Rational Sumchecks*

Siyao Guo[1]**, Pavel Hubáček[2]***, Alon Rosen[3]†, and Margarita Vald[4]‡

[1] Chinese University of Hong Kong
syguo@cse.cuhk.edu.hk
[2] Weizmann Institute of Science
pavel.hubacek@weizmann.ac.il
[3] IDC Herzliya
alon.rosen@idc.ac.il
[4] Tel Aviv University
margarita.vald@cs.tau.ac.il

Abstract. Rational proofs, introduced by Azar and Micali (STOC 2012) are a variant of interactive proofs in which the prover is neither honest nor malicious, but rather rational. The advantage of rational proofs over their classical counterparts is that they allow for extremely low communication and verification time. In recent work, Guo et al. (ITCS 2014) demonstrated their relevance to delegation of computation by showing that, if the rational prover is additionally restricted to being computationally bounded, then every language in NC1 admits a single-round delegation scheme that can be verified in sublinear time.

We extend the Guo et al. result by constructing a single-round delegation scheme with sublinear verification for all languages in P. Our main contribution is the introduction of *rational sumcheck protocols*, which are a relaxation of classical sumchecks, a crucial building block for interactive proofs. Unlike their classical counterparts, rational sumchecks retain their (rational) soundness properties, *even if the polynomial being verified is of high degree* (in particular, they do not rely on the Schwartz-Zippel lemma). This enables us to bypass the main efficiency bottleneck in classical delegation schemes, which is a result of sumcheck protocols being inapplicable to the verification of the computation's input level.

As an additional contribution we study the possibility of using rational proofs as efficient blocks within classical interactive proofs. Specifically, we show a composition theorem for substituting oracle calls in an interactive proof by a rational protocol.

* Part of this work done while authors were visiting IDC Herzliya, supported by the European Research Council under the European Union's Seventh Framework Programme (FP 2007-2013), ERC Grant Agreement n. 307952.

** Work partially supported by RGC GRF grants CUHK410112 and CUHK410113.

*** Supported by the I-CORE Program of the Planning and Budgeting Committee and The Israel Science Foundation (grant No. 4/11).

† Supported by ISF grant no. 1255/12 and by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) ERC Grant Agreement n. 307952. Work in part done while the author was visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant #CNS-1523467.

‡ Work supported by the Check Point Institute for Information Security and by ISF grant no. 1255/12.

1 Introduction

The availability of on-demand computational power and the ubiquitous connectivity of small devices are some of the main driving forces behind the move to the model of cloud computing. In this model a client faces a computationally demanding task and relies on the assistance of an external server with sufficient computational power, e.g. a cluster of machines. When the weak client asks the powerful server to perform a computation on its behalf it would like to have some guarantees on the correctness of the provided result. This scenario is addressed by the model of *verifiable delegation of computation*. In this setting, the server provides the client with the result of the computation together with a proof of its correctness. Since the client must be able to verify the proof despite its limited computational resources, the verification should be much easier than running the computation itself, or else there is no point in outsourcing it.

Interactive Proofs and Arguments. A setting where an all-powerful entity aims to convince a computationally bounded one of the correctness of a computational statement was studied in the context of *interactive proof systems*. In this model interaction and randomization enable the prover to efficiently convince the verifier. The IP = PSPACE theorem [22,28] showed that it is possible for the prover to convince the verifier about large classes of languages, in particular any language computable in polynomial time. However, this result is not efficient enough to be practically applicable to the problem of verifiable delegation. In this context, one aims to minimize multiple complexity measures at once, such as communication complexity (both in the number and size of exchanged messages), running time of the verifier and prover efficiency.

For higher complexity classes, the round-complexity/prover-efficiency of interactive proofs is a limiting factor to their use in practice. The notion of *interactive arguments* considers a setting where the prover is computationally bounded, allowing to circumvent these efficiency shortcomings. The work of Kilian [20] gave four round interactive arguments for all languages in NP. Micali [23], relying on random oracles proposed a non-interactive version of this protocol. More recently, there has been significant effort to obtain more efficient non-interactive arguments for NP (see e.g. [5,10] and the references therein). One limitation of all such known constructions is that they are based on non-standard assumptions (cf. [24]). The problem of constructing efficient non-interactive arguments for NP under standard assumptions is still open, though there is some evidence that non-standard assumptions are unavoidable [13].

Unlike in the case of arguments for non-deterministic computation, the situation for tractable languages (which actually correspond to problems common in real-life delegation scenarios) is significantly better. The first evidence that one can attain delegation schemes for restricted complexity classes is the work of Goldwasser, Kalai and Rothblum [14], who gave a single-round argument that allows to verifiably delegate any bounded depth computation with quasi-linear verification time. Recently, the work of Kalai, Raz and Rothblum [18] achieved a single-round argument (under standard assumptions) with quasi-linear verification time for any language in P.

In some scenarios quasi-linear verification time may not be good enough. For instance, if the input $x \in \{0,1\}^n$ is a large database and the output $f(x)$ of the outsourced computation is a concise aggregation of its statistics, then it is desirable if the verifier

does not need to read the whole database to verify correctness. In such cases one would prefer to have a delegation scheme with verification time *sublinear in the input size n*, preferably even as low as polylog(n). As was pointed out in the literature, delegation schemes with sublinear verification are in general not achievable with respect to the standard notion of soundness (cf. Rothblum, Vadhan and Wigderson [27]), which led to introduction of alternative relaxed models that would enable sublinear verification time.

Rational Proofs and Arguments. One recent notion that opens the door for sublinear verification is that of *rational arguments* [15]. This model follows the paradigm of rational proofs introduced by Azar and Micali [2], who relax the prover in interactive proof systems to be rational. In rational proofs the verifier pays the prover according to the quality of the provided answer, and the reward is set up so that it is irrational for the prover to report an incorrect result of the computation. Azar and Micali [2] illustrated the power of rational proofs by giving a single-round rational proof for any problem in #P and in general a constant round rational proof for any level of the counting hierarchy. In subsequent work, Azar and Micali [3] gave a "scaled-down" version of their #P-protocol that leads to constant round rational interactive proof with sublinear ($O(\log n)$ time) verification for the class of log-time uniform TC^0, i.e., the class of constant-depth, log-time uniform polynomial-size circuits with threshold gates. They also argue that such efficient rational proofs capture precisely the class of log-time uniform TC^0.

More recently, Guo, Hubáček, Rosen and Vald [15] put forward the notion of rational arguments, by further restricting the rational prover to be computationally bounded. They then showed how to construct single-round rational arguments with sublinear (polylog(n) time) verification for the class NC^1, of search problems computable by log-time uniform Boolean circuits of $O(\log n)$-depth.

1.1 Our Results

We extend the results of Guo *et al.* [15] and give a single-round rational argument with sublinear (polylog(n) time) verification for any language in P. Our initial observation is that both the non-interactive arguments for NC of Goldwasser *et al.* [14] and the non-interactive arguments for P of Kalai *et al.* [18] have for the most part sublinear verification time, with the exception of a single heavy verification step that ultimately induces quasi-linear running time for the verifier. If we could substitute this step by a more efficient procedure that does not dominate the rest of the protocol then we would achieve sublinear verification time.

Our proposal is to use a rational proof with sublinear verification for the heavy step and get a rational version of the original protocol which enjoys sublinear verification time. There are two main issues that we will need to address: 1) construct sublinear rational proofs for the heavy step; 2) argue how the rationality can be preserved under composition.

Our main contribution is the introduction of *rational sumcheck protocols*, which are a relaxation of classical sumchecks, a crucial building block for interactive proofs. To show that our approach yields the desired result, we pin down sufficient conditions for our transformation to work and prove that the protocol of Goldwasser *et al.* [14] (respectively Kalai *et al.* [18]), with the rational sumcheck replacing the heavy step, yields the sought after rational argument for NC (respectively for P).

It should be noted that our main efficiency gains are not due to the fact that rational sumcheck protocols are more efficient than their classical counterparts (though we do gain some efficiency by making sumcheck protocols non-interactive). Indeed, one of the key observations behind the works on efficient delegation [14,18] is that one could verify correctness of computation via very efficient sumcheck protocols. The one place where rational sumcheck protocols turn out to be more useful than classical ones is at the input layer, where usage of the latter would entail a total break-down of soundness.

We show that a rational version of sumcheck protocols is in fact sufficient to carry out verification, *even without reading the entire input*. This is something that was not possible to achieve using classical sumcheck protocols, since the input layer does not satisfy the structural properties (low-degree) that would guarantee soundness when verifying via classical sumchecks. Our (equally efficient) rational sumcheck protocols, on the other hand, give a meaningful soundness guarantee even when such structural properties are absent.

Sumcheck Protocols. At a high level, a classical sumcheck protocol allows the verifier to check a sum of evaluations of a given low-degree polynomial $h : \mathbb{F}_q^m \to \mathbb{F}_q$ on a certain subset $S \subset \mathbb{F}_q^m$ of its domain (e.g. $S = \{0,1\}^m$). The source of the protocol's power is that it makes it sufficient for the verifier to evaluate h on a *single* randomly chosen point $p \in \mathbb{F}_q^m$, rather than on the entire subset S. This results in significant efficiency gains, since instead of requiring the evaluation of h on $|S|$ points it reduces the problem of verification to the evaluation on a single point (at the cost of $m = \log(|S|)$ rounds of communication).

Previous works on delegation [14,18] make extensive use of sumcheck protocols in order to efficiently verify the low degree extensions $\tilde{W}$ of intermediate levels of computation.[5] Specifically, it is possible to write $\tilde{W}(z) = \sum_{p \in S} \beta_z(p) W(p)$, where $\beta_z(p)$ is a low-degree function and $W(p)$ is an appropriate encoding of the corresponding level. This reduces the task of verifying the correctness of evaluating $\tilde{W}$ on z to the problem of performing a sumcheck on individual inner summands $\beta_z(p)W(p)$. In intermediate levels of the computation, we are guaranteed that W is of low degree, and hence so is $\beta_z(p)W(p)$. However, at the input level the function $W(p) = W_x(p)$ corresponds to a straightforward bit-wise representation of the input $x \in \{0,1\}^n$. The problem is that this representation might result in a high-degree polynomial. Not being of low degree, $\beta_z(p)W_x(p)$ cannot be verified by a classical sumcheck protocol. This means that the input x needs to be read in its entirety, or else the protocol is not sound.

Rational Sumcheck Protocols. To circumvent the above issue, we leverage the power of rational proofs, in which soundness relies on rationality of the prover. We give a rational sumcheck protocol that allows to efficiently verify summation of any function over a fixed set, as long as evaluating the function on a single point can be performed efficiently (see Section 3 for details). Not only that our rational sumcheck protocol preserves the efficiency of classical sumchecks, but it can also be performed without any communication overhead (it is in fact non-interactive). The main feature of rational

[5] In Goldwasser *et al.* [14] this is performed layer by layer over the circuit computing the function, whereas in Kalai *et al.* [18] the reduction to sumchecks is done via a global encoding of the transcript of the computation.

sumchecks, however, is that they give a meaningful (rational) soundness guarantee *even if the degree of the polynomial is high*, which implies that unlike their classical counterparts they are also applicable at the input layer.

Technically speaking, the reason for which the new rational protocols work regardless of the polynomial's degree is because the soundness analysis does not necessitate invoking the Schwartz-Zippel lemma. Instead, we rely on a specially-tailored reward function that is designed to translate sums of finite field elements to numerical values that are used to determine the reward. The challenge in designing the reward function originates from the fact that modular sums lose information about the summands, whereas the reward is required to reflect this information in its entirety.

Composition of Classical and Rational Interactive Proofs. To make the above fit into a general purpose protocol, we need to carefully show how to plug a rational subprotocol into a larger one while retaining rational soundness. To this end, we show a composition theorem for substituting oracle calls in an interactive proof by a rational protocol. This allows us to use the classical interactive proofs almost as a black-box. This approach may turn out to be useful elsewhere.

Putting the Pieces Together. At a high level, the structure of our construction of single-round rational arguments for P follows the delegation scheme of Kalai *et al.* [18]. In particular, we define and construct δ-no-signaling rational multi-prover proofs (RMIPs) by using our composition theorem and relying on rational sumchecks as a subprotocol. We then show a general efficient transformation that uses any sub-exponentially secure *Fully Homomorphic Encryption* (FHE) scheme to transform no-signaling RMIPs into single-round rational arguments (in a manner similar to Kalai *et al.* [18]). Crucial to our transformation is the *reward gap* of the underlying rational protocol, which roughly captures the utility loss of the prover as a result of misreporting the function's value. Unlike early rational proofs of Azar and Micali [2] and akin to Guo *et al.* [15], both our sumchecks and the overall composed protocol enjoy noticeable reward gap. This is sufficient for the overall transformation to go through (enabling a reduction from the security of the FHE scheme), and results in the sought-after single-round rational argument for P with sublinear verification time.

Beyond being of importance in the transformation from rational proofs to non-interactive rational arguments, noticeable reward gap is also crucial for incentivizing the prover to report the correct value of the computation, as otherwise he might be tempted to avoid performing the work while risking very little penalty (see Section 2 and Guo *et al.* [15] for an extended discussion of the subject).

1.2 Comparison to Alternative Delegation Schemes

The classical interactive proof for NC of Goldwasser *et al.* [14] has quasi-linear verification time. The running time of the verifier in their protocol appears to be optimal in the standard model, in the sense that achieving sublinear verification time with standard soundness guarantee seems unlikely without reading the whole input (even for a simple function such as parity). To circumvent this limitation Rothblum, Vadhan and Wigderson [27] considered interactive proofs of proximity, a relaxation of interactive proofs motivated by property testing, and show that it is possible to achieve sublinear

verification for NC in this new model (since the protocol does not need to provide soundness guarantee for all instances).

An alternative relaxation was studied by Azar and Micali [3] and Guo *et al.*[15]. These works considered delegation in the setting of rational proofs and proposed schemes whith both sublinear verification (as small as polylogarithmic) and (rational) soundness guarantees, which in contrast to proofs of proximity hold for all instances.Whereas their protocols work only for NC^1, our new rational proof, which is a combination of classical and rational proofs, works for the entirety of NC while preserving the desired properties of sublinear verification and rational soundness (see Table 1 for a detailed comparison). By composing classical and rational proofs, we obtain a rational multi-prover proof (secure against no-signaling provers) with sublinear verification for any deterministic computation akin to the classical proof of Kalai *et al.* [18] (see Table 2 for a detailed comparison). We remark it is possible to transform the above classical proofs and rational proofs into one-round classical and rational arguments.

Table 1. efficiency comparison of results for NC

	Queries[6]	Rounds	Communication	Verification time	Depth
Goldwasser *et al.* [14] (interactive proofs)	n	$\tilde{O}(d)$	$\tilde{O}(d)$	$\tilde{O}(n)$	$d=\mathrm{polylog}(n)$
Rothblum *et al.* [27] (proofs of D-proximity)	$\left(\frac{n}{\mathrm{D}}\right)^{1+o(1)}$	$\tilde{O}(d)$	$\mathrm{D}\left(\frac{n}{\mathrm{D}}\right)^{o(1)}\cdot\tilde{O}(d)$	$\left(\frac{n}{\mathrm{D}}+\mathrm{D}\right)^{1+o(1)}\tilde{O}(d)$	$d=\mathrm{polylog}(n)$
Azar and Micali [3] (rational proofs)	1	d	$\tilde{O}(d)$	$\tilde{O}(d)$	$d=O(\log n)$
Guo *et al.* [15] (rational proofs)	1	d	d	$\tilde{O}(d)$	$d=O(\log n)$
This work (rational proofs)	1	$\tilde{O}(d)$	$\tilde{O}(d)$	$\tilde{O}(d)$	$d=\mathrm{polylog}(n)$

Table 2. efficiency comparison of results for P

	Queries	Number of provers	Communication	Verification time	Remarks
Kalai *et al.* [18] (MIP)	n	$\mathrm{polylog}(t)$	$\mathrm{polylog}(t)$	$n\cdot\mathrm{polylog}(t)$	DTIME(t)
This work (rational MIP)	$\mathrm{polylog}(t)$	$\mathrm{polylog}(t)$	$\mathrm{polylog}(t)$	$\mathrm{polylog}(t)$	DTIME(t)

1.3 Other Related Work

To give a complete overview of works on verifiable delegation of computation is out of the scope of this paper, an interested reader can find many related results in the recent survey by Blumberg and Walfish [6].

[6] By queries we denote the number of input bits read by the verifier.

An alternative approach for *interactive proofs with sublinear verification* was given in Rothblum, Vadhan and Wigderson [27] who introduced *interactive proofs of proximity* and Gur and Rothblum [16] who considered their non-interactive analogues. Since both works studied a protocol analogue of property testing, their protocols provide guarantees only for instances that are either in the language or far from being in the language. Independently an in parallel to our work, Kalai and Rothblum [19] studied proofs of proximity with computationally bounded provers and introduced *arguments of proximity*.

Besides the mentioned works in the context of rational proofs, Zheng and Blanton [29] study the specific problem of delegating matrix multiplication and give also a rational argument for this task. The work of Chen, McCauley and Singh [9] introduces the model of rational interactive proofs with multiple provers.

Alternative approaches for incentivizing correct computation can be found in the work of Bentov and Kumaresan [21] who consider a model for incentivizing computation over Bitcoin. Alternatively, Belenkiy *et al.* [4] or Pham, Khouzani and Cid [25] study a model where the verifier infrequently performs the whole computation to verify the correctness of prover's output.

The treatise of general composition of rational protocols in scientific literature is limited. The work of Garay, Katz, Maurer, Tackman and Zikas [12] provides some insights on composition of protocols secure in the presence of a single *central rational adversary*. The framework of Canetti and Vald [8] studies a notion sufficient for preserving rationality under composition by imposing strong restrictions on the information available to distinct adversarial entities.

2 Preliminaries

Throughout the rest of the paper we use the following notation and definitions. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1,\ldots,n\}$. A function $g : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if it tends to 0 faster than any inverse polynomial, i.e., for all $c \in \mathbb{N}$ there exists $k_c \in \mathbb{N}$ such that for every $k > k_c$ it holds that $g(k) < k^{-c}$. We use $\mathrm{negl}(\cdot)$ to talk about negligible function if we do not need to specify its name.

Rational Proofs In a rational proof, Arthur pays Merlin a randomized reward according to the transcript of the communication, and the communication constitutes a rational Merlin Arthur game if the correct evaluation $y = f(x)$ can be derived from a transcript that maximizes the expected reward.

For a pair of interactive Turing machines, P and V, we denote by $(P,V)(x)$ the random variable representing the transcript between P and V when interacting on common input x. Let $\mathsf{reward}(\cdot)$ denote a randomized function computed by V that given a transcript calculates a reward for P, and by $\mathsf{output}((P,V)(x))$ the output of V after interacting with P on common input x. In this setting, the goal of a *rational* P is to maximize the expected value of $\mathsf{reward}(\cdot)$, while the goal of V is to learn (and output) the true evaluation of the desired function f on x. We consider the setting where a rational prover first declares his answer to $f(x)$, and only then tries to prove the correctness of the reported value.

Definition 1. *[Functional Rational Merlin Arthur] Let $C,T:\mathbb{N}\to\mathbb{R}$ be some functions. A function $f:\{0,1\}^*\to\{0,1\}^*$ is in* $\mathrm{FRMA}[r,C,T]$ *if there exists an r-round public-coin protocol (P,V), referred as rational proof, and a randomized reward function* $\mathsf{reward}:\{0,1\}^*\to\mathbb{R}_{\geq 0}$ *such that for all inputs $x\in\{0,1\}^*$:*

(a) $\Pr[\mathsf{output}((P,V)(x))=f(x)]=1$.

(b) For every round i and for any prover P^* that misreports $f(x)$ and behaves as P up to round i and differs on round i'th message it holds that: $\mathrm{E}[\mathsf{reward}((P,V)(x))]>\mathrm{E}[\mathsf{reward}((P^*,V)(x))]$, *where the expectation is taken over the random coins of the verifier and the prover.*

(c) The communication complexity of P is $C(|x|)$.

(d) The running time of V is $T(|x|)$.

No-signaling provers. In this work we use the heuristic suggested by Aiello *et al.* [1] for transforming a multi-prover proof into a single round argument using an efficient Private Information Retrieval (PIR) scheme (or alternatively a Fully Homomorpic Encryption scheme), though in the rational setting. As pointed out in the work of Dwork *et al.* [11], the bottleneck when proving soundness of the resulting argument is the possibility for the prover to correlate the answers in an undetectable way. Such no-signaling strategies (introduced as "spooky interactions" in the work of Dwork *et al.* [11]) need to be accounted for in the proof of soundness, as shown in Kalai, Raz and Rothblum [17].

Thus, we extend Definition 1 to the setting with multiple provers restricted to δ-no-signaling strategies. In contrast to the classical multi-prover setting, where each prover strategy is completely independent of other provers' queries, δ-no-signaling strategies can be correlated as long as for any subset of provers their answers do not contain information about the queries of provers outside the subset.

Definition 2 (Statistically No-Signaling Distributions). *Let D be a query alphabet and let Σ be an answer alphabet. For every $q=(q_1,\ldots,q_k)\in D^k$, let $\mathcal{A}_q$ be a distribution over Σ^k. We think of $\mathcal{A}_q$ as the distribution of the answers for queries q. We say that the family of distributions $\{\mathcal{A}_q\}_{q\in D^k}$ is δ-no-signaling if for every subset $S\subset[k]$ and every two sequences of queries $q,q'\in D^k$, such that $q_S=q'_S$, the following two random variables are δ-close: $\{a_S:a\leftarrow\mathcal{A}_q\}$ and $\{a'_S:a'\leftarrow\mathcal{A}_{q'}\}$.*

The rational no-signaling multi-prover proof consists of only one round. Given an input, the verifier generates queries, one for each prover, and sends them to the k provers. Each prover responds with an answer that might depend on all the queries, as long as the provers' strategies are no-signaling. Finally, the verifier computes the reward based on the received answers (as well as the input and the randomness used).

Definition 3 (One-Round Rational Multi-Prover Interactive Proof). *Let $C,T:\mathbb{N}\to\mathbb{R}$ be some functions. A function $f:\{0,1\}^*\to\{0,1\}^*$ is in* $\mathrm{FRMIP}[k,\delta,C,T]$ *if there exists a one-round public-coin protocol $(\overrightarrow{P},V)=(P_1,\ldots,P_k,V)$, referred as* multi-prover rational proof*, and a randomized reward function* $\mathsf{reward}:\{0,1\}^*\to\mathbb{R}_{\geq 0}$ *such that for all inputs $x\in\{0,1\}^*$:*

(a) $\Pr[\mathsf{output}((\overrightarrow{P},V)(x))=f(x)]=1$.

(b) For every set of provers $P_1^*, \ldots, P_k^*$ with δ-no-signaling distributions that misreport $f(x)$ it holds that: $\mathrm{E}[\mathsf{reward}((P_1, \ldots, P_k, V)(x))] > \mathrm{E}[\mathsf{reward}((P_1^*, \ldots, P_k^*, V)(x))]$, *where the expectation is taken over the random coins of the verifier and the provers.*

(c) The communication complexity from any of the provers to V is at most is $C(|x|)$.

(d) The running time of V is $T(|x|)$.

Reward gap. We note that once computation incurs some cost to the prover the Definitions 1 and 3 of rational proofs do not rule out a "lazy behavior" of the prover corresponding to outputting a fixed default value. Having this in mind, Guo *et al.* [15] proposes the notion of *reward gap* that measures how big is the loss of a prover that always reports $f(x)$ incorrectly. A noticeable gap in expectation between such a prover and the prescribed behavior then assures that it is beneficial for the prover to perform the computation to significantly increase its expectation.

Definition 4 (Reward Gap). *Let $f \in \mathrm{FRMA}[r, C, T]$ be some function and let (P, V) and $\mathsf{reward}(\cdot)$ be the guaranteed protocol and reward function. The* reward gap *of $\mathsf{reward}(\cdot)$ is a function $\Delta_{\mathsf{reward}} : \mathbb{N} \to \mathbb{R}$, such that for every $n \in \mathbb{N}$,*

$$\Delta_{\mathsf{reward}}(n) = \min_{x \in \{0,1\}^n} \min_{P^* \in S} \left(\mathrm{E}[\mathsf{reward}((P, V)(x))] - \mathrm{E}[\mathsf{reward}((P^*, V)(x))] \right),$$

where the expectation is taken over the random coins of the verifier and the prover, and S is the set of all P^* such that $\Pr[\mathsf{output}((P^*, V)(x)) \neq f(x)] = 1$.

We emphasize that scaling the reward does not imply a real improvement in the reward gap. In order to have a robust notion we always work with a *normalized reward gap*, i.e., reward gap divided by the maximal value of the reward function. An alternative approach (taken for example in Azar and Micali [3]) that prevents the use of scaling to improve the reward gap might be to assume that the verifier has a fixed budget. We use the natural extension of reward gap to rational multi-prover interactive proofs.

Rational Arguments. *Rational arguments* were defined by Guo *et al.* [15] to capture the behavior of a rational prover that is computationally bounded. The definition of rational arguments allows negligible gains over the reward guaranteed by the prescribed behavior (but not more), since the rational prover might not follow the prescribed strategy, and it would try to solve the underlying hard problems (see item (b) in Definition 5).

Another important issue needed to be addressed in the computational setting is the cost of computing $f(x)$. As in the unbounded setting, it must rule out a prover that always gives some default (possibly incorrect) output, without performing any computation, while getting just slightly less than the expectation of the prescribed behavior. To address this shortcoming the definition of rational arguments "pins down" the profitability of deviation explicitly by appropriately adapting the notion of reward gap to the computationally bounded setting (see item (c) in Definition 5).

Definition 5 (Rational Argument). *A function $f : \{0,1\}^* \to \{0,1\}^*$ admits a* rational argument *with security parameter $\kappa : \mathbb{N} \to \mathbb{N}$ if there exists a protocol (P, V) and a randomized reward function $\mathsf{reward} : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ such that for any input $x \in \{0,1\}^*$ and any prover P^* of size $\leq \mathrm{poly}(2^{\kappa(|x|)})$ the following hold:*

(a) $\Pr[\mathsf{output}((P,V)(x)) = f(x)] = 1$.

(b) *There exists a negligible function* $\varepsilon(\cdot)$ *such that* $\mathrm{E}[\mathsf{reward}((P,V)(x))] + \varepsilon(|x|) \geq \mathrm{E}[\mathsf{reward}((P^*,V)(x))]$.

(c) *If there exists a polynomial* $p(\cdot)$ *such that* $\Pr[\mathsf{output}((P^*,V)(x)) \neq f(x)] \geq p(|x|)^{-1}$ *then there exists a polynomial* $q(\cdot)$ *such that* $\mathrm{E}[\mathsf{reward}((P^*,V)(x))] + q(|x|)^{-1} \leq \mathrm{E}[\mathsf{reward}((P,V)(x))]$.

The expectations and the probabilities are taken over the random coins of the respective prover and verifier. We say that the rational argument is efficient *if the running time of* V *is* $o(|x|)$ *for every* $x \in \{0,1\}^*$.

3 Rational Sumcheck Protocols

Sumcheck protocols are an important building block in many classical interactive proofs. In particular, they play a crucial role in the IP = PSPACE theorem [22,28]. Informally, a sumcheck protocol allows a verifier to efficiently check that a summation of evaluations of a polynomial of low degree on a given set of points is equal to a certain value (e.g. zero). In this section we show how to construct a rational sumcheck protocol that is sound (against a rational prover) even when applied on a polynomial of high degree. An important property of rational proofs is the *reward gap*, that captures the minimal loss in reward of the prover that always misreports the value of the function (formal definitions of rational proofs and reward gap are provided in Section 2). All of our rational proofs achieve noticeable reward gap.

Before describing our rational sumchecks, we show how to solve a simpler related problem: the verifier is given a bound M and n integers $x_1,\ldots,x_n \in \{0,\ldots,M-1\}$, the verifier's goal is to learn the sum of $x_1,\ldots,x_n$. In the even more restricted case when $x_1,\ldots,x_n$ are bits (i.e., $M = 2$), one could solve this binary counting problem using an analogue of the rational proof of Azar and Micali [2]. In particular, the verifier can use a strictly proper scoring rule (e.g. the Brier's score [7]) to reward the quality of the prover's answer $y = \sum_{i=1}^{n} x_i$ as a prediction of the binary random variable b defined by outputting a uniformly random x_i. The intuition behind such protocol is that the Boolean random variable b encodes the information about the number of ones within $x_1,\ldots,x_n$; specifically, the probability of $b = 1$ is exactly the number of ones divided by n. Since the reward is defined according to a strictly proper scoring rule, a rational prover will uniquely maximize its expected reward by reporting the correct $y = \sum_{i=1}^{n} x_i$ (it describes the true distribution of b) as long as it is possible to efficiently sample b.

When $M > 2$, the mean of the random variable defined by outputting a uniformly random x_i still encodes the sum of $x_1,\ldots,x_n$. However, b is not necessarily Boolean and, unlike in the case when $x_1 \ldots,x_n$ are bits, the problem can no longer be solved by the protocol of Azar and Micali [2]. In order to use the Brier's score, it is necessary to appropriately modify the procedure of sampling b. Our more general protocol is given in Figure 1. The verifier picks a random i from $\{1,\ldots,n\}$, and sets $b = 1$ with probability x_i/M and otherwise sets $b = 0$. After this normalization the probability of $b = 1$ is $\sum_{i=1}^{n} x_i/(nM)$ which still encodes the sum of $x_1,\ldots,x_n$, and since b is a Boolean variable it is possible to use the same reward function to incentivize any rational prover to report

On common input $x_1,\ldots,x_n \in \{0,\ldots,M-1\}$:

1. The prover sends integer $y = \sum_{i=1}^{n} x_i$ to the verifier.
2. The verifier samples i from $\{1,\ldots,n\}$, it sets $b = 1$ with probability $\frac{x_i}{M}$ (otherwise it sets $b = 0$) and outputs the reward for the prover in the following way

$$R(y) = \begin{cases} 2\left(\frac{y}{nM}\right) - \left(\frac{y}{nM}\right)^2 - \left(1 - \frac{y}{nM}\right)^2 + 1, & \text{if } b = 1, \\ 2\left(1 - \frac{y}{nM}\right) - \left(\frac{y}{nM}\right)^2 - \left(1 - \frac{y}{nM}\right)^2 + 1, & \text{if } b = 0. \end{cases}$$

Fig. 1. Rational proof for summation of n non-negative integers.

correct description of b. Therefore, the protocol in Figure 1 is a non-interactive rational proof for the simplified problem of summation of n bounded non-negative values.

Lemma 1 (Rational Proof for Summation). *For any integer $M \geq 2$, let $f(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i$ be the function that computes the sum of any n-tuple of integers $x_1,\ldots,x_n \in \{0,\ldots,M-1\}$. Then $f \in \mathrm{FRMA}\left[1, \log(nM), O(\mathrm{polylog}(nM))\right]$ with reward gap at least $\frac{1}{(nM)^2}$.*

Proof. Consider the protocol in Figure 1. The expected reward when prover sends y is

$$\mathrm{E}[R(y)] = -2\left(\frac{y}{nM} - \frac{\sum_{i=1}^{n} x_i}{nM}\right)^2 + 2\left(\frac{\sum_{i=1}^{n} x_i}{nM}\right)^2 - 2\left(\frac{\sum_{i=1}^{n} x_i}{nM}\right) + 2\,,$$

therefore the expected reward of the prover is uniquely maximized when $y = \sum_{i=1}^{n} x_i$.

For any integer $y^* \neq \sum_{i=1}^{n} x_i$,

$$\mathrm{E}\left[R\left(\textstyle\sum_{i=1}^{n} x_i\right)\right] - \mathrm{E}[R(y^*)] = 2\left(\frac{y^*}{nM} - \frac{\sum_{i=1}^{n} x_i}{nM}\right)^2 \geq \frac{2}{(nM)^2}\,,$$

where the equality holds when $y^* = \sum_{i=1}^{n} x_i \pm 1$. The reward function has maximal value 2, hence the (normalized) reward gap is $\frac{1}{(nM)^2}$. Because $y = \sum_{i=1}^{n} x_i \leq nM$, y can be represented using $\log(nM)$ bits which upper bounds the total communication. The verifier only needs to access a single x_i where i is chosen uniformly and randomly from $\{1,\ldots,n\}$. After accessing to x_i, the computation of the reward can be done in $O(\mathrm{polylog}(nM))$ time. □

Note that for any polynomially bounded M, the protocol in Figure 1 achieves sublinear verification (the verifier only needs to access a single value) and noticeable reward gap. Moreover, based on the protocol in Figure 1, we can construct an efficient rational proof for any problem which can be reduced to summation of several bounded values. For example, we immediately obtain a rational proof for addition of n elements over a finite field $\mathbb{Z}_p$ of prime characteristic p. Given $x_1,\ldots,x_n \in \mathbb{Z}_p$:

1. The prover sends to the verifier the sum $s = \sum_{i=1}^{n} x_i$ over $\mathbb{Z}$ (i.e., without performing the modulo operation) together with $y = (s \mod p)$, where s serves as the proof of correctness of y.

On common input $x_1,\ldots,x_n \in \mathbb{F}_{p^m}$:

1. The prover sends to the verifier a vector $(y^1,\ldots,y^m) \in \mathbb{Z}_p^m$ corresponding to $y = \sum_{i=1}^n x_i \in \mathbb{F}_{p^m}$ and a vector $s = (s^1,\ldots,s^m)$ such that $s^j = \sum_{i=1}^n x_i^j$ for all $j \in \{1,\ldots,m\}$.
2. The verifier checks if $y^j = (s^j \mod p)$ for all $j \in \{1,\ldots,m\}$. If not the verifier pays 0, and otherwise the verifier samples $j \in \{1,\ldots,m\}$ and $i \in \{1,\ldots,n\}$, it sets $b = 1$ with probability $\frac{x_i^j}{p}$ and computes the reward for the prover as

$$R(s^j) = \begin{cases} 2\left(\frac{s^j}{np}\right) - \left(\frac{s^j}{np}\right)^2 - \left(1-\frac{s^j}{np}\right)^2 + 1, & \text{if } b = 1, \\ 2\left(1-\frac{s^j}{np}\right) - \left(\frac{s^j}{np}\right)^2 - \left(1-\frac{s^j}{np}\right)^2 + 1, & \text{if } b = 0. \end{cases}$$

Fig. 2. Rational proof for summation of n elements over a finite field.

2. If $y \neq (s \mod p)$ then the verifier pays reward 0, and otherwise the verifier computes the reward for s as in the rational proof for summation of $x_1,\ldots,x_n$ with $M = p$ (as described in Figure 1).

To deal with general summation over a finite field $\mathbb{F}_q$ of prime power characteristic $q = p^m$, we leverage the fact that the additive group of $\mathbb{F}_{p^m}$ is isomorphic to $(\mathbb{Z}_p, +_{\mod p})^m$, where $+_{\mod p}$ denotes addition over $\mathbb{Z}_p$. Thus, we can work with the representation of elements in $\mathbb{F}_{p^m}$ as vectors over $\mathbb{Z}_p^m$, i.e., we represent any $x \in \mathbb{F}_{p^m}$ as $(x^1,\ldots,x^m) \in \mathbb{Z}_p^m$. This allows us to get a rational proof for the function $\sum_{i=1}^n x_i$ that computes the sum of any n-tuple of elements $x_1,\ldots,x_n \in \mathbb{F}_{p^m}$ over $\mathbb{F}_{p^m}$ simply by applying the rational protocol for summation over $\mathbb{Z}_p$ on a randomly chosen coordinate of the vector representation $(y^1,\ldots,y^m) \in \mathbb{Z}_p^m$ of the output $y \in \mathbb{F}_{p^m}$ declared by the prover. The protocol is given in Figure 2.

Corollary 1 (Rational Proof for Addition over Finite Fields). *For any integer $m \geq 1$ and any prime $p \in \mathbb{N}$. Let $f(x_1,\ldots,x_n) = \sum_{i=1}^n x_i$ be the function that computes the sum of any n-tuple of elements $x_1,\ldots,x_n \in \mathbb{F}_{p^m}$ over the field $\mathbb{F}_{p^m}$. Then $f \in \mathrm{FRMA}[1, \log(np^m), O(m \cdot \mathrm{polylog}(np))]$ with reward gap at least $\frac{1}{m(np)^2}$.*

Proof. Consider the protocol in Figure 2. Let y and s denote the vectors sent by the prover when he tells the truth. It is easy to check the expected reward of the prover is maximized at y, s. When prover answers $\tilde{y} \neq y$ and $\tilde{s}$, if $\tilde{y} \neq (\tilde{s} \mod p^m)$ then the prover gets reward 0, otherwise s and $\tilde{s}$ must differ in at least one entry and the expected reward of the prover is

$$\begin{aligned} \mathrm{E}_j[R(\tilde{s}^j)] &= \mathrm{E}_j\left[-2\left(\frac{\tilde{s}^j}{np} - \frac{s^j}{np}\right)^2 + 2\left(\frac{s^j}{np}\right)^2 - 2\left(\frac{s^j}{np}\right) + 2\right] \\ &\leq \mathrm{E}_j\left[2\left(\frac{s^j}{np}\right)^2 - 2\left(\frac{s^j}{np}\right) + 2\right] - \frac{2}{m(np)^2}\ . \end{aligned}$$

Note the reward function has maximal value 2 therefore the reward gap is at least $\frac{1}{m(np)^2}$. □

Note that a sumcheck protocol is used to verify a sum of evaluations of a polynomial on a given set of points. Corollary 1 immediatelly gives rise to a *non-interactive* rational sumcheck protocol, where the verifier needs to evaluate the polynomial on a single point from the subset.

Corollary 2 (Rational Sumcheck Protocol). *For any finite field $\mathbb{F}$ and integer $m \geq 1$. Let $S \subseteq \mathbb{F}^m$ be a non-empty subset of $\mathbb{F}^m$. Let $\sum_{z \in S} f(z)$ be the function that sums evaluations of a given polynomial $f : \mathbb{F}^m \to \mathbb{F}$ (of arbitrary degree) on S. Then $f \in$ FRMA$[1, \log(|S||\mathbb{F}|), O(t + \text{polylog}(|S||\mathbb{F}|))]$, where t is the time it takes to evaluate f on any $z \in \mathbb{F}^m$. The rational proof has reward gap at least $1/(\log(|\mathbb{F}|) \cdot (|S||\mathbb{F}|)^2)$.*

Proof. Using the protocol in Figure 2 with field $\mathbb{F}$ and setting $n = |S|$, we obtain a rational proof for $\sum_{z \in S} f(z)$ with reward gap $\frac{1}{\log(|\mathbb{F}|) \cdot (|S||\mathbb{F}|)^2}$, verification time $O(t + \text{polylog}(|S||\mathbb{F}|))$, and communication $\log(|S||\mathbb{F}|)$ bits. □

4 Composition of Classical and Rational Interactive Proofs

In this section we investigate on the possibility of composition of classical interactive proofs with rational interactive proofs. In particular, we show a composition theorem for replacing oracle calls in a certain type of classical interactive proofs by a rational proof implementing the oracle. The composition is presented for both interactive proofs and δ-no-signaling multi-prover interactive proofs (for formal definition see Definition 3 in Section 2) resulting in their respective rational counterparts. The obtained rational proof has minimal loss in the reward gap that is proportional to the soundness of the classical interactive proof.

4.1 Substituting Oracle by Rational Proof in Interactive Proof

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function implicitly defining language $L_f = \{(x,y)|y = f(x)\}$. Let $\pi^g = (P_\pi, V_\pi^g)$ be an interactive proof for L_f where the verifier has oracle access to function $g : \{0,1\}^* \to \{0,1\}^*$. Let $\varphi = (P_\varphi, V_\varphi)$ be a rational interactive proof for g with reward function reward_φ. We denote by $\pi^\varphi = (P,V)$ with a reward function R the protocol between the prover P and verifier V given in Figure 3. We define the reward in the resulting protocol as the average of the rewards obtained for each rational proof implementing an oracle query, though we note that this is not crucial for our results. The new reward function can be defined in other natural ways depending on the application.

We concentrate on a class of *query independent interactive proofs* in which the queries to the oracle can depend only on the input and the randomness of the verifier. Aditionally, once a query is submitted to the oracle the prover also recives the query.

Definition 6 (Query Independent Interactive Proofs). *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function and let $\pi^g = (P_\pi, V_\pi^g)$ be an interactive proof for $L_f = \{(x,y)|y = f(x)\}$ with V_π^g having oracle access to some function $g : \{0,1\}^* \to \{0,1\}^*$. We say that π^g is a* query independent interactive proof *if for any input x the following holds:*

1. *Only one query is issued by V_π^g to g and it depends only on the input x and on the randomness of V_π^g.*

On common input x:

1. The prover P sends $y = f(x)$ to the verifier.
2. The prover and the verifier initiate the interactive protocol π^g with P_π proving that $(x,y) \in L_f$.
3. Whenever V_π^g would issue a query q to g, the verifier V sends q to the prover P and they run φ on input q. P answers the query of V_π^g using the output of P_φ and continues with simulating π^g.
4. After V_π^g terminates, V outputs $(y, R(\tau))$, where τ is the transcript of communication between P and V and R is computed as:

$$R(\tau) = \begin{cases} R(\tau_\phi) & \text{if } V_\pi^g \text{ on } \tau \text{ accepts} \\ 0 & \text{otherwise.} \end{cases}$$

where τ_ϕ is the transcript corresponding to executions of φ in π^φ and $R(\tau_\phi)$ is the average of rewards computed by applying reward_φ to the transcript of each execution of φ separately.

Fig. 3. Rational proof $\pi^\varphi = (P,V)$ resulting from interactive proof $\pi^g = (P_\pi, V_\pi^g)$ with oracle calls to g substituted by a rational proof $\varphi = (P_\varphi, V_\varphi)$.

2. *The query issued by V_π^g is send to P_π in the next round.*

Theorem 1 (Oracle Substitution in IP). *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function and let $\pi^g = (P_\pi, V_\pi^g)$ be a query independent interactive proof for $L_f = \{(x,y) | y = f(x)\}$ with V_π^g having oracle access to some function $g : \{0,1\}^* \to \{0,1\}^*$. If π^g has perfect completeness and soundness s then for any rational interactive proof $\varphi = (P_\varphi, V_\varphi)$ for g with reward gap Δ, the composed protocol $\pi^\varphi = (P,V)$ is a rational proof for f with reward gap $\Delta(1-s)$.*

Proof. The reward in the rational protocol π^φ (defined in Figure 3) is equal to the reward in the rational proof φ for evaluating the oracle query if the verifier accepts and zero otherwise. In order to show that π^φ is a rational proof with the claimed reward gap, we show that for every x the expectation of any prover P^* that reports $y' \neq f(x)$ (i.e., (x,y') is not in L_f) can be bound. To simplify the notation, we define three events that might happen during the execution of the protocol π^φ:

- E_0 corresponds to the event when V_π^g (simulated by V) accepts and P^* supplies *a correct answer* to the oracle query q (i.e., $(P^*, V_\pi^g)(x) = 1 \wedge \mathsf{output}(P^*, V_\varphi)(q) = g(q)$).
- E_1 corresponds to the event when V_π^g (simulated by V) accepts and P^* supplies *an incorrect answer* to the oracle query q (i.e., $(P^*, V_\pi^g)(x) = 1 \wedge \mathsf{output}(P^*, V_\varphi)(q) \neq g(q)$).
- E_2 corresponds to the event when V_π^g (simulated by V) rejects.

We can express the expectation of P^* as

$$\begin{aligned} \mathrm{E}[\mathsf{reward}(P^*,V)(x)] = \Pr[E_0] \cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x) | E_0] \\ + \Pr[E_1] \cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x) | E_1] + \Pr[E_2] \cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x) | E_2] \,. \end{aligned}$$

Since the expected reward is zero in case of event E_2 (the verifier V_π^g rejects), the above is equal to

$$\Pr[E_0]\cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_0] + \Pr[E_1]\cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_1]\ .$$

We can bound $\Pr[E_1]$ by $1-\Pr[E_0]$, so

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(P^*,V)(x)] \leq \Pr[E_0]\cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_0] \\ + (1-\Pr[E_0])\cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_1]\ .\end{aligned}$$

We use the following two claims to conclude the proof.

Claim 1. $\Pr[E_0] \leq s$.

Proof (of Claim 1). The interactive protocol with oracle access (P_π, V_π^g) is query independent in the sense of Definition 6, hence the prover in the composed protocol π^φ does not gain any additional information from the verifier's query to the oracle for g. It follows that in the case when the prover P^* supplies a correct answer to the oracle query the verifier accepts at most with the same probability as in the interactive proof with an oracle access, and the claim follows from the soundness of the interactive proof (P_π, V_π^g). ■

Claim 2. $\mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_1] \leq \mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\varphi)(q)] - \Delta$.

Proof (of Claim 2). Assume that the claim does not hold, then the prover P^* achieves for some q a higher reward than $\mathrm{E}[\mathsf{reward}(P_\varphi,V_\varphi)(q)] - \Delta$. P^* can be used in the rational proof (P_φ,V_φ) for evaluating the oracle in order to achieve a higher reward than what is guaranteed by the reward gap of (P_φ,V_φ), since the oracle query is completely independent of the transcript. ■

We use Claim 2 to bound the expectation as:

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(P^*,V)(x)] \leq \Pr[E_0]\cdot \mathrm{E}[\mathsf{reward}(P^*,V)(x)|E_0] \\ + (1-\Pr[E_0])\cdot (\mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\phi)(q)] - \Delta)\ .\end{aligned}$$

Notice that the expectation when event E_0 materializes is equal to $\mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\phi)(q)]$, and hence we can rewrite the right side of the above inequality:

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(P^*,V)(x)] \leq \Pr[E_0]\cdot \mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\phi)(q)] \\ + (1-\Pr[E_0])\cdot (\mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\phi)(q)] - \Delta)\ .\end{aligned}$$

The distribution of oracle queries q is independent of the communication between the prover and the verifier and we can merge the expressions on the right side of the inequality.

$$\mathrm{E}[\mathsf{reward}(P^*,V)(x)] \leq \mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\varphi)(q)] - (1-\Pr[E_0])\cdot\Delta\ ,$$

Finally, by Claim 1:

$$\mathrm{E}[\mathsf{reward}(P^*,V)(x)] \leq \mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\varphi)(q)] - (1-s)\cdot\Delta\ .$$

By observing that for all x it holds that $\mathrm{E}_q[\mathsf{reward}(P_\varphi,V_\varphi)(q)] = \mathrm{E}[\mathsf{reward}(P,V)(x)]$ (since the distribution of queries produced by V is independent of x), we get the sought after bound on the reward gap of the resulting rational proof. □

On common input x:

1. The prover P_1 sends $y = f(x)$ to the verifier V.
2. The verifier V simulates the verifier V_π^g on input (x,y) to obtain k queries $\mathbf{q}$ for $\overrightarrow{P_\pi}$ and sends them to k provers in $\overrightarrow{P}$ (one query to each prover). The k provers answer their queries $\mathbf{q}$ using the corresponding outputs of $\overrightarrow{P_\pi}$.
3. For every query q^* that V_π^g issues to g, the verifier V simulates the verifier V_φ on input q^* to obtain k' queries $\omega(q^*)$ for $\overrightarrow{P_\varphi}$ and sends them to a new set of k' provers in $\overrightarrow{P}$ (one query to each prover). The k' provers answer their queries $\omega(q^*)$ using the corresponding outputs of $\overrightarrow{P_\varphi}$.
4. After V receives answers to all the queries from $\overrightarrow{P}$, the verifier V outputs $(y, R(\tau))$, where τ is the transcript of communication between $\overrightarrow{P}$ and V and R is computed as:

$$R(\tau) = \begin{cases} R(\tau_\varphi) & \text{if } V_\pi^g \text{ on } \tau \text{ accepts} \\ 0 & \text{otherwise.} \end{cases}$$

where τ_φ is the transcript corresponding to executions of φ within π^φ and $R(\tau_\varphi)$ is the average of rewards computed by applying the reward function reward_φ to the transcript of each execution of φ separately.

Fig. 4. Rational multi-prover proof $\pi^\varphi = (\overrightarrow{P}, V)$ resulting from multi-prover proof $\pi^g = (\overrightarrow{P_\pi}, V_\pi^g)$ with oracle calls to g substituted by a (multi-prover) rational proof $\varphi = (\overrightarrow{P_\varphi}, V_\varphi)$ for evaluating g.

4.2 Substituting Oracle by Rational Multi-Prover Proof in Multi-Prover Proof

The composition theorem holds for oracle substitution also in the setting of δ-no-signaling multi-prover proofs. Given a k-prover interactive proof $\pi^g = (\overrightarrow{P_\pi}, V_\pi^g)$ for function f with an oracle access to a function g and a "rational" k'-prover implementation $\varphi = (\overrightarrow{P_\varphi}, V_\varphi)$ of the function g, a new rational protocol $\pi^\varphi = (\overrightarrow{P}, V)$ with $(k+k')$ provers can be obtained by executing the rational protocol φ instead of the oracle call with a new set of k' provers, as defined in Figure 4. We define the reward in the resulting protocol analogously to the single prover setting and take the average of the rewards.

Similarly to the previous setting we require the oracle queries to depend only on the input and the randomness of the verifier and the queries to the provers to be independent of the answers of the oracle. Definition 6 of query independent interactive proofs naturally extends to multi-prover interactive proofs and we refer to multi-prover interactive proofs with this analogous property as *query independent*. Note that item 1 in Definition 6 is no longer required since we only deal with no-signaling strategies. In order to enable submission of all queries at once in the composed protocol, we must require independence of the queries to the provers from the oracle answers.

Definition 7 (Query Independent Multi-Prover Proofs). *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function and let $\pi^g = (\overrightarrow{P_\pi}, V_\pi^g)$ be a multi-prover proof for $L_f = \{(x,y) | y = f(x)\}$ with V_π^g having oracle access to some function $g : \{0,1\}^* \to \{0,1\}^*$. We say that π^g is a* query independent multi-prover proof *if for any input x the following holds:*

1. *Only a single query q is issued by V_π^g to g and it depends only on the input x and on the randomness of V_π^g.*

2. *The queries of V_π^g to $\overrightarrow{P_\pi}$ are independent of the oracle answer to the query q.*

We show that the composition theorem holds also for oracle substitution in the setting of query independent multi-prover proofs. Note that our composition theorem shows that when dealing with δ-no-signaling strategies, a loss in the reward gap proportional to δ is incurred in the resulting composed protocol.

Theorem 2 (Oracle Substitution in MIP). *Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function and let $\pi^g = (\overrightarrow{P_\pi}, V_\pi^g)$ be a query independent k-prover* MIP *for $L_f = \{(x,y)|y = f(x)\}$ with V_π^g having oracle access to some function $g : \{0,1\}^* \to \{0,1\}^*$. If π^g has perfect completeness and soundness s against δ-statistically no-signaling strategies then for any rational k'-prover* RMIP *$\varphi = (\overrightarrow{P_\varphi}, V_\varphi)$ for evaluating g with reward gap Δ in presence of δ'-statistically no-signaling strategies, the composed protocol $\pi^\varphi = (\overrightarrow{P}, V)$ is a $(k+k')$-prover* RMIP *for evaluating f with reward gap $\Delta(1-s-\delta'')$ against δ''-no-signaling strategies, where $\delta'' = \min\{\delta, \delta'\}$.*

Proof. The reward in the rational protocol π^φ (defined in Figure 4) is equal to the reward in the rational proof φ for evaluating the oracle query if the verifier accepts and zero otherwise. In order to show that π^φ is a multi-prover rational proof for evaluating f with the claimed reward gap, we show that for every x the expectation of any set of provers $\overrightarrow{P^*}$ that report $y' \neq f(x)$ (i.e., (x,y') is not in L_f) can be bounded. To simplify the notation, we define three events that might happen during the course of the protocol π^φ:

- E_0 corresponds to the event when V_π^g (simulated by V) accepts and $\overrightarrow{P^*}$ supply *a correct answer* to the oracle query q^* (i.e., $(\overrightarrow{P^*}, V_\pi^g)(x) = 1 \wedge \mathsf{output}(\overrightarrow{P^*}, V_\varphi)(q^*) = g(q^*)$).
- E_1 corresponds to the event when V_π^g (simulated by V) accepts and $\overrightarrow{P^*}$ supply *an incorrect answer* to the oracle query q^* (i.e., $(\overrightarrow{P^*}, V_\pi^g)(x) = 1 \wedge \mathsf{output}(\overrightarrow{P^*}, V_\varphi)(q^*) \neq g(q^*)$).
- E_2 corresponds to the event when V_π^g (simulated by V) rejects.

We can express the expectation of $\overrightarrow{P^*}$,

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)] = \Pr[E_0] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_0] \\ + \Pr[E_1] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_1] + \Pr[E_2] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_2]\ .\end{aligned}$$

Since the expected reward in case of event E_2 is zero, the above is equal to

$$\Pr[E_0] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_0] + \Pr[E_1] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_1]\ .$$

We can bound the $\Pr[E_1]$ by $1 - \Pr[E_0]$, so

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)] \leq \Pr[E_0] \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_0] \\ + (1 - \Pr[E_0]) \cdot \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_1]\ .\end{aligned}$$

We use the two following claims to complete the proof.

Claim 3. $\Pr[E_0] \leq s + \delta''$.

Proof (of Claim 3). For any q^*, an oracle query of V_π^g, define $\omega(q^*)$ to be the queries to $\overrightarrow{P_\varphi}$ generated by V_φ on input q^*. Let $\mathsf{A} = \{\mathsf{A}_{\mathbf{q},\omega(q^*)}\}$ denote the δ''-no-signaling family of distributions, where $\mathsf{A}_{\mathbf{q},\omega(q^*)}$ is the distribution of answers of $\overrightarrow{P^*}$ given queries $(\mathbf{q},\omega(q^*))$. We fix an arbitrary set of queries $\mathbf{w}$ of V_φ to $\overrightarrow{P_\varphi}$ and consider the family of distributions $\mathsf{B} = \{\mathsf{B}_\mathbf{q}\}$, where $\mathsf{B}_\mathbf{q}$ is defined by sampling uniformly and randomly $(\mathbf{a},\mathbf{z}) \leftarrow \mathsf{A}_{\mathbf{q},\mathbf{w}}$ and outputting $\mathbf{a}$.

First, we show that B is δ-no-signaling. Let S be an arbitrary subset of $[k]$ and $\mathbf{q},\mathbf{q}'$ be two arbitrary queries such that $\mathbf{q}_S = \mathbf{q}'_S$. Since the projections of $\mathsf{A}_{\mathbf{q},\mathbf{w}}$ and $\mathsf{A}_{\mathbf{q}',\mathbf{w}}$ on the coordinates in S are δ''-close (by the fact that A is δ''-no-signaling), the statistical distance between $\mathsf{B}_\mathbf{q}$ and $\mathsf{B}_{\mathbf{q}'}$ when projected on S is

$$\begin{aligned}\frac{1}{2}\sum_\beta \left| \Pr_{\mathbf{a}\leftarrow\mathsf{B}_\mathbf{q}}[\mathbf{a}_S = \beta] - \Pr_{\mathbf{a}'\leftarrow\mathsf{B}_{\mathbf{q}'}}[\mathbf{a}'_S = \beta] \right| &= \frac{1}{2}\sum_\beta \left| \Pr_{\mathbf{a}\leftarrow\mathsf{A}_{\mathbf{q},\mathbf{w}}}[\mathbf{a}_S = \beta] - \Pr_{\mathbf{a}'\leftarrow\mathsf{A}_{\mathbf{q}',\mathbf{w}}}[\mathbf{a}'_S = \beta] \right| \\ &\leq \delta'' \\ &\leq \delta\ ,\end{aligned}$$

where the last inequality follows from δ'' being defined as $\min\{\delta,\delta'\}$. Hence, B is δ-no-signaling.

Let $\overrightarrow{P^*_\pi}$ be the set of provers in π^g that follow the δ-no-signaling strategies B. By the soundness of π^g in the presence of δ-no-signaling strategies, $\Pr[(\overrightarrow{P^*_\pi}, V_\pi^g)(x) = 1] \leq s$. Assume that the claim does not hold, then

$$\begin{aligned}\delta'' &< \Pr[E_0] - \Pr[(\overrightarrow{P^*_\pi}, V_\pi^g)(x) = 1] \\ &= \Pr_{(\mathbf{a},\mathbf{z})\leftarrow\mathsf{A}_{\mathbf{q},\omega(q^*)}}[V_\pi^g(x,\mathbf{a},\mathbf{z}_1) = 1 \wedge \mathbf{z}_1 = g(q^*)] - \Pr_{(\mathbf{a},\mathbf{z})\leftarrow\mathsf{A}_{\mathbf{q},\mathbf{w}}}[V_\pi^g(x,\mathbf{a},g(q^*)) = 1]\end{aligned}$$

A contradiction to A being δ''-no-signaling. ■

Claim 4. *For all x it holds that* $\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_1] \leq \mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_\varphi}, V_\varphi)(q^*)] - \Delta$.

Proof (of Claim 4). Assume that the claim does not hold. By an averaging argument over the randomness of the verifier V for generating queries to the provers $\overrightarrow{P}$, there exists an x and a fixed choice of randomness for generating the queries such that

$$\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*}, V)(x)|E_1, (\mathbf{q},\omega(q^*))] > \mathrm{E}[\mathsf{reward}(\overrightarrow{P_\varphi}, V_\varphi)(q^*)] - \Delta\ ,$$

where $\mathbf{q}$ and q^* are fixed. Let $\mathsf{A}_{\mathbf{q},\omega(q^*)}$ denote the δ''-no-signaling distribution of answers of $\overrightarrow{P^*}$ to the queries $(\mathbf{q},\omega(q^*))$. Consider the family of distributions $\mathsf{B} = \{\mathsf{B}_{\omega(q^*)}\}$, where $\mathsf{B}_{\omega(q^*)}$ is defined by sampling uniformly and randomly $(\mathbf{a},\mathbf{z}) \leftarrow \mathsf{A}_{\mathbf{q},\omega(q^*)}$ and outputting $\mathbf{z}$.

First, we show that B is δ'-no-signaling. Let S be an arbitrary subset of $[k']$ and $\mathbf{w},\mathbf{w}'$ be two sets of queries such that $\mathbf{w}_S = \mathbf{w}'_S$. Since the projections of $\mathsf{A}_{\mathbf{q},\mathbf{w}}$ and $\mathsf{A}_{\mathbf{q},\mathbf{w}'}$ on the

coordinates $S' = \{k+i \colon i \in S\}$ are δ''-close, the statistical distance between $\mathsf{B}_{\mathbf{w}}$ and $\mathsf{B}_{\mathbf{w}'}$ is

$$\begin{aligned}\frac{1}{2}\sum_{\beta}\left|\Pr_{\mathbf{z}\leftarrow\mathsf{B}_{\mathbf{w}}}[\mathbf{z}_S=\beta]-\Pr_{\mathbf{z}'\leftarrow\mathsf{B}_{\mathbf{w}'}}[\mathbf{z}'_S=\beta]\right| &= \frac{1}{2}\sum_{\beta}\left|\Pr_{\mathbf{z}\leftarrow\mathsf{A}_{\mathbf{q},\mathbf{w}}}[\mathbf{z}_S=\beta]-\Pr_{\mathbf{z}'\leftarrow\mathsf{A}_{\mathbf{q},\mathbf{w}'}}[\mathbf{z}'_S=\beta]\right| \\ &\leq \delta'' \\ &\leq \delta' \ .\end{aligned}$$

Where the last inequality follows from δ'' being defined as $\min\{\delta,\delta'\}$, and hence B is δ'-no-signaling.

Let $\overrightarrow{P^*_{\varphi}}$ behave according to B, then on input q^*,

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*_{\varphi}},V_{\varphi})(q^*)] &= \mathrm{E}[\mathsf{reward}(\overrightarrow{P^*},V)(x)|E_1,(\mathbf{q},\omega(q^*))] \\ &> \mathrm{E}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]-\Delta \ .\end{aligned}$$

Therefore, $\overrightarrow{P^*_{\varphi}}$ is a set of δ'-no-signaling provers that break the reward gap guarantee of φ, a contradiction. ■

We use Claim 4 to bound the expectation as:

$$\begin{aligned}\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*},V)(x)] \leq \Pr[E_0]\cdot\mathrm{E}[\mathsf{reward}(\overrightarrow{P^*},V)(x)|E_0] \\ +(1-\Pr[E_0])\cdot(\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]-\Delta) \ .\end{aligned}$$

Notice that due to the query independence of the protocol π^g the expectation when event E_0 materializes is equal to $\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]$. Hence, we can rewrite the right side of the above inequality as

$$\begin{aligned}&\Pr[E_0]\cdot\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]+(1-\Pr[E_0])\cdot(\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]-\Delta) \\ &=\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]-(1-\Pr[E_0])\cdot\Delta \ .\end{aligned}$$

Finally, by Claim 3:

$$\mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P^*},V)(q^*)] \leq \mathrm{E}_{q^*}[\mathsf{reward}(\overrightarrow{P_{\varphi}},V_{\varphi})(q^*)]-(1-s-\delta'')\cdot\Delta \ .$$

Therefore, we get the sought after bound on the reward gap of the multi-prover rational proof π^{φ} resulting from the composition of π^g and φ. □

5 Rational Delegation for NC

The work of Guo *et al.* [15] showed how to efficiently delegate computation performed by low-depth circuits in the rational setting, and in particular constructed a rational proof with noticeable reward gap for any language in NC^1. However, the reward gap in their construction is proportional to the depth of the evaluated circuit (the reward is scaled proportionally to the depth) and this prevents to use their rational proof with meaningful (noticeable) reward gap beyond the class NC^1. In this section we give a rational proof with sublinear verification time for any function computable by log-space uniform NC by composing the rational sumcheck protocol from Section 3 with the classical protocol of Goldwasser *et al.* [14].

5.1 The Protocol of Goldwasser, Kalai and Rothblum [14]

In their work Goldwasser, Kalai and Rothblum [14] gave a protocol that allows to delegate computation of any function computable by log-space uniform circuits via an interactive proof with a polynomial prover and a quasi-linear verifier. In particular, they showed the following theorem:

Theorem 3 (Theorem 1.1.1. in [26]). *Let L be a language computable by a family of $O(\log(S(n)))$-space uniform boolean circuits of size $S(n)$ and depth $d(n)$. L has an interactive proof where:*

1. *The prover runs in time* $\mathrm{poly}(S(n))$. *The verifier runs in time* $n \cdot \mathrm{poly}(d(n), \log(S(n)))$ *and space* $O(\log(S(n)))$. *Moreover, if the verifier is given oracle access to the low degree extension of its input, then its running time is only* $\mathrm{poly}(d(n), \log(S(n)))$.
2. *The protocol has perfect completeness and soundness* $1/2$.
3. *The protocol is public-coin, with communication complexity* $d(n) \cdot \mathrm{polylog}(S(n))$.
4. *Each message of the prover depends only on* $O(\log(n))$ *random bits sent by the verifier.*

Their interactive proof builds on arithmetization techniques and employs efficient sumcheck protocols in order to establish correctness of the output. The sumcheck is run on multivariate polynomials of low degree that encode the values of intermediate layers of computation to allow the verifier to efficiently check consistency of the prover's answers.

Let $w = (w_1, \ldots, w_k)$ be k bits. The vector w defines a function $W : \{1, \ldots, k\} \to \{0,1\}$ such that $W(i) = w_i$ for all $i \in \{1, \ldots, n\}$. Let $\mathbb{H}$ be an extension field of $\mathbb{GF}[2]$, m be an integer such that $k \leq |\mathbb{H}|^m$, and let $\mathbb{F}$ be an extension field of $\mathbb{H}$. The *low degree extension* of w is the unique m-variate polynomial $\tilde{W} : \mathbb{F}^m \to \mathbb{F}$ of degree at most $|\mathbb{H}| - 1$ in each variable that agrees with W on $\mathbb{H}^m$. It is a useful fact that the low degree extension can be expressed as sum over $\mathbb{H}^m$, where each term is efficiently computable (for the details see Appendix A).

Here we provide a high-level overview of the protocol (for the full exposition see e.g. [26]):

1. The prover P evaluates the circuit C on input x received from the verifier V, and computes a low degree extension $\widetilde{W}_i$ for every layer i of the circuit C.
2. For $1 \leq i \leq d$, in each phase i the prover initiates an interactive sumcheck protocol to convince the verifier that $\widetilde{W_{i-1}}(z_{i-1}) = r_{i-1}$. In the first phase $z_0 = (0, \ldots, 0)$ and $r_0 = (C(x), 0, \ldots, 0)$. To complete the i-th sumcheck protocol the verifier would need to evaluate $\widetilde{W}_i$ on two random points ω_1, ω_2, but to avoid the related computational burden this task is reduced to another sumcheck performed in phase $i+1$. In particular, the prover and the verifier run an interactive procedure using ω_1, ω_2, the verifier picks a random z_i and the prover reports a corresponding value $r_i = \widetilde{W}_i(z_i)$. The protocol proceeds to phase $i+1$.
3. In phase $d+1$ the verifier evaluates the low degree extension $\widetilde{W_d}$ (of the input x) on the random point z_d and checks that it is equal to r_d reported by the prover. This is the final phase and the only point at which the verifier evaluates a low degree extension.

The running time of the verifier in the first d phases is $\text{poly}(d(n), \log(S(n)))$, and it is the evaluation of the low degree extension of the input in the last step that induces the overall quasi-linear overhead of $n \cdot \text{poly}(d(n), \log(S(n)))$ for the verifier. Hence, given oracle access to the low degree extension of the input the verification can be performed in sublinear time. Moreover, the protocol of Goldwasser *et al.* [14] is query independent in the sense of Definition 6, i.e., after receiving the answer to its query the verifier can send the query (the random point z_d) to the prover and the soundness is preserved. This allows us to use our composition framework from Section 4 in order to substitute the oracle call with our rational sumcheck protocol.

Substituting the Low Degree Extension Oracle with a Rational Proof First, we show that our rational sumcheck protocol from Section 3 can evaluate an arbitrary low degree extension.

Proposition 1 (Rational Protocol for Evaluating Low Degree Extension). *The low degree extension $\tilde{W} : \mathbb{F}^m \to \mathbb{F}$ of $(w_1, \ldots, w_k) \in \{0,1\}^k$ admits a rational proof with verification time* $\text{poly}(|\mathbb{H}|, m)$*, assuming oracle access to $(w_1, \ldots, w_k)$, with reward gap* $1/4(\log|\mathbb{F}|)|\mathbb{H}^m|^2$.

Proof. By Proposition 2 (given in Appendix A), for any $z \in \mathbb{F}^m$, $\tilde{W}(z)$ is a summation of $|\mathbb{H}|^m$ terms of the form $\sum_{p \in \mathbb{H}^m} \tilde{\beta}(z,p) \cdot W(p)$, where the addition is over $\mathbb{F}$ and $\mathbb{F}$ is a extension field of $\mathbb{GF}[2]$. Moreover, for every (z,p), $\tilde{\beta}(z,p)$ can be computed in time $\text{poly}(|\mathbb{H}|, m)$, therefore $\tilde{\beta}(z,p) \cdot W(p)$ can be computed in time $\text{poly}(|\mathbb{H}|, m)$. By Corollary 2, $\tilde{W}(z)$ admits rational proof with reward gap $1/(\log|\mathbb{F}|)(2|\mathbb{H}^m|)^2 = 1/4(\log|\mathbb{F}|)|\mathbb{H}^m|^2$ and verification time $\text{poly}(|\mathbb{H}|, m)$. □

Finally, we use the above efficient rational proof in the protocol of Goldwasser *et al.* [14] to allow the verifier to avoid reading the whole input when evaluating the low degree extension of the input.

Theorem 4 (Rational Interactive Proof for NC**).** *For any function $f : \{0,1\}^* \to \{0,1\}$, if $L_f = \{(x,y) | y = f(x)\}$ is computable by a family of $O(\log(S(n)))$-space uniform Boolean circuits of size $S(n)$ and depth $d(n) = O(\text{polylog}(n))$ then $f \in \text{FRMA}[d(n) \cdot \text{polylog}(n), d(n) \cdot \text{polylog}(S(n)), \text{poly}(d(n), \log(S(n)))]$ with a public-coin rational interactive proof with a noticeable reward gap, where the prover runs in time* $\text{poly}(S(n))$ *and the verifier runs in space $O(\log(S(n)))$.*

Proof. For $f \in NC$, we let $\pi^g = (P_\pi, V_\pi^g)$ be the interactive proof for $L_f = \{(x,y) | y = f(x)\}$ defined in Theorem 3 where g is the low degree extension of x with $|\mathbb{F}| = \text{poly}(n,d)$ and $|\mathbb{H}^m| = \text{poly}(n)$, the soundness is $1/2$ and the completeness is 1. Let $\varphi = (P_\varphi, V_\varphi)$ be the rational proof for g as defined in Proposition 1 with reward gap $\Delta = 1/(4\log|F|)(|\mathbb{H}|^{2m})$. Note that V_π^g only issues a single query and for all x the communication between P_π and V_π^g is independent of $(q, g(q))$. By Theorem 1, π^φ is a rational proof for f with regard gap $\Delta(1-s) = \Delta/2 = 1/\text{poly}(n)$.

The running time of the prover or verifier is at most the sum of the running time of P_π in Theorem 3 and the running time of P_φ. The total running time is $\text{poly}(S(n))$. The verifier runs in at most V_π^g and the running time of V_φ. Therefore the running time of verifier is upper bounded by $\text{poly}(d(n), \log(S(n)))$. The total communication

is the communication of φ and the communication of π which is upper bounded by $d(n) \cdot \text{poly}(S(n))$. □

5.2 Single-Round Rational Arguments for NC

Guo *et al.* [15] gave an efficient transformation from any rational proof with noticeable reward gap to single-round rational argument. The transformation uses an efficient Private Information Retrieval (PIR) scheme (for formal definition see the full version) in order to submit all the round queries to the prover at once.

Theorem 5 (Theorem 6 in [15]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a function in* FRMA $[r, C, T]$. *Assume the existence of a PIR scheme with communication complexity* $\text{poly}(\kappa)$ *and receiver work* $\text{poly}(\kappa)$, *where $\kappa \geq \max\{C(n), \log n\}$ is the security parameter. If f has an admissible rational proof with noticeable reward gap Δ, then f admits single-round rational argument which has the following properties:*

(a) The verifier runs in time $C(n) \cdot \text{poly}(\kappa) + O(T(n))$.
(b) The communication complexity is $r \cdot \text{poly}(\kappa, \lambda)$ where λ is the longest message sent by the prover.

By applying the above transformation of Guo *et al.* [15] on the rational interactive proofs in Theorem 4, we obtain single-round rational arguments for NC with sublinear verification.

Corollary 3 (Rational Argument for NC). *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by log-space uniform* NC *of size $S(n) = \text{poly}(n)$ and depth $d(n) = O(\text{polylog}(n))$. Assume the existence of a PIR scheme with communication complexity* $\text{poly}(\kappa)$ *and receiver work* $\text{poly}(\kappa)$, *where $\kappa \geq d(n) \cdot \text{polylog}(S(n))$ is the security parameter. Then f admits single-round efficient rational argument which has the following properties:*

1. *The verifier runs in* $\text{poly}(\kappa, d(n), \log(S(n)))$ *and the prover runs in* $\text{poly}(\kappa, S(n))$.
2. *The length of the prover's message and the verifier's challenge is $d(n) \cdot \text{poly}(\kappa, \log(S(n)))$. The verifier's challenge depends only on his random coins and is independent of the input x.*

6 Rational Delegation for P

Recently, Kalai, Raz and Rothblum [18] gave a single-round delegation scheme for every language computable in time $t(n)$, where the running time of the verifier is $n \cdot \text{polylog}(t(n))$. For languages in P where $t(n) = \text{poly}(n)$, the verification time is $O(n \cdot \text{polylog}(n))$. The efficiency bottleneck for achieving sublinear verification for P in Kalai *et al.* [18] (similarly to the protocol for NC of Goldwasser *et al.* [14]) is that the verifier needs to evaluate a low degree extension of the input which takes quasi-linear time. We show that it is possible to improve the verification time to be sublinear in the rational setting.

6.1 The Protocol of Kalai, Raz and Rothblum [18]

Recently, Kalai *et al.* [18] gave an MIP secure against no-signaling provers for any deterministic computation.

Theorem 6 (Theorem 4 in [18]). *Suppose that* $L \in \mathrm{DTIME}(t(n))$*, where* $t = t(n)$ *satisfies* $\mathrm{poly}(n) \leq t \leq \exp(n)$*. Then, for any integer* $(\log t)^c \leq k \leq \mathrm{poly}(n)$*, where c is some (sufficiently large) universal constant, there exists an* MIP *for L with* $k \cdot \mathrm{polylog}(t)$ *provers where:*

1. *The verifier runs in time* $n \cdot k^2 \cdot \mathrm{polylog}(t)$ *and the provers run in time* $\mathrm{poly}(t,k)$*. Moreover, if the verifier is given oracle access to the low degree extension of its input, then its running time is only* $t' \cdot k^2 \cdot \mathrm{polylog}(t)$*, where* t' *is the cost of the oracle access.*
2. *The protocol has perfect completeness and soundness* 2^{-k} *against* $2^{-k \cdot \mathrm{polylog}(t)}$*-no-signaling strategies.*
3. *Each query and answer is of length* $k \cdot \mathrm{polylog}(t)$*.*

Here we give a high level overview of the MIP construction of Kalai *et al.* [18]. It is obtained in three steps:

1. *No-signaling PCP with Oracle.* They first construct a Probabilisticaly Checkable Proof (PCP) with oracle access to a function which makes at most k queries and is secure against no-signaling provers. The construction of the PCP is the most technical part of their work and we refer to Kalai *et al.* [18] for the construction and analysis of this PCP. The total number of oracle queries is at most $k \cdot \mathrm{polylog}(t)$ and the running time of the verifier is $k \cdot \mathrm{polylog}(n)$.
2. *No-signaling* MIP *with Oracle.* Based on the PCP, they construct in a straightforward way an MIP with $k_{\max} \leq k \cdot \mathrm{polylog}(t)$ provers secure against no-signaling strategies given oracle access to the same function as for the PCP. In this MIP, the verifier simulates the PCP verifier, and the i-th prover prepares the PCP proof and answers the i-th query according to the PCP. The running time of the verifier is $O(k \cdot \mathrm{polylog}(t))$.
3. *No-signaling* MIP *without Oracle.* In order to remove the oracle, they employ an MIP for the oracle which is secure against no-signaling provers. They replace the number of queries to the oracle one by one, each time reducing one query to the oracle by letting the verifier run the MIP for the oracle with additional provers. At the end, they obtain an MIP without oracle access which is secure against no-signaling provers. To construct the MIP for the oracle, they observe that any interactive proof gives rise to an MIP secure against no-signaling provers by sending the first i messages to the i-th prover and letting the i-th prover answer the message in the i-th round. Observed that the oracle is computable by linear space, we have IP for this oracle so that we can obtain an MIP against no-signaling provers. The running time of the verifier is $k \cdot \mathrm{polylog}(t) + n \cdot k^2 \cdot \mathrm{polylog}(t)$. Moreover, if the verifier can compute the low degree extension of the input in time t', then the running time can be further improved into $k \cdot \mathrm{polylog}(n) + t' \cdot k^2 \cdot \mathrm{polylog}(t)$.

Note that for languages in P where $t = \mathrm{poly}(n)$, we can let $k = \mathrm{polylog}(t)$ so that the verifier runs in time $n \cdot \mathrm{polylog}(n)$. Moreover, the running time can be improved

to $t' \cdot \text{polylog}(n)$ when the verifier is given oracle access to evaluate the low degree extension and t' is the cost of the oracle access. Therefore, the task of constructing delegation scheme for P with sublinear verification can be reduced to constructing a delegation scheme for low degree extension with sublinear verification.

6.2 No-Signaling Rational Multi-Prover Proofs for Deterministic Computations

In this section, we present our RMIPs for deterministic computations which are secure against no-signaling provers. Recall from the previous section that the efficiency bottleneck for achieving sublinear verification for P is that the evaluation of low degree extension runs in quasi-linear time. To overcome the efficiency bottleneck we combine the no-signaling MIP of Kalai *et al.* [18] with our sublinear rational proofs for evaluating the low degree extension of the input (Proposition 1). Unlike the oracle simulation mentioned in the third step of the work of Kalai *et al.* [18], we reduce all queries to the low degree extension oracle at once and only increase the number of provers by 1. To do this, we view the queries to the oracle as a single query consisting of many points to a larger oracle that evaluates the low degree extension of inputs on all the points and returns the answers at once.

For a function $g\colon \mathbb{F}^n \to \mathbb{F}$, we let $g^l\colon (\mathbb{F}^n)^l \to (\mathbb{F})^l$ be the function that on any l-tuple $(x_1,\ldots,x_l) \in (\mathbb{F}^n)^l$ outputs $(g(x_1),\ldots,g(x_l))$. For a rational proof $\varphi = (V_\varphi, P_\varphi)$ for g with input $x \in \mathbb{F}^n$, we define another rational proof $\varphi^l = (V_{\varphi^l}, P_{\varphi^l})$ for g^l with input $(x_1,\ldots,x_l) \in (\mathbb{F}^n)^l$, where the verifier V_{φ^l} simulates V_φ on x_i for all $i \in \{1,\ldots,l\}$ and pays the average reward outputted by V_ϕ on the l inputs and P_{φ^l} simulates P_φ on x_i for all $i \in \{1,\ldots,l\}$. It is easy to see that if g admits rational proof φ with reward gap Δ, then g^l admits a rational proof φ^l with reward gap Δ/l.

Theorem 7. *Suppose that $f : \{0,1\}^n \to \{0,1\}$ is a function computable by deterministic Turing machine in time $t(n)$, where $t = t(n)$ satisfies $\text{poly}(n) \le t \le \exp(n)$. Then, for any integer $(\log t)^c \le k \le \text{poly}(n)$, where c is some (sufficiently large) universal constant, there exists an* RMIP *for f with $k \cdot \text{polylog}(t) + 1$ provers where:*

1. *The provers run in time $\text{poly}(t,k)$ and the verifier runs in time $k^2 \cdot \text{polylog}(t)$.*
2. *The protocol has reward gap $1/k \cdot \text{poly}(\log(t), n)$ against $2^{-k \cdot \text{polylog}(t)}$-no-signaling strategies.*
3. *Each query and answer is of length $k \cdot \text{polylog}(t)$.*

Proof. Let $\pi^g = (\overrightarrow{P_\pi}, V_\pi^g)$ be the MIP for $L_f = \{(x,y) | y = f(x)\}$ from Theorem 6, which has soundness $s = 2^{-k}$ against $\delta = 2^{-k \cdot \text{polylog}(t)}$-no-signaling strategies and perfect completeness, where $g\colon \mathbb{F}^m \to \mathbb{F}$ is the low degree extension of inputs with parameters $\mathbb{F}, \mathbb{H}, m$ such that $|\mathbb{H}| \le |\mathbb{F}| \le \text{polylog}(t)$, $|\mathbb{H}^m| = \text{poly}(n)$. As noted in [18], the total number of the queries to g is $l \le k \cdot \text{polylog}(t)$. We consider $\pi^{g^l} = (\overrightarrow{P_\pi}, V_\pi^{g^l})$ where $V_\pi^{g^l}$ behaves exactly as V^g except that V^{g^l} only makes a single query which consists all the queries of V^g to the oracle for g. Because the queries made by V_π^g are independent of each other, it is possible to query them at once and conclude that π^{g^l} is also an MIP for L_f with the same guarantee.

By Proposition 1, g admits a rational proof $\varphi = (P_\varphi, V_\varphi)$ with reward gap $\Delta = 1/(4\log|\mathbb{F}|)(|\mathbb{H}^m|^2)$. Therefore g^l admits a rational proof φ^l with reward gap $\Delta' = \Delta/l$.

Note that φ^l is also an RMIP with reward gap Δ in presence of $\delta' = 1$-no-signaling strategies.

Note that $V_\pi^{g^l}$ only issues a single query and for all x the communication between $\overrightarrow{P_\pi}$ and $V_\pi^{g^l}$ is independent of $(q, g(q))$. By Theorem 2, π^{φ^l} is an RMIP for f with reward gap $\Delta'(1-s-\min(\delta,\delta')) = \Omega(\Delta/l) = 1/k \cdot \text{poly}(\log(t), n)$, in presence of $\delta'' = \delta$-no-signaling strategies.

The running time of the prover is at most the sum of the running time of $\overrightarrow{P_\pi}$ in Theorem 6 and the running time of P_{φ^m} which is upper bounded by $\text{poly}(t,k)$. The verifier runs in at most $t' \cdot k^2 \cdot \text{polylog}(t)$ where the t' is the running time of V_φ upper bounded by $\text{poly}(\mathbb{H}, m) \leq \text{polylog}(t)$. Therefore the running time of verifier is $k^2 \cdot \text{polylog}(t)$. The maximal length of queries and answers in π^{g^l} is $k \cdot \text{polylog}(t)$ by Theorem 6, and the maximal length of queries and answers in π^{g^l} is $(m \log \mathbb{F}) \cdot l \leq k \cdot \text{polylog}(t)$. Therefore the maximal length of queries and answers in π^{g^l} is bounded by $k \cdot \text{polylog}(t)$. □

6.3 Single-Round Rational Arguments for P.

We show how to transform any RMIP secure against no-signaling provers into a single-round rational argument using a sub-exponentially secure Fully Homomorphic Encryption (see Definition 8), and as a result obtain a single-round rational argument with sublinear verification for any language in P. For that we extend the transformation of Guo *et al.* [15] to the multi-prover setting.

Theorem 8. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function in* FRMIP$[k, \delta, C, T]$. *Assume f has a* RMIP *with noticeable reward gap Δ and negligible no-signaling parameter δ, and let λ denote the length of the longest message sent by the verifier. If there exists a secure FHE scheme, where $\kappa \geq \max\{\text{polylog}(n), \lambda, C\}$ is the security parameter, then f admits single-round rational argument which has the following properties:*

1. *The verifier runs in time* $\text{poly}(\kappa) + O(T(n))$.
2. *The prover runs in time* $\text{poly}\left(\kappa, n, T_{\overrightarrow{P_{MIP}}}\right)$, *where $T_{\overrightarrow{P_{MIP}}}$ is the sum of the running times of the provers in the* RMIP.
3. *The length of prover's message and the verifier's challenge is $\ell \cdot \text{poly}(\kappa)$.*

The proof of Theorem 8 follows by the following lemma (due to space restrictions, we provide the proof of Lemma 2 in the full version).

Lemma 2. *Let $(\overrightarrow{P_{MIP}}, V_{MIP})$ be a δ-no signaling* RMIP *protocol for a function f with ℓ provers. Let λ be the longest query size and C be the answer size. Let* $\mathsf{reward}(\cdot)$ *and Δ be the reward function and the corresponding reward gap. Assume the existence of a (Z, δ')-secure FHE with correctness $1-\gamma$ (where γ is some negligible function), and let $\gamma_0 = \gamma \cdot \ell$. If $\delta' \leq \delta/\ell$ and the security parameter $\kappa = \kappa(n) \geq \max\{\text{poly}\log(n), \lambda, C\}$ and $Z = Z(\kappa) \geq \kappa$ such that $Z \geq \max\left\{n, 2^{\ell \cdot C}\right\}$, then there exists a one-round protocol (P_A, V_A) with the following properties:*

(a) $\Pr[\mathsf{output}((P_A, V_A)(x)) = f(x)] = 1$.

(b) $\mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] \geq \mathrm{E}[\mathsf{reward}((\overrightarrow{P_{MIP}}, V_{MIP})(x))] \cdot (1-\gamma_0))$.

(c) The length of P_A's message and the V_A's challenge is $\ell \cdot \text{poly}(\kappa)$.

(d) The verifier V_A runs in time $\text{poly}(\kappa) + O(T_{V_{MIP}})$, where $T_{V_{MIP}}$ is the running time of V_{MIP}.

(e) The prover P_A runs in time $\text{poly}\left(\kappa, n, T_{\overrightarrow{P_{MIP}}}\right)$, where $T_{\overrightarrow{P_{MIP}}}$ is the sum of the running times of the provers in $\overrightarrow{P_{MIP}}$.

(f) For any prover P^* of size $\leq \text{poly}(Z(\kappa))$ that achieves

$$\mathrm{E}[\mathsf{reward}((P^*, V_A)(x))] = \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] + \delta^* ,$$

let $\mu = \Pr[\mathsf{output}((P^*, V_A)(x)) \neq f(x)]$. It holds that
(a) (Utility gain) $\delta^* \leq \gamma_0$, *and*
(b) (Utility loss) $(-\delta^*) \geq \mu\Delta - \gamma_0$.

From interactive rational proofs to rational arguments. Let $(\overrightarrow{P_{MIP}}, V_{MIP})$ be a δ-no-signaling rational MIP with ℓ provers $P^1_{MIP}, \ldots, P^\ell_{MIP}$ for evaluating some function f, as in the statement of the Lemma 2. Recall that λ denotes length of the longest message sent by V_{MIP} in $(\overrightarrow{P_{MIP}}, V_{MIP})$. For simplicity of exposition (and without loss of generality) we assume that the first prover P^1_{MIP} sends $f(x)$, and all queries are of size exactly λ.

Fix any security parameter $\kappa \geq \max\{\text{polylog}(n), \lambda, C\}$ and let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ be a (Z, δ')-secure FHE scheme, with respect to security parameter κ. The one-round rational argument (P_A, V_A) is constructed as follows:

1. On common input $x \in \{0,1\}^n$, the verifier V_A proceeds as follows:
   (a) Emulate the verifier V_{MIP} and obtain queries $m_1, \ldots, m_\ell \in \{0,1\}^\lambda$ to be sent by V_{MIP}.[7]
   (b) Compute key-pairs $(pk_i, sk_i) \leftarrow \mathsf{Gen}(1^\kappa)$ and encryptions $q_i \leftarrow \mathsf{Enc}(pk_i, m_i)$ for $1 \leq i \leq \ell$.
   Send $pk = (pk_1, \ldots, pk_\ell)$ and $q = (q_1, \ldots, q_\ell)$ to P_A.
2. Upon receiving keys $pk = (pk_1, \ldots, pk_\ell)$ and queries $q = (q_1, \ldots, q_\ell)$ from V_A, the prover P_A operates as follows:
   (a) Emulate provers $\overrightarrow{P_{MIP}}$ to obtain $f(x)$.
   (b) For each $1 \leq i \leq \ell$, compute $P_{x,i}$, a Boolean circuit that on input query m computes the function $P^i_{MIP}(x, m)$.
   (c) For each $1 \leq i \leq \ell$, compute $a_i \leftarrow \mathsf{Eval}(pk_i, P_{x,i}, q_i)$ and send the message $(f(x), a_1, \ldots, a_\ell)$ to V_A.
3. Upon receiving the message $(f(x), a_1, \ldots, a_\ell)$ from P_A, the verifier V_A operates as follows:
   (a) For every $1 \leq i \leq \ell$, compute $b'_i \leftarrow \mathsf{Dec}(sk_i, a_i)$.
   (b) Emulate V_{MIP} on $\left(f(x), b'_1, \ldots, b'_\ell\right)$, as if each b'_i is P^i_{MIP}'s response.
   (c) Output whatever V_{MIP} outputs (i.e., $f(x)$ and '1' with probability of the computed reward).

[7] These queries can be computed in advance since in the protocol $(\overrightarrow{P_{MIP}}, V_{MIP})$ all the messages sent by V_{MIP} depend only on V_{MIP}'s random coin tosses.

Proof (of Theorem 8). The running time of the verifier, the communication complexity, and property (a) of Definition 5 of rational arguments are all explicitly provided by Lemma 2. It remains to show property (b) and property (c) of definition of rational arguments.

The utility gain is $\delta^* \leq \gamma_0 \leq \kappa \cdot \mathrm{negl}(\kappa) = \mathrm{negl}(n)$. By the definition of δ^* we have, $\mathrm{negl}(n) + \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] \geq \delta^* + \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))]$ which is equal to $\mathrm{E}[\mathsf{reward}((P^*, V_A)(x))]$. Hence, the property (b) of rational arguments holds.

To show property (c) of Definition 5, we assume that $\mu \geq p^{-1}(|x|)$ for some polynomial $p(\cdot)$. Due to the noticeable Δ, we know that $\mu\Delta \geq q_1^{-1}(|x|)$ for some polynomial $q_1(\cdot)$. From the utility loss bound we obtain that

$$(-\delta^*) \geq \mu\Delta - \gamma_0 = \mu\Delta - \mathrm{negl}(n) \geq q_1^{-1}(|x|) - \mathrm{negl}(n) \geq q_1^{-1}(|x|)/2 \ .$$

By defining polynomial $q(\cdot)$ to be $q(|x|) = 2q_1(|x|)$ we get

$$\begin{aligned} \mathrm{E}[\mathsf{reward}((P_A, V_A)(x))] &= \mathrm{E}[\mathsf{reward}((P^*, V_A)(x))] - \delta^* \\ &\geq \mathrm{E}[\mathsf{reward}((P^*, V_A)(x))] + q^{-1}(|x|) \ , \end{aligned}$$

as desired. □

By applying the above transformation on the no-signaling RMIP protocol presented in Theorem 7, we obtain the following single-round rational arguments for P with sublinear verification.

Corollary 4 (Rational Argument for P). *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by deterministic Turing machine in time $\mathrm{poly}(n) \leq T(n) \leq \exp(n)$ and let $k = \mathrm{polylog}(T(n))$. Let $\kappa \geq \mathrm{polylog}(T(n)) \cdot k$ be a security parameter and let $Z = Z(\kappa)$ be such that $2^{(\log T(n))^c} \leq Z \leq 2^\kappa$ for sufficiently large constant c. If there exists $(Z, 2^{-k^2 \cdot \mathrm{polylog}(T(n))})$-secure FHE scheme then f admits single-round efficient rational argument which has the following properties:*

1. *The verifier runs in time $\mathrm{poly}(\kappa, \log(T(n)))$ and the prover runs in $\mathrm{poly}(\kappa, T(n))$.*
2. *The length of prover's message and the verifier's challenge is $k \cdot \mathrm{poly}(\kappa, \log(T(n)))$. The verifier's challenge depends only on his random coins and is independent of the input x.*

Proof. Suppose that $f \in \mathrm{DTIME}(T)$, where $T = T(n)$ satisfies $\mathrm{poly}(n) \leq T \leq \exp(n)$ and set $k = \mathrm{polylog}(T)$. Let $\kappa = \kappa(n)$ be a security parameter such that $k \cdot \mathrm{polylog}(T) \leq \kappa$. Let $Z = Z(\kappa)$ such that $2^{(\log T)^c} \leq Z \leq 2^\kappa$ for sufficiently large universal constant c satisfying $Z \geq \max\{n, 2^{k^2 \cdot \mathrm{polylog}(T)}\}$. Let $\delta' = 2^{-k^2 \mathrm{polylog}(T)}$. By applying Theorem 7 (with respect to the parameter k) to the function f, we obtain an RMIP for f with $k \cdot \mathrm{polylog}(T)$ provers and reward gap $1/k \cdot \mathrm{poly}(\log(T), n)$ against $2^{-k \cdot \mathrm{polylog}(T)}$-no-signaling strategies. The verifier of the RMIP runs in time $k^2 \cdot \mathrm{polylog}(T)$ and the provers run in time $\mathrm{poly}(T, k)$. Each query and answer is of length $k \cdot \mathrm{polylog}(T)$. Assume that there exists an (Z, δ')-secure FHE.

By Theorem 8, we obtain that f has a 1-round rational argument. The running time of the verifier is $\mathrm{poly}(\kappa, \log(T)$ and the running time of the prover is $\mathrm{poly}(\kappa, T)$. The message of the prover and the verifier is of length $k \cdot \mathrm{poly}(\kappa, \log T)$. □

We remark that Corollary 3 could be alternatively obtained using our new transformation presented in Theorem 8. This is done by first transforming the rational interactive proof for NC to RMIP (with only negligible loss in the reward gap) and then applying Theorem 8 on the resulted RMIP.

References

1. 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 *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 463–474, 2000.
2. Pablo D. Azar and Silvio Micali. Rational proofs. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 1017–1028. ACM, 2012.
3. Pablo D. Azar and Silvio Micali. Super-efficient rational proofs. In Michael Kearns, R. Preston McAfee, and Éva Tardos, editors, *ACM Conference on Electronic Commerce*, pages 29–30. ACM, 2013.
4. Mira Belenkiy, Melissa Chase, C. Christopher Erway, John Jannotti, Alptekin Küpçü, and Anna Lysyanskaya. Incentivizing outsourced computation. In *Proceedings of the ACM SIGCOMM 2008 Workshop on Economics of Networked Systems, NetEcon 2008, Seattle, WA, USA, August 22, 2008*, pages 85–90, 2008.
5. Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, pages 315–333, 2013.
6. Andrew J. Blumberg and Michael Walfish. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, 2015.
7. Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.
8. Ran Canetti and Margarita Vald. Universally composable security with local adversaries. In Ivan Visconti and Roberto De Prisco, editors, *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, volume 7485 of *Lecture Notes in Computer Science*, pages 281–301. Springer, 2012.
9. Jing Chen, Samuel McCauley, and Shikha Singh. Rational proofs with multiple provers. *CoRR*, abs/1504.08361, 2015.
10. Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 54–74. Springer, 2012.
11. Cynthia Dwork, Michael Langberg, Moni Naor, Kobbi Nissim, and Omer Reingold. Succinct NP proofs and spooky interactions. *Available at* `www.openu.ac.il/home/mikel/papers/spooky.ps`, 2004.
12. Juan A. Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 648–657. IEEE Computer Society, 2013.
13. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 99–108. ACM, 2011.
14. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27, 2015.
15. Siyao Guo, Pavel Hubáček, Alon Rosen, and Margarita Vald. Rational arguments: single round delegation with sublinear verification. In Moni Naor, editor, *Innovations in Theoretical*

Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014, pages 523–540. ACM, 2014.
16. Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. In Tim Roughgarden, editor, *Innovations in Theoretical Computer Science, ITCS'15, Rehovot, Israel, January 11-13, 2015*, pages 133–142. ACM, 2015.
17. Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 565–574. ACM, 2013.
18. Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 485–494. ACM, 2014.
19. Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 422–442. Springer, 2015.
20. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *STOC*, pages 723–732. ACM, 1992.
21. Ranjit Kumaresan and Iddo Bentov. How to use bitcoin to incentivize correct computations. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 30–41. ACM, 2014.
22. Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.
23. Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.
24. Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2003.
25. Viet Pham, M. H. R. Khouzani, and Carlos Cid. Optimal contracts for outsourced computation. In Radha Poovendran and Walid Saad, editors, *Decision and Game Theory for Security - 5th International Conference, GameSec 2014, Los Angeles, CA, USA, November 6-7, 2014. Proceedings*, volume 8840 of *Lecture Notes in Computer Science*, pages 79–98. Springer, 2014.
26. Guy N. Rothblum. *Delegating computation reliably: paradigms and constructions*. PhD thesis, Massachusetts Institute of Technology, 2009.
27. Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 793–802. ACM, 2013.
28. Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.
29. Yihua Zhang and Marina Blanton. Efficient secure and verifiable outsourcing of matrix multiplications. In Sherman S. M. Chow, Jan Camenisch, Lucas Chi Kwong Hui, and Siu-Ming Yiu, editors, *Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings*, volume 8783 of *Lecture Notes in Computer Science*, pages 158–178. Springer, 2014.

A Building Blocks

Here we provide only the main claim about efficiency of evaluation of a low degree extension. For an in-depth exposition see e.g. Rothblum [26].

Proposition 2 ([26]). *There exists a Turing machine that takes as input an extension field $\mathbb{H}$ of $\mathbb{GF}[2]$, an extension field $\mathbb{F}$ of $\mathbb{H}$, an integer m, and $w = (w_0, \ldots, w_{m-1}) \in \mathbb{H}^m$. The machine runs in time $\mathrm{poly}(|\mathbb{H}|, m)$ and space $O(\log|\mathbb{H}| + \log m)$, and it outputs the unique $2m$-variate polynomial $\tilde{\beta} : \mathbb{F}^m \times \mathbb{F}^m \to \mathbb{F}$ of degree at most $|\mathbb{H}| - 1$ in each variable (represented as an arithmetic circuit of degree at most $|\mathbb{H}| - 1$ in each variable), such that for every $z \in \mathbb{F}^m$, it holds for the unique low degree extension $\tilde{W} : \mathbb{F}^m \to \mathbb{F}$ of w that $\tilde{W}(z) = \sum_{p \in \mathbb{H}^m} \tilde{\beta}(z,p) \cdot W(p)$, where $W : \mathbb{H}^m \to \mathbb{F}$ is the function corresponding to $(w_0, \ldots, w_{n-1})$ defined using the lexicographic ordering α of $\mathbb{H}^m$ as $W(p) = w_{\alpha(p)}$ if $\alpha(p) \leq n-1$ and otherwise 0. Moreover, $\tilde{\beta}$ can be evaluated in time $\mathrm{poly}(|\mathbb{H}|, m)$ and space $O(\log|\mathbb{H}| + \log m)$. Namely, there exists a Turing machine with above time and space bounds, that takes an input parameters $\mathbb{H}, \mathbb{F}, m$ and a pair $(z,p) \in \mathbb{F}^m \times \mathbb{F}^m$ and outputs $\tilde{\beta}(z,p)$.*

Fully homomorphic encryption. A public-key fully homomorphic encryption scheme consists of four probabilistic polynomial-time algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$. The key generation algorithm Gen, when given as input a security parameter 1^κ, outputs a pair (pk, sk) of public and secret keys. The encryption algorithm, Enc, on input a public key pk and a message $m \in \{0,1\}^{poly(\kappa)}$, outputs a ciphertext q, The homomorphic evaluation algorithm, Eval, on input the public-key pk, a circuit $C : \{0,1\}^a \to \{0,1\}^b$, where $a, b \leq \mathrm{poly}(\kappa)$, and a ciphertext q that is an encryption of a message $m \in \{0,1\}^a$ with respect to pk, outputs a ciphertext $\tilde{q}$ of length $\mathrm{poly}(\kappa, a, b)$ that is an evaluation of C over q. The decryption algorithm, Dec, when given a ciphertext q and the secret key sk, outputs the original message m. We allow the decryption process to fail with negligible probability (over the randomness of all algorithms).

Definition 8 (Fully homomorphic encryption). *A public-key fully homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$, satisfies the following properties.*

Completeness *For every security parameter κ, for every message $m \in \{0,1\}^{poly(\kappa)}$ and for every circuit C taking inputs of length $|m|$,*

$$\Pr_{(pk,sk) \leftarrow \mathsf{Gen}(1^\kappa)} \left[C(m) = \mathsf{Dec}(sk, \tilde{q}) \,\middle|\, \begin{array}{c} q \leftarrow \mathsf{Enc}(pk, m) \\ \tilde{q} \leftarrow \mathsf{Eval}(pk, C, q) \end{array} \right] = 1 - \mathrm{negl}(\kappa).$$

Security *For every polynomial $p(\cdot)$ and every polynomial size distinguisher $\mathcal{D}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every sufficiently large security parameter κ and every pair of messages $m_0, m_1 \in \{0,1\}^{p(\kappa)}$*

$$\left| \Pr_{\substack{(pk,sk) \leftarrow \mathsf{Gen}(1^\kappa) \\ b \leftarrow \{0,1\}}} [\mathcal{D}(pk, \mathsf{Enc}(pk, m_b)) = b] - \frac{1}{2} \right| < \mathrm{negl}(\kappa)$$

where the probability is also over the random coin tosses of Enc.

We say that the encryption scheme is (S, δ)-secure, for a function $S : \mathbb{N} \to \mathbb{N}$ and a negligible function $\delta : \mathbb{N} \to [0,1]$, if the security property holds for every adversary of size $\mathrm{poly}(S(\kappa))$, with distinguishing gap at most $\delta(\kappa)$.