Constant-Round Arguments from One-Way Functions

Constant-Round Arguments from One-Way Functions

Noga Amit
Weizmann Institute of Science
Rehovot, Israel
noga.amit@weizmann.ac.il

Guy N. Rothblum
Apple
Cupertino, CA, USA
rothblum@alum.mit.edu

ABSTRACT

We study the following question: what cryptographic assumptions are needed for obtaining constant-round computationally-sound argument systems? We focus on argument systems with almost-linear verification time for subclasses of P, such as depth-bounded computations. Kilian's celebrated work [STOC 1992] provides such 4-message arguments for P (actually, for NP) using collision-resistant hash functions. We show that *one-way functions* suffice for obtaining constant-round arguments of almost-linear verification time for languages in P that have log-space uniform circuits of linear depth and polynomial size. More generally, the complexity of the verifier scales with the circuit depth. Furthermore, our argument systems (like Kilian's) are doubly-efficient; that is, the honest prover strategy can be implemented in polynomial-time. Unconditionally sound interactive proofs for this class of computations do not rely on any cryptographic assumptions, but they require a linear number of rounds [Goldwasser, Kalai and Rothblum, STOC 2008]. Constant-round interactive proof systems of linear verification complexity are not known even for NC (indeed, even for AC^1).

CCS CONCEPTS

• **Theory of computation → Interactive proof systems**; **Cryptographic protocols**.

KEYWORDS

Interactive arguments, delegation, one-way functions, doubly-efficient proof systems

ACM Reference Format:
Noga Amit and Guy N. Rothblum. 2023. Constant-Round Arguments from One-Way Functions. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3564246.3585244

1 INTRODUCTION

A proof-system allows an untrusted prover to convince a verifier that a complex claim is true. The claim is usually framed as the membership of an input x in a language $\mathcal{L}$, where verification should be more efficient than deciding membership in $\mathcal{L}$. In a computationally sound *argument system* [7], soundness is relaxed to hold only against polynomial-time cheating provers, under cryptographic assumptions. If $x \notin \mathcal{L}$, then no polynomial-time cheating prover should be able to get the verifier to accept (except with small probability). Understanding the cryptographic assumptions needed to construct argument systems, and the expressive power of these protocols (i.e., which languages have argument systems) is a central question in the foundations of cryptography.

We study this question, focusing on efficient argument systems that have a constant number of rounds and almost-linear communication and verification time, and on constructing such argument systems for subclasses of P, such as depth-bounded computations. Kilian's [25] celebrated work showed that, assuming the existence of collision-resistant hash functions (CRHs), every language in P has a 4-message argument system with sublinear communication and almost-linear verification time (actually, this result applies to all of NP, but for now we focus on P and its subclasses). Kilian's result demonstrated that CRHs are sufficient for constructing argument systems that go well beyond what is known for *unconditionally sound* interactive proof systems (IPs) [15] (and, in some regimes, beyond what is plausible for IPs [10, 13]). It is not well understood, however, whether such argument systems can be based on assumptions that are significantly weaker than collision-resistant hashing. The quest to understand the minimal assumptions needed for implementing cryptographic primitives is a central theme in the theoretical study of cryptography. Considering argument systems through this lens, we ask:

Can constant-round computationally sound argument systems with almost-linear communication and verification time be based on the "minimal"[1] assumption of one-way functions (OWFs)?
Does their power extend beyond what is known (or plausible) using unconditionally sound IPs?

We answer these questions in the affirmative. Our main result is a constant-round argument system, whose security only relies on the existence of one-way functions, where the communication and the verification time grow linearly with the depth of the circuits computing the language:

Theorem 1.1 (Constant-round arguments from one-way functions). *If one-way functions exist, then for every language $\mathcal{L}$ that is computable by log-space uniform circuits of fan-in 2, depth $D(n)$ and polynomial size, and for every desired constant $\sigma \in (0, 1]$, there is a constant-round public-coin argument system, with perfect completeness and negligible soundness error against* $\mathrm{poly}(n)$*-time cheating provers, where n is the input length. The protocol's complexities are:*

[1] One-way functions are often referred to as a "minimal" assumption for cryptography, and in many cases one-way functions are *essential* for constructing cryptographic primitives [20]. We note, however, that it is *not* known whether one-way functions are *essential* for constructing arguments. Wee [38] studies the relationship between non-trivial arguments and various assumptions.

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585244

- *Constant round complexity* $O(1/\sigma^3)$,[2]
- *Communication complexity* $O(n^\sigma \cdot D(n))$,
- *The verifier runs in time* $O(n^\sigma \cdot D(n) + n^{1+\sigma})$, *while the honest prover runs in* $\mathrm{poly}(n)$ *time.*

The number of rounds is a constant, where this constant depends on the desired communication complexity and verification time. For linear-depth computations, for any desired constant σ, the communication and the verification time can be $O(n^{1+\sigma})$ using $O(1/\sigma^3)$ rounds. The protocol is *doubly-efficient*: the honest prover runs in polynomial time. Thus, this argument system can be used for delegating computation to an untrusted server, and obtaining a proof that a claimed output of the computation is correct [14]. For simplicity, we take the security parameter to be a small polynomial in the input length throughout (obtaining security against $\mathrm{poly}(n)$-time adversaries). More generally, the communication, verifier runtime and prover runtime depend polynomially on the security parameter. Finally, having established that one-way functions suffice for extending the power of argument systems beyond what is known for interactive proof systems, a natural question for future work is whether one-way functions suffice for constructing arguments that can be verified in almost-linear time for all of P (or even for all of NP).

Comparison to known argument systems. As noted above, assuming the existence of CRHs, there exist 4-message doubly-efficient arguments with sublinear communication and almost-linear verification time for all of P [25]. In fact, this celebrated protocol can be used for all of NP, i.e., a much richer class of computations. Indeed, such arguments exist even beyond NP, as was shown by Micali [28] (under stronger assumptions) and by Barak and Goldreich [3]. One major focus, starting with [28], is reducing the round complexity to "non-interactive" or to 2-message protocols, under minimal cryptographic assumptions. This vast literature is too vast for us to survey here. See e.g. Kalai, Raz and Rothblum [24] and Choudhuri, Jain and Jin [8], as well as the recent expositions by Thaler [37] and Ishai [22, 23], and the references therein. Many of these works construct protocols for P or for subclasses of P. The vast majority of works on argument systems use assumptions that (at the very least) imply CRHs. One exception is works by Bitansky, Kalai and Paneth [6] and by Komargodski, Naor and Yogev [26], who construct argument systems based on the existence of the more relaxed primitive of multi-collision-resistant hash functions, though the recent work of Rothblum and Vasudevan [35] indicates that the gap between collision-resistance and multi-collision-resistance might not be wide.

The main distinction in our work is that *we rely only on the existence of one-way functions*, but our result is for a more restricted class of depth-bounded computations, and the round complexity, while constant, is larger than in Kilian's 4-message protocol (let alone the subsequent works that further reduce the interaction using stronger assumptions). OWFs are generally considered to be a considerably more relaxed assumption than CRHs. Simon [36] showed a black-box separation between the two notions. We remark that while the long-standing open question of constructing CRHs from OWFs is well beyond the current state of the art, Holmgren and Lombardi [19] do show that an exponentially hard variant of OWFs is sufficient for constructing CRHs.

Comparison to Interactive Proofs. A parallel body of work studies the expressive power of unconditionally sound interactive proof systems. For the class of log-space uniform poly-size depth D circuits, known doubly-efficient IPs (DEIPs) require round complexity that is quasi-linear in D [14], compared with the constant round complexity in our new protocol (though we note that our protocol is computationally sound, assumes the existence of one-way functions, and its communication complexity and verification time are larger by a small polynomial factor). See 1e 1 for a full comparison. The gap in the round complexity is not merely for known protocols: we find it quite plausible to conjecture that there do not exist constant-round interactive proofs for linear depth computations (indeed, one could conjecture that they do not even exist for AC^1 circuits):

Remark 1.2 (IPs for linear depth.). *In an interactive proof, we require a constant gap between the completeness and the soundness error. A significantly more relaxed requirement is to only have some infinitesimal gap between the completeness and the soundness error. Essentially all known results for IPs become straightforward if one is willing to make this relaxation (in particular, the gap can be smaller than the inverse of the computation size). Nonetheless, for linear-depth computations, it is not known how to construct a constant-round DEIP with* **any** *gap between the completeness and the soundness error. We find it plausible to conjecture that no such proof system exists. Indeed, even for* AC^1 *no such proof system is known. Of course, this assumption precludes the possibility of getting a DEIP with a* constant gap*, but our main result shows this* is *possible for an argument system (assuming OWFs).*

In Table 1 we compare the power and complexity of known doubly-efficient IPs.

Elaborating on the relationship to known constant-round DEIPs: Reingold, Rothblum and Rothblum [31] showed constant-round interactive proofs for languages that can be decided in bounded-polynomial space and polynomial time. This class is incomparable to the class of languages in our main result (languages with linear-depth polynomial-size circuits). Though the classes are incomparable, we view the round complexity in our construction as much smaller. Fixing a constant σ, to get communication complexity $O(n^\sigma \cdot \mathrm{poly}(S))$ for an S-space computation, the RRR protocol uses $\exp(\tilde{O}(1/\sigma))$ many rounds. In our protocol, on the other hand, $O(1/\sigma^3)$ rounds suffice for obtaining $O(n^\sigma \cdot D)$ communication for a depth-D computation. Goldreich and Rothblum [12] constructed constant-round protocols for highly uniform variants of the complexity classes $\mathsf{AC}^0[\oplus]$, using $O(1/\sigma)$ rounds, and NC^1, using $O(1/\sigma^2)$ rounds. The main distinction with our work is that our new protocol applies to computations well beyond NC^1. We remark, however, that the constant-round GR protocol for $\mathsf{AC}^0[\oplus]$ plays an important role in our construction.

Zero-Knowledge Arguments. Based on Theorem 1.1, we show that the existence of one-way functions suffices for succinct *constant*

[2] By $O(1/\sigma^3)$ we mean that there exists a universal constant c s.t. the round complexity is at most c/σ^3.

Table 1: Comparison of doubly-efficient proof systems, where $\sigma > 0$ is a desired constant for bounding the communication. The proof systems from prior works are unconditionally sound, whereas our new result is computationally sound assuming the existence of one-way functions.

class	(in)	# rounds	communication	verifier time	uniformity
depth D, size S	[14]	$O(D \cdot \log S)$	$D \cdot \text{polylog}(S)$	$n \cdot \text{poly}(D, \log S)$	log-space uniform
Space S, poly time	[31]	$\exp(\tilde{O}(1/\sigma))$	$n^{\sigma} \cdot \text{poly}(S)$	$n^{\sigma} \cdot \text{poly}(S) + \tilde{O}(n)$	none (Turing machine)
$\text{AC}^0[\oplus]$	[12]	$O(1/\sigma)$	$n^{\sigma+o(1)}$	$n^{1+o(1)}$	adjacency predicate[a]
NC^1	[12]	$O(1/\sigma^2)$	$n^{\sigma+o(1)}$	$n^{1+o(1)}$	incidence function[b]
depth D, poly-size	(this)	$O(1/\sigma^3)$	$O(n^{\sigma} \cdot D)$	$O(n^{\sigma} \cdot D + n^{1+\sigma})$	log-space uniform

[a]The circuit's *adjacency predicate* should be computable by a $n^{o(1)}$-size formula that can be constructed in $n^{1+o(1)}$-time.
[b]The circuit's *incidence function* should be computable by a $n^{o(1)}$-size formula that can be constructed in $n^{1+o(1)}$-time.

round zero-knowledge argument systems, with perfect completeness and constant soundness error, for NP relations whose verification circuit has bounded-polynomial depth. Succinctness means that the communication is nearly-linear in the witness length.

Theorem 1.3 (Constant-round succinct zero-knowledge for bounded-polynomial depth relations from one-way functions). *If one-way functions exist, then for every language $\mathcal{L} \in \text{NP}$ whose relation is computable by log-space uniform circuits of fan-in 2, depth $D(n)$ and polynomial size, and for every desired constant $\sigma \in (0, 1]$, there is a constant-round computational zero-knowledge argument system as follows. The argument system is public-coin and has perfect completeness and constant soundness error against $\text{poly}(n)$-time cheating provers, where n is the input length. The protocol has constant round complexity $O(1/\sigma^3)$. Taking $M(n)$ to be the witness length, the communication complexity is $O(n^{\sigma} \cdot (M(n) + D(n)))$ and the verifier runtime is $O(n^{\sigma} \cdot (n + M(n) + D(n)))$. Given a witness w for x's membership in $\mathcal{L}$, the honest prover runs in $\text{poly}(n)$ time.*

Proof sketch. The zero-knowledge argument system follows from the public-coin protocol of Theorem 1.1, using a standard transformation for public-coins protocols [5, 21]. The prover and the verifier run the protocol of Theorem 1.1 on the circuit computing the NP relation, with respect to input (x, w) (where x is the input and w is the witness). Rather than sending its messages in the clear, the prover sends computationally hiding and statistically binding bit commitments to the witness and to its messages. Such bit commitments can be constructed from one-way functions [29]. After completing the protocol, the prover and the verifier run a non-succinct zero-knowledge proof to show that the verifier would accept if it saw the witness and the messages inside the prover's commitments (we need to use refinements to the 3-coloring protocol of [11] to get constant soundness with communication and verification time that are nearly-linear in the circuit *size* for this final statement). □

The constant soundness error in Theorem 1.3 can be amplified, but we need to use sequential repetition to maintain zero-knowledge. Indeed, constructing even non-succinct constant-round zero-knowledge arguments with negligible soundness error from one-way functions is a long-standing open problem. We also remark that we could use *statistically hiding* and computationally binding commitments, to obtain an analogous succinct statistical zero-knowledge argument, but known constructions of such commitments from OWFs [18] are not constant-round (indeed, there are black-box lower bounds [17]). The resulting protocol would have a small polynomial number of rounds.

Comparison to prior work on succinct ZK from one-way functions. The comparison to the state of the art from prior work on constructing zero-knowledge from one-way functions is analogous to the comparison of Theorem 1.1 to prior work on DEIPs (all ZK protocols we compare to here are unconditionally sound proof systems). For NP relations computable by poly-size linear-depth circuits, the GKR protocol gives a succinct proof system with $\tilde{O}(n)$ rounds. The RRR protocol gives constant-round succinct zero-knowledge for an incomparable class of poly-time bounded-polynomial space relations. The GR protocol imply constant-round succinct zero-knowledge for $\text{AC}^0[\oplus]$ and NC^1 relations.

2 TECHNICAL OVERVIEW

We outline the key ideas underlying our constant-round argument construction (Theorem 1.1). We begin with a brief review on universal one-way hash functions (UOWHFs) and hash trees.

2.1 Background: UOWHFs, Local Openings and Holographic Proof-Systems

UOWHFs. A family $\mathcal{H}$ of universal one-way hash functions (UOWHFs), introduced by Naor and Yung [30], is a family of shrinking functions with the following property: fixing an input x, and drawing a random hash function h from the family $\mathcal{H}$, it is hard to find a "second preimage" $x' \neq x$ s.t. $h(x') = h(x)$. This is sometimes referred to as second-preimage collision resistance, or targeted collision resistance. Note that the order of events is important: the input x should be fixed *before* the hash function $h \sim \mathcal{H}$ is selected. This is a considerable relaxation to collision-resistant families, where even after h is chosen, it should be hard to find *any* collision (in a UOWHF, after h is revealed, it may well be possible to adaptively compute a pair x', x'' that collide, but they will not collide with any input that was fixed before h was chosen). Indeed, Rompel [33] showed that UOWHFs can be constructed from any one-way function (Naor and Yung showed a construction from one-way permutations). Our construction uses a family of UOWHFs that

map inputs in $\{0,1\}^{\kappa^2}$ to outputs in $\{0,1\}^{\kappa}$, where the security parameter κ is generally taken to be n^{ε} for a small constant $\varepsilon \in (0,1]$, and the "shrinkage" factor is $1/\kappa = 1/n^{\varepsilon}$.

Local opening. UOWHFs can be used in a hash tree to hash an M-bit string $x \in \{0,1\}^M$ to a short commitment (or hash root) $y \in \{0,1\}^{\kappa}$. The root y can later be used to *locally open* any desired bit x_i. The construction divides the string x into "chunks" of length κ and places them on the leaves of a binary tree of depth $\ell \approx \log(M/\kappa)$ (i.e., at layer ℓ of the tree). For each internal node in the tree, its value is the output of a UOWHF applied to (the concatenation of) its children's values. Thus, nodes are hashed up the tree, and the value at the root is the commitment or hash-root. Later, we can "open" $x[i]$ (the i^{th} bit of x) by revealing the hash values along the path from the root to the leaf containing the i^{th} bit, together with the value of the sibling of each node along this path. The two important properties of this construction are:

(1) **Local opening:** given any i, the local opening for $x[i]$ only requires sending $O(\kappa \cdot \log M)$ bits and can be verified in $\text{poly}(\log M, \kappa)$ time.
(2) **Local targeted collision resistance:** for any string x fixed before choosing the hash functions (see below), taking $\text{root}(x)$ to be the (correct) hash root according to x, it is hard to find an index i and a valid opening for any value of the i^{th} bit that is different from $x[i]$.

Bellare and Rogaway [4] observed that using a single UOWHF in a straightforward hash tree [9, 27] might not be secure. However, the construction is secure if we use a separate hash function for each layer of the tree ($(\ell-1)$ UOWHFs in all).

Our construction uses a d-ary hash tree, which is a straightforward extension of the binary tree described above. The depth is $\ell \approx \log_d(|x|/\kappa)$, and local opening requires sending $O(\ell \cdot d \cdot \kappa)$ bits. We comment that this construction is not a "standard" commitment scheme in the sense that it is not necessarily hiding (and, as described in the second item, only satisfies the relaxed "targeted" binding property when comparing to the one implied by CRH).

Holographic proof systems. In a *holographic* proof system, the verifier is not given access to the main input explicitly. Instead, it outputs a claim about the *encoding of the input* under a high distance error correcting code. Given this redundancy, the verifier should run in sublinear time in the input length. Holographic proof systems were introduced by Babai *et al.* [2] in the context of PCPs. Holographic Interactive Proofs were formalized and generalized by Gur and Rothblum [16].

In our work, the code used for the "holographic input" x will always be the low-degree extension (LDE, see the full version for a formal definition). After interacting with the prover, the verifier either rejects, or it outputs a claim (r, v) about the input's encoding $\text{LDE}(x)$, where r is a location in $\text{LDE}(x)$, and v is a claim about the value of $\text{LDE}(x)$ at location r. Completeness means that if the prover's statement is true and the prover follows the protocol, then the verifier doesn't reject and it holds that $\text{LDE}(x)[r] = v$. Soundness means that if the prover's statement is false, then the probability that the verifier doesn't reject and $\text{LDE}(x)[r] = v$ is small. The statement can either be that the input x is in a language $\mathcal{L}$, or that $f(x) = y$ for a specified function f and claimed output y (verifying membership in a language corresponds to the case where f is a Boolean-valued function).

Holographic interactive proofs are at the heart of many IP systems. In particular, all the proof systems in Table 1 have holographic variants, where the verifier's runtime is reduced to being nearly-linear in the communication complexity, while the number of rounds, communication complexity, and prover runtime are unchanged.[3] If the claim is about evaluating a function f with output length $|y|$ (rather than membership in a language), then the communication complexity and verification time grow by an additive term that is nearly-linear in $|y|$.

2.2 An Argument-System Template

We begin by describing a "template protocol" for constructing argument systems, which allows us both to prove a warm-up for our main result, and also to introduce important ideas and concepts from the full protocol. In particular, we isolate a new primitive: a *holographic hash root* protocol, which suffices for constructing argument systems that have a small number of interaction rounds.

The warm-up in this section gives an argument system for linear-depth circuits with round complexity $\exp(\tilde{O}(1/\sigma))$, and nearly-linear communication and verification time. We note that while the number of rounds is an exponentially larger constant than in our main result, it already goes well beyond what is known (and, under plausible assumptions, beyond what is possible) using unconditionally sound interactive proofs. Let C be a log-space uniform circuit ensemble with size $S = S(n) = \text{poly}(n)$ and depth $D = D(n)$. Without loss of generality, we assume that the circuit is layered, where the gates in layer i are the ones at distance $(i-1)$ from the circuit's output gate, and for each i, the gates in layer i are fed (only) by gates in layer $(i+1)$. We also pad the circuit so that each layer is exactly of width S. On input x, the template protocol proceeds as follows:

(1) The verifier chooses UOWHFs $\bar{h}$ for a hash tree on $M = \text{poly}(S)$ bits and sends them to the prover.
(2) For each layer i of the circuit C, let $V_i \in \{0,1\}^S$ be the values of the gates in layer i when the circuit is evaluated on the input x. Let $\widehat{V_i} \in \{0,1\}^M$ be the encoding of the i^{th} layer using the low-degree extension (see above). For each $i \in [1, \ldots, D-1]$, the prover computes $\widehat{V_i}$, hashes it using the hash tree, and sends the root $y_i = \text{root}(\widehat{V_i})$ to the verifier.
(3) The verifier receives alleged hash roots $\{\widetilde{y_i}\}_{i=1}^{D-1}$. The prover and the verifier run in parallel $(D-1)$ executions of an unconditionally sound holographic interactive proof (HIP, see above). The i^{th} execution is on (holographic) input V_{i+1} (the values of gates at layer $(i+1)$), and proves that $\widetilde{y_i}$ is the correct hash root for the low-degree extension of the values of the gates *in the i^{th} layer*, where these latter values (of layer i) are computed by applying the gates in the i^{th} circuit layer to the string V_{i+1} that is the input to the proof system.
The outputs of these $(D-1)$ executions are claims $\{(r_i, v_i)\}_{i=2}^{D}$, where the i^{th} claim alleges that the value of the low-degree extension $\widehat{V_i}$ at location r_i has value v_i. We also add the claim

[3] The GR proof-system for AC^0 is not holographic as-is, but modifying it to be holographic is straightforward, see the full version for more details.

that the circuit accepts as a claim (r_1, v_1) about the LDE of the output layer.

(4) The verifier accepts if the following checks pass:
   (a) none of the HIP executions rejected.
   (b) the verifier asks the prover to perform a local opening for each $\widetilde{y_i}$: opening the r_i^{th} location in the hashed string, and showing that its value is v_i.
   (c) For the input layer, the verifier also checks that $\widehat{x}[r_D] = v_D$ (this requires evaluating the input's low-degree extension at a single point).

Completeness and Soundness. Completeness follows by construction. For soundness, let $\widehat{V_i}$ be the correct LDE of the gates in layer i of the circuit (when evaluated on the input x), and consider the hash roots $\{\widetilde{y_i}\}_{i=1}^{D-1}$. A critical insight in our analysis is that the values $\{\widehat{V_i}\}$ *are fixed before the verifier chooses its hash functions*. Thus, the hash tree's local targeted collision resistance applies, and for each i, if $\widetilde{y_i}$ is "correct", i.e. if it is the real value at the root of the hash tree on $\widehat{V_i}$, then in Step (4b), the prover cannot open $\widetilde{y_i}$ to any other value except $\widehat{V_i}[r_i]$. This will be quite helpful for catching the prover if it is cheating.

We proceed as follows: if the first hash root $\widetilde{y_1}$ is correct, then the prover has committed to a string indicating that the circuit rejects the input! This commitment is binding, and in particular the verifier will reject when it checks the opening of the commitment in Step (4b). Otherwise, if $\widetilde{y_1}$ is incorrect, then there are two possibilities: either there is some layer $i^* \in [1, \ldots, D-2]$ where $\widetilde{y_{i^*}}$ is incorrect, but $\widetilde{y_{i^*+1}}$ is correct. The soundness of the HIP implies that the $i^{*\text{th}}$ execution will yield a false claim, i.e. $\widetilde{V_{i^*+1}}[r_{i^*+1}] \neq v_{i^*+1}$. But since the prover sent the correct hash root for the $(i^*+1)^{\text{th}}$ layer, in Step (4b), it cannot open the hash root to any value except the correct value (which is different from v_{i^*+1}), and the verifier will reject. The remaining possibility is that the prover was cheating on $\widetilde{y_{D-1}}$: in this case, w.h.p. the HIP for layer $(D-1)$ outputs a false claim about the LDE of the input x, and the verifier will reject in Step (4c).

Complexity analysis. The communication complexity is D times the communication complexity in the HIP, plus $(D \cdot \text{poly}(\log(n), \kappa))$ for sending the hash roots and openings. Similarly, the verifier runtime is D times the runtime in the HIP, plus $(D \cdot \text{poly}(\log(n), \kappa))$ for the hash roots, and openings and another almost-linear term for the final LDE check in Step (4c) (this final term can be omitted if we give the verifier query access to the LDE of the input x). Finally, the round complexity is dominated by the round complexity of the HIP executions of Step (3), but the key point is that these are *performed in parallel*, and the proof in each execution is for a computation whose depth is independent of the depth of the circuit C.

The HIP. The i^{th} HIP is performed to check the following computational claim: let C_i^{layer} be the circuit that computes the i^{th} layer of the circuit C. Let LDE be the circuit that takes an input $V \in \{0,1\}^S$ and computes its low-degree extension $\widehat{V} \in \{0,1\}^M$. Finally, let root be the circuit that takes an M-bit string $\widehat{V}$ and outputs the root of the hash tree computed on $\widehat{V}$ (with the hash functions $\bar{h}$ sent by the verifier). The i^{th} execution uses the HIP to verify a claim about the value of the function $(\text{root} \circ \text{LDE} \circ C_i^{\text{layer}})$, where we think of the input to this function as the values of the $i+1^{\text{st}}$ layer of the circuit. Any HIP that can perform this computation efficiently may be used here. For example, we can take the holographic variants of the GKR or RRR protocols (see Table 1, and note that the verifier time is reduced in the holographic case). In particular, since this function can be computed in polynomial time and $\text{poly}(\kappa)$ space, RRR can yield $\exp(\tilde{O}(1/\sigma))$ rounds, $(n^{\sigma+o(1)} \cdot \text{poly}(\kappa))$ communication and verification time, and polynomial prover time. Alternatively, since this function is also computable by log-space uniform circuits of depth $\text{poly}(\kappa)$, GKR can yield $\text{poly}(\log n, \kappa)$ rounds, communication, and verification time. Under stronger cryptographic assumptions, one could also use the GR protocols, see Remark 2.1.

Digest. Our protocol uses the UOWHF to force the prover to send a commitment for each layer of the circuit, where a cheating prover has two (bad) choices, either (a) send a correct hash root for that layer's gate values. In this case, the commitment is binding, and the prover's hands are forevermore tied when it makes claims about this layer's LDE, or (b) the prover sends an *incorrect* hash root, where at the very least the prover needs to send an incorrect hash root for the output layer (otherwise it will be caught immediately). Since there must be *some* layer where the prover is cheating on the root of layer i but we can access the correct LDE of layer $(i+1)$ (either because the prover sent a correct hash root, which is binding, or because layer $(i+1)$ is the input layer), verification can be reduced to checking consistency between a hash root and the layer below it. I.e., we have reduced verifying the deep / complex computation of C, to verifying (in parallel) many simpler computations. Each of these simpler computations evaluates one circuit layer, and composes it with a computation of the low-degree extension and the hash tree. Thus, our goal is constructing efficient HIPs for these simpler computations (moreover, as we will see below, these simpler computations have nice structure that facilitates the construction of very efficient proof systems).

2.3 Holographic Hash Root (HHR) Protocol

In a *holographic hash root (*HHR*)* protocol, the prover and the verifier are given a claim of the form $(\bar{h}, y)$ and a holographic input w. After interacting with the prover, the verifier (who never accesses w) either rejects or outputs a claim (r, v) about the LDE $\widehat{w}$ of w. If y is the correct hash-root of w w.r.t. the hash functions $\bar{h}$, then $\widehat{w}[r] = v$. If y is *not* the correct hash root, then w.h.p. either the verifier rejects, or $\widehat{w}[r] \neq v$. The HHR protocols in this work have information-theoretic soundness (though computational soundness would suffice for the template protocol).

On a conceptual level, our work identifies HHR protocols as a very useful component for constructing argument-systems with small round-complexity. Once we have a HHR protocol, we can compose it sequentially with a HIP for verifying a claim about the computation that takes as input a vector V of values for the gates at layer $(i+1)$, computes the values V' that V induces for the gates in layer i, and checks a single claim about the LDE of V'. We can construct a constant-round HIP for the latter task (evaluating a single circuit layer and then computing a low-degree extension) using the GR protocol (see Table 1). Thus, *the round complexity of the template protocol is dominated by the round complexity that can be achieved for* HHR*s.*

A better HHR *protocol.* Our main technical contribution is an HHR protocol whose round complexity is only $(1/\sigma^3)$. Our main result (Theorem 1.1) follows by plugging the HHR protocol into the template protocol of Section 2.2. The HHR protocol closely follows the construction of a hash tree. Fixing a small constant $\delta > 0$ (set below), we use a family of UOWHF functions $\{h : \{0,1\}^{n^{2\delta}} \to \{0,1\}^{n^{\delta}}\}_{h\in\mathcal{H}}$ (the security parameter κ is set to be a sufficiently small power of n). We use these hash functions in an $d = n^{\delta}$-ary hash tree (see Section 2.1). The tree has $\ell = O(1/\delta)$ layers, where layer j in the tree is n^{δ}-times smaller than layer $j+1$. Given the root of the tree, local opening requires sending $O(n^{2\delta})$ bits to the receiver/verifier (for each node on the path from the root to the leaf, the values of all $(d-1)$ of its siblings need to be sent).

The HHR protocol sequentially "strips away" the layers of the hash tree, beginning with a claim about the hash root and ending with a claim about the leaves of the tree. This is achieved by means of a *tree-layer sub-protocol*: a $O(1/\delta^2)$-round protocol that begins with an input claim about the LDE of the tree nodes in layer j, and ends with an output claim about the LDE of the tree nodes in layer $(j+1)$. If the input claim is correct and the prover follows the protocol, then the output claim will also be correct. If, however, the input claim is incorrect, then (no matter what strategy a cheating prover utilizes) the output claim will also be incorrect. This structure is inspired by, and similar to, the GKR protocol, but we emphasize that the computation for moving from one layer to another is *not* of constant depth, since it involves applying the hash function, which is an arbitrary $\mathrm{poly}(\kappa) = n^{O(\delta)}$-time computation, whereas we want a $O(1/\delta^2)$ round protocol.

Several remarks are in order. We emphasize that we run the tree-layer sub-protocols *sequentially*, starting from the output layer (layer 1), and ending with the bottom of the tree (layer $(\ell-1)$). There are $O(1/\delta)$ tree layers, so the total round complexity is $O(1/\delta^3)$. Theorem 1.1 is derived by taking δ to be a small enough constant multiple of the desired σ, so the $n^{O(\delta)}$ term in the verification time and communication complexity ends up being n^{σ}. Finally, the alert reader will have noticed that we need to begin the HHR with a claim about the LDE of the hash root, and we will end it with a claim about the LDE of the values in the leaves. For the first point: the verifier, who knows the claimed hash root y, can choose a random location r and take $v = \mathsf{LDE}(y)[r]$ to be the input claim to the first sub-protocol. If y is not the correct hash root, then w.h.p. over the choice of r the first input claim will be false (since the low-degree extension is a high-distance error-correcting code). Second, by definition of the HHR, the values in layer ℓ are already a low-degree extension of the string w. In the final sub-protocol, we will directly get a claim about $\mathsf{LDE}(w)$ (rather than a claim about $\mathsf{LDE}(\mathsf{LDE}(w))$.

Remark 2.1 (Stronger assumptions). *If the UOWHF were computable in highly uniform* $\mathsf{AC}^0[\oplus]$*, we could instead simply use the constant-round GR protocol to move from one layer to another (see Table 1). In fact, a UOWHF computable in highly uniform* NC^1 *should suffice, since the GR protocol can also work on highly uniform* NC^1 *circuits. Indeed, it is possible to prove that the circuit that computes the UOWHF tree can satisfy this stronger uniformity condition (see Footnotes a and b for the exact uniformity conditions), and it would be an* NC^1 *circuit thanks to the polynomial shrinkage of the UOWHF, that promises that the tree has a constant number of layers.*

UOWHFs in such low classes have been conjectured to exist (see e.g. [1], but note that we need super-linear shrinkage in our construction, because we want the tree to be of constant depth). Regardless, in this work, we do not want to assume anything beyond the existence of one-way functions.

The Tree-Layer Sub-Protocol. We briefly sketch some of the ideas in this final sub-protocol. Let $n_{in} = n^{2\delta}$ and $n_{out} = n^{\delta}$ be the input and output lengths of the hash function. We take $w_i \in \{0,1\}^{n^{i\cdot\delta}}$ to be the vector of the values of nodes in layer i of the hash tree, and let $k = k(i) = |w_{i+1}|/n_{in}$ be the number of nodes in layer i. As described above, given a claim (i, r_i, v_i) about w_i, and given also a holographic input w_{i+1}, the goal of our sub-protocol is for the verifier (to reject or) to output a claim (r_{i+1}, v_{i+1}) about the holographic input. The crux of the matter is doing this using only $O(1/\delta^2)$ rounds, which we accomplish by utilizing the particular structure of the hash tree's computation: the tree operates independently and in parallel on blocks of w_{i+1}. Dividing the layers into blocks we have:

$$w_i = y_1, ..., y_k \text{ for } |y_j| = n_{out},$$

where y_j is the value of the j^{th} node in layer i, and

$$w_{i+1} = z_1, ..., z_k \text{ for } |z_j| = n_{in},$$

where each z_j is the concatenation of the values of the j^{th} node's children. Taking h_j to be the hash function for layer j, we can now restate the claim about layer j:

$$\left(\bigwedge_{j=1}^{k} y_j = h_j(z_j)\right) \bigwedge \left(\mathsf{LDE}(y_1, \ldots, y_k)[r_i] = v_i\right). \tag{1}$$

Thus, there are k "mini-claims", each about a single evaluation of the hash function, tied together by a " global claim" about the low-degree extension of (the concatenation of) the resulting outputs. We use a *batch-verification* protocol to verify the k mini-claims, together with the global claim about the LDE, at a cost that is not much larger than verifying a single claim (each single claim is about single a $\mathrm{poly}(\kappa)$-size computation, so the verifier can verify it on its own). Our protocol is inspired by the UP batching protocol of Reingold, Rothblum and Rothblum [32]. The idea is to proceed in sequential iterations, where in each iteration we run a "reducing" sub-sub-protocol to restrict the claims being made to a smaller subset $S' \subseteq [k]$ of the k initial mini-claims, tied together with a "global claim" about the computations in the set S'. The size of the set is reduced by a factor of roughly n^{δ} in each iteration, so after $O(1/\delta)$ iterations, the final set has only a few surviving mini-claims. The prover can send to the verifier the values of the surviving tree nodes, and the verify can verify the remaining claims by brute force in $n^{O(\delta)}$ time and communication (there is a technical issue here: this is a holographic protocol, so we need to reduce these final mini-claims to claims about the LDE of the $(i+1)^{\text{st}}$ tree layer).

As in [32], the "reducing" sub-sub-protocol is performed using an interactive proof of proximity (IPP) [34], where a claim about a large implicit input X (the sequence of y_i's and z_i's) is reduced to a claim about a subset of X's bits. We elaborate briefly on how this is done in our context. We use an IPP where the verifier (on

top of having implicit input) has *holographic* input, and at the end of the interaction, the verifier outputs a claim about its encoding. We view the k hash outputs (in layer i) as the implicit input, and the k hash inputs (in layer $(i+1)$) as the holographic input. The IPP lets us reduce a claim about a set S of input-output pairs to: (i) a claim of the same form about a smaller subset of the pairs, and (ii) a holographic claim about the encoding of the inputs. We "set aside" the holographic claims generated by the IPPs, and at the end of the protocol we reduce all of them to a single claim about the LDE of the $(i+1)^{\text{st}}$ layer. The reducing sub-sub-protocol has $O(1/\delta)$ rounds, $n^{O(\delta)}$ communication and verification time, and a polynomial prover. Rolling these complexities back to the HHR protocol and the template protocol gives the result claimed in Theorem 1.1.

We remark that there are significant technical hurdles that need to be overcome in the full construction. The main reason is that we want the reducing sub-sub-protocol to run in only $O(1/\delta)$ rounds. Thus, we can only afford to use (an extension of) the GR protocol for highly-uniform $\mathsf{AC}^0[\oplus]$ circuits in the IPP.[4] Thus, we need to carefully argue that all the computations being verified can be performed via highly uniform low-complexity circuits. For example, we need to augment the implicit y_i inputs in the IPP with the entire tableau of the hash function's computation, so that verification can be in $\mathsf{AC}^0[\oplus]$. We also need to carefully argue about the structure of the "global claims" tying together the mini-claims in each iteration, to ensure they can be verified by a highly-uniform low-depth circuit.

This concludes our high-level sketch of the sub-sub-protocol's structure, and we direct the reader to the full version for a more detailed overview and the full details.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702) and from the Simons Collaboration on The Theory of Algorithmic Fairness. Part of this work was done while G.N.R. was at the Weizmann Institute of Science.

REFERENCES

[1] Benny Applebaum and Yoni Moses. 2013. Locally Computable UOWHF with Linear Shrinkage. *IACR Cryptol. ePrint Arch.* (2013), 423. http://eprint.iacr.org/2013/423

[2] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. 1991. Checking Computations in Polylogarithmic Time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*. 21–31. https://doi.org/10.1145/103418.103428

[3] Boaz Barak and Oded Goldreich. 2008. Universal Arguments and their Applications. *SIAM J. Comput.* 38, 5 (2008), 1661–1694. https://doi.org/10.1137/070709244

[4] Mihir Bellare and Phillip Rogaway. 1997. Collision-Resistant Hashing: Towards Making UOWHFs Practical. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings (Lecture Notes in Computer Science, Vol. 1294)*, Burton S. Kaliski Jr. (Ed.). Springer, 470–484. https://doi.org/10.1007/BFb0052256

[5] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. 1988. Everything Provable is Provable in Zero-Knowledge. In *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*. 37–56. https://doi.org/10.1007/0-387-34799-2_4

[4] In fact, $\mathsf{AC}^0[\oplus]$ isn't a sufficiently rich for our purposes, so we extend the protocol to apply to constant-depth *arithmetic* circuits over large fields with bounded fan-in multiplication gates .

[6] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. 2018. Multi-collision resistance: a paradigm for keyless hash functions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, Ilias Diakonikolas, David Kempe, and Monika Henzinger (Eds.). ACM, 671–684. https://doi.org/10.1145/3188745.3188870

[7] Gilles Brassard, David Chaum, and Claude Crépeau. 1988. Minimum Disclosure Proofs of Knowledge. *J. Comput. Syst. Sci.* 37, 2 (1988), 156–189. https://doi.org/10.1016/0022-0000(88)90005-0

[8] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. 2021. SNARGs for P from LWE. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 68–79. https://doi.org/10.1109/FOCS52979.2021.00016

[9] Ivan Damgård. 1989. A Design Principle for Hash Functions. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings (Lecture Notes in Computer Science, Vol. 435)*, Gilles Brassard (Ed.). Springer, 416–427. https://doi.org/10.1007/0-387-34805-0_39

[10] Oded Goldreich and Johan Håstad. 1998. On the Complexity of Interactive Proofs with Bounded Communication. *Inf. Process. Lett.* 67, 4 (1998), 205–214. https://doi.org/10.1016/S0020-0190(98)00116-1

[11] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1991. Proofs that Yield Nothing But Their Validity for All Languages in NP Have Zero-Knowledge Proof Systems. *J. ACM* 38, 3 (1991), 691–729. https://doi.org/10.1145/116825.116852

[12] Oded Goldreich and Guy N. Rothblum. 2020. Constant-Round Interactive Proof Systems for AC0[2] and NC1. In *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, Oded Goldreich (Ed.). Lecture Notes in Computer Science, Vol. 12050. Springer, 326–351. https://doi.org/10.1007/978-3-030-43662-9_18

[13] Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. 2002. On interactive proofs with a laconic prover. *Comput. Complex.* 11, 1-2 (2002), 1–53. https://doi.org/10.1007/s00037-002-0169-0

[14] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. 2015. Delegating Computation: Interactive Proofs for Muggles. *J. ACM* 62, 4 (2015), 27. https://doi.org/10.1145/2699436

[15] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1989. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.* 18, 1 (1989), 186–208. https://doi.org/10.1137/0218012

[16] Tom Gur and Ron D. Rothblum. 2017. A Hierarchy Theorem for Interactive Proofs of Proximity. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*. 39:1–39:43. https://doi.org/10.4230/LIPIcs.ITCS.2017.39

[17] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. 2015. Finding Collisions in Interactive Protocols - Tight Lower Bounds on the Round and Communication Complexities of Statistically Hiding Commitments. *SIAM J. Comput.* 44, 1 (2015), 193–242. https://doi.org/10.1137/130938438

[18] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. 2009. Statistically Hiding Commitments and Statistical Zero-Knowledge Arguments from Any One-Way Function. *SIAM J. Comput.* 39, 3 (2009), 1153–1218. https://doi.org/10.1137/080725404

[19] Justin Holmgren and Alex Lombardi. 2018. Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications). In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 850–858. https://doi.org/10.1109/FOCS.2018.00085

[20] Russell Impagliazzo and Michael Luby. 1989. One-way Functions are Essential for Complexity Based Cryptography (Extended Abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*. IEEE Computer Society, 230–235. https://doi.org/10.1109/SFCS.1989.63483

[21] Russell Impagliazzo and Moti Yung. 1987. Direct Minimum-Knowledge Computations. In *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings (Lecture Notes in Computer Science, Vol. 293)*, Carl Pomerance (Ed.). Springer, 40–51. https://doi.org/10.1007/3-540-48184-2_4

[22] Yuval Ishai. 2020. Zero-Knowledge Proofs from Information-Theoretic Proof Systems - Part I. Available at https://zkproof.org/2020/08/12/information-theoretic-proof-systems/.

[23] Yuval Ishai. 2020. Zero-Knowledge Proofs from Information-Theoretic Proof Systems - Part II. Available at https://zkproof.org/2020/10/15/information-theoretic-proof-systems-part-ii/.

[24] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. 2022. How to Delegate Computations: The Power of No-Signaling Proofs. *J. ACM* 69, 1 (2022), 1:1–1:82. https://doi.org/10.1145/3456867

[25] Joe Kilian. 1992. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In *STOC*. 723–732.

[26] Ilan Komargodski, Moni Naor, and Eylon Yogev. 2018. Collision Resistant Hashing for Paranoids: Dealing with Multiple Collisions. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications*

of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II (Lecture Notes in Computer Science, Vol. 10821), Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, 162–194. https://doi.org/10.1007/978-3-319-78375-8_6

[27] Ralph C. Merkle. 1989. One Way Hash Functions and DES. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings (Lecture Notes in Computer Science, Vol. 435)*, Gilles Brassard (Ed.). Springer, 428–446. https://doi.org/10.1007/0-387-34805-0_40

[28] Silvio Micali. 1994. CS Proofs (Extended Abstracts). In *FOCS*. 436–453.

[29] Moni Naor. 1991. Bit Commitment Using Pseudorandomness. *J. Cryptology* 4, 2 (1991), 151–158. https://doi.org/10.1007/BF00196774

[30] Moni Naor and Moti Yung. 1989. Universal One-Way Hash Functions and their Cryptographic Applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, David S. Johnson (Ed.). ACM, 33–43. https://doi.org/10.1145/73007.73011

[31] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. 2016. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. 49–62. https://doi.org/10.1145/2897518.2897652

[32] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. 2018. Efficient Batch Verification for UP. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*. 22:1–22:23. https://doi.org/10.4230/LIPIcs.CCC.2018.22

[33] John Rompel. 1990. One-Way Functions are Necessary and Sufficient for Secure Signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, Harriet Ortiz (Ed.). ACM, 387–394. https://doi.org/10.1145/100216.100269

[34] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. 2013. Interactive proofs of proximity: delegating computation in sublinear time. In *STOC*. 793–802.

[35] Ron D. Rothblum and Prashant Nalini Vasudevan. 2022. Collision-Resistance from Multi-Collision-Resistance. In *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 13509)*, Yevgeniy Dodis and Thomas Shrimpton (Eds.). Springer, 503–529. https://doi.org/10.1007/978-3-031-15982-4_17

[36] Daniel R. Simon. 1998. Finding Collisions on a One-Way Street: Can Secure Hash Functions Be Based on General Assumptions?. In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding (Lecture Notes in Computer Science, Vol. 1403)*, Kaisa Nyberg (Ed.). Springer, 334–345. https://doi.org/10.1007/BFb0054137

[37] Justin Thaler. 2022. Proofs, Arguments, and Zero-Knowledge. Available at https://people.cs.georgetown.edu/jthaler/ProofsArgsAndZK.html.

[38] Hoeteck Wee. 2005. On Round-Efficient Argument Systems. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings (Lecture Notes in Computer Science, Vol. 3580)*, Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung (Eds.). Springer, 140–152. https://doi.org/10.1007/11523468_12

Received 2022-11-07; accepted 2023-02-06