting (except that soundness is $O(1/\min\{\mathfrak{p}, \mathfrak{q}\})$ instead of $O(1/|\mathbb{F}|)$).

We also consider a notion of targeted malleability, weaker than linear-only encryption, that is closer to the definition template of Boneh et al. [BSW12]. In such a notion, the extractor is replaced by a simulator. Relying on this weaker variant, we are only able to prove the security of our preprocessing SNARKs against *non-adaptive* choices of statements (and still prove soundness, though

not proof of knowledge, if the simulator is allowed to be inefficient). Nonetheless, for natural instantiations, even adaptive security seems likely to hold for our construction, but we do not know how to prove it. One advantage of working with this weaker variant is that it seems to allow for more efficient candidates constructions. Concretely, the linear-only property rules out any encryption scheme where ciphertexts can be sampled obliviously; instead, the weaker notion does not, and thus allows for shorter ciphertexts. For example, we can consider a standard ("non-sparsified") version of Paillier encryption. We will get back to this point in Section 1.3.

Publicly-verifiable preprocessing SNARKs from LIPs with low-degree verifiers. Next, we identify properties of LIPs that are sufficient for a transformation to *publicly-verifiable* preprocessing SNARKs. Note that, if we aim for public verifiability, we cannot use semantically-secure encryption to encode the message of the LIP verifier, because we need to "publicly test" (without decryption) certain properties of the plaintext underlying the prover's response. The idea, implicit in previous publicly-verifiable preprocessing SNARK constructions, is to use linear-only encodings (rather than encryption) that do allow such public tests, while still providing certain one-wayness properties. When using such encodings with an LIP, however, it must be the case that the public tests support evaluating the decision algorithm of the LIP and, moreover, the LIP remains secure despite some "leakage" on the queries. We show that LIPs with *low-degree verifiers* (which we call *algebraic* LIPs), combined with appropriate one-way encodings, suffice for this purpose.

More concretely, like [Gro10, Lip12, GGPR12], we consider candidate encodings in bilinear groups under similar knowledge-of-exponent and computational Diffie-Hellman assumptions; for such encoding instantiations, we must start with an LIP where the degree d_D of the decision algorithm D_{LIP} is at most quadratic. (If we had multilinear maps supporting higher-degree polynomials, we could support higher values of d_D.) In addition to $d_D \leq 2$, to ensure security even in the presence of certain one-way leakage, we need the query algorithm Q_{LIP} to be of polynomial degree.

Both of the LIP constructions from LPCPs described in Section 1.3 satisfy these requirements. When combined with the above transformation, these LIP constructions imply new constructions of publicly-verifiable preprocessing SNARKs, one of which can be seen as a simplification of the construction of [Gro10] and the other as a reinterpretation (and slight simplification) of the construction of [GGPR12].

Zero-knowledge. In all aforementioned transformations to preprocessing SNARKs, if we start with an HVZK LIP (such as those mentioned in Section 1.3) and additionally require a rerandomization property for the linear-only encryption/encoding (which is available in all of the candidate instantiations we consider), we obtain preprocessing SNARKs that are (perfect) zero-knowledge in the CRS model. In addition, for the case of *publicly-verifiable* (perfect) zero-knowledge prepro-

cessing SNARKs, the CRS can be tested, so that (similarly to previous works [Gro10, Lip12, GGPR12]) we also obtain succinct ZAPs.

New efficiency features for SNARKs. We obtain the following concrete improvements in communication complexity for preprocessing SNARKs.

"Single-ciphertext preprocessing SNARKs". If we combine the LIPs that we obtained from traditional PCPs (where the prover returns only a single field element) with "non-sparsified" Paillier encryption, we obtain (non-adaptive) preprocessing SNARKs that consist of *a single Paillier cipherext*. Moreover, when using the query-efficient PCP from [HK05] as the underlying PCP, even a standard-size Paillier ciphertext (with plaintext group $\mathbb{Z}_{\mathfrak{p}\mathfrak{q}}$ where $\mathfrak{p}, \mathfrak{q}$ are 512-bit primes) suffices for achieving soundness error $2^{-\lambda}$ with $\lambda = 100$. (For the case of [HK05], due to the queries' dependence on the input, the reference string of the SNARK also depends on the input.) Alternatively, using the sparsified version of Paillier encryption, we can also get security against adaptively-chosen statements with only *two Paillier ciphertexts*.

Towards optimal succinctness. A fundamental question about succinct arguments is how low can we push communication complexity. More accurately: what is the optimal tradeoff between communication complexity and soundness? Ideally, we would want succinct arguments that are *optimally succinct*: to achieve $2^{-\Omega(\lambda)}$ soundness against $2^{O(\lambda)}$-bounded provers, the proof length is $O(\lambda)$ bits long.

In existing constructions of succinct arguments, interactive or not, to provide $2^{-\Omega(\lambda)}$ soundness against $2^{O(\lambda)}$-bounded provers, the prover has to communicate $\omega(\lambda)$ bits to the verifier. Concretely, PCP-based (and MIP-based) solutions require $\Omega(\lambda^3)$ bits of communication. This also holds for known preprocessing SNARKs, because previous work and the constructions discussed above are based on bilinear groups or Paillier encryption, both of which suffer from subexponential-time attacks.

If we had a candidate for (linear-only) homomorphic encryption that did not suffer from subexponential-time attacks, our approach could perhaps yield preprocessing SNARKs that are optimally succinct. The only known such candidate is Elgamal encryption (say, in appropriate elliptic curve groups) [PQ12]. However, the problem with using Elgamal decryption in our approach is that it requires, in general, to compute discrete logarithms.

One way to overcome this problem is to ensure that honest proofs are always decrypted to a known polynomial-size set. This can be done by taking the LIP to be over a field $\mathbb{F}_{\mathfrak{p}}$ of only polynomial size, and ensuring that any honest proof $\boldsymbol{\pi}$ has small ℓ_1-norm $\|\boldsymbol{\pi}\|_1$, so that in particular, the prover's answer is taken from a set of size at most $\|\boldsymbol{\pi}\|_1 \cdot \mathfrak{p}$. For example, in the two LPCP-based constructions described in Section 1.3, this norm is $O(s^2)$ and $O(s)$ respectively for a circuit of size s. This approach, however, has two caveats: the soundness of the underlying LIP is only $1/\mathrm{poly}(\lambda)$ and moreover, the verifier's running time is proportional to s, and not independent of it, as we usually require.

A very interesting related question that may lead to a solution circumventing the aforementioned caveats is whether there exist LIPs where the decision algorithm has *linear* degree. With such an LIP, we would be able to directly use Elgamal encryption because linear tests on the plaintexts can be carried out "in the exponent", without having to take discrete logarithms.

Finally, a rather generic approach for obtaining "almost-optimal succintness" is to use (linear-only) Elgamal encryption in conjunction with any linear homomorphic encryption scheme (perhaps not having the linear-only property) that is sufficiently secure. Concretely, the verifier sends his LIP message encrypted under both encryption schemes, and then the prover homomorphically evaluates the affine function on both. The additional ciphertext can be efficiently decrypted, and can assist in the decryption of the Elgamal ciphertext. For example, there are encryption schemes based on Ring-LWE [LPR10] that are conjectured to have quasiexponential security; by using these in the approach we just discussed, we can obtain $2^{-\Omega(\lambda)}$ soundness against $2^{O(\lambda)}$-bounded provers with $\widetilde{O}(\lambda)$ bits of communication.

Strong knowledge and reusability. Designated-verifier SNARKs typically suffer from a problem known as the *verifier rejection problem*: security is compromised if the prover can learn the verifier's responses to multiple adaptively-chosen statements and proofs. For example, the PCP-based (or MIP-based) SNARKs of [BCCT12a, GLR11, DFH12, BC12] suffer from the verifier rejection problem because a prover can adaptively learn the encrypted PCP (or MIP) queries, by feeding different statements and proofs to the verifier and learning his responses, and since the secrecy of these queries is crucial, security is lost.

Of course, one way to avoid the verifier rejection problem is to generate a new reference string for each statement and proof. Indeed, this is an attractive solution for the aforementioned SNARKs because generating a new reference string is very cheap: it costs $\text{poly}(\lambda)$. However, for a designated-verifier *preprocessing* SNARK, generating a new reference string is not cheap at all, and being able to *reuse* the same reference string across an unbounded number of adaptively-chosen statements and proofs is a very desirable property.

A property that is satisfied by all algebraic LIPs , which we call *strong knowledge*, is that such attacks are impossible. Specifically, for such LIPs, every prover either makes the verifier accept with probability 1 or with probability less than $O(\text{poly}(\lambda)/|\mathbb{F}|)$. (In the full version of this paper, we also show that traditional "unstructured" PCPs cannot satisfy this property.) Given LIPs with strong knowledge, it seems that designated-verifier SNARKs that have a reusable reference string can be constructed. Formalizing the connection between strong knowledge and reusable reference string actually requires notions of linear-only encryption that are somewhat more delicate than those we have considered so far.

1.4 Structured PCPs In Other Works

Ishai et al. [IKO07] proposed the idea of constructing argument systems with nontrivial efficiency properties by using "structured" PCPs and cryptographic primitives with homomorphic properties, rather than (as in previous approaches) "unstructured" polynomial-size PCPs and collision-resistant hashing. We have shown how to apply this basic approach in order to obtain succinct non-interactive arguments with preprocessing. We now compare our work to other works that have also followed the basic approach of [IKO07].

Strong vs. weak linear PCPs. Both in our work and in [IKO07], the notion of a "structured" PCP is taken to be a *linear PCP*. However, the notion of a linear PCP used in our work does not coincide with the one used in [IKO07]. Indeed there are two ways in which one can formalize the intuitive notion of a linear PCP. Specifically:

- A *strong* linear PCP is a PCP in which the honest proof oracle is guaranteed to be a linear function, and soundness is required to hold for *all* (including non-linear) proof oracles.
- A *weak* linear PCP is a PCP in which the honest proof oracle is guaranteed to be a linear function, and soundness is required to hold *only* for linear proof oracles.

In particular, a weak linear PCP assumes an algebraically-bounded prover, while a strong linear PCP does not. While Ishai et al. [IKO07] considered strong linear PCPs, in our work we are interested in studying algebraically-bounded provers, and thus consider weak linear PCPs.

Arguments from strong linear PCPs. Ishai et al. [IKO07] constructed a four-message argument system for NP in which the prover-to-verifier communication is short (i.e., an argument with a *laconic* prover [GVW02]) by combining a strong linear PCP and (standard) linear homomorphic encryption; they also showed how to extend their approach to "balance" the communication between the prover and verifier and obtain a $O(1/\varepsilon)$-message argument system for NP with $O(n^{\varepsilon})$ communication complexity. Let us briefly compare their work with ours.

First, in this paper we focus on the non-interactive setting, while Ishai et al. focused on the interactive setting. In particular, in light of the negative result of Gentry and Wichs [GW11], this means that the use of non-standard assumptions in our setting (such as linear targeted malleability) may be justified; in contrast, Ishai et al. only relied on the standard semantic security of linear homomorphic encryption (and did not rely on linear targeted malleability properties). Second, we focus on constructing (non-interactive) succinct arguments, while Ishai et al. focus on constructing arguments with a laconic prover. Third, by relying on weak linear PCPs (instead of strong linear PCPs) we do not need to perform (explicitly or implicitly) linearity testing, while Ishai et al. do. Intuitively, this is because we rely on the assumption of linear targeted malleability, which ensures that a prover is algebraically bounded (in fact, in our case, linear); not having to perform

proximity testing is crucial for preserving the algebraic properties of a linear PCP (and thus, e.g., obtain public verifiability) and obtaining $O(\text{poly}(\lambda)/|\mathbb{F}|)$ soundness with only a constant number of encrypted/encoded group elements. (Recall that linearity testing only guarantees constant soundness with a constant number of queries.)

Turning to computational efficiency, while their basic protocol does not provide the verifier with any saving in computation, Ishai et al. noted that their protocol actually yields a *batching argument*: namely, an argument in which, in order to simultaneously verify the correct evaluation of ℓ circuits of size S, the verifier may run in time S (i.e., in time S/ℓ per circuit evaluation). In fact, a set of works [SBW11, SMBW12, SVP+12, SBV+12] has improved upon, optimized, and implemented the batching argument of Ishai et al. [IKO07] for the purpose of verifiable delegation of computation.

Finally, [SBV+12] have also observed that QSPs can be used to construct weak linear PCPs; while we compile weak linear PCPs into LIPs, [SBV+12] (as in previous work) compile weak linear PCPs into strong ones. Indeed, note that a weak linear PCP can always be compiled into a corresponding strong one, by letting the verifier additionally perform linearity testing and self-correction; this compilation does not affect proof length, increases query complexity by only a constant multiplicative factor, and guarantees constant soundness.

Remark 1.1. The notions of (strong or linear) PCP discussed above should not be confused with the (unrelated) notion of a *linear PCP of Proximity* (linear PCPP) [BSHLM09, Mei12], which we now recall for the purpose of comparison.

Given a field $\mathbb{F}$, an $\mathbb{F}$-*linear circuit* [Val77] is an $\mathbb{F}$-arithmetic circuit $C\colon \mathbb{F}^h \to \mathbb{F}^\ell$ in which every gate computes an $\mathbb{F}$-linear combination of its inputs; its *kernel*, denoted $\ker(C)$, is the set of all $w \in \mathbb{F}^h$ for which $C(w) = 0^\ell$. A *linear PCPP* for a field $\mathbb{F}$ is an oracle machine V with the following properties: (1) V takes as input an $\mathbb{F}$-linear circuit C and has oracle access to a vector $w \in \mathbb{F}^h$ and an auxiliary vector π of elements in $\mathbb{F}$, (2) if $w \in \ker(C)$ then there exists π so that $V^{w,\pi}(C)$ accepts with probability 1, and (3) if w is far from $\ker(C)$ then $V^{w,\pi}(C)$ rejects with high probability for every π.

Thus, a linear PCPP is a proximity tester for the kernels of linear circuits (which are not universal), while a (strong or weak) linear PCP is a PCP in which the proof oracle is a linear function.

References

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

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

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in FOCS '92.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *Proceedings of the 32nd Annual International Cryptology Conference*, CRYPTO '12, pages 255–272, 2012.

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

[BCCT12a] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 326–349, 2012.

[BCCT12b] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKs and proof-carrying data. Cryptology ePrint Archive, Report 2012/095, 2012.

[BCI+12] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Paneth Omer. Succinct non-interactive arguments via linear interactive proofs. Cryptology ePrint Archive, Report 2012, 2012.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.

[BG08] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM Journal on Computing*, 38(5):1661–1694, 2008. Preliminary version appeared in CCC '02.

[BGV11] Siavosh Benabbas, Rosario Gennaro, and Yevgeniy Vahlis. Verifiable delegation of computation over large datasets. In *Proceedings of the 31st Annual International Cryptology Conference*, CRYPTO '11, pages 111–131, 2011.

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

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

[BSCGT12] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. On the concrete-efficiency threshold of probabilistically-checkable proofs, 2012. Electronic Colloquium on Computational Complexity, TR12-045.

[BSHLM09] Eli Ben-Sasson, Prahladh Harsha, Oded Lachish, and Arie Matsliah. Sound 3-query PCPPs are long. *ACM Transactions on Computation Theory*, 1(2):7:1–7:49, 2009. Preliminary version appeared in ICALP '08.

[BSW12] Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: Homomorphic encryption for restricted computations. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 350–366, 2012.

[BV07] Andrej Bogdanov and Emanuele Viola. Pseudorandom bits for polynomials. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 41–51, 2007.

[CKV10] Kai-Min Chung, Yael Kalai, and Salil Vadhan. Improved delegation of computation using fully homomorphic encryption. In *Proceedings of the 30th Annual International Cryptology Conference*, CRYPTO '10, pages 483–501, 2010.

[CMT12] Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Practical verified computation with streaming interactive proofs. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 90–112, 2012.

[CRR11] Ran Canetti, Ben Riva, and Guy N. Rothblum. Two 1-round protocols for delegation of computation. Cryptology ePrint Archive, Report 2011/518, 2011.

[CTY11] Graham Cormode, Justin Thaler, and Ke Yi. Verifying computations with streaming interactive proofs. *Proceedings of the VLDB Endowment*, 5(1):25–36, 2011.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In *Proceedings of the 11th Annual International Cryptology Conference*, CRYPTO '92, pages 445–456, 1992.

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

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In *Proceedings of the 9th Theory of Cryptography Conference*, TCC '12, pages 54–74, 2012.

[DFK+92] Cynthia Dwork, Uriel Feige, Joe Kilian, Moni Naor, and Shmuel Safra. Low communication 2-prover zero-knowledge proofs for NP. In *Proceedings of the 11th Annual International Cryptology Conference*, CRYPTO '92, pages 215–227, 1992.

[DGW09] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

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

[EG85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FG12] Dario Fiore and Rosario Gennaro. Publicly verifiable delegation of large polynomials and matrix computations, with applications. Cryptology ePrint Archive, Report 2012/281, 2012.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Proceedings of the 6th Annual International Cryptology Conference*, CRYPTO '87, pages 186–194, 1987.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: outsourcing computation to untrusted workers. In *Proceedings of the 30th Annual International Cryptology Conference*, CRYPTO '10, pages 465–482, 2010.

[GGPR12] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. Cryptology ePrint Archive, Report 2012/215, 2012.

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

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

[GLR11] Shafi Goldwasser, Huijia Lin, and Aviad Rubinstein. Delegation of computation without rejection problem from designated verifier CS-proofs. Cryptology ePrint Archive, Report 2011/456, 2011.

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

[GR05] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 407–418, 2005.

[Gro10] Jens 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*, ASIACRYPT '10, pages 321–340, 2010.

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

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC '11, pages 99–108, 2011.

[HK05] Johan Håstad and Subhash Khot. Query efficient PCPs with perfect completeness. *Theory of Computing*, 1(1):119–148, 2005.

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

[IKO07] Yuval Ishai, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient arguments without short PCPs. In *Proceedings of the Twenty-Second Annual IEEE Conference on Computational Complexity*, CCC '07, pages 278–291, 2007.

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

[KR08] Yael Kalai and Ran Raz. Interactive PCP. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, ICALP '08, pages 536–547, 2008.

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

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *Proceedings of the 9th Theory of Cryptography Conference*, TCC '12, pages 169–189, 2012.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '10, pages 1–23, 2010.

[Mei12] Or Meir. Combinatorial PCPs with short proofs. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, CCC '12, 2012.

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

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

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

[NN90] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, STOC '90, pages 213–223, 1990.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference On Theory And Application Of Cryptographic Techniques*, EUROCRYPT '99, pages 223–238, 1999.

[PQ12] Christophe Petit and Jean-Jacques Quisquater. On polynomial systems arising from a Weil descent. In *Proceedings of the 18th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '12, 2012.

[SBV+12] Srinath Setty, Benjamin Braun, Victor Vu, Andrew J. Blumberg, Bryan Parno, and Michael Walfish. Resolving the conflict between generality and plausibility in verified computation. Cryptology ePrint Archive, Report 2012/622, 2012.

[SBW11] Srinath Setty, Andrew J. Blumberg, and Michael Walfish. Toward practical and unconditional verification of remote computations. In *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*, HotOS '13, pages 29–29, 2011.

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

[SMBW12] Srinath Setty, Michael McPherson, Andrew J. Blumberg, and Michael Walfish. Making argument systems for outsourced computation practical (sometimes). In *Proceedings of the 2012 Network and Distributed System Security Symposium*, NDSS '12, 2012.

[SVP+12] Srinath Setty, Victor Vu, Nikhil Panpalia, Benjamin Braun, Andrew J. Blumberg, and Michael Walfish. Taking proof-based verified computation a few steps closer to practicality. In *Proceedings of the 21st USENIX Security Symposium*, Security '12, 2012.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical Foundations of Computer Science*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. 1977.

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