Succinct Arguments from Multi-Prover Interactive Proofs and their Efficiency Benefits

Nir Bitansky[⋆] and Alessandro Chiesa

[1] TAU, nirbitan@tau.ac.il
[2] MIT, alexch@csail.mit.edu

Abstract. *Succinct arguments of knowledge* are computationally-sound proofs of knowledge for NP where the verifier's running time is independent of the time complexity of the NP nondeterministic machine for the considered language.

Existing succinct argument constructions are, typically, based on techniques that combine cryptographic hashing and probabilistically-checkable proofs (PCPs), and thus, in light of today's state-of-the-art PCP technology, are quite inefficient: either one uses long PCP proofs with lots of redundancy to make the verifier fast but at the cost of making the prover slow, or one uses short PCP proofs to make the prover fast but at the cost of making the verifier slow.

To obtain better efficiency, we propose to investigate the alternative approach of constructing succinct arguments based on multi-prover interactive proofs (MIPs) and stronger cryptographic techniques:

(1) We construct a one-round succinct MIP of knowledge protocol where (i) each prover is highly efficient in terms of time AND space, and ALSO (ii) the verifier is highly efficient.

(2) We show how to transform any one round MIP protocol to a succinct four-message argument (with a single prover), while preserving the time and space efficiency of the original MIP protocol.

As a main tool for this transformation, we construct a *succinct multi-function commitment* that (a) allows the sender to commit to a vector of functions in time and space complexity that are essentially the same as those needed for a single evaluation of the functions, and (b) ensures that the receiver's running time is essentially independent of the function. The scheme is based on fully-homomorphic encryption (and no additional assumptions are needed for our succinct argument).

(3) In addition, we revisit the problem of *non-interactive* succinct arguments of knowledge (SNARKs), where known impossibilities rule out solutions based on black-box reductions to standard assumptions. We formulate a natural (though non-standard) variant of homomorphic encryption that has a *homomorphism-extraction property*. We then show that his primitive essentially allows to "squash" our interactive protocol, while again preserving time and space efficiency. We further show that this variant is, in fact, implied by the existence of SNARKs.

[⋆] This research was done while visiting Boston University and funded by the Check Point Institute for Information Security, by Marie Curie grant PIRG03-GA-2008-230640, an ISF grant 0603805843, and the Fulbright program.

1 Introduction

Interactive proofs & succinctness. Interactive proofs [GMR89] are central to modern cryptography and complexity theory. One extensively-studied aspect of interactive proofs is their expressibility; this study culminated with the celebrated result that $\mathsf{IP} = \mathsf{PSPACE}$ [Sha92]. Another aspect, which is the focus of this work, is that proofs for NP-statements can potentially be verified much faster than by directly checking an NP witness.

Unfortunately, in interactive proofs with statistical soundness, any non-trivial savings in verification time is unlikely (see, e.g., [BHZ87, GH98, GVW02, Wee05]). However, if we settle for proof systems with only *computational* soundness (also known as argument systems [BCC88]), then significant savings can be made.

Indeed, using collision-resistant hash functions (CRHs) and probabilistically-checkable proofs (PCPs) [BFLS91], Kilian [Kil92] showed a four-message interactive argument where 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 and a valid witness, p is a fixed polynomial independent of L, and k is a security parameter. Following tradition, we call such argument systems *succinct*.

A natural strengthening of computational soundness is (computational) *proof of knowledge*: it requires that, when 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. Proof of knowledge is a natural property (satisfied by most proof system constructions, including the aforementioned one of Kilian [BG08]) that is very useful in many applications of succinct arguments.

A special case of succinct arguments that has received a lot of attention is the one of succinct *non-interactive* arguments of knowledge (SNARKs). Indeed, SNARKs are known for their powerful applications including: non-interactive delegation of computation, succinct non-interactive secure computation, extractable cryptographic primitives [BCCT11], and constructions of proof-carrying data [CT10, BCCT12].

1.1 Existing Succinct Arguments and their Efficiency

Kilian's four-message succinct argument of knowledge works as follows: the prover first uses a Merkle hash tree to bind itself to a polynomial-size PCP oracle for the statement to be proven, and then answers the PCP verifier's queries while demonstrating consistency with the previous Merkle tree commitment.

Most SNARK constructions [Mic00, DCL08, BCCT11, DFH11, GLR11] are based on techniques for "squashing" Kilian's protocol into a non-interactive one, and hence are also based on the "commit to a PCP oracle and then reveal" paradigm.[3]

Room for improvement. The current state of affairs is unsatisfying in two respects: on the one hand, from an "understanding perspective" it is unsatisfying that the only

[3] An exception are the SNARKs constructed in [BCCT12], which rely on recursive composition and "bootstrapping" of SNARKs with an expensive offline phase [Gro10, Lip11, GGPR12]; indeed, these SNARKs do not invoke the PCP theorem. Unfortunately the techniques of [BCCT12] do not seem to the extend to the interactive setting.

way we know how to build succinct arguments (at least from standard assumptions) is through Kilian-type protocols; on the other hand, from a practical perspective, this lack of alternative constructions is forcing us to use PCPs, and thus compromise on efficiency — indeed, the concrete efficiency of PCPs is currently poorly understood (though recent progress was shown in [BSCGT12b, BSCGT12a]).

Thus, there is room to consider alternative succinct argument constructions that are potentially more efficient. Moreover, even for the weaker (but still very desirable) goal of delegating deterministic polynomial-time computations (rather than NP computations), we *also* do not have solutions with satisfying efficiency. Any progress on the practicality of succinct arguments will have direct implications for the practicality of delegation protocols (which can naturally be constructed based on succinct arguments).

In light of the above discussion, we consider the following questions:

- *Can we obtain succinct arguments via a construction not of the Kilian type?*
- *Can we avoid the inefficiencies of PCPs?*

The question of whether succinct arguments can be built by using tools that are "lighter" than polynomial-size PCPs (perhaps at the expense of relying on stronger cryptographic primitives) was raised by Ishai et al. [IKO07]. Specifically, Ishai et al. showed how to use Hadamard-based PCPs (together with additively-homomorphic encryption) to obtain simpler but only "semi-succinct" arguments (i.e., arguments where only the prover-to-verifier communication is succinct). We follow the same path and seek techniques for obtaining "fully-succinct" arguments via constructions that are simpler, avoid the use of polynomial-size PCPs, and are potentially more efficient.

Note that a potential obstruction to achieving our goal is that Rothblum and Vadhan [RV09] proved that PCPs are in a certain sense inherent to succinct argument constructions: they showed that any succinct argument (even if interactive) can be transformed into a PCP, as long as its security is established via a black-box reduction to standard cryptographic assumptions. So perhaps the inefficiencies of PCPs cannot be avoided.

The transformation of Rothblum and Vadhan, however, incurs significant overhead in the general case. Thus, it is *still* possible for there to exist a succinct argument construction that is more efficient than any construction directly relying on PCPs (e.g., a Kilian-type one); this holds *even* if such a construction induces a corresponding PCP. Looking ahead, one of the results of this paper is that this is indeed the case.

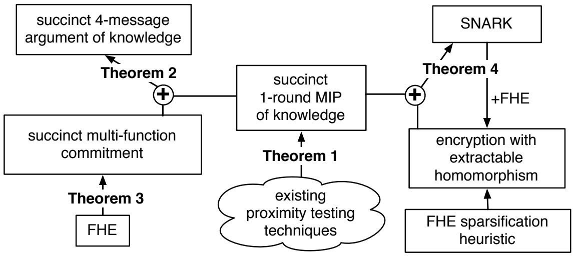
succinct 4-message argument of knowledge
Theorem 2
succinct multi-function commitment
Theorem 3
FHE
succinct 1-round MIP of knowledge
Theorem 1
existing proximity testing techniques
SNARK
Theorem 4
+FHE
encryption with extractable homomorphism
FHE sparsification heuristic

Fig. 1. Summary of our results.

2 Summary of Our Results

At high-level, we show how PCPs in succinct arguments can be replaced by *multi-prover interactive proofs* (MIPs), when combined with the proper cryptographic tools, thus achieving new constructions with several efficiency benefits. Specifically (and as summarized in Figure 1):

We revisit the MIP model and prove that, using existing proximity testing techniques:

> **Theorem 1**: "There is a one-round succinct MIP of knowledge where, to prove correctness of a t-time s-space computation, every prover runs in $t \cdot \mathrm{polylog}(t)$ time and $s \cdot \mathrm{polylog}(t)$ space, and the verifier runs in $\mathrm{polylog}(t)$ time."

Our construction does not hide any large constants and is quite simple and efficient. We then proceed to the task of trying to use our first theorem to construct succinct arguments (with a single prover) that are potentially more efficient than those constructed via PCP-based techniques. We show that:

> **Theorem 2**: "Assuming the existence of fully-homomorphic encryption, there exists a four-message succinct argument of knowledge with the same efficiency as our MIP protocol, up to polynomial factors in the security parameter."

We stress that succinct arguments with the time and space efficiency as in our construction are *not* known to be achievable via Kilian-type constructions that use PCPs.

The main tool in the above theorem is a *succinct multi-function commitment*, which enables a sender to commit to a vector of functions in time and space complexity that are essentially the same as those needed for a single evaluation of the functions, and where the receiver's running time is essentially independent of the function.

> **Theorem 3**: "Assuming the existence of fully-homomorphic encryption, there is a succinct multi-function commitment."

In addition, we explore methods to construct SNARKs based on MIPs. Here, known impossibilities rule out solutions based on black-box reductions to standard assumptions [GW11]. We formulate and study a natural (but non-standard) variant of homomorphic encryption that we call *encryption with extractable homomorphism*, and suggest a candidate construction for this primitive. We show that:

> **Theorem 4**: "Assuming the existence of fully-homomorphic encryption with extractable homomorphism, there exists a SNARK (with the same efficiency as our MIP protocol, up to polynomial factors in the security parameter). Furthermore, such encryption schemes are implied by the existence of SNARKs."

In the following sections, we discuss and explain in somewhat more detail our results. For technical details, see the full version of this paper [BC12].

3 A Simple 1-Round Succinct MIP of Knowledge

A beautiful proof model that has not played so far a major role in the development of succinct arguments is that of *multi-prover* interactive proofs (or MIP protocols), originally introduced by Ben-Or et al. [BOGKW88]. In this model, a verifier conducts a simultaneous interactive protocol with several provers that, crucially, are assumed to be unable to communicate during the protocol.

Because MIPs have not been studied with an eye towards concrete efficiency (e.g., the construction in [BFL90] does not seem so practical), we revisit MIPs and show that, despite the fact that PCPs can be used to construct MIPs and vice versa [TS96], our present ability to exhibit practical constructions for the two is vastly different. Specifically, when studying the problem of making PCPs very efficient, we are faced with two difficult challenges:

- **Proof length vs soundness tradeoff.** We know how to construct either PCPs with great soundness but proofs that are long [RS97, MR08] (and thus with fast verifiers but slow provers), or PCPs with short proofs but soundness that is low [BSS08, BSGH^{+}05] (and thus with not-as-slow provers but quite slow verifiers — due to the large number of repetitions needed to achieve, say, constant soundness).

  Obtaining a PCP that *simultaneously* has short proof length and great soundness is an exciting open problem (though [BSCGT12b] show recent progress).
- **Prover high space complexity.** We know of PCPs where the prover runs in $\tilde{O}(t)$ time but such running time is achieved via the use of FFT-like methods [BSCGT12b], which unfortunately demand $\Omega(t)$ space — essentially, the prover is asked to evaluate a $\Omega(t)$-degree polynomial over a domain of $\Omega(t)$ size. For large t, space usage is a severe problem in practice (more so than time is).

  One may wonder if a prover could do better in its space usage. After all, naïvely verifying a theorem y with witness w may take time t but only space s with $s \ll t$. This is for example what we would expect if y encodes the correctness of some program that runs, given input w, for a long time on a single computer. Thus, it is reasonable to ask whether we can have PCP provers with space complexity that is only $O(s)$ (or even $s \cdot \mathrm{polylog}(t)$) instead of $\Omega(t)$.

  A natural suggestion to alleviate the space requirements would be to allow the prover to naïvely evaluate the polynomial at every single one of the $\Omega(t)$ points in the hopes that, by not using FFTs, it could run in smaller space. This suggestion could go through as long as the "witness reduction" from w to the polynomial to evaluate could be done in polylog space; this is for example the case by using the computational Levin reduction[4] from random-access machines of [BSCGT12a] (which relies on the existence of collision-resistant hash functions).

  However, while the resulting prover would run in polylog space, its running time would now be $\Omega(t^2)$ — too slow.

On the other hand, we show that:

[4] Levin reductions guarantee that there is not only an instance reduction, but a witness reduction that "goes both ways"; this ensures that proof of knowledge is preserved.

Theorem 1. *There is a one-round succinct MIP of knowledge with $O((\log t)^2)$ number of provers and constant soundness error, where:*

- *the verifier is very efficient, AND*
- *each honest prover runs in quasilinear time.*

Moreover, by using a computational reduction (so to obtain a Multi-Prover Interactive Argument), each honest prover runs in BOTH quasilinear time and polylog space.

While the above theorem may be quite surprising (since PCPs with similar efficiency as our construction are not known), it has a natural high-level explanation of how it circumvents the two difficulties afflicting PCPs:

- **NO proof length vs soundness tradeoff.** With MIPs, there is no such thing as proof length, because the provers are *functions*. In other words, provers do not have to write down long proofs, but only answer specific questions of the verifier, which amounts to just a few evaluations of certain polynomials. Thus, we can design a simple and efficient MIP protocol by using the proximity testing tools at the basis of PCPs with long proofs and relatively high soundness, because we will not be paying for proof length. For example, we can leverage the high soundness of the Subspace vs. Point Test of Raz and Safra [RS97] (and its detailed analysis of Moshkovitz and Raz [MR08]) without worrying about the field size or proximity proof length.
- **NO prover with high space complexity.** Each honest prover is asked only a few evaluations of a polynomial, and thus naïve evaluation of the polynomial suffices; this, coupled with the appropriate (computational) reduction of [BSCGT12a] from random-access machines, will yield provers that only require polylog space.

Our construction ensures that the verifier is "adaptive" (namely, can generate queries without knowing in advance the theorem to be proved) and also ensures a proof of knowledge. Both properties play an important role in the other results in this paper.

Construction outline. We follow a paradigm of probabilistic checking that is by now standard. We identify a convenient succinct *algebraic* constraint satisfaction problem (involving properties of polynomials) and then build on a low-degree proximity test to construct a probabilistic verifier (in our case, relying on the help of multiple non-communicating provers) for this problem.[5]

More concretely, the main ingredients of our construction are the low-degree test of Raz and Safra [RS97] for the Reed-Muller code (i.e., multivariate low-degree polynomials), a technique of Ben-Sasson and Sudan [BSS08, Lemma 4.5] for transforming a low-degree test for the Reed-Muller code to a low-degree test for the *Vanishing* Reed-Muller code (namely, the subcode consisting of those polynomials vanishing on a certain subset of the field), and a simple consistency check.

The technical challenge when constructing an MIP protocol instead of a PCP is that provers are functions and not strings, so that querying a given function at several locations becomes problematic. In our construction, we show how to "distribute" the

[5] Of course, one must also ensure that there are sufficiently efficient reductions from the universal language on random-access machines to this problem. That this is the case is not clear a priori. Nonetheless, we observe that the aforementioned reduction of [BSCGT12a] followed by an arithmetization of [Har04] will suffice.

various functions across as few provers as possible while at the same time ensuring that we do not have to query any given alleged function more than once. (Actually, we will have to query a function more than once only for the consistency check, and additional care will be needed to do so.)

For details, see full version of the paper.

Implementing MIPs with a single prover. Our MIP construction is simple and efficient, and so it suggests the possibility of obtaining alternative succinct argument constructions that are more efficient than PCP-based ones; thus, the question is:

Can we "implement" the MIP model with a *single prover* using cryptographic means, in a way that preserves its efficiency benefits?

(In particular, we are "not allowed" to deduce a PCP system [TS96] and rely on previous Kilian-type constructions!) We show two results in this direction, respectively described in Section 4 and Section 5.

Prior MIP work. The question of implementing the MIP model for the purpose of constructing succinct arguments was asked by Dwork et al. [DLN$^+$04] with the motivation being non-interactivity (rather than efficiency). Specifically, after pointing out the unsoundness of the proposed SNARK protocol of Aiello et al. [ABOR00], Dwork et al. suggested that an approach to constructing SNARKs would be to implement the MIP model by encrypting the query for each MIP prover using independent PIR instances and send all the encrypted queries to the same prover. They point out that for this approach to work (without additional assumptions on the MIP), the PIR should be immune to a specific class of attacks, called "spooky interactions". They were not able to prove any PIR scheme to have this property, nor exhibit a counterexample. By now we know that such PIR schemes cannot be constructed via black-box reductions to falsifiable assumptions [GW11], but such PIR schemes are implied by the existence of SNARKs.

Later, Ishai et al. [IKO07] constructed a compiler from *linear* MIP protocols to four-message interactive arguments by leveraging *commitments with linear decommitment* (which can be obtained, e.g., from additively-homomorphic encryption schemes); by plugging into their compiler a linear MIP based on the Hadamard code, Ishai et al. obtained an argument where the prover-to-verifier communication is small, but the verifier running time is not succinct.

4 Four-Message Succinct Arguments from MIPs

A simple but wasteful solution. Given a one-message succinct MIP (of knowledge) protocol and a collision-resistant hash family, it is easy to construct a four-message succinct argument (of knowledge): in the first message the verifier sends to the prover a seed for a collision-resistant hash; then the prover commits (separately and in parallel) to the evaluation table of each MIP prover by sending a Merkle commitment for each; then the verifier sends to the prover the desired message for each MIP prover, and finally the prover replies with an answer to each message, accompanied by an authentication path relative to the appropriate root. Committing to each evaluation table of each MIP prover before any messages are sent by the verifier ensures that a malicious prover cannot "correlate" its answers depending on the messages sent by the verifier in the following message. In other words, one can simply run a copy of Kilian's protocol for each MIP prover, in parallel.

However, the above approach *does not preserve* the efficiency properties of the MIP protocol. For example, suppose that the first honest prover P_1 of the protocol is a function $f\colon A \to B$ that requires time t to evaluate at a single point. Then the above approach would require the prover to evaluate f at every point of A, which could in principle require as much time as $|A| \cdot t$. While the MIP protocol could have started out as quite efficient, now having to "think" of each prover as a table of values, instead of as a function, could sink the resulting construction into complete impracticality. (For example our MIP construction would indeed become too slow if we were to evaluate each prover function everywhere on its domain, even if the resulting string is "only" of polynomial size.[6])

Even if there are algorithms for evaluating f everywhere on A that are faster than the naïve "point-by-point" evaluation, these faster algorithms may come with hefty space requirements compared to the space requirement of a single evaluation of f. (As discussed in the previous section, this is for example the case when f is a polynomial and we seek to evaluate it faster, at many points, via the use of FFT techniques.)

Thus, the approach of using Merkle trees to implement the MIP protocol does not seem to take us any further, in terms of efficiency, than previous PCP-based protocols.

Ideally, we would want a way to implement the MIP protocol that *preserves* its efficiency: namely, the resulting succinct argument prover and verifier should have time and space complexities that are the same as in the corresponding MIP protocol, up to $\mathrm{poly}(k)$ factors, where k is the security parameter (and ideally $\mathrm{poly}(k)$ is also small.)

With the above goal in mind, we prove the following result:

Theorem 2. *Assuming the existence of fully-homomorphic encryption, there exists an* *efficiency-preserving* *compiler that transforms a given succinct MIP of knowledge into a corresponding four-message succinct argument of knowledge.*[7] *(Furthermore, the resulting succinct argument is "universal" in the sense that there is* one *protocol for* all NP *languages.)*

Our efficiency-preserving solution. The tool that enables the above theorem is a generic construction of a *Succinct Multi-Function Commitment* (SMFC), a commitment that is, informally, an "efficiency-preserving analogue" of a Merkle tree commitment for functions. We next elaborate on this construction.

Generalizing the idea of commitments with linear decommitment of Ishai et al. [IKO07] to arbitrary functions, we define an SMFC to be a commitment scheme that

[6] Furthermore, for functions with superpoly-size domain, writing out the evaluation table of the function is simply not feasible! (For example, this is the case in [IKO07].)

[7] More precisely, the preservation of efficiency holds as long as each MIP prover has a sufficiently tight *deterministic* reduction to circuits. This technical condition is quite mild, and does hold for our MIP construction.

works as follows: given a vector of ℓ functions $\vec{f}\colon A \to B$ where f_i can be evaluated in time t_i:[8]

- During a commitment phase, the sender and receiver interact; at the end of this phase, the sender has committed to the vector of functions $\vec{f}$; both parties maintain a private state for the next phase.
- During a decommitment phase, the receiver may request the value of $\vec{f}$ at a vector $\vec{\alpha} \in A^\ell$ by interacting with the receiver. At the end of this phase, the receiver either outputs $\vec{\beta} \in B^\ell$ or $\perp$.

The commitment guarantees that, if both parties are honest, $\vec{\beta} = \vec{f}(\vec{\alpha})$; moreover, it has the following *computational binding* property: for any efficient malicious sender, after the commitment phase, there is some vector of functions $\vec{f}^*$ such that, for any query $\vec{\alpha}$, the receiver either outputs $\vec{f}^*(\vec{\alpha})$ or rejects (except with negligible probability).

The commitment is *succinct* in the sense that the sender runs in time $\text{poly}(k, \vec{t})$ and the receiver in time $\ell \cdot (\log |A| + \log |B|) \cdot \text{poly}(k)$. Crucially, the receiver running time does not depend on the size of the description or running time of $\vec{f}$.

We do not make the additional requirement that the commitment is hiding (although our commitment can be easily enhanced to also be function-hiding).

Theorem 3. *Assuming the existence of fully-homomorphic encryption, there is a succinct multi-function commitment. Moreover, if each function in $\vec{f}$ can be computed "gate-by-gate" as a circuit by an evaluator algorithm in time t and space s, then the sender runs in time $\ell \cdot t \cdot \text{poly}(k)$ and space $\ell \cdot s \cdot \text{poly}(k)$ — we call such a property* strongly *succinct.*[9]

Succinct arguments from multi-function commitments. Before moving on to describe some of the details behind Theorem 3, we briefly describe how to obtain succinct arguments given succinct multi-function commitments. The protocol essentially consists of two phases: in the first, the prover commits to ℓ functions corresponding to the provers $P_1, \ldots, P_\ell$ given by the one round MIP protocol. Then, in the second phase the verifier produces the queries $q_1, \ldots, q_\ell$ to the MIP provers, and asks for the corresponding decommitments. In case all the decommitments are valid, and the MIP verifier is satisfied by the corresponding answers, the verifier accepts.

The fact that the prover is committed in advance to each one of the ℓ functions, ensures that it cannot correlate the answers according to the joint vector of queries, allowing to prove security in a rather direct way.

The second part of Theorem 3 is what enables us, through our concrete MIP construction in the proof of Theorem 1, to perform the above transformation in an efficiency-preserving way, as claimed in Theorem 2.

[8] We can of course consider functions with different domains and ranges, but for simplicity here we set them equal.

[9] Succinct multi-function commitments can of course be constructed generically using succinct arguments of knowledge for NP together with Merkle hashing, but the efficiency of existing succinct argument constructions is not good enough to imply the strong efficiency properties that we need for succinct multi-function commitments (captured by the *strong* succinctness of the second part of the theorem).

We now provide more details regarding our succinct multi-function commitments (given by Theorem 3) and then elaborate on how efficiency is preserved.

The first step towards a multi-function commitment scheme is noting that it is enough to construct a single-function commitment. Once this is achieved a commitment to a vector of functions is done by independently committing to each one of the functions. Hence, we focus on the single-function case.

The starting point of our construction is the cut-and-choose delegation scheme of Chung et al. [CKV10], which works as follows:

- given a function f (to be delegated), during a setup phase, the verifier generates k independent encryptions $c_1, \ldots, c_k$ of 0 and then homomorphically computes

$$\hat{c}_1 := \mathsf{Eval}(f, c_1), \ldots, \hat{c}_k := \mathsf{Eval}(f, c_k) \ ;$$

- afterwards, during the "online" phase, if the verifier wishes to delegate the computation of f on a given input x, the verifier will encrypt k times the input x to obtain $c_{k+1}, \ldots, c_{2k}$, and will send, in random order, $c_1, \ldots, c_{2k}$ to the prover;
- the prover should then homomorphically evaluate f on each ciphertext to obtain $\hat{c}'_1, \ldots, \hat{c}'_{2k}$ (in some permuted order) and send them to the verifier; and
- the verifier will check that $\hat{c}'_1 = \hat{c}_1, \ldots, \hat{c}'_k = \hat{c}_k$ and that $\hat{c}'_{k+1}, \ldots, \hat{c}'_{2k}$ all decrypt to the same value.

Crucially, the verifier's check on the challenges is done "on the ciphertexts"; this enables the security reduction to go through (and there is an attack if the check is instead performed on the underlying ciphertexts).

In our setting, we do not have a prover and a weak verifier, but a sender and a weak receiver. In particular, we are not interested in the sender computing a specific function f, but instead we are interested in the sender committing to *some* function (more precisely, vector of functions). Furthermore, we are not willing to allow the receiver to engage with the sender in an expensive setup phase.

We show that that in a sense we can "delegate" the expensive setup phase of the Chung et al. protocol to obtain an interactive function commitment protocol where the receiver is completely succinct. More precisely, because the sender is the one deciding the function to compute, we can simply ask him during the first round to evaluate the challenges $\hat{c}_1 := \mathsf{Eval}(f, c_1), \ldots, \hat{c}_k := \mathsf{Eval}(f, c_k)$ himself for some function f of his choosing; of course, we cannot do this "in the clear" (as the sender would see the secret challenges), so we simply conduct the first round under another layer of fully-homomorphic encryption. In the resulting protocol, the receiver is indeed "fully succinct".

While the intuition for the security of our modified protocol is strong, proving its binding property does not appear to be trivial. Our security reduction works as follows:

- We first prove a "weak" binding property for a non-amplified version of the protocol. We show that an adversary that is able, with high probability, to open to two different values of the function for the same query during two independent invocations of the decommitment phase can be used to break semantic security in a certain two-prover game.

- We then augment the weakly binding protocol so that we can apply a parallel repetition theorem for interactive arguments (such as [Hai09] or [CL10]), and get a protocol that is strongly binding. This is done by also having the decommitment phase executed under a (second) layer of fully-homomorphic encryption. Unlike typical commitments, where this transformation is rather straight forward, in our case this transformation turns out to be more involved, because the decommitment phase is interactive (and naively making it non-interactive causes an undesired computational overhead).

It may be surprising that, while the security reduction of the Chung et al. [CKV10] protocol is quite straightforward, a simple (and "direct") proof of security for our protocol seems much harder to find. Perhaps the increased difficulty may be attributed to the fact that the Chung et al. [CKV10] protocol only has one round and in this case parallel repetition is, typically, "simpler" [CHS05]; in contrast, in our case proving security of the protocol amounts to proving a special case of parallel repetition for interactive arguments with at least four messages for a protocol that is not of the public coin type (or more generally, one for which sampling random protocol continuations [Hai09] is quite challenging). For more details on the construction and its proof of security the reader is referred to the full version of this paper [BC12].

Prover complexity. The sender in our SMFC protocol is required to (doubly) homomorphically evaluate each function in the vector $\vec{f}$; in general, we do not know how to homomorphically evaluate a function that can be evaluated in time t in less than $t^2 \cdot \mathrm{poly}(k)$ time [BSCGT12a], and that is why, for general functions, the running time of the sender of our protocol is only bounded by $\mathrm{poly}(k, \vec{t})$.

If, however, we apply our SMFC protocol to a vector of functions $\vec{f}$ where we do know that every function can be computed by a circuit of size at most t, then we can improve the time bound of the sender to $\ell \cdot t \cdot \mathrm{poly}(k)$. While it is clear that a circuit C can be homomorphically evaluated in time $|C| \cdot \mathrm{poly}(k)$, it is not as immediate that this is *also* the case for *double* homomorphic evaluation (and this fact is what enables the better time bound); indeed, the homomorphic evaluation algorithm is in fact two "local algorithms", homomorphic addition and multiplication over $\mathbb{F}_2$, iteratively applied to the gates of the circuit; each of these local algorithms can be generically reduced with a quadratic blow up to a corresponding circuit, but this time around we do not mind the quadratic blow up because the running time that we are squaring is only $\mathrm{poly}(k)$ (and not, say, dependent on $|C|$).

Furthermore, if we also know that each function in our vector of functions $\vec{f}$ can be evaluated as a circuit in time t and space s (by some "gate-by-gate" evaluator algorithm), then we can also bound the space complexity of the sender by $\ell \cdot s \cdot \mathrm{poly}(k)$. Whenever $s \ll t$ (for example, when the circuits computing the functions have a somewhat succinct representation), this better space bound is very attractive.

Thus, overall, in our application of the SMFC protocol in Theorem 2, we use homomorphic encryption in a way that enables us to preserve the efficiency of the MIP protocol we construct in Theorem 1. (Though, of course that current FHE constructions are still quite inefficient in practice.) For details, see full version of the paper [BC12].

Interpretation. As discussed in the introduction, the motivation of Ishai et al. [IKO07] was to construct simpler succinct arguments by using simpler probabilistic checking tools, possibly at the expense of stronger cryptographic assumptions. (In their case, they relied on additively-homomorphic encryption instead of only collision-resistant hash functions.) Because both MIPs and PCPs based on Hadamard codes are very simple, working with Hadamard codes is a natural choice. However, techniques based on (the exponentially-long) Hadamard codes are somehow "too simple": their simplicity comes from great proximity testing properties at the expense of a very poor rate; this poor rate makes it very difficult to construct succinct arguments (as merely producing a query indexing into the code is as expensive as the entire computation).

One interpretation of our Theorem 1 and Theorem 2 is that Ishai et al. in a sense "overshot", and one can already find great simplicity while at the same time succeed at obtaining succinct arguments by using standard Reed-Muller proximity testing techniques to construct an MIP protocol and then implement it (based on the stronger cryptographic assumption of fully-homomorphic encryption).

We note that, unlike Kilian's protocol, the interactive protocol we obtain in Theorem 2 is *not* of the public-coin type. It is an interesting open question to understand if there is an efficiency-preserving transformation from MIPs that yields a public-coin succinct argument. (Note that this question is interesting even in the random-oracle model, where it is also not clear how to obtain a public-coin function commitment!)

5 SNARKs from MIPs

Having discussed in Section 4 how to construct an interactive succinct argument from an MIP protocol, we next consider the natural question of whether we can succeed in the analogous task of constructing a succinct *non-interactive* argument of knowledge (SNARK) from an MIP protocol.

5.1 Known SNARK constructions

Before we discuss our results in this direction, we briefly recall existing constructions.

In the random-oracle model. 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 paradigm [FS87] to Kilian's protocol; later, Valiant [Val08] showed that Micali's protocol is a proof of knowledge.

In the plain model. Micali's protocol is essentially as good as one can hope for in the random oracle model. In the plain model, such "totally non-interactive" succinct arguments (against non-uniform provers) do not exist except for "quasi-trivial" languages (i.e., languages in $\mathsf{BPtime}(n^{\mathrm{polylog} n})$), because the impossibility results for statistical soundness can be directly extended to this case. Nonetheless, known impossibility results leave open the possibility of succinct non-interactive arguments in a slightly more relaxed model, where a generator (run by the verifier or a trusted entity) produces ahead of time a short *reference string* σ for the prover and a short *verification state* τ for the verifier (and both strings are independent of the statements to be proven later). Indeed, the definition of SNARKs in the plain model refers to this more relaxed setting.

A set of works [BCCT11, DFH11, GLR11] showed how to construct designated-verifier SNARKs (i.e., τ needs to remain secret) from a non-standard cryptographic primitive called *extractable collision-resistant hashes*.The protocol used in these works revisits a previous protocol proposed by Di Crescenzo and Lipmaa [DCL08] who, in turn, followed up on ideas of Dwork et al. [DLN+04] and Aiello et al. [ABOR00] for "squashing" Kilian's protocol using succinct private information retrieval schemes.

Provable limitations. Gentry and Wichs [GW11] showed that no non-interactive succinct argument can be proven to be (adaptively) sound via a black-box reduction to a falsifiable assumption (as defined in [Nao03]), even in the designated-verifier case. Their result suggests that non-standard assumptions, such as knowledge (extractability) assumptions may be inherent for constructing succinct *non-interactive* arguments (even if we were to drop the proof of knowledge requirement). Thus, constructing SNARKs by relying on (reasonable) knowledge assumptions might be justified.

In light of the [GW11] impossibility, it seems that a SNARK construction (which would likely rely on a non-standard assumption) would circumvent the result of Rothblum and Vadhan [RV09]. This suggests that exhibiting a SNARK construction that does not rely on PCPs at all (not even implicitly so) is a possibility. In fact, [BCCT12] show that this is indeed the case. In this paper, we seek to also do so but we restrict our focus to the problem of constructing SNARKs directly from MIP protocols; indeed, despite "implicitly" using PCPs, MIP protocols have constructions (such as the one in the proof of our Theorem 1) that seem much more efficient than current PCP constructions.

5.2 Our Theorem

We adopt a similar strategy to the one in [BCCT11, DFH11, GLR11]:

1. In Kilian's protocol, the main cryptographic tool is collision-resistant hash functions, which are used to commit to the PCP string. By defining the non-standard cryptographic primitive of extractable collision-resistant hashes (ECRH), the works mentioned above show that it is possible to commit and reveal to the PCP string in a single message.
   In our protocol from Section 4, the main cryptographic tool is fully-homomorphic encryption, which is used to succinctly commit to a vector of functions. Our goal is to define a (non-standard) cryptographic primitive for revealing values of a vector of functions in a *single* message (as opposed to via an interactive commitment scheme) and show that this is enough to obtain a SNARK from MIPs.
2. Then, [BCCT11] show that ECRHs are necessary to the construction of SNARKs and exhibit several candidate constructions.
   We will also aim at showing that our primitive is necessary and at presenting candidate constructions.

The extractable primitive. We formulate a natural (though non-standard) primitive of *encryption with extractable homomorphism*: an encryption scheme where homomorphic operations cannot be performed "obliviously". Specifically, these are (public- or secret-key) encryption schemes that are homomorphic (with respect to some given class of function $\mathcal{F}$). The scheme has a specialized decryption algorithm Dec that takes as input a "source" cipher $\mathsf{c} = \mathsf{Enc}_{\mathsf{pk}}(m)$ and an evaluated cipher $\hat{\mathsf{c}}$ (which is allegedly the homomorphic evaluation of some function f on c). If $\hat{\mathsf{c}}$ is indeed such an evaluation, then $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{c}, \hat{\mathsf{c}}) = f(m)$; however, if the adversary "obliviously" computes some function of c, we require that $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{c}, \hat{\mathsf{c}}) = \perp$. More precisely, the guarantee is given in terms of extraction:

For any efficient $\mathcal{A}$ there is an efficient extractor $\mathcal{E}_{\mathcal{A}}$ such that:
if $\mathcal{A}$, given $\mathsf{c} = \mathsf{Enc}_{\mathsf{pk}}(m)$, outputs $\hat{\mathsf{c}}$ such that $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{c}, \hat{\mathsf{c}}) = v \neq \perp$,
then $\mathcal{E}_{\mathcal{A}}$, given the same input c, outputs a function f, such that $f(m) = v$.

We stress that the extractable-homomorphism property does *not* provide any guarantee regarding the extracted function f (e.g, it may not be in the family $\mathcal{F}$). In particular, the property does not ensure targeted malleability in the sense of, e.g., [BSW11].

The requirement can then be extended to multiple ciphers (and multiple corresponding functions). Coupled with semantic-security, such a primitive effectively yields a two-message succinct multi-function commitment. Indeed, whenever the adversary manages to transform $\mathsf{c}_1 = \mathsf{Enc}_{\mathsf{pk}_1}(q_1), \ldots, \mathsf{c}_\ell = \mathsf{Enc}_{\mathsf{pk}_\ell}(q_\ell)$ into evaluations $\hat{\mathsf{c}}_1, \ldots, \hat{\mathsf{c}}_\ell$ that are accepted by the decryption algorithm, the extractor will output corresponding functions $f_1, \ldots, f_\ell$ that "explain" the evaluations; moreover, because of semantic security, these functions are independent of the plaintext queries. More precisely, for any two vectors of queries $\vec{q}$ and $\vec{q}\,'$ and malicious adversary $\mathcal{A}$,

$$\left\{ \vec{f} \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k) \\ \vec{\mathsf{c}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\vec{q}) \\ \vec{\hat{\mathsf{c}}} \leftarrow \mathcal{A}(\mathsf{c}) \\ \vec{f} \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{c}) \end{array} \right\} \quad \text{and} \quad \left\{ \vec{f}\,' \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k) \\ \vec{\mathsf{c}}\,' \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\vec{q}\,') \\ \vec{\hat{\mathsf{c}}}\,' \leftarrow \mathcal{A}(\mathsf{c}') \\ \vec{f}\,' \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{c}') \end{array} \right\}$$

are computationally indistinguishable distributions over functions.

SNARKs from HEE. Given a fully-homomorphic encryption with extractable homomorphism, we are able to obtain a SNARK construction that is, in a sense, a "squashed" version of the interactive argument described in Section 4: the SNARK verifier simply generates MIP queries and sends them encrypted, using an extractable-homomorphism encryption scheme, to the SNARK prover. The prover, in turn, homomorphically evaluates each of the MIP provers on the corresponding encrypted query. The resulting scheme can be seen as an efficiency-preserving variant of the PIR-based protocol suggested by Dwork et al. [DLN$^+$04] as a heuristic for squashing Kilian's protocol.

We thus prove the following theorem (for details, see the full version of this paper):

Theorem 4. *Assuming the existence of fully-homomorphic encryption with extractable homomorphism, there exists an* <u>*efficiency-preserving*</u> *compiler that transforms a given succinct one-round MIP of knowledge into a corresponding SNARK.*[10]

[10] More precisely, as in Theorem 2, the preservation of efficiency holds as long as each MIP prover has a sufficiently tight deterministic reduction to circuits, which is the case for our MIP construction. (See Footnote 7.)

Furthermore, the existence of fully-homomorphic encryption and SNARKs implies the existence of fully-homomorphic encryption with extractable homomorphism.

Is the existence of encryption with extractable homomorphism plausible? Leaving efficiency aside and purely from an existential perspective, homomorphism-extractable encryption (HEE) schemes exist as long as extractable collision-resistant hashes (ECRHs) and FHE schemes exist: [BCCT11] showed how to construct SNARKs from ECRHs, and we will show that SNARKs and FHE together imply HEE schemes.

However, because our motivation for studying HEE schemes is the construction of SNARKs that are potentially more efficient than those built using PCPs, we believe that a more interesting question is to understand how plausible are "direct" constructions of HEE schemes. Consider, for example, the fully-homomorphic encryption (FHE) scheme of [BV11], based on LWE. Their FHE scheme likely does *not* have the extractable homomorphism property: consider an adversary that, given a ciphertext c of some message m, does not apply the honest evaluation algorithm to some function but instead applies some arbitrary transformation to c to obtain a new ciphertext $\hat{\mathsf{c}}$. With high probability, the new ciphertext $\hat{\mathsf{c}}$ will decrypt to some valid plaintext, but such an adversary may not "know" a corresponding function that can be homomorphically evaluated on the original plaintext c to result in the same ciphertext $\hat{\mathsf{c}}$.

Intuitively, the reason is that the ciphertext space of [BV11] is not "sparse": by merely applying an arbitrary operation in the ciphertext space, an adversary can "jump" from a valid ciphertext to another without being aware of what function he is applying on the underlying plaintext.[11]

Nonetheless, there *is* a natural method to generate sparsity in a given FHE scheme, and this method seems to serve as a heuristic for ensuring that the resulting encryption scheme can be plausibly assumed to have the extractable-homomorphism property.

The idea is to create sparsity via "amplification": we consider a new encryption scheme, built out of the old one, that, in order to encrypt a given message m, encrypts m under many different independently-generated public keys, and the new decryption algorithm will check that a given vector of ciphertexts decrypts to a vector of messages that are *all equal*. For this amplified encryption scheme, it seems that the oblivious adversary described earlier does not work anymore: if the adversary only applies algebraic operations to the ciphertexts, he is likely to obtain ciphertexts that decrypt to different values, that is, an invalid ciphertext of the new scheme.

Linear extractable homomorphism from KEA. In addition, if we consider encryption schemes that only support *linear homomorphism*, we can prove the extractable homomorphism property from a previously-studied knowledge assumption. Concretely, Damgård [Dam92] describes a variant of the El-Gamal encryption scheme which Bellare and Palacio [BP04b] prove to be plaintext aware assuming a Knowledge of Exponent assumption [Dam92, BP04a]. Under the same Knowledge of Exponent assumption we can show that Damgård's variant does satisfy the extractable homomorphism property. Unfortunately, linear homomorphism is not enough for use in our Theorem 4 (where full homomorphism is needed).

[11] Interestingly, sparsity also arises as a necessary condition when studying candidate constructions of ECRHs [BCCT11].

Acknowledgements

We thank Eli Ben-Sasson for discussions about MIP constructions. We also thank Ran Canetti, Omer Paneth, and Ben Riva for valuable discussions on MIP-based SNARKs.

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.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. Cryptology ePrint Archive, 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.

[BCCT11] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. Cryptology ePrint Archive, Report 2011/443, 2011.

[BCCT12] 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.

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, SFCS '90, pages 16–25, 1990.

[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.

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

[BOGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, STOC '88, pages 113–131, 1988.

[BP04a] 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.

[BP04b] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In *Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '04, pages 48–62, 2004.

[BSCGT12a] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. Fast reductions from RAMs to delegatable succinct constraint satisfaction problems, 2012. Cryptology ePrint Archive, Report Report 2012/071.

[BSCGT12b] 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.

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

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

[BSW11] Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: Homomorphic encryption for restricted computations. Cryptology ePrint Archive, Report 2011/311, 2011.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '11, 2011.

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *Proceedings of the 2nd Theory of Cryptography Conference*, TCC '05, pages 17–33, 2005.

[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.

[CL10] Kai-Min Chung and Feng-Hao Liu. Parallel repetition theorems for interactive arguments. In *Proceedings of the 7th Theory of Cryptography Conference*, TCC '10, pages 19–36, 2010.

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

[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.

[DFH11] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. Cryptology ePrint Archive, Report 2011/508, 2011.

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

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

[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.

[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.

[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.

[Hai09] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 241–250, 2009.

[Har04] Prahladh Harsha. *Robust PCPs of Proximity and Shorter PCPs*. PhD thesis, MIT, EECS, September 2004.

[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.

[Lip11] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. Cryptology ePrint Archive, Report 2011/009, 2011.

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

[MR08] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *Journal of the ACM*, 57:1–29, June 2008. Preliminary version appeared in FOCS '08.

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

[RS97] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, STOC '97, pages 475–484, 1997.

[RV09] Guy N. Rothblum and Salil Vadhan. Are PCPs inherent in efficient arguments? In *Proceedings of the 24th IEEE Annual Conference on Computational Complexity*, CCC '09, pages 81–92, 2009.

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

[TS96] Amnon Ta-Shma. A note on PCP vs. MIP. *Information Processing Letters*, 58:135–140, May 1996.

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

[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.