Fiat-Shamir Goes Rational

Matteo Campanelli[1] and Agni Datta[2]

[1] No affiliation
[2] SECURE–CoE, VIT Bhopal University, India

Abstract. This paper investigates the open problem of how to construct non-interactive rational proofs. Rational proofs, introduced by Azar and Micali (STOC 2012), are a model of interactive proofs where a computationally powerful server can be rewarded by a weaker client for running an expensive computation $f(x)$. The honest strategy is enforced by design when the server is rational: any adversary claiming a false output $y \neq f(x)$ will lose money on expectation.
Rational proof constructions have appealing properties: they are simple, feature an extremely efficient verifier—reading only a sublinear number of bits of the input x—and do not require any collateral from the prover. Currently, all non-trivial constructions of rational proofs are interactive. Developing non-interactive rational protocols would be a game-changer, making them practical for use in smart contracts, one of their most natural applications.
Our investigation revolves around the Fiat-Shamir transform, a common approach to compiling interactive proofs into their non-interactive counterparts. We are the first to tackle the question:

Can Fiat-Shamir be successfully applied to rational protocols?

We find negative evidence by showing that, after applying Fiat-Shamir in the random oracle model to two representative protocols in literature (AM13 and CG15) these lose their security guarantees. Our findings point to more general impossibility theorems, which we leave as future work.
To achieve our results we first need to address a fundamental technical challenge: the standard Fiat-Shamir transform does not apply to protocols where the verifier has only oracle access to its input x (a core feature of the rational setting). We propose two versions of Fiat-Shamir for this setting, a "vanilla" variant and a "stronger" variant (where the verifier has access to an honestly computed digest of its input). We show that neither variant is sufficient to ensure that AM13 or CG15 are secure in the non-interactive setting.
Finally, as an additional contribution, we provide a novel, and arguably simpler, definition for the soundness property of rational proofs (interactive or non-interactive) of independent interest.

1 Introduction

Consider an untrusted server claiming that a certain computation f executed on the client's data x produces the output y. For instance, f may represent a machine learning training algorithm, x a training dataset, and y the resulting model. We are interested in approaches to checking the server's response without the need to recompute $f(x)$ from scratch (which could be unfeasible due to limited resources or too costly). A technique widely employed to address this issue is that of cryptographic proofs [28,31]:

the server (the *prover*) can transmit a certificate π that the client (the *verifier*) can check efficiently; if the server claims a false output, this discrepancy will be detected with overwhelming probability via π. Despite the enormous progress in their design (see, e.g., [31]) cryptographic proofs still have a series of limitations. They demand significant resources (a typical overhead for proving a computation f is three or more order of magnitudes on top of f's execution) and may require strong or unscrutinized security assumptions [30,18].

RATIONAL PROOFS. Our focus in this work is an alternative approach to certifying computation: the model of *rational proofs* introduced by Pablo Azar and Silvio Micali [2]. In contrast to purely cryptographic counterparts, they work under the assumption that the prover is *rational* and *economically incentivized*. In order to ascertain the result of a computation $f(x)$, the prover and verifier engage in an interaction, after which the verifier will determine the reward for the prover based on the interaction transcript. The protocol is designed so that a cheating prover, in expectation, will receive a lower reward (in this model, a "verifier" is effectively just a "rewarder"; in this paper we adopt this common yet slightly imprecise terminology).

In scenarios where we can assume the server to be economically incentivized, rational proofs present two primary advantages: a *simple and efficient* prover and, a *sublinear* verifier. A rational proof construction—in contrast to a cryptographic one—may incur almost no overhead beyond the computation of f itself (proving may often involves providing specific segments of the computation without further processing). Rational protocols are also easier to implement and understand. One of the main advantages of rational proofs stems from one their most peculiar features: the verifier *will not have to access the entirety of its input* x (hence the term "sublinear"). In particular, they assume that the verifier has random access (or "oracle" access) to x, of which it may only need to query a few bits[3][4].

The features above make rational proofs especially appealing in practice. There is however a major challenge to their applicability—interaction—a challenge we gradually introduce in the rest of this section and which is the focus of our work.

THE SYNERGY RATIONAL PROOFS $\leftrightarrow$ SMART CONTRACTS. When, in 2012, Azar and Micali first proposed the model of rational proofs, they relied on the assumption that the verifier would simply follow the protocol and reward the prover. This is not an assumption one can apply universally. A dishonest (or rational) verifier may bias the coins chosen throughout the protocol or simply walk away. In their work, Azar and Micali left open how to *enforce* this behavior instead of simply assuming it[5]. The advent of *smart contracts* provided a solution: these are programs collectively run on a "decentralized computer",

[3] An intuition for why this may be a reasonable model: consider a client that, after collecting data x, uploads it to some external memory which is assumed to be trusted in a restricted: it will later return the original bits of x. We can think of the verifier querying this memory through random access during the interaction.

[4] A couple clarifications: our of focus is on *non-trivial* rational proofs, i.e., where the verifier is sublinear (this excludes standard interactive proofs which do otherwise satisfy the definition). Also, throughout the introduction—and only there—we will, for simplicity, talk about *proofs* meaning more inclusively both *arguments* (secure against bounded adversaries only) and *proofs* (secure against any adversary).

[5] We are aware of only one work studying security for when the verifier is also *rational*, namely [25]. However, it applies only to specific threshold circuits.

with native abstractions for transferring assets and for enforcing agreements (hence the term "contracts").

Smart contracts not only offer solutions for rational proofs, but the reverse is also true. Various cryptographic proofs are already being verified within smart contracts, but the efficiency advantages of rational proofs make them particularly well-suited for this environment. As a consequence we could run better incentivized outsourced computation on smart contracts *through* rationals proofs. For instance, imagine a contract that releases funds to a user if they can provide value y such that $y = f(x)$. Instead of directly checking $y = f(x)$, the prover could submit a rational proof, and the contract would then automatically execute the reward[6]. For a broader discussion on this topic, see [9].

The challenge of non-interaction. A major obstacle to applying rational arguments in smart contracts is that all known constructions in the literature require multiple rounds of interaction, at least for non-trivial computations. Non-interactivity is typically a critical requirement for systems deployed on distributed ledgers. Beyond that, non-interactive protocols are generally preferable due to their efficiency: fewer rounds mean reduced latency, and they impose fewer demands on the prover, who can submit a proof and disengage, rather than staying online for the entire interaction.

This work: is Fiat-Shamir secure in rational arguments? We turn our attention to the de facto standard tool for transforming interactive protocols into non-interactive ones, the celebrated *Fiat-Shamir transform* [16], which replaces the interaction with the verifier (who is only supposed to send uniformly random challenges) by the invocation of a public hash function. This means that the challenge at round i is computed as $e_i = \mathcal{H}(\mathsf{pp}, \mathsf{x}, \tau_{i-1})$ where pp are some parameters, x is the statement whose truth we are ascertaining and τ_{i-1} is the transcript so far. If the hash function $\mathcal{H}$ is assumed to be random—a *random oracle*—one can hope that the prover has no easier time producing "successful" proofs for false statements than when interacting with an actual verifier. While the Fiat-Shamir heuristic does not always guarantee soundness [19]—the standard security property for cryptographic proofs—we do know of sufficient conditions for it to do so (which are also satisfied by widely used protocols) [1].

Unfortunately, results on Fiat-Shamir in cryptographic proofs do not apply directly to our setting. This is due to two reasons (which we already hinted at):

a) The soundness definition for rational protocols is very different from the more traditional one for interactive proofs.

b) In a rational proof the verifier does not read the whole input x. Instead, it queries relatively few bits from it at the end of the interaction. Yet Fiat-Shamir requires us to feed the public input at each step. This means that the standard Fiat-Shamir transform does not apply to our setting *even from a purely syntactical point of view!*

Item *a)* suggests we need a completely different security treatment, but item *b)* requires that, first, we also need to reformalize Fiat-Shamir (FS) for our setting, where the verifier has this special access to its public input (to which we hereby refer to as *holographic* for succinctness).

[6] This application scenario is being concretely considered by various organization. As an example, consider the case where x is data for classification, f is a machine learning model and y is the resulting output. This intuitive scenario is being explored, among others, by the gensyn project [17].

Our results in a nutshell (or, FS is insecure for rational arguments) We provide a negative answer to the question: *Is Fiat-Shamir in the random oracle model (ROM) generally secure for rational protocols?* Below, we outline our findings.

First, we formalize a natural variant of Fiat-Shamir for verifiers with holographic inputs. This variant, which we term *simple* Fiat-Shamir, computes challenges using the entire visible transcript and the parameters accessible to the verifier up to that point. Specifically, if the prover claims $f(x) = y$, then the challenge in each round i is computed as $e_i = \mathcal{H}(\mathsf{pp}, f, y, \tau_{i-1})$ (if the verifier accesses any input bits, these are incorporated into $\mathcal{H}$ for any subsequent randomized stages of the protocol). We then show:

> **Result 1**. *There exist (non-contrived) secure, interactive rational arguments that become insecure when compiled with the "simple" Fiat-Shamir in the ROM.*

The "non-contrived" constructions above are such because they represent established constructions in the literature for non-trivial subsets of P: the rational protocol for threshold circuits of $\mathcal{O}(1)$ depth by Azar and Micali [3] and the protocol for NC^1 circuits (boolean circuits with logarithmic depth) by Campanelli and Gennaro [10].

We observe that the attacks breaking "simple" Fiat-Shamir have a recognizable patter which stems from lack of dependence of the challenges from the whole input. We then design a different ("stronger") Fiat-Shamir for holographic inputs where the same attacks are not possible.

We find, however, that the same constructions mentioned above can be compromised (through a different approach) even in this stronger scenario:

> **Result 2**. *There exist (non-contrived) secure, interactive rational arguments that become insecure when compiled with the "strong" Fiat-Shamir in the ROM.*

1.1 Technical Overview

Background on Rational Proofs. In rational proofs, the prover and verifier engage in an interactive protocol, at the end of which the prover receives a monetary reward for the prover. The verifier is *public-coin*, that is her messages consist of uniformly random bit strings (and this is what makes the Fiat-Shamir transform applicable, at least in principle). The final step of the protocol involves the verifier invoking the function $\mathsf{reward}^x(\mathsf{pp}, f, y)$, which returns a non-negative amount $\tilde{R}$—the superscript notation for x indicates that this function possesses random—*holographic*—access to the public input x. The property of "rational soundness" mandates that a cheating prover, claiming $\tilde{y} \neq f(x)$, receives a reward $\tilde{R}$ such that $\mathbb{E}[R_{\text{hon}}] - \mathbb{E}\left[\tilde{R}\right]$ is deemed "substantial", where R_{hon} represents the reward of the honest prover. A formal treatment of rational soundness typically requires several additional notions, including that of the "reward gap" as articulated in [20]. We propose what we believe to be a simpler model based on a security game, which we demonstrate to be equivalent to prior ones (Appendix B).

A simple Fiat-Shamir for holographic inputs. Our first attempt in defining Fiat-Shamir for the rational setting is the natural one: we apply its underlying principle by properly adapting it ("hash all the public inputs and messages up to the current point of the interaction"). See Fig. 3. The astute reader will notice that this, however, cannot

include the full input x from the beginning—x is queried only at the end and partly. We now discuss how this is problematic.

True-to-false statement attacks. We identify a class of potential attacks stemming from our previous observation. Consider an input x, a function f, and the honest (Fiat-Shamir) proof π_{hon} used to assert the output $y_{\text{hon}} = f(x)$. By the definition of rational proofs, π_{hon} assures the "maximum expected reward." Yet, we can show that this same proof may yield the same reward even for a fraudulent output. Here is how: let $\tilde{x}$ be an input that is identical to x except in certain positions specified by the set $\mathcal{I}$, under the condition that $f(\tilde{x}) \neq f(x)$. If the positions in $\mathcal{I}$ are not among those observed by the verifier, one can receive the *honest* reward for the *false* claim $y = f(\tilde{x})$ (example: let x with parity 1 and let x' be as x but with one non-queried bit that has been flipped). The reason why this "true-to-false statement" attack is potentially feasible is that the input positions queried by the verifier are determined by the random oracle (RO) invocations and rely solely on public parameters y and f, which remain consistent in both scenarios[7].

The Π_{AM13} and Π_{CG15} Constructions. From the observations above, we are aware of which potential attacks may occur, but it remains unclear whether these attacks manifest in concrete protocols. We show, however, that such attacks are actually possible. Our case studies focus on Π_{AM13} from [3], which applies to constant-depth threshold circuits, and Π_{CG15} from [10], which targets logarithmic-depth Boolean circuits. Both protocols proceed in a layer-by-layer fashion, but differ in their reward mechanisms: Π_{AM13} utilizes scoring rules[8] (specifically, Brier's rule [8]), whereas Π_{CG15} employs a specially designed spot-checking method. We show that both protocols are vulnerable to true-to-false statement attacks and are therefore insecure when compiled using the strengthened FS transform discussed above.

Strengthening FS by Incorporating the Full Statement. The attacks we just described rely on a key observation: two *distinct* statements, x and x', can (intuitively) have the prover face the *same* challenges during the proof, and therefore the same set of queried positions. Naturally, this raises the question of whether strengthening the Fiat-Shamir transform could offer a solution. To address this, we consider an augmented model where the verifier retains its usual "spot" access to x, but also takes as input information depending on x in its entirety—a digest[9] δ_x. We strengthen the Fiat-Shamir transform by including δ_x in each challenge. As a result, two distinct inputs will now yield distinct digests, effectively preventing the aforementioned attacks.

[7] Variants of such attacks are recognised in traditional cryptographic proof settings (see, e.g., [14]). While astute readers may find such attacks plausible, the implications for verification and rewarding mechanisms are not immediately apparent. The soundness notions in traditional versus rational interactive proofs diverge significantly; the former permits the verifier to access the *entire* input. Accordingly, previous analyses of the Fiat-Shamir approach do not necessarily illuminate our context.

[8] A (proper) scoring rule rewards a forecast provided by an expert and has the key property that any forecast deviating from the true distribution of events (modelled as a probability distribution) will cause the expert to incur a loss. Azar and Micali show how these techniques can be applied to gates in a circuit [2,3].

[9] Intuitively, this can be understood as a compact hash value that the verifier computes once before uploading its input to some memory. See also Footnote 3.

Further Attacks. The goal of this "stronger" FS approach is to provide a litmus test: they make things harder for the adversary assuming *more* from the model (a digest δ_x); if we can show security of rational schemes in this stronger model, then there may be hope to later develop an intermediate Fiat-Shamir approach based on slightly less stringent assumptions. Unfortunately, this is not the case. We can show that both Π_{AM13} and Π_{CG15} are vulnerable to other forms of attack beyond the true-to-false statement manipulations. These new attacks allow an adversary to fix an attempted target input $\tilde{x}$ and then identify a pair (f, y) such that: *i)* $f(\tilde{x}) \neq y$, and *ii)* the reward obtained from proving this false claim is nearly as high as what an honest prover would earn.

1.2 Discussion and Future Work

Our findings have two key implications. First, they demonstrate that adapting the Fiat-Shamir transform to the context of holographic verifiers is non-trivial—this is evident from our "simple" FS construction and the associated attacks. Second, they suggest that such an adaptation may not be feasible at all, since even imposing stronger requirements on the Fiat-Shamir transform, different attacks on the protocols we analyzed still emerged. These attacks are not limited to the specific cases of Π_{AM13} and Π_{CG15}, and likely extend to other similar constructions. Formalizing these generalizations is left as future work.

It is important to emphasize that rational proofs are not the only class of interactive proofs involving *holographic verifiers*. Indeed, proofs of proximity [23,22,6,29,32] also allow for sublinear verification, and the feasibility of applying Fiat-Shamir in that context remains an open question. Our work provides initial tools that may help explore this area further.

1.3 Related Work

Azar and Micali [2] introduced rational proofs and showed their power for large complexity classes constructing a single-round scheme for all of #P (with a non-polynomial prover); they propose efficient protocols for restricted classes in [3]. The work of Campanelli and Gennaro [10,11] constructs rational proofs with composition properties (e.g., that are reusable for multiple executions) for bounded-depth circuits and bounded-space computations. The work of Guo, Hubacek, Rosen, and Vald [20] restrict the rational prover to be computationally bounded, obtaining the notion of rational arguments and proposing constructions with a sublinear verifier rational arguments for the class NC^1. This work was extended to the class P in [21]. Recently, Campanelli, Ganesh, and Gennaro have proposed extractability properties for rational arguments [9]. Another line of work achieves verifiable computation against rational parties in an indirect manner through approaches based on fine-grained cryptographic primitives [12].

A long line of work has studied Fiat-Shamir in traditional interactive arguments. A partial list includes [16,15,4,19,7,26,24,27].

Paper outline We introduce our model of rational arguments in Section 2 (both interactive and non-interactive in the ROM). We describe our Fiat-Shamir variants in Section 3; the same section discusses generic attacks. Finally, Section 4 provides background on our case study schemes and formal results on their insecurity.

2 Modeling Rational Arguments

Interactive Rational Arguments Here we describe our model interactive arguments. Notice that a rational *proof* (where the adversary may be unbounded) also satisfies this definition with an empty setup; we therefore provide only a definition of arguments. In the appendix, we discuss the difference between our model and earlier ones showing they are equivalent.

Syntactically, a rational argument consists of a tuple of (PPT, possibly interactive) algorithms $(\mathsf{Setup}, \mathcal{P}, \mathcal{V}, \mathsf{reward})$ that work as follows:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: outputs parameters pp on input security parameter λ;
- $\langle \mathcal{P}, \mathcal{V} \rangle \to \tau$: $(\mathcal{P}, \mathcal{V})$ are an interactive protocol producing transcript τ. Both parties take parameters pp and $\mathcal{V}$ only returns random challenges (it is public-coin). $\mathcal{P}$ also takes inputs (x, f, y), where x is the input, y the purported output $f(x)$, and $f \in \mathcal{F}_\lambda$ is a function from a family $\mathcal{F}_\lambda$ parametrized by λ.
- $\mathsf{reward}^x(\mathsf{pp}, f, y, \tau) \to R \in \mathbb{R}_{\geq 0}$: produces a non-negative real number R representing the reward on input pp, f, y, transcript τ and with *oracle access* to input x.

We require two properties, roughly corresponding to traditional completeness and soundness. Rational completeness intuitively captures the idea that the honest strategy is the one obtaining (essentially) the highest reward.

Definition 1 (Rational Completeness). *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for every probabilistic polynomial-time algorithm* $\mathcal{P}^*$*, for all* $\lambda \in \mathbb{N}$*, inputs* $x \in \{0,1\}^*$*, functions* $f \in \mathcal{F}_\lambda$*, strings* $y \in \{0,1\}^*$*, s.t. the transcripts* $\tau^* \leftarrow \langle \mathcal{P}^*(\mathsf{pp}, x, f, y), \mathcal{V} \rangle(\mathsf{pp})$*,* $\tau \leftarrow \langle \mathcal{P}(\mathsf{pp}, x, f, f(x)), \mathcal{V} \rangle(\mathsf{pp})$ *satisfy:*

$$\mathbb{E}[\mathsf{reward}^x(\mathsf{pp}, f, y, \tau^*)] - \mathbb{E}[\mathsf{reward}^x(\mathsf{pp}, f, f(x), \tau)] \leq \mathsf{negl}(\lambda), \text{ w/ } \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$$

Rational soundness, on the other hand, ensures that a dishonest prover should suffer a substantial loss. This property is defined within an adaptive reward game, where an adversary seeks to maximize its reward by deviating from honest behaviour.

Game $\mathsf{ExpReward}^{\mathcal{A}}(\lambda)$**:**

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$;
$(f, x, y) \leftarrow \mathcal{A}(\mathsf{pp})$;
$R_{\mathcal{P}} \leftarrow \mathsf{reward}^x(\mathsf{pp}, f, f(x), \langle \mathcal{P}, \mathcal{V} \rangle)$ **if** $y \neq f(x) \wedge f \in \mathcal{F}_\lambda$ **then**
  $R_{\mathcal{A}} \leftarrow \mathsf{reward}^x(\mathsf{pp}, f, y, \langle \mathcal{A}, \mathcal{V} \rangle)$
else
  $R_{\mathcal{A}} \leftarrow 0$;
return $(R_P - R_{\mathcal{A}})/R_P$;

Fig. 1: Game $\mathsf{ExpReward}^{\mathcal{A}}$

Game $\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{RO}}(\lambda)$**:**

$\mathcal{H} \leftarrow \mathsf{SampleRO}(1^\lambda)$;
$\mathsf{pp} \leftarrow \mathsf{Setup}^{\mathcal{H}}(1^\lambda)$;
$(f, x, y, \pi_{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathcal{H}}(\mathsf{pp})$;
$\pi_{\mathcal{P}} \leftarrow \mathcal{P}^{\mathcal{H}}(\mathsf{pp}, f, x, f(x))$;
$R_{\mathcal{P}} \leftarrow \mathsf{reward}^{\mathcal{H},x}(\mathsf{pp}, f, f(x), \pi_P)$;
if $y \neq f(x) \wedge f \in \mathcal{F}_\lambda$ **then**
  $R_{\mathcal{A}} \leftarrow \mathsf{reward}^{\mathcal{H},x}(\mathsf{pp}, f, y, \pi_{\mathcal{A}})$;
else
  $R_{\mathcal{A}} \leftarrow 0$;
return $(R_P - R_{\mathcal{A}})/R_P$;

Fig. 2: Game $\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{RO}}$

Definition 2 (Rational Soundness). *A rational argument system satisfies rational soundness for a function domain* $\mathcal{F}$ *if, for all stateful PPT adversaries* $\mathcal{A}$*, there exists a polynomial* $q(\cdot)$ *s.t. for all security parameters* $\lambda \in \mathbb{N}$*:*

$$\mathbb{E}\left[\mathsf{ExpReward}^{\mathcal{A}}(\lambda)\right] \geq 1/q(\lambda)$$

where $\mathsf{ExpReward}$ *is defined in Fig. 1.*

This definition guarantees that, in expectation, even an adversary with adaptive capabilities cannot avoid incurring a loss that increases polynomially with the security parameter.

Remark 1. For a rational argument to be non-trivial, we require that the reward be polynomial in λ and the verifier run in time $o(|x|)$.

Non-Interactive Rational Arguments in the ROM Here we present a variant of the rational arguments model for the non-interactive case in the random oracle model (ROM). A non-interactive rational argument in the ROM is a tuple of algorithms $(\mathsf{Setup}, \mathcal{P}, \mathsf{reward})$ which work almost as in Section 2 with the exception that $\mathcal{P}$ is a non-interactive algorithm returning a proof π, there is no interactive challenger $\mathcal{V}$, all algorithms have access to a random oracle $\mathcal{H}$. Rational completeness in the ROM is a straightforward variant of Definition 1 and we do not present it; however, we explicitly show an analogue of Definition 2:

Definition 3 (Rational Soundness in the ROM). *We say that a non-interactive rational argument satisfies rational soundness (in the ROM) w.r.t. a function domain $\mathcal{F}$ if, for all (potentially non-uniform)* $\mathsf{PPT}\mathcal{A}$*, there exists a polynomial $q(\cdot)$ s.t. for all security parameters $\lambda \in \mathbb{N}$*

$$\mathbb{E}\left[\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{RO}}(\lambda)\right] \geq 1/q(\lambda)$$

where $\mathsf{ExpReward}_{\mathsf{RO}}$ *is defined in Fig. 2.*

3 Fiat-Shamir Transforms for Rational Arguments

Here we present two versions of Fiat-Shamir. The "simple" variant (sFS) directly adapts the standard FS to the holographic setting. The "stronger" variant (digFS) works in a model where the verifier is also given a digest to its input. Naturally, for it to be interesting, an FS transform for holographic settings should preserve the sublinearity of the verifier—this is the case for our two compilers.

3.1 A Simple Fiat-Shamir for Rational Arguments

Let Π be a (standard) interactive rational argument. We denote by $\mathsf{sFS}[\Pi]$ the non-interactive rational argument in the random oracle mode obtained by applying the transform in Fig. 3[10].

[10] Notice we make a change from the syntax in Section 2 that is without loss of generality: we make explicit the part of the reward function that simply queries the input x (Query) and the part without access to x that simply outputs the reward. This makes it easier to see why some of (in)security claims hold in later sections.

$\mathcal{P}^H(\mathsf{pp}_{\mathsf{RP}}, f, x, y)$:
  $m_1 \leftarrow \mathcal{P}(\mathsf{pp}_{\mathsf{RP}}, x, y)$;
  for $i = 2$ **to** r **do**
    $e_{i-1} \leftarrow H(\mathsf{pp}_{\mathsf{RP}}, f, y, m_1, \ldots, m_{i-1})$;
    $m_i \leftarrow \mathcal{P}(e_{i-1})$;
  return $\pi \leftarrow (m_1, \ldots, m_r)$;

$\mathcal{V}^{H,x}(\mathsf{pp}_{\mathsf{RP}}, f, y, \pi)$:
  $\pi \leftarrow (m_1, \ldots, m_r)$;
  for $i = 1$ **to** r **do**
    $e_i \leftarrow H(\mathsf{pp}_{\mathsf{RP}}, f, y, m_1, \ldots, m_i)$;
  $\tau \leftarrow (m_1, e_1, \ldots, m_r, e_r)$;
  $I \leftarrow \mathsf{Query}^x(\tau)$;
  I is a set of indices;
  $\tilde{x} \leftarrow x_I$;
  $R \leftarrow \mathsf{reward}(\mathsf{pp}_{\mathsf{RP}}, f, y, \tau, \tilde{x})$;
  return R;

Fig. 3: "Simple" FS Transform for Rational Proofs. Setup is unchanged.

$\mathcal{P}^H(\mathsf{pp}_{\mathsf{RP}}, f, x, y)$:
  $\delta \leftarrow H(x)$;
  $m_1 \leftarrow \mathcal{P}(\mathsf{pp}_{\mathsf{RP}}, x, y)$;
  for $i = 2$ **to** r **do**
    $e_{i-1} \leftarrow H(\mathsf{pp}_{\mathsf{RP}}, f, \delta, y, m_1, \ldots, m_{i-1})$;
    $m_i \leftarrow \mathcal{P}(e_{i-1})$;
  return $\pi \leftarrow (m_1, \ldots, m_r)$;

$\mathcal{V}^{H,x}(\mathsf{pp}_{\mathsf{RP}}, f, \delta, y, \pi)$:
  $\pi \leftarrow (m_1, \ldots, m_r)$;
  for $i = 1$ **to** r **do**
    $e_i \leftarrow H(\mathsf{pp}_{\mathsf{RP}}, f, \delta, y, m_1, \ldots, m_i)$;
  $\tau \leftarrow (m_1, e_1, \ldots, m_r, e_r)$;
  $I \leftarrow \mathsf{Query}^x(\tau)$;
  I is a set of indices;
  $\tilde{x} \leftarrow x_I$;
  $R \leftarrow \mathsf{reward}(\mathsf{pp}_{\mathsf{RP}}, f, \delta, y, \tau, \tilde{x})$;
  return R;

Fig. 4: "Stronger" FS Transform for Rational Proofs in the Input-Digest Model (changes from Fig. 3 are in blue). Setup is unchanged.

True-to-False-Input Attacks Here we describe a class of general attacks on schemes compiled with sFS.

Definition 4 (Input Query Set). *Let Π denote a rational argument*[11]*. The input query set for Π is defined as the set of indices and values queried during the reward stage of the rational argument. Specifically, for a function f, input x, alleged output y, proof π, and parameters* pp*:*

$$\mathsf{QSet}_{\mathsf{pp}}(f, x, y, \pi) := (i, x[i])_{i \in \mathcal{I}} \quad \textit{where } \mathcal{I} \leftarrow \mathsf{Query}^x(\mathsf{pp}, f, y, \pi)$$

We will often omit the parameters pp *when their context is clear.*

The following lemma introduces the input query set as a valuable abstraction. Its proof follows immediately.

Lemma 1. *Let Π, f, y, and π be as in Definition 4. If the inputs x and x' satisfy $\mathsf{QSet}_{\mathsf{pp}}(f, x, y, \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', y, \pi)$, then we have $\mathsf{reward}^x(f, x, y, \pi) = \mathsf{reward}^{x'}(f, x', y, \pi)$.*

Next, we define a *true-to-false-input* adversary. Informally, such an adversary, upon receiving a function f and input x (as appropriately quantified), efficiently outputs $x' \neq x$ such that: *(i)* $\mathsf{QSet}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', f(x), \pi)$ for a given π; and *(ii)* $f(x)$ is a false output for x', i.e., $f(x') \neq f(x)$.

[11] The rational argument may be interactive or non-interactive in general. The distinction in this definition is straightforward. For results concerning true-to-false-input adversaries, we assume interactive arguments for QSet, while we assume non-interactive arguments in the ROM for the robustness definition.

Definition 5 (True-to-false-input adversary). *A true-to-false-input adversary against an interactive rational argument Π is a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ such that for all security parameters λ, there exists*[12] *a function $f \in \mathcal{F}_\lambda(\Pi)$ and an input x such that for all transcripts π, the adversary $\mathcal{A}(\mathsf{pp}, f, x, \pi)$ outputs x', with overwhelming probability, satisfying $\mathsf{QSet}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', f(x), \pi) \wedge x \neq x' \wedge f(x) \neq f(x')$, where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$.*

Theorem 1 (True-to-false-input adversaries break the FS transform). *Let Π be a secure interactive rational argument. If there exists a true-to-false-input adversary $\mathcal{A}_T$ against Π, then the compiled protocol $\mathsf{sFS}[\Pi]$ is insecure.*

Proof. We construct a true-to-false-input adversary $\mathcal{A}^*(\mathsf{pp})$ as follows (note that this adversary has access to a random oracle, as in Fig. 2):

- Let f and x be as in Definition 5. We assume that $\mathcal{A}^*$ is non-uniform and has these strings embedded in its code for each value of λ.
- The honest prover $\mathcal{P}^{\mathcal{H}}(\mathsf{pp}, f, x, f(x))$ is executed, yielding the proof π_{hon}.
- The adversary computes $x_{\text{dis}} \leftarrow \mathcal{A}(\mathsf{pp}, f, x, \pi_{\text{hon}})$.
- It then returns $(f, x_{\text{dis}}, f(x), \pi_{\text{hon}})$.

Applying Lemma 1, we observe that with overwhelming probability,

$$\mathsf{reward}^{x_{\text{dis}}}(\mathsf{pp}, f, f(x), \pi_{\text{hon}}) = \mathsf{reward}^{x_{\text{hon}}}(\mathsf{pp}, f, f(x), \pi_{\text{hon}})$$

implying that Definition 3 does not hold. This concludes the proof. □

3.2 Strengthening Fiat-Shamir: the Input-Digest Model

The attacks above depend on the ability of the adversary to "have the same challenges on the same inputs". If Fiat-Shamir could somehow compute challenges that depend on the *whole* input, we might be able to prevent them. This is not immediate in the holographic setting since we do not want to read the whole input. We therefore propose a model where the verifier takes as input also a (collision resistant) digest δ_x. For simplicity we let the digest be the RO invocation on x. We dub this the *input-digest* model. The latter is a straightforward adaptation of the baseline notion in Section 2 and is described in the appendix (Fig. 4 also implicitly illustrates how it works)[13].
Let Π be a (standard) interactive rational argument. We denote by $\mathsf{digFS}[\Pi]$ the non-interactive rational argument in the input-digest model obtained by applying the transform in Fig. 4.

Robustness A simple sanity check for digFS is that it should prevent the attacks from the previous section. The following notion captures protocols not susceptible to true-to-false-input attacks. Notice that it assumes a simple adaptation of Definition 5 (Definition 13 in appendix) since the latter assumes interaction.

[12] Alternatively, one could define this adversary to output $(f, x, x', \dots)$ satisfying the properties mentioned above. This definition highlights that the adversary $\mathcal{A}$ transitions from the honest input $(x, f(x))$ to a dishonest one $(x', f(x))$, which underpins the vulnerability of the basic Fiat-Shamir transformation.

[13] The input-digest model could also be interesting for the case of interactive rational arguments. For us it is a tool to study the security of Fiat-Shamir; we leave studying this model in the interactive setting as an open problem.

Definition 6 (Robust protocol). *We say that a rational argument in the input-digest model is robust if there exist no true-to-false-input adversaries (as by the variant of Definition 5 in Definition 13).*

The following theorem actually shows that, for our target schemes, digFS leads to robust protocols. We do not generalize this result but it can be done; it extends to all rational arguments with query sets with enough entropy.

Theorem 2. *Let* $\Pi \in \{\Pi_{\mathrm{AM13}}, \Pi_{\mathrm{CG15}}\}$*, then* $\mathsf{digFS}[\Pi]$ *is robust.*

Unfortunately, as we show in the next section (Theorem 6 and Theorem 8) it is the case that robustness $\not\Rightarrow$ soundess (where soundness is in the rational sense).

4 Negative Results on the Security of Fiat-Shamir

4.1 Background on our case studies

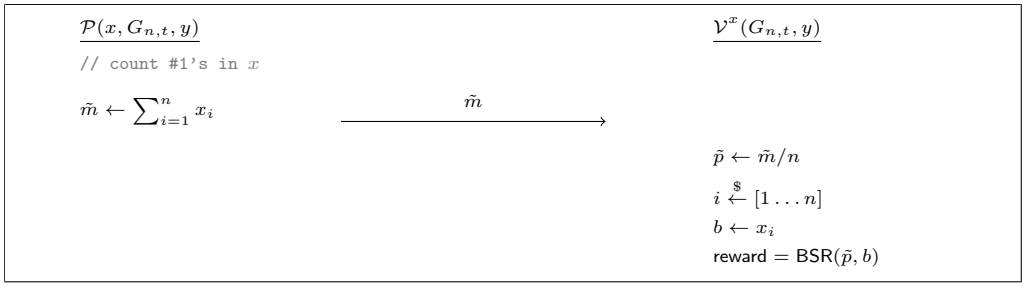

Fig. 5: Construction Π_{AM13} for threshold gates.

Azar-Micali Rational Interactive Protocol [3]. The protocol evaluates a threshold gate $G_{n,t}(x_1, \dots, x_n)$, where n is the number of Boolean inputs, and t is the minimum number of 1s needed for an output of 1. The Prover first announces $\tilde{m}$, the claimed number of inputs set to 1, allowing the Verifier to compute the gate's output based on $\tilde{m}$ and t.

The Verifier randomly selects an index $i \in [1, \dots, n]$ and observes the corresponding input bit $b = x_i$, which helps evaluate the Prover's claim. The Verifier then computes $\tilde{p} = \tilde{m}/n$, the Prover's claimed probability that a random input is 1. The reward mechanism, based on Brier's Scoring Rule (BSR) [8], adjusts according to b: if $b = 1$, the reward is $\mathsf{BSR}(\tilde{p}, 1) = 2\tilde{p}(2 - \tilde{p})$; if $b = 0$, $\mathsf{BSR}(\tilde{p}, 0) = 2(1 - \tilde{p}^2)$.

The expected reward is $\mathbb{E}[\mathsf{reward}] = p \cdot \mathsf{BSR}(\tilde{p}, 1) + (1 - p) \cdot \mathsf{BSR}(\tilde{p}, 0)$, where m is the true number of 1s, and $p = m/n$. The reward is maximized when $\tilde{p} = p$, encouraging honest reporting. Misreporting reduces the reward by at least $2(p - \tilde{p})^2$, implying a minimum penalty of $2/n^2$.

For a dishonest Prover $\tilde{\mathcal{P}}$, misreporting with probability $\epsilon_{\tilde{\mathcal{P}}}$, the penalty $\delta_{\tilde{\mathcal{P}}}$ must satisfy $\delta_{\tilde{\mathcal{P}}} > 2\epsilon_{\tilde{\mathcal{P}}}/n^2$. This ensures the penalty for dishonesty exceeds any gains, affirming

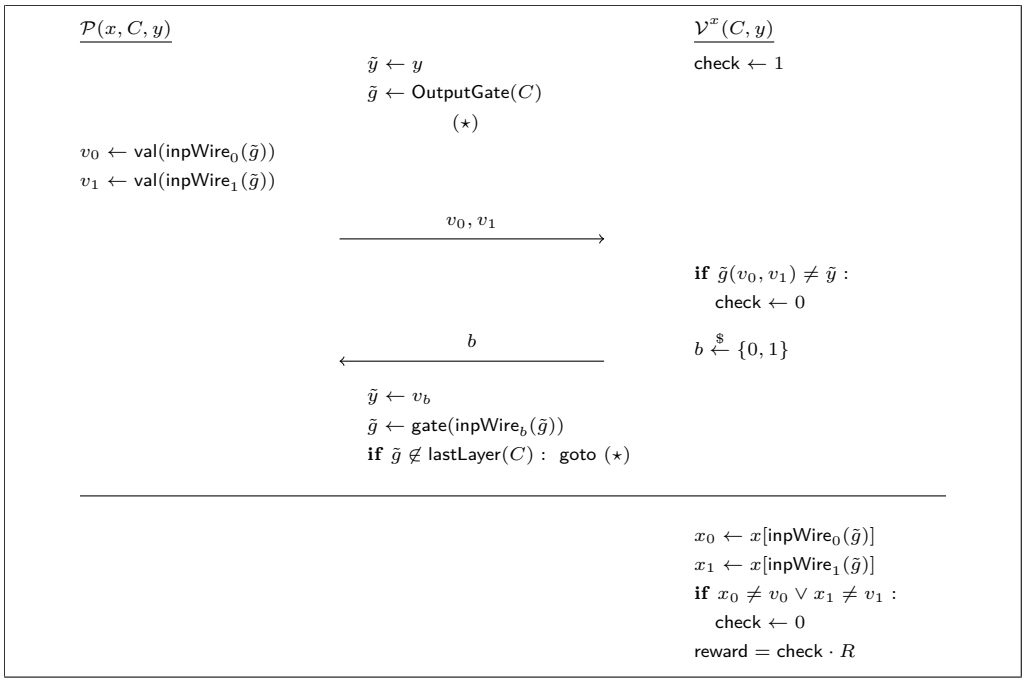

Fig. 6: The construction Π_{CG15} from [10]. Ours is a more specific variant for a (layered) formula C of logarithmic depth. Operations in the centre are run by both parties. $\mathsf{val}(\mathsf{w})$ is the value of the wire w in the evaluation $C(x)$, $\mathsf{inpWire}_0(g)$ (resp. $\mathsf{inpWire}_1(g)$) returns the left (resp. right) input wire of gate g, $g(u, w)$ returns the output of gate g on inputs bits u, w, $\mathsf{lastLayer}$ is the formula's last layer of gates (whose input wires are from x). The quantity R is some fixed positive reward which depends only on the parameters.

the protocol's rationality. Therefore, honesty maximizes the Prover's reward, with the quadratic penalty and random verification enforcing an efficient, secure protocol.

Theorem 3 ([3]). *The protocol Π_{AM13} in Fig. 5 is a secure interactive rational argument for threshold gates with constant rounds, $\mathcal{O}(\log n \log\log n)$[14] verification time and $\mathcal{O}(1)$ queries to the input x.*

Campanelli-Gennaro Rational Interactive Protocol [10]. Π_{CG15} is an efficient interactive proof system for a Boolean formula $f : \{0,1\}^n \to \{0,1\}$, computed by a circuit C with size $|C|$, depth d, and fan-in 2. The protocol is defined as an interaction between $(\mathcal{P}(x, C, f(x)), \mathcal{V}^x(C, f(x)))$, where $x \in \{0,1\}^n$ is the input. For more details, see Fig. 6.

Intuitively speaking, the protocol begins with the Prover sending the claimed output y for input x to the Verifier, who then enters a recursive process. The Prover provides the values y_L and y_R for a specific output gate g, enabling the Verifier to check if $g(y_L, y_R) = y$. The Verifier randomly selects y_L or y_R to continue the recursion for other gates in the

[14] Azar and Micali show that the running time can be optimised to $\mathcal{O}(\log n)$ through the application of a randomised variant of Brier's scoring rule (see [3]).

subcircuit. For uniform circuits—whose structure can be efficiently computed—the Verifier remains particularly low.

Theorem 4 ([10]). *The protocol Π_{CG15} in Fig. 6 is a secure interactive rational argument for the class of formulas C of logarithmic depth for an n-bit input x. It requires logarithmic rounds and assuming the circuits are T-uniform, the verifier runs in $\mathcal{O}(\log n \cdot T(n))$ time, with $\mathcal{O}(1)$ queries to the input.*

4.2 Negative Results for Π_{AM13}

True-to-false-input attacks against simple Fiat-Shamir The proof of the following theorem is in the appendix and closes an approach similar to that for Theorem 7.

Theorem 5. *The protocol $\mathsf{sFS}[\Pi_{\mathrm{AM13}}]$ is insecure.*

Attacks in the input-digest model

Theorem 6. *The construction $\mathsf{digFS}[\Pi_{\mathrm{AM13}}]$ is insecure.*

Proof. At the high level, we build an adversary that can find a function for which the transcript through Fiat-Shamir is going to be "benign" for them (i.e., it is going to lead to a reward as high as in the case the output had been honest). The adversary can do this after fixing a target input x and that is intuitively why its digest is not going to help. We construct an adversary as follows:

- Let x be a string with half of its inputs 1s and the rest 0s, i.e., $x = 1^{n/2} \parallel 0^{n/2}$.
- Compute the digest for x, i.e., $\delta \leftarrow \mathcal{H}(x)$.
- Let the (false) claimed output be $y \leftarrow 1$.
- We iterate over all possible threshold gates that output 0 on x, i.e. for $t \in \{n/2 + 1, \ldots, n\}$:
  - Let $\tilde{m}_t := t$ be the prover's message (that is the adversary is claiming there are exactly t 1-s in x; see Fig. 5);
  - Compute challenge $e^* \leftarrow \mathcal{H}(G_{n,t}, \delta, y, \tilde{m}_t)$
  - If $e^* \leq n/2$ (i.e., if $x[n/2] = 1$) return $(G_{n,t}, x, y, \pi := \tilde{m}_t)$
- Return $\perp$ if all iterations above fail.

We first claim that the probability that the loop finds a "good" cheating challenge e^* after a polynomial number of steps is overwhelming. Let us bound the probability that this event does *not* happen. Now, this is the event:

$$\forall t \in \{n/2 + 1, \ldots, n\} \bigwedge_{i \in [n/2]} \mathcal{H}(G_{n,t}, \delta, y, \tilde{m}_t) \neq i$$

This probability of this event is at most $2^{-n/2}$ (negligible) and our claim follows.

The second observation we need is that for a $\tilde{m}_t$ returned by the adversary, the reward is the highest possible reward for that particular input (by inspection of Π_{AM13} we can see that there are two possible rewards and the highest one in this case is given by a queried bit 1 in the input, which is exactly what a "good" challenge e^* provides). Combining this observation with the previous claim we see we built an *always* cheating adversary (as in *always returning a false output*) achieving in expectation $(1 - \mathsf{negl}(n)) \cdot R_{\mathrm{hon}}$ with R_{hon} the honest reward. This violates the noticeable loss required by Definition 3 and concludes the proof. □

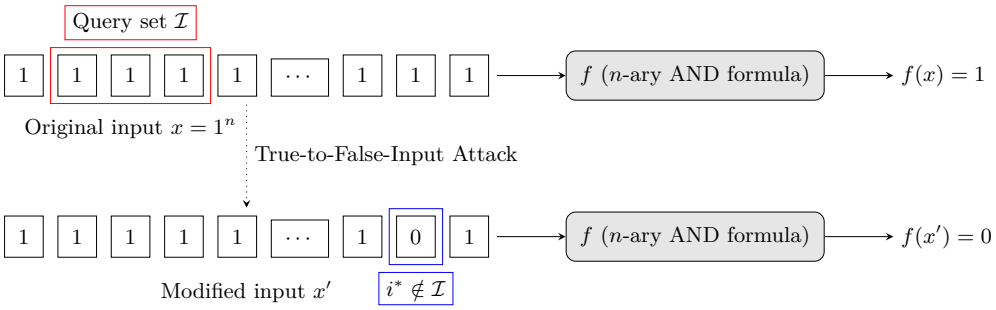

Fig. 7: Example of a true-to-false-input attack on Π_{CG15}. The original input x is transformed into a modified input x' by changing a bit, affecting the function's output f. The query set $\mathcal{I}$ does not include the index of the changed bit i^*.

4.3 Negative Results for Π_{CG15}

True-to-false-input attacks against simple Fiat-Shamir

Theorem 7. *The construction* $\mathsf{sFS}[\Pi_{\mathrm{CG15}}]$ *is insecure.*

Proof. We show a true-to-false-input attacker against Π_{CG15}. Consider the function $f : \{0,1\}^n \to \{0,1\}$, which is composed of n-ary AND gates. Let $x \coloneqq 1^n$, representing the input where every bit is set to 1. Let π denote an arbitrary transcript. We define the adversary true-to-false-input $\mathcal{A}$ as follows: First, the adversary retrieves the query set $\mathcal{I}$ by executing $\mathsf{Query}^x(\mathsf{pp}, f, f(x), \pi)$. Next, the adversary selects the smallest index i^* from the set $\{1, \ldots, n\} \setminus \mathcal{I}$, ensuring that $i^* \notin \mathcal{I}$. The adversary then constructs a modified input x' by setting $x'[i^*] = 0$, while leaving all other bits unchanged.

We now verify that $\mathcal{A}$ satisfies the required properties of an adversary, as defined in Definition 5. First, we show that the query set is preserved, i.e., $\mathcal{Q}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathcal{Q}_{\mathsf{pp}}(f, x', f(x), \pi)$. Since $x[i] = x'[i]$ for all $i \in \mathcal{I}$, the query set remains identical. Next, we observe that $x \neq x'$, as $x'[i^*] = 0 \neq 1 = x[i^*]$. Lastly, we establish that $f(x) \neq f(x')$: since f is an n-ary AND function, we know that $f(x) = 1$; however, because x' has a 0 at index i^*, we have $f(x') = 0$, as the AND operation outputs 0 if any input bit is 0. This completes the proof. The argument can be trivially extended to n-ary OR gates. □

Attacks in the input-digest model The proof of the next result is a slightly more complicated variant of that of Theorem 6 which we defer to Appendix E.

Theorem 8. *The construction* $\mathsf{digFS}[\Pi_{\mathrm{CG15}}]$ *is insecure.*

Acknowledgement. We thank Ron Rothblum for engaging in useful discussions and steering us towards what became our stronger variant of the Fiat-Shamir transform. Matteo Campanelli would like to thank Pavel Hubáček who indirectly made this work possible by sending Agni Datta his way (Matteo's way) for thesis supervision. Matteo also thanks Chaya Ganesh and Rosario Gennaro with whom, during another project, he had shared conversations on the initial hunch that Fiat-Shamir "may definitely not have worked" in the rational setting. Agni would like to thank Matteo Campanelli for

supervising him and for motivating and enabling him to work in theory and cryptography. We designate both authors as first authors due to their substantial contributions to the research and manuscript preparation.

References

1. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022: 20th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 13747, pp. 113–142. Springer, Cham, Switzerland, Chicago, IL, USA (Nov 7–10, 2022). https://doi.org/10.1007/978-3-031-22318-1_5
2. Azar, P.D., Micali, S.: Rational proofs. In: Karloff, H.J., Pitassi, T. (eds.) 44th Annual ACM Symposium on Theory of Computing. pp. 1017–1028. ACM Press, New York, NY, USA (May 19–22, 2012). https://doi.org/10.1145/2213977.2214069
3. Azar, P.D., Micali, S.: Super-efficient rational proofs. In: Proceedings of the Fourteenth Acm Conference on Electronic Commerce. EC '13, ACM (Jun 2013). https://doi.org/10/gtxc77
4. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd Annual Symposium on Foundations of Computer Science. pp. 106–115. IEEE Computer Society Press, Las Vegas, NV, USA (Oct 14–17, 2001). https://doi.org/10.1109/SFCS.2001.959885
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93: 1st Conference on Computer and Communications Security. pp. 62–73. ACM Press, Fairfax, Virginia, USA (Nov 3–5, 1993). https://doi.org/10.1145/168588.168596
6. Berman, I., Rothblum, R.D., Vaikuntanathan, V.: Zero-knowledge proofs of proximity. In: Karlin, A.R. (ed.) ITCS 2018: 9th Innovations in Theoretical Computer Science Conference. vol. 94, pp. 19:1–19:20. LIPIcs, Cambridge, MA, USA (Jan 11–14, 2018). https://doi.org/10.4230/LIPIcs.ITCS.2018.19
7. Bitansky, N., Kalai, Y.T., Paneth, O.: Multi-collision resistance: a paradigm for keyless hash functions. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th Annual ACM Symposium on Theory of Computing. pp. 671–684. ACM Press, Los Angeles, CA, USA (Jun 25–29, 2018). https://doi.org/10.1145/3188745.3188870
8. Brier, G.W.: Verification of forecasts expressed in terms of probability. Monthly Weather Review **78**(1), 1–3 (Jan 1950). https://doi.org/10.1175/1520-0493(1950)0782.0.co;2
9. Campanelli, M., Ganesh, C., Gennaro, R.: How to make rational arguments practical and extractable. IACR Communications in Cryptology (CiC) **1**(1), 19 (2024). https://doi.org/10.62056/a63zl86bm
10. Campanelli, M., Gennaro, R.: Sequentially Composable Rational Proofs, pp. 270–288. Springer International Publishing (2015). https://doi.org/10/gtx3j5
11. Campanelli, M., Gennaro, R.: Efficient Rational Proofs for Space Bounded Computations, pp. 53–73. Springer International Publishing (2017). https://doi.org/10/gt7hrx
12. Campanelli, M., Gennaro, R.: Fine-grained secure computation. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018: 16th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 11240, pp. 66–97. Springer, Cham, Switzerland, Panaji, India (Nov 11–14, 2018). https://doi.org/10.1007/978-3-030-03810-6_3
13. Dao, Q., Grubbs, P.: Spartan and bulletproofs are simulation-extractable (for free!). In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part II. Lecture Notes in Computer Science, vol. 14005, pp. 531–562. Springer, Cham, Switzerland, Lyon, France (Apr 23–27, 2023). https://doi.org/10.1007/978-3-031-30617-4_18
14. Dao, Q., Miller, J., Wright, O., Grubbs, P.: Weak fiat-shamir attacks on modern proof systems. In: 2023 IEEE Symposium on Security and Privacy. pp. 199–216. IEEE Computer

Society Press, San Francisco, CA, USA (May 21–25, 2023). https://doi.org/10.1109/SP46215.2023.10179408
15. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.J.: Magic functions. In: 40th Annual Symposium on Foundations of Computer Science. pp. 523–534. IEEE Computer Society Press, New York, NY, USA (Oct 17–19, 1999). https://doi.org/10.1109/SFFCS.1999.814626
16. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO'86. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer, Berlin, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
17. Gensyn: Gensyn litepaper (Feb 2022), https://docs.gensyn.ai/litepaper
18. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd Annual ACM Symposium on Theory of Computing. pp. 99–108. ACM Press, San Jose, CA, USA (Jun 6–8, 2011). https://doi.org/10.1145/1993636.1993651
19. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th Annual Symposium on Foundations of Computer Science. pp. 102–115. IEEE Computer Society Press, Cambridge, MA, USA (Oct 11–14, 2003). https://doi.org/10.1109/SFCS.2003.1238185
20. Guo, S., Hubácek, P., Rosen, A., Vald, M.: Rational arguments: single round delegation with sublinear verification. In: Naor, M. (ed.) ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science. pp. 523–540. Association for Computing Machinery, Princeton, NJ, USA (Jan 12–14, 2014). https://doi.org/10.1145/2554797.2554845
21. Guo, S., Hubácek, P., Rosen, A., Vald, M.: Rational sumchecks. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A: 13th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 9563, pp. 319–351. Springer, Berlin, Heidelberg, Germany, Tel Aviv, Israel (Jan 10–13, 2016). https://doi.org/10.1007/978-3-662-49099-0_12
22. Gur, T., Rothblum, R.D.: Non-interactive proofs of proximity. In: Roughgarden, T. (ed.) ITCS 2015: 6th Conference on Innovations in Theoretical Computer Science. pp. 133–142. Association for Computing Machinery, Rehovot, Israel (Jan 11–13, 2015). https://doi.org/10.1145/2688073.2688079
23. Gur, T., Rothblum, R.D.: A hierarchy theorem for interactive proofs of proximity. In: Papadimitriou, C.H. (ed.) ITCS 2017: 8th Innovations in Theoretical Computer Science Conference. vol. 4266, pp. 39:1–39:43. LIPIcs, Berkeley, CA, USA (Jan 9–11, 2017). https://doi.org/10.4230/LIPIcs.ITCS.2017.39
24. Holmgren, J., Lombardi, A., Rothblum, R.D.: Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In: Khuller, S., Williams, V.V. (eds.) 53rd Annual ACM Symposium on Theory of Computing. pp. 750–760. ACM Press, Virtual Event, Italy (Jun 21–25, 2021). https://doi.org/10.1145/3406325.3451116
25. Inasawa, K., Yasunaga, K.: Rational proofs against rational verifiers. Cryptology ePrint Archive, Report 2017/270 (2017), https://eprint.iacr.org/2017/270
26. Jawale, R., Kalai, Y.T., Khurana, D., Zhang, R.Y.: SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. In: Khuller, S., Williams, V.V. (eds.) 53rd Annual ACM Symposium on Theory of Computing. pp. 708–721. ACM Press, Virtual Event, Italy (Jun 21–25, 2021). https://doi.org/10.1145/3406325.3451055
27. Kiyoshima, S.: Public-coin 3-round zero-knowledge from learning with errors and keyless multi-collision-resistant hash. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022, Part I. Lecture Notes in Computer Science, vol. 13507, pp. 444–473. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 15–18, 2022). https://doi.org/10.1007/978-3-031-15802-5_16
28. Micali, S.: Computationally sound proofs. SIAM Journal on Computing **30**(4), 1253–1298 (2000)
29. Rothblum, G.N., Vadhan, S.P., Wigderson, A.: Interactive proofs of proximity: delegating computation in sublinear time. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th

Annual ACM Symposium on Theory of Computing. pp. 793–802. ACM Press, Palo Alto, CA, USA (Jun 1–4, 2013). https://doi.org/10.1145/2488608.2488709
30. Thaler, J.: Measuring snark performance, https://a16zcrypto.com/posts/article/measuring-snark-performance-frontends-backends-and-the-future/
31. Thaler, J., et al.: Proofs, arguments, and zero-knowledge. Foundations and Trends® in Privacy and Security **4**(2–4), 117–660 (2022)
32. Vaudenay, S.: Proof of proximity of knowledge. Cryptology ePrint Archive, Report 2014/695 (2014), https://eprint.iacr.org/2014/695

A Prior Definitions of Rational Proofs and Arguments

In this appendix, we provide the definitions of rational arguments as introduced by Guo et al. [20] (and slightly adapted in [9]). As we emphasized in the main text, our approach diverges by using a security game-based definition instead of the reward gap-based definitions we present here.

Definition 7 (Rational Argument [20,9]). *Let $f : \{0,1\}^n \to \{0,1\}^*$ be a circuit. We say that f admits a rational argument if there exists an interactive protocol $(\mathcal{P}, \mathcal{V})$ and a reward function $\mathsf{reward} : \{0,1\}^* \to \mathbb{R}_{\geq 0}$ such that, for every input $x \in \{0,1\}^n$ and prover $\hat{\mathcal{P}}$ of size at most $2^{\lambda(n)}$, the following hold:*

- ***Correctness:*** *The protocol achieves correctness[15] with high probability:*

$$\mathsf{Pr}[\mathsf{output}((\mathcal{P}, \mathcal{V})(x)) = f(x)] \geq 1 - \mathsf{negl}(\lambda).$$

- ***Reward Bound:*** *The expected reward of any alternative prover $\hat{\mathcal{P}}$ is close to that of the honest prover:*

$$\mathbb{E}[\mathsf{reward}((\hat{\mathcal{P}}(x), \mathcal{V}^{[x]}))] \leq \mathbb{E}[\mathsf{reward}((\mathcal{P}(x), \mathcal{V}^{[x]}))] + \mathsf{negl}(\lambda).$$

- ***Noticeable Reward Gap for Incorrect Outputs:*** *The protocol has a noticeable reward gap (defined in Definition 8)*

Reward Gap. The reward gap quantifies the incentive for honest behaviour by measuring the expected loss that provers incur when they consistently deviate from correctly reporting $f(x)$.

Definition 8 (Reward Gap). *Let $(\mathcal{P}, \mathcal{V})$ be a rational argument for the function $f : \{0,1\}^n \to \{0,1\}^*$ with a reward function reward. Define the following quantities:*

- $\epsilon_{\hat{\mathcal{P}}} = \mathsf{Pr}[\mathsf{output}((\hat{\mathcal{P}}, \mathcal{V})(x)) \neq f(x)]$, *representing the probability that an alternative prover $\hat{\mathcal{P}}$ outputs an incorrect result.*
- $\delta_{\mathcal{P}^*}(x) = \mathbb{E}[\mathsf{reward}((\mathcal{P}(x), \mathcal{V}^{[x]}))] - \mathbb{E}[\mathsf{reward}((\hat{\mathcal{P}}(x), \mathcal{V}^{[x]}))]$, *representing the expected reward difference between the honest prover $\mathcal{P}$ and an alternative prover $\mathcal{P}^*$.*

[15] Differently from the treatment in the main text, here there is an explicit output from the verifier. We find this unnecessary and model this by implicitly assuming that the verifier directly reports the claimed output.

The reward gap $\Delta : \mathbb{N} \to \mathbb{R}$ *is defined as:*

$$\Delta(n) = \min_{x} \min_{\mathcal{P}^*:\epsilon_{\mathcal{P}^*}=1} [\delta_{\mathcal{P}^*}(x)],$$

where the inner minimisation is over all provers $\mathcal{P}^*$ *that consistently output incorrect results* ($\epsilon_{\mathcal{P}^*} = 1$)*.*

Remark 2. It holds for any prover $\hat{\mathcal{P}}$ that:

$$\delta_{\hat{\mathcal{P}}}(x) \geq \epsilon_{\hat{\mathcal{P}}} \cdot \Delta(x).$$

Therefore, an implication of the requirement on reward gap is that any adversary cheating with noticeable probability will incur a noticeable reward loss.

B Equivalence Between Our Model and Earlier Ones

Here we compare our model of rational arguments with previous models (e.g., the ones from Appendix A).

Remark 3 (Advantages compared to previous definitions). A previous definition of "meaningful" rational soundness, due to [20], is based on the notion of *reward gap* (see Appendix A). This is defined as the minimum over all possible inputs and all possible adversaries of the gap between the honest and adversarial reward. This notion was meant to fix the excessively strong requirements of the original definition by Azar and Micali [2].

Arguably, both approaches have limitations: Azar and Micali's is not well-defined, whereas the one based on the reward gap requires this arguably complicated notion and adds additional requirements to the rational soundness definition (namely that is limited to adversaries with noticeable deviations).

Our definition has the advantage of being arguably simpler and being in the spirit of cryptographic games. Most importantly, as we show, in Theorem 9 it also captures the definition in [20]. Finally, it does not require us to concern ourselves with the probability of cheating of the adversary because of Lemma 2.

It is easy to see that the correctness-like properties correspond in the two models. A minor difference is that the previous definitions have a notion of "output" of the verifier while we do not (we assume w.l.o.g. the verifier will always output the claimed out from the prover). We ignore this minor difference in the formal comparison below where we show that the notions of soundness of the two models are equivalent.

Theorem 9. *A rational argument satisfies rational soundness as for Definition 2 if and only if it has a noticeable reward gap as defined Definition 8.*

Proof. For simplicity, we assume the reward to be in the interval $[0, 1]$, with the expected reward for the honest prover being 1 (the general case can be obtained by proper scaling). This allows us to think of the difference $R_P - R_{\mathcal{A}}$ as the same as $(R_P - R_{\mathcal{A}})/R_P$ simplifying certain steps. We will also crucially use the fact n, the size of x, is a polynomial in λ.

Let $(\mathcal{P}, \mathcal{V})$ be a rational argument system for a function f, and let reward be its reward function. We start by proving the forward direction. Assume the system

satisfies rational soundness as per Definition 2. We need to show it has a noticeable reward gap. By Definition 2, there exists a polynomial $q(\cdot)$ such that for all $\lambda \in \mathbb{N}$, we have $\mathbb{E}[\mathsf{ExpReward}^{\mathcal{A}}(\lambda)] \geq 1/q(\lambda)$. From the game definition in Fig. 1, we have $\mathsf{ExpReward}^{\mathcal{A}}(\lambda) = 1 - R_{\mathcal{A}}$, where R_P is the reward for the honest prover and $R_{\mathcal{A}}$ is the reward for the adversary. For any adversary $\mathcal{A}$ that consistently outputs incorrect results (i.e., $\epsilon_{\mathcal{A}} = 1$), we have $\mathbb{E}[R_P - R_{\mathcal{A}}] \geq 1/q(\lambda)$. This implies $\mathbb{E}[R_P] - \mathbb{E}[R_{\mathcal{A}}] \geq 1/q(\lambda)$. By the definition of reward gap $\Delta(\lambda)$ in Definition 8, we have $\Delta(n) = \min_x \min_{\mathcal{P}^*:\epsilon_{\mathcal{P}^*}=1}[\delta_{\mathcal{P}^*}(x)] \geq 1/q(\lambda)$. Since $q(\lambda)$ is a polynomial the reward gap is noticeable.

Now we prove the reverse direction. Assume the system has a noticeable reward gap. We need to show it satisfies rational soundness. By the noticeable reward gap assumption, there exists a polynomial $p(\cdot)$ such that for all $n \in \mathbb{N}$, we have $\Delta(n) \geq \frac{1}{p(n)}$. For any adversary $\mathcal{A}$, in the adaptive reward game $\mathsf{ExpReward}^{\mathcal{A}}$, we have:

$$\mathbb{E}[\mathsf{ExpReward}^{\mathcal{A}}(\lambda)] = \mathbb{E}[R_P] - \mathbb{E}[R_{\mathcal{A}}] \geq \mathbb{E}[R_P] - \mathbb{E}[R_{\mathcal{A}'}] \geq \Delta(n) \geq \frac{1}{p(n)}$$

where $\mathcal{A}'$ is as by Lemma 2. This concludes the proof. □

Lemma 2. *Let $\mathcal{A}$ be a PPT adversary satisfying $\mathbb{E}[\mathsf{ExpReward}^{\mathcal{A}}(\lambda)] \geq g(\lambda)$ for all security parameters λ, where g is a given function. Assume further that $\mathcal{A}$ behaves honestly with non-zero probability, i.e.,*

$$\mathsf{Pr}[f(x) = y \wedge f \in \mathcal{F}_\lambda \mid (f, x, y) \leftarrow \mathcal{A}(\mathsf{pp})] > 0$$

Then, there exists a PPT adversary $\mathcal{A}'$ such that $\mathbb{E}[\mathsf{ExpReward}^{\mathcal{A}'}(\lambda)] \leq g(\lambda)$ for all λ and such that $\mathcal{A}'$ deviates from honest behaviour with probability one, i.e.,

$$\mathsf{Pr}[f(x) = y \wedge f \in \mathcal{F}_\lambda \mid (f, x, y) \leftarrow \mathcal{A}'^{\mathcal{H}}(\mathsf{pp})] = 0$$

Proof. We construct the adversary $\mathcal{A}'$ by modifying $\mathcal{A}$ as follows: whenever $\mathcal{A}$ would produce an honest output y, $\mathcal{A}'$ instead outputs an incorrect response $y' \neq y$. In all other instances, $\mathcal{A}'$ replicates the function, input, and proof as generated by $\mathcal{A}$. Notice that in scenarios where $\mathcal{A}$ would incur zero reward (i.e. $R_{\mathcal{A}} = 0$ in Fig. 1), the adversary $\mathcal{A}'$ receives some non-negative reward $R_{\mathcal{A}'}$. This ensures that the output of the experiment involving $\mathcal{A}'$ yields $R_P - R_{\mathcal{A}'} \geq R_P - 0$, thereby concluding the proof. □

C Standard Fiat-Shamir Transform

Here we provide more background on the standard Fiat-Shamir transform for the reader's reference. The Fiat-Shamir transform, initially introduced by Fiat and Shamir [16], is a cornerstone technique of cryptographic protocol design. It allows to transform an interactive proof systems into a non-interactive one. This transformation is often used within the idealised Random Oracle Model (ROM) [5], where cryptographic hash functions are modelled as random oracles (i.e. a random function). In this setting, the verifier's random challenges, crucial in interactive protocols, are systematically replaced by the outputs of the random oracle.

Below is a formalisation of the Fiat-Shamir transformation, following the Dao and Grubbs [13]. Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ denote a public-coin interactive argument with r rounds (and hence $2r + 1$ messages=. The complete transcript of this interaction is denoted as $\tau = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$, where $a_i \in \{0,1\}^*$ represents the i-th prover message and $c_i \in \{0,1\}^*$ denotes the i-th verifier challenge. By convention, we assume $c_0 = \epsilon$, indicating that the prover initiates the communication and concludes the protocol.

Definition 9 (Fiat-Shamir Transformation). *Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be a public-coin $(2r+1)$-message interactive argument of knowledge for a relation $\mathcal{R}$, with transcript $\tau = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$. The Fiat-Shamir transformation FS converts Π into a non-interactive argument $\Pi_{\mathsf{FS}} = (\mathsf{Setup}_{\mathsf{FS}}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ in the Random Oracle Model as by Fig. 8.*

1. **Setup:** $\mathsf{Setup}_{\mathsf{FS}}(\lambda)$
   - Generate the public parameters pp by executing $\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda)$.
   - Sample a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$.
   - Output the transformed public parameters $\mathsf{pp}_{\mathsf{FS}} := (\mathsf{pp}, \mathcal{H})$.
2. **Prover:** $\mathcal{P}_{\mathsf{FS}}(\mathsf{pp}_{\mathsf{FS}}, x, w)$
   - Parse the public parameters $\mathsf{pp}_{\mathsf{FS}}$ as $(\mathsf{pp}, \mathcal{H})$.
   - Initialise the prover $\mathcal{P}(x, w)$ using the statement x and the witness w.
   - For each round $i = 1$ to r:
     - Generate the i-th prover message $a_i \leftarrow \mathcal{P}$.
     - Compute the corresponding verifier challenge $c_i := \mathcal{H}(\mathsf{pp}, x, a_1, \ldots, a_i)$ using the random oracle.
     - Input the challenge c_i into $\mathcal{P}$ for subsequent computations.
   - Generate the final prover message $a_{r+1} \leftarrow \mathcal{P}$.
   - Output the non-interactive proof $\pi := (a_1, \ldots, a_r, a_{r+1})$.
3. **Verifier:** $\mathcal{V}_{\mathsf{FS}}(\mathsf{pp}_{\mathsf{FS}}, x, \pi)$
   - Parse $\mathsf{pp}_{\mathsf{FS}}$ as $(\mathsf{pp}, \mathcal{H})$ and the proof π as $(a_1, \ldots, a_r, a_{r+1})$.
   - For each round $i = 1$ to r:
     - Calculate the verifier challenge $c_i := \mathcal{H}(\mathsf{pp}, x, a_1, \ldots, a_i)$.
   - Evaluate the interactive verifier $\mathcal{V}$ with the complete transcript, returning $\mathcal{V}(\mathsf{pp}, x, (a_1, c_1, \ldots, a_r, c_r, a_{r+1}))$.

Fig. 8: Standard Fiat-Shamir Transform.

D Rational Arguments in the Input-Digest Model

A rational argument in the input-digest model is non-interactive and assumes the ROM, but in addition, lets the prover have a digest to its input (we can imagine the verifier preprocessed it once and for all). Here we present the small changes from the other definitions in the paper.

A rational argument in the input-digest model consists of a tuple of (PPT, possibly interactive) algorithms $(\mathsf{Setup}, \mathcal{P}, \mathsf{reward})$ that work as follows (all algorithms have access to a random oracle $\mathcal{H}$):

– $\mathsf{Setup}^{\mathcal{H}}(1^\lambda) \to \mathsf{pp}$: outputs parameters pp on input security parameter λ;
– $\mathcal{P}^{\mathcal{H}}(\mathsf{pp}, x, f, y) \to \pi$ produces a proof.
– $\mathsf{reward}^{x,\mathcal{H}}(\mathsf{pp}, f, \delta, y, \pi) \to R \in \mathbb{R}_{\geq 0}$: provides a reward from usual inputs, a proof π and a digest of the input $\delta \leftarrow \mathcal{H}(x)$.

Definition 10 (Rational Completeness). *There exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for every PPT algorithm* $\mathcal{P}^*$*, for all* $\lambda \in \mathbb{N}$*, inputs* $x \in \{0,1\}^*$*, functions* $f \in \mathcal{F}_\lambda$*, strings* $y \in \{0,1\}^*$*, s.t. the proofs* $\pi^* \leftarrow \mathcal{P}^*(\mathsf{pp}, x, f, y)$*,* $\pi \leftarrow \mathcal{P}(\mathsf{pp}, x, f, f(x))$ *satisfy the condition:*

$$\mathbb{E}[\mathsf{reward}^x(\mathsf{pp}, f, \mathcal{H}(x), y, \pi)] - \mathbb{E}[\mathsf{reward}^x(\mathsf{pp}, f, \mathcal{H}(x), f(x), \pi)] \leq \mathsf{negl}(\lambda)$$

where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$*.*

Definition 11 (Rational Soundness in the input-digest model). *We say that a non-interactive rational argument satisfies rational soundness (in the input-digest model) w.r.t. a function domain* $\mathcal{F}$ *if, for all (potentially non-uniform) PPT* $\mathcal{A}$*, there exists a polynomial* $q(\cdot)$ *s.t. for all security parameters* $\lambda \in \mathbb{N}$

$$\mathbb{E}\left[\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{dig}}(\lambda)\right] \geq q(\lambda)$$

where $\mathsf{ExpReward}_{\mathsf{dig}}$ *is defined in Fig. 9.*

Game $\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{dig}}(\lambda)$**:**
- $\mathcal{H} \leftarrow \mathsf{SampleRO}(1^\lambda)$;
- $\mathsf{pp} \leftarrow \mathsf{Setup}^{\mathcal{H}}(1^\lambda)$;
- $(f, x, y, \pi_{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathcal{H}}(\mathsf{pp})$;
- $\pi_{\mathcal{P}} \leftarrow \mathcal{P}^{\mathcal{H}}(\mathsf{pp}, f, x, f(x))$;
- $R_{\mathcal{P}} \leftarrow \mathsf{reward}^{\mathcal{H},x}(f, \mathcal{H}(x), f(x), \pi_P)$;
- **if** $y \neq f(x) \wedge f \in \mathcal{F}_\lambda$ **then**
  - $R_{\mathcal{A}} \leftarrow \mathsf{reward}^{\mathcal{H},x}(\mathsf{pp}, f, \mathcal{H}(x), y, \pi_{\mathcal{A}})$;
- **else**
  - $R_{\mathcal{A}} \leftarrow 0$;
- **return** $R_P - R_{\mathcal{A}}$;

Fig. 9: Game $\mathsf{ExpReward}^{\mathcal{A}}_{\mathsf{dig}}$ for rational soundness in the input-digest model.

We adapt the definition of input query set and true-to-false-input adversary to the input-digest model.

Definition 12 (Input Query Set). *Let* Π *denote a rational argument in the input-digest model. The input query set for* Π *is defined as the set of indices and values queried during the reward stage of the rational argument. Specifically, for a function* f*, input* x*, alleged output* y*, proof* π*, and parameters* pp*:*

$$\mathsf{QSet}_{\mathsf{pp}}(f, x, y, \pi) := (i, x[i])_{i \in \mathcal{I}} \;\; \textit{where } \mathcal{I} \leftarrow \mathsf{Query}^{x,\mathcal{H}}(\mathsf{pp}, f, \mathcal{H}(x), y, \pi)$$

We will often omit the parameters pp *when their context is clear.*

Definition 13 (True-to-false-input adversary). *A true-to-false-input adversary against a non-interactive rational argument Π in the input-digest model is a PPT algorithm $\mathcal{A}$ such that for all security parameters λ, there exists a function $f \in \mathcal{F}_\lambda(\Pi)$ and an input x such that for all proofs π, the adversary $\mathcal{A}^{\mathcal{H}}(\mathsf{pp}, f, x, \pi)$ outputs x', with overwhelming probability, satisfying $\mathsf{QSet}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', f(x), \pi) \wedge x \neq x' \wedge f(x) \neq f(x')$, where $\mathsf{pp} \leftarrow \mathsf{Setup}^{\mathcal{H}}(1^\lambda)$ and QSet is from Definition 12.*

E Missing Proofs

E.1 Proof of Theorem 2

We start by approaching the robustness of $\mathsf{digFS}[\Pi_{\mathrm{AM13}}]$. Our goal is to show that there does not exist a successful true-to-false-input adversary $\mathcal{A}$ against $\mathsf{digFS}[\Pi_{\mathrm{AM13}}]$. According to Definition 13, this implies that for every security parameter $\lambda \in \mathbb{N}$, there exists a function f in the domain $\mathcal{F}_\lambda(\mathsf{digFS}[\Pi_{\mathrm{AM13}}])$, an input $x \in \{0,1\}^*$, and for every proof π, the adversary $\mathcal{A}^{\mathcal{H}}(\mathsf{pp}, f, x, \pi)$ outputs a distinct input x' with overwhelming probability $1 - \mathsf{negl}(\lambda)$. This result should satisfy the conditions that $\mathsf{QSet}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', f(x), \pi)$, $x \neq x'$, and $f(x) \neq f(x')$, where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and QSet is defined as per Definition 12.

In order for $\mathcal{A}$ to succeed, it must identify an input x' such that $f(x) \neq f(x')$, implying that x and x' contain different quantities of 1s, while ensuring that $\mathsf{QSet}_{\mathsf{pp}}(f, x, f(x), \pi) = \mathsf{QSet}_{\mathsf{pp}}(f, x', f(x), \pi)$. This requires that the queried bit b remain identical for both x and x'. However, in $\mathsf{digFS}[\Pi_{\mathrm{AM13}}]$, the selection of index i for querying b is naturally linked to $\delta = \mathcal{H}(x)$ (see Definition 12). Given that $x \neq x'$, the collision resistance property of the random oracle $\mathcal{H}$ guarantees that the probability of $\mathcal{H}(x)$ being equal to $\mathcal{H}(x')$ is noticeable, but this is far from being overwhelming, proving our statement.

For the case of Π_{CG15} we can proceed analogously as above observing that the queried position is determined completely by the challenges of the verifier. Here, again, it is easy to observe that a collision can only occur with probability $1/n$, which is not overwhelming. □

E.2 Proof of Theorem 5

We prove the result by showing that there exists a true-to-false-input adversary against Π_{AM13}. Let $G_{n,t} : \{0,1\}^n \to \{0,1\}$ be a threshold gate with positive threshold t. Notice that $G_{n,t}(x) = 1$ if and only if the Hamming weight (denoted by $\mathfrak{H}$) of x is at least t. Let $x \in \{0,1\}^n$ be an input such that $G_{n,t}(x) = 1$ and $n > t$. Such an x always exists; for instance, we can choose x to have exactly t ones and $n - t$ zeros. Let π be a Π_{AM13} proof for the claim $G_{n,t}(x) = 1$ where the prover's messages have been computed honestly. Let $\mathsf{QSet}(G_{n,t}, x, 1, \pi)$ be the query set for this claim given the proof π. We define two crucial quantities: u, the minimum number of bit flips (switching 1 to 0 in x) required to falsify the claim, and v, the maximum number of such bit flips that are possible without having to touch the indices in the query set. We can observe that $u = \mathfrak{H}(x) - t + 1$ and $v = n - |\mathsf{QSet}(G_{n,t}, x, 1, \pi)|$.

A true-to-false-input adversary succeeds if and only if $u \leq v$. This condition can be rewritten as $\mathfrak{H}(x) - t + 1 \leq n - |\mathsf{QSet}(G_{n,t}, x, 1, \pi)|$, or equivalently, $n - |\mathsf{QSet}(G_{n,t}, x, 1, \pi)| -$

$1 + (t - \mathfrak{H}(x)) \geq 0$. Given our choice of x with $\mathfrak{H}(x) = t$, this inequality simplifies to $|\mathsf{QSet}(G_{n,t}, x, 1, \pi)| \leq n - 1$. This inequality always holds in the AM13 protocol, as it queries at most $n - 1$ bits of the public input. Accordingly, we can construct a true-to-false-input adversary $\mathcal{A}$ as follows: On input $(G_{n,t}, x, \pi)$, $\mathcal{A}$ computes $\mathsf{QSet}(G_{n,t}, x, 1, \pi)$ and constructs x' by flipping u bits of x that are not in the query set. Such bits always exist due to the proven inequality. By construction, $G_{n,t}(x') = 0 \neq G_{n,t}(x)$, $x' \neq x$, and $\mathsf{QSet}(G_{n,t}, x, G_{n,t}(x), \pi) = \mathsf{QSet}(G_{n,t}, x', G_{n,t}(x), \pi)$. Therefore, $\mathcal{A}$ satisfies all conditions of a true-to-false-input adversary against the AM13 protocol. □

E.3 Proof of Theorem 8

At the high level, we build an adversary that is able to find a function for which the transcript through Fiat-Shamir is going to be "benign" for them (i.e., it is going to lead to a reward as high as in the case the output had been honest). The adversary is able to do this after fixing a target input x and that is intuitively why its digest is not going to help. We construct an adversary as follows :

- Let x be a string with half of its inputs 1s and the rest 0s, i.e., $x = 0^{n/2} \parallel 1^{n/2}$.
- Compute the digest for x, i.e., $\delta \leftarrow \mathcal{H}(x)$.
- Let the (false) claimed output be $y \leftarrow 1$.
- Define the formula C^{AND} as the one computing the n-ary AND through a tree of binary AND gates. Notice that the bottom layer of gates is composed of $n/2$ AND gates. For $i \in [n/2]$ denote by $C^{\mathsf{AND}}_{i \to \mathsf{OR}}$ the formula that is exactly like C^{AND} except that we replace the i-th AND gate in the bottom layer with an OR gate. Observe that, for $i \in [n/4]$ $C^{\mathsf{AND}}_{i \to \mathsf{OR}}(x) = 0$. This is because $C^{\mathsf{AND}}(x) = 0$ and changing the i-th gate to an OR in the "left half" of the tree ($i \in [n/4]$) will not change any of the internal wires.
- We iterate over some of the possible $C^{\mathsf{AND}}_{i \to \mathsf{OR}}$ formulas that output 0 on x, i.e. for $i \in [n/4]$:
  - Construct a proof π by at each round always providing $v_0, v_1 = (1, 1)$ (see Fig. 6) and computing the challenges as prescribed by $\mathsf{digFS}[\Pi_{\mathrm{CG15}}]$;
  - Let b_1 be the first (bit) challenge, that is $b_1 = \mathcal{H}(C^{\mathsf{AND}}_{i \to \mathsf{OR}}, \delta, y, (1, 1))$ where $\delta = \mathcal{H}(x)$;
  - If $b_1 = 1$ (i.e. if the protocol continues on the right side of the formula) then return $\left(C^{\mathsf{AND}}_{i \to \mathsf{OR}}, x, y, \pi\right)$;
- Return $\perp$ if all iterations above fail.

We first observe that, if $b_1 = 1$ in some proof π as above, then the adversary has "won": it will receive the full reward since the verifier will end up on the right side of the formula tree where the input wires will be consistent with the claimed outputs (also notice that by returning $(1, 1)$ at each level, the adversary ensures all the intermediate AND checks will be satisfied).

Changing one gate to an OR is for the adversary just a strategy to have completely fresh challenges (variations of this approach are possible). All that is left is to claim that this allows them to find a "good" cheating challenge b_1 with overwhelming probability. Let us bound the probability that this event does *not* happen. Now, this is the event:

$$\forall i \in \{1, \ldots, n/4\}\ \mathcal{H}(C^{\mathsf{AND}}_{i \to \mathsf{OR}}, \delta, y, (1, 1)) \neq 1$$

This probability of the event above is at most $2^{-n/4}$, which is negligible. Therefore, the claim follows.

We can finally observe that we built an *always* cheating adversary (as in *always returning a false output*) achieving in expectation $(1 - \mathsf{negl}(n)) \cdot R_{\text{hon}}$ where R_{hon} is the honest reward. This violates the noticeable loss required by Definition 3 and concludes the proof. □