Fiat–Shamir Bulletproofs are Non-Malleable (in the Algebraic Group Model)

Chaya Ganesh[1], Claudio Orlandi[2], Mahak Pancholi[2], Akira Takahashi[2], and Daniel Tschudi[3]

[1] Indian Institute of Science
chaya@iisc.ac.in
[2] Aarhus University
{orlandi,mahakp,takahashi}@cs.au.dk
[3] Concordium
dt@concordium.com

Abstract. Bulletproofs (Bünz et al. IEEE S&P 2018) are a celebrated ZK proof system that allows for short and efficient proofs, and have been implemented and deployed in several real-world systems. In practice, they are most often implemented in their *non-interactive* version obtained using the Fiat-Shamir transform, despite the lack of a formal proof of security for this setting.
Prior to this work, there was no evidence that *malleability attacks* were not possible against Fiat-Shamir Bulletproofs. Malleability attacks can lead to very severe vulnerabilities, as they allow an adversary to forge proofs re-using or modifying parts of the proofs provided by the honest parties.
In this paper, we show for the first time that Bulletproofs (or any other similar multi-round proof system satisfying some form of *weak unique response* property) achieve *simulation-extractability* in the *algebraic group model*.
This implies that Fiat-Shamir Bulletproofs are *non-malleable*.

Keywords: Non-interactive Zero-knowledge, Simulation-extractability, Fiat-Shamir

Table of Contents

1 Introduction

Zero-knowledge (ZK) proof systems [GMR85] are one of the most fascinating ideas in modern cryptography, as they allow a prover to persuade a verifier that some statement is true without revealing any other information. In recent years we have observed a new renaissance for ZK proofs, motivated in large part by their applications to advanced Blockchain applications. This has led, among other things, to a standardization effort for ZK proofs.[4]

A celebrated modern ZK proof system is Bulletproofs [BBB+18]. Bulletproofs offer transparent setup, short proofs and efficient verification (and it is therefore a *zero-knowledge succinct argument of knowledge* or zkSNARK) using only very well established computational assumptions, namely the hardness of discrete logarithms. At the heart of Bulletproofs lies an "inner product" component. This can be used then for general purpose proofs (i.e., where the statement is described as an arithmetic circuit) or for specific purpose proofs (i.e., range proofs, which are the most common use case in practice). Bulletproofs have been implemented in real world systems, especially for confidential transaction systems, like Monero, Mimblewimble, MobileCoin, Interstellar, etc.

Most practical applications of Bulletproofs utilize their non-interactive variant which, since Bulletproofs is a public-coin proof system, can be obtained using the Fiat-Shamir heuristic [FS87] e.g., the interaction with the verifier (who is only supposed to send uniformly random challenges) is replaced by interacting with a public hash function. Under the assumption that the hash function is a random oracle, one can hope that the prover has no easier time producing proofs for false statements (or for statements for which they do not know a witness) than when interacting with an actual verifier.

While the Fiat-Shamir heuristic has been around for decades, its formal analysis has only been performed much later. It is first in [FKMV12] that it was formally proven that the Fiat-Shamir heuristic is indeed sound. However, this proof only applies to classic Σ-protocols [CDS94], which are a special class of ZK protocols with only 3 moves. Therefore this analysis does not cover the case of Bulletproofs, which is a multi-round protocol.

For the case of Bulletproofs, it was first in [GT21], that it was shown that Fiat-Shamir Bulletproofs are indeed arguments of knowledge e.g., it is not possible for the prover to produce a valid proof without knowing a witness for the statement (a similar result, but with less tight bounds, appeared concurrently also in [BMM+19]). However, the results in [GT21] only consider a malicious prover "in isolation", whereas in most practical applications of Bulletproofs, several provers are producing and exchanging proofs at the same time (e.g., on a Blockchain).

The notion of *non-malleability* in cryptography was introduced in [DDN91], and the notion of non-malleability for zero-knowledge proofs was introduced in [Sah99]. In a nutshell, a malleability attack is one in which the adversary gets to see proofs from honest parties, and then modifies or re-uses parts of the proofs output by the honest parties to forge a proof on some statement for which they do not know a witness. Malleability attacks can have very serious consequences, such as the famous MtGox attack of 2014 [DW14].

Therefore, it is worrisome that Fiat-Shamir Bulletproofs have been implemented in the wild without any solid evidence that malleability attacks are not possible against them.

Luckily, in this paper we are able to show that Fiat-Shamir Bulletproofs satisfy a strong notion of *simulation-extractability* which in particular implies *non-malleability*. We do so in the *algebraic group model* (AGM) which is a model that only considers restricted classes of adversaries that, in a nutshell, output a group element $z \in \mathbb{G}$ together with its representations $[z]$ w.r.t. all elements they have seen so far. This is a limitation that our result shares with previous results in this area [BMM+19, GT21] that studied concrete knowledge-soundness of Fiat–Shamir Bulletproofs. [5]

1.1 Technical Overview

As already argued, in applications where proof systems are deployed, an adversary who tries to break the system has access to proofs provided by other parties using the same scheme. Thus, any reasonable security notion must require that a ZK proof system be secure against adversaries that potentially see and utilise proofs generated by different parties. *Simulation-soundness* (SIM-SND) and *simulation-extractability* (SIM-EXT) are the notions that guarantee soundness (the prover cannot prove false statements) or the

[4] https://zkproof.org

[5] A recent ePrint of Attema, Fehr and Klooß [AFK21] formally proved the *assumed* concrete knowledge error of Fiat–Shamir Bulletproofs in the standard (random oracle) model. As we discuss in Section 1.3 it is an interesting open question how our simulation-extractability analysis can be built on top of their results.

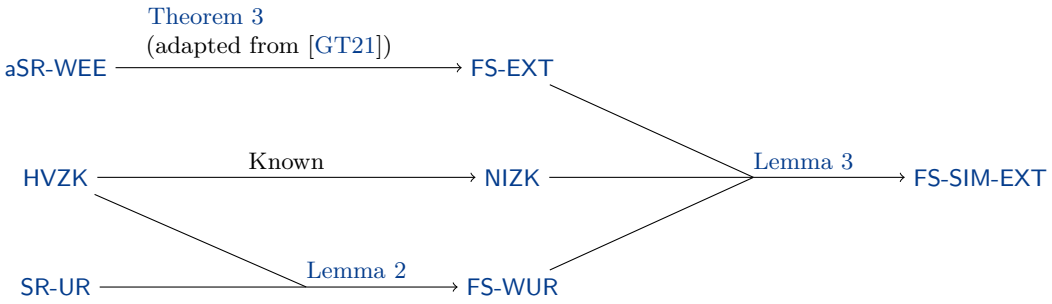

Fig. 1: Overview of modular security analysis towards simulation-extractability of multi-round Fiat–Shamir NIZK.

stronger property knowledge-soundness (the prover cannot prove statements without knowing a witness) to hold against adversaries who may see many (simulated) proofs.

Our starting point is the work of [GT21], that proves that the Fiat-Shamir transform of Bulletproofs (henceforth BP) is knowledge-sound in the AGM and random oracle (RO) model. They do this by first proving that the interactive version of BP satisfies a stronger property of state-restoration witness extended emulation (SR-WEE), where the prover is allowed to rewind the verifier a polynomial number of times (hence the name since the prover can "restore" the state of the verifier). They then turn this into a result for Fiat-Shamir BP by showing that for any adversary who breaks the knowledge-soundness of Fiat-Shamir BP, there exists an adversary for the SR-WEE property of the interactive BP.

The natural question is then, can their proof be easily extend to the case of SIM-EXT (where the result needs to hold even when the simulator has to provide the adversary with simulated proofs on statements of their choice)? To see why this is not straightforward, consider the following natural approach: just answer the proof queries of the adversary by running the honest verifier zero-knowledge simulator of BP, and then program the RO with the challenges returned by the simulator. The RO queries, on the other hand, are simply forwarded to the state-restoration oracle as before. This simple approach works if the underlying protocol satisfies "unique response", which informally means that the adversary cannot generate two distinct accepting transcripts that share a common prefix. (This notion has already been used to prove simulation-extractability of Σ-protocols [FKMV12], multi-round public coin interactive protocols [DFM20, KZ21], and Sonic and Plonk [KZ21]). However, BP does not have unique response under their definition: this is simple to see since randomized commitments are sent from the prover during the third round. Therefore, if the forged proof returned by the adversary has a matching prefix as one of the simulated proofs, this forged proof cannot be used to break SR-WEE.[6]

The next natural attempt might then be to "de-randomize" later rounds of BP e.g., by letting the prover choose and commit all their random coins in the first round, and then prove consistency of all future rounds with these coins. This of course introduces new challenges, since these additional consistency proofs must themselves not use any additional randomness in rounds other than the first one. While these technical challenges could be overcome using the right tools, the final solution would be all but satisfactory. First of all, the new protocol would be less efficient than the original BP. And perhaps more importantly, all real-world implementations of BP would have to decide whether to switch to the new protocol without any evidence that the original BP is insecure.

Instead, we present a new approach here that allows us to prove that Fiat-Shamir BP *as is* satisfies SIM-EXT, which has wide-reaching impact for systems based on BP that are already in use. The diagram in Fig. 1 summarizes our modular security analysis towards simulation-extractability of multi-round Fiat–Shamir NIZK. We discuss our new security notions and a chain of implications below.

Unique response. We introduce two new definitions: state-restoration unique response (SR-UR), and weak unique response (FS-WUR), which are the interactive, and non-interactive definitions for showing unique response of protocols. We show that these two notions are tightly related, i.e., FS-WUR tightly reduces to SR-UR of the interactive protocol (Lemma 2). Both notions require that it should be hard for the adversary, on input a simulated proof, to output a proof which shares a prefix with it. This is opposed to the previous notion of unique response that requires it should be infeasible for the adversary to come

[6] In a nutshell, this is because the forged proof may not be an accepting transcript in the SR-WEE game since the shared prefix is a partial transcript that has not been queried to the oracle before. Hence, the oracle has no knowledge of the simulated proofs and therefore any partial transcript that has a matching prefix with a simulated proof.

up with two different proofs that share a prefix. As an analogy, our notion is akin to second preimage resistance for hash functions, while the previous notion is akin to collision resistance. Clearly, it is easier to show that an existing protocol satisfies the weaker definition. But it is in turn harder to show that the weaker definition is enough to achieve the overall goal. However, note that the weaker variant of the definition is also somewhat closer to the intuitive goal of non-malleability: we do not want the adversary to be able to reuse parts of proofs generated by other parties to forge new proofs.

Simultation-extractability of multi-round Fiat–Shamir. Once we have FS-EXT (i.e., extractability), FS-WUR, and NIZK for a non-interactive protocol, we are able to show its online simulation-extractability (Lemma 3). Putting together, we prove a general theorem showing that:

Theorem 1 (General Theorem (Informal)). *If a multi-round public-coin interactive protocol satisfies: (1)* adaptive *state-restoration witness extended emulation (aSR-WEE), (2) perfect HVZK with an algebraic simulator, and (3) state-restoration unique responses (SR-UR), then the non-interactive version of the protocol achieved via the Fiat-Shamir transform, is online simulation-extractable (FS-SIM-EXT) in the algebraic group model and the random oracle model.*

While our framework has been built with Bulletproofs as its main use case, we believe that it is general enough and could be used to show simulation-extractability for other public-coin protocols in the literature.

Non-malleable Bulletproofs. We use our definitional foundation to show that Fiat-Shamir BP is non-malleable and give concrete security bounds for it. The main technical contribution here is to show that BP satisfies our (weaker) definition of unique response, namely SR-UR. For the other assumptions in the theorem, we rely on existing knowledge with some adjustments: BP is already known to satisfy SR-WEE (from [GT21]), however in our theorem we require a stronger (adaptive) version of the definition, namely aSR-WEE, but it turns out that the proof of SR-WEE in [GT21] can be used to show the stronger definition as well. Finally, BP is already known to admit a perfect HVZK simulator, which we have to extend to the algebraic setting. Thus, using the general theorem, we get our result. We do this for two versions of BP, namely Bulletproofs for arithmetic circuits (in Section 4) and range-proofs Bulletproofs (in Appendix D).

1.2 Related work

Goldwasser and Kalai [GK03] show that the Fiat-Shamir heuristic is not sound in general, by showing explicit – and somewhat contrived – counterexamples that cannot be proven secure for any hash function. However, there is no evidence that any *natural* construction using the Fiat-Shamir heuristic is insecure.

Faust et al. [FKMV12] are the first to analyze SIM-SND and SIM-EXT of Fiat–Shamir NIZK from Σ-protocols. Kohlweiss and Zając [KZ21] extend their result to multi-round protocols with $(n_1, \ldots, n_r)$-special soundness where all-but-one n_i's are equal to 1, which is the case for some modern zkSNARKs (cf. [MBKM19, GWC19]), but is not the case for Bulletproofs-style recursive protocols.

Don et al. [DFM20] study multi-round Fiat–Shamir in the quantum random oracle model, but their generic claim (Corollary 15) incurs at least a multiplicative factor $O(q^r)$[7] in the loss in soundness due to Fiat–Shamir, even if the result is downgraded to the classical setting. Hence their result leads to a super-polynomial loss when the number of rounds r depends on the security parameter as in Bulletproofs. They also showed SIM-EXT of multi-round Fiat–Shamir proofs in the QROM assuming the unique response property of the underlying interactive protocols. As we shall see later, Bulletproofs do not meet their definition of unique responses and we are thus motivated to explore alternative paths towards SIM-EXT, but in the classical ROM and the AGM.

There are a limited number of works that analyze the concrete soundness loss incurred by Fiat–Shamir when applied to *non-constant round* protocols. Ben-Sasson et al. [BCS16] show that if the underlying *interactive oracle proof* protocol satisfies *state-restoration soundness* (SR-SND) (a stronger variant of soundness where the prover is allowed to rewind the verifier states) then Fiat–Shamir only introduces $3(q^2+1)2^{-\lambda}$ of additive loss both in soundness (SND) and proof of knowledge (EXT). Canetti et al. [CCH+18, CCH+19] propose the closely related notion of *round-by-round soundness* (RBR-SND) which is sufficient to achieve soundness, even without round oracles. Following these works, Holmgren [Hol19] shows SR-SND and RBR-SND are equivalent.

[7] Here and below q is the number of queries to the random oracle, r is the number of rounds, and λ is the security parameter.

The latest works on this line of research are due to Ghoshal and Tessaro [GT21] and Bünz et al. [BMM+19]. They both provide a detailed analysis of *non-interactive* Bulletproofs in the algebraic group model (AGM) [FKL18] and, in particular, the former shows *state-restoration witness extended emulation* (SR-WEE) of interactive Bulletproofs in the AGM and uses it to argue that EXT of non-interactive Bulletproofs results in $(q+1)/2^{\mathsf{sLen}(\lambda)}$ in additive loss, where $\mathsf{sLen}(\lambda)$ is the bit length of the shortest challenge. However, none of these works explore SIM-SND or SIM-EXT of non-constant round Fiat–Shamir.

There are also other zkSNARKs that satisfy simulation-extractability such as e.g., [GM17] and [Gro16, KZ21]. However, these constructions are very different than Bulletproofs since they rely on a structured reference string which comes with a trapdoor, the knowledge of which compromises the soundness. [BKSV20] show techniques to make [Gro16] black-box weakly SIM-EXT NIZK using verifiable encryption. A generic framework to turn existing zkSNARKs into SIM-EXT zkSNARKs was presented in [ARS20], but Bulletproofs is not covered by their result since their transform only works for schemes with trusted setup.

1.3 Open question

Concurrent works due to Wikström [Wik21] and Attema, Fehr and Klooß [AFK21] show that, assuming an $(n_1, \ldots, n_r)$-special sound interactive protocol Π, one can relate the knowledge soundness of Π to that of Π_{FS} without the AGM, while retaining only $q+1$ multiplicative loss in the knowledge error. Although our generic chain of reductions in Section 3 is tailored to the AGM in order to invoke the results of [GT21], we expect a large body of the analysis and definitions may be reusable without assuming the AGM. In fact, our Lemma 2 and Lemma 3 even hold without the AGM because these reductions do not exploit the fact that adversaries are algebraic—what is open is whether Bulletproofs satisfy the *assumptions* required by these lemmas (i.e., SR-UR and FS-WUR) without the AGM. Hence it is an interesting open question whether a chain of implications similar to Fig. 1 can be built on top of [Wik21, AFK21] to prove SIM-EXT of Bulletproofs only in the ROM.

2 Preliminaries

We denote by $\mathbb{N} = \{0, 1, 2, \ldots\}$ the natural numbers, and $\mathbb{Z}$ the integers. We use bold face to denote vectors, e.g., $\mathbf{a} \in \mathbb{Z}^n$ is a vector of n integers. For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$ we write $\langle \mathbf{a}, \mathbf{b} \rangle$ for the inner product. For a group $\mathbb{G}$ where $(g_1, \ldots, g_n) = \mathbf{g} \in \mathbb{G}^n$ and $(a_1, \ldots, a_n) = \mathbf{a} \in \mathbb{Z}^n$, we denote by $\mathbf{g}^{\mathbf{a}} = \prod_{i=1}^{n} g_i^{a_i}$. The Hadamard product of vectors $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ is $\mathbf{a} \circ \mathbf{b} = (a_1 b_1, \ldots, a_n b_n)$. For $z \in \mathbb{Z}_p^*$, denote $\mathbf{z}^n$ as the vector $(1, z, \ldots, z^{n-1})$, and $\mathbf{z}^{-n} = (1, z^{-1}, \ldots, z^{-n+1})$.

The security parameter λ is 1^λ in unary. We use $\mathsf{negl}(\lambda)$ to denote a negligible function in λ. We write $X_\lambda \equiv Y_\lambda$ to denote that probabilistic ensembles $\{X_\lambda\}_\lambda$ and $\{Y_\lambda\}_\lambda$ are equivalent.

Lemma 1 (Schwartz-Zippel Lemma). *Let $\mathbb{F}$ be a finite field and let $f \in \mathbb{F}[X_1, \ldots, X_n]$ be a non-zero n variate polynomial with maximum degree d. Let S be a subset of $\mathbb{F}$. Then $\Pr[f(x_1, \ldots, x_n) = 0] \leq d/|S|$, where the probability is taken over the choice of $x_1, \ldots, x_n$ according to $x_i \xleftarrow{\$} S$.*

For a univariate polynomial $f \in \mathbb{Z}_p[X]$ of degree d, where p is prime, let $x \xleftarrow{\$} \mathbb{Z}_p^*$, then $\Pr[f(x) = 0] \leq d/(p-1)$.

2.1 Discrete Logarithm Problems

We follow the formulations of [GT21]. Let $\{\mathbb{G}_\lambda\}_\lambda$ be a family of groups with prime order $p(\lambda)$, identity $e_{\mathbb{G}_\lambda}$ and generators $\mathbb{G}_\lambda^* = \mathbb{G}_\lambda \setminus \{e_{\mathbb{G}_\lambda}\}$. We can use game G^{DL} in Figure 2 to define the advantage of an adversary in the discrete logartihm problem over $\{\mathbb{G}_\lambda\}_\lambda$.

Definition 1 (DL). *The advantage of a non-uniform adversary $\{\mathcal{A}_\lambda\}_\lambda$ against the* discrete logaritm problem *in $\{\mathbb{G}_\lambda\}_\lambda$ is*

$$\mathbf{Adv}^{DL}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) = \Pr\left[\mathsf{G}^{DL}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) = 1\right].$$

Similarly, we can define the advantage for the discrete logarithm relation problem using game $\mathsf{G}_n^{\mathsf{DL\text{-}REL}}$ in Figure 2.

Definition 2 (DL-REL). *The advantage of a non-uniform adversary $\{\mathcal{A}_\lambda\}_\lambda$ against the* discrete logaritm relation problem *in $\{\mathbb{G}_\lambda\}_\lambda$ is*

$$\mathbf{Adv}^{DL\text{-}REL}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) = \Pr\left[\mathsf{G}_n^{DL\text{-}REL}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) = 1\right].$$

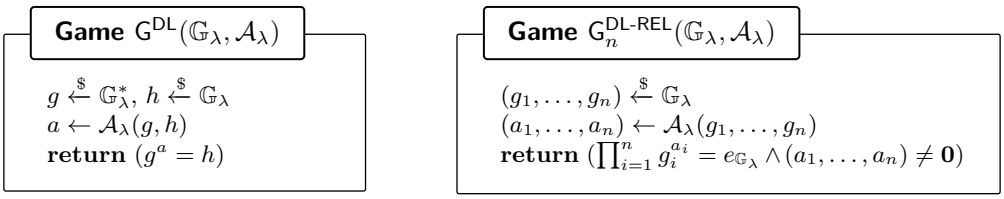

Fig. 2: The games for the discrete logarithm problem and the discrete logarithm relation problem. Here $\{\mathcal{A}_\lambda\}_\lambda$ is a non-uniform adversary, $\{\mathbb{G}_\lambda\}_\lambda$ a family of cyclic groups where $\mathbb{G}_\lambda^*$ denotes the set of generators of $\mathbb{G}_\lambda$ and $e_{\mathbb{G}_\lambda}$ is the identity.

2.2 Relations

Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be an efficiently decidable relation. For given public parameters pp, we call w a *witness* for *statement* x if $(\mathsf{pp}, x, w) \in \mathcal{R}$. The associated language for pp is $\mathcal{L}_{\mathsf{pp}} = \{x \mid \exists w \text{ s.t. } (\mathsf{pp}, x, w) \in \mathcal{R}\}$.

Relation for inner product argument. The relation proven by the inner-product argument in [BBB+18] is

$$\begin{aligned}\mathcal{R}_{\mathsf{InPrd}} &= \left\{((n, \mathbf{g}, \mathbf{h}, u), (P), (\mathbf{a}, \mathbf{b})) \mid P = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}}u^c \wedge c = \langle \mathbf{a}, \mathbf{b}\rangle\right\} \\ &\subseteq (\mathbb{N} \times \mathbb{G}^n \times \mathbb{G}^n \times G) \times (\mathbb{G}) \times (\mathbb{Z}_p^n \times \mathbb{Z}_p^n)\end{aligned} \tag{1}$$

where $\mathbf{g}, \mathbf{h}$ are vectors of independent generators.

Relation for arithmetic circuit satisfiability. Any arithmetic circuit with n multiplication gates can be represented using a constraint system (cf. [BCCJG16]). The constraint system consists of three vectors $\mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O \in \mathbb{Z}_p^n$ representing the left inputs, right inputs, and outputs of multiplication gates respectively, so that $\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O$, and additional $Q \leq 2n$ linear constraints. The linear constraints can be represented as $\mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_O = \mathbf{c}$, where $\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O \in \mathbb{Z}_p^{Q \times n}$ and $\mathbf{c} \in \mathbb{Z}_p^Q$. This results in the following relation for arithmetic circuit satisfiability

$$\begin{aligned}\mathcal{R}_{\mathsf{ACS}} &= \left\{ \begin{array}{l} ((n, Q), (\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O, \mathbf{c}), (\mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O)) \mid \\ \mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O \wedge \mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_O = \mathbf{c} \end{array} \right\} \\ &\subseteq (\mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^Q) \times (\mathbb{Z}_p^n \times \mathbb{Z}_p^n \times \mathbb{Z}_p^n)\end{aligned}$$

For simplicity, we also include the Bulletproofs setup resulting in the relation

$$\begin{aligned}\mathcal{R}_{\mathsf{ACSPf}} &= \left\{ \begin{array}{l} ((n, Q, g, h, u, \mathbf{g}, \mathbf{h}), (\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O, \mathbf{c}), (\mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O)) \mid \\ \mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O \wedge \mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_O = \mathbf{c} \end{array} \right\} \\ &\subseteq (\mathbb{Z}^2 \times \mathbb{G}^3 \times \mathbb{G}^n \times \mathbb{G}^n) \times (\mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^{Q \times n} \times \mathbb{Z}_p^Q) \times (\mathbb{Z}_p^n \times \mathbb{Z}_p^n \times \mathbb{Z}_p^n)\end{aligned} \tag{2}$$

where $\mathbb{G} = \{\mathbb{G}_\lambda\}_{\lambda \in \mathbb{N}}$ is a family of groups of order $p = p(\lambda)$.

2.3 Interactive Proofs

An *interactive proof* [GMR85] is a tuple of three algorithms $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$. On input 1^λ, the setup algorithm Setup produces public parameters pp, also known as common reference string. The transcript of the interaction between prover $\mathcal{P}$ with input x and PPT verifier $\mathcal{V}$ with input y is denoted by $\mathsf{tr} \leftarrow \langle \mathcal{P}(\mathsf{pp}, x)\,,\, \mathcal{V}(\mathsf{pp}, y)\rangle$. We write $b = \langle \mathcal{P}(\mathsf{pp}, x)\,,\, \mathcal{V}(\mathsf{pp}, y)\rangle$ for the decision bit of the verifier, i.e., $b = 1$ means the verifier accepts. An interactive proof where the prover is PPT is also known as *argument*.

Intuitively an interactive proof for relation $\mathcal{R}$ allows a prover knowing a witness w to convince the verifier of a statement x. Given $(\mathsf{pp}, x, w) \in \mathcal{R}$ an honest prover should always convince the verifier (completeness) while a prover should not be able to convince the verifier of x if x is not true (soundness), nor without 'knowing' w such that $(\mathsf{pp}, x, w) \in \mathcal{R}$ (proof of knowledge). Finally, the proof should not reveal any more information than the validity of the statement (zero-knowledge).

Definition 3 (Public Coin). *An interactive proof $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ is* public coin *if all messages sent by the verifier $\mathcal{V}$ during the interaction with $\mathcal{P}$ in round $2i$ are chosen uniform-randomly from the challenge space Ch_i (where the challenge spaces Ch_i are specified by Setup). In particular, they are independent of the provers messages.*

Definition 4 (Completeness). *An interactive proof* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *satisfies* perfect completeness *if for all* λ *and for all PPT adversaries* $\mathcal{A}$ *it holds that*

$$\Pr\left[1 = \langle \mathcal{P}(\mathsf{pp}, x, w)\,,\, \mathcal{V}(\mathsf{pp}, x)\rangle \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (x, w) \leftarrow \mathcal{A}(\mathsf{pp}) \text{ s.t. } (\mathsf{pp}, x, w) \in \mathcal{R} \end{array}\right] = 1.$$

Definition 5 (Perfect Honest-Verifier Zero-Knowledge). *A public coin argument of knowledge* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for relation* $\mathcal{R}$ *is perfect* honest verifier zero-knowledge *(HVZK) if there exists a PPT simulator* $\mathcal{S}$ *such that for all PPT adversaries* $\mathcal{A}$ *it holds that*

$$\langle \mathcal{P}(\mathsf{pp}, x, w)\,,\, \mathcal{V}(\mathsf{pp}, x)\rangle \equiv \mathcal{S}(\mathsf{pp}, x)$$

where $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(x, w) \leftarrow \mathcal{A}(\mathsf{pp})$ *such that* $(\mathsf{pp}, x, w) \in \mathcal{R}$.

Min-entropy of commitments We want to assess how likely it is for first round messages of an interactive protocol to collide with a fixed value.

Definition 6 (Min-entropy of commitments). *Let* λ *be a security parameter and* $\mathcal{L}_{\mathsf{pp}}$ *be an NP-language with relation* $\mathcal{R}$. *Consider a pair* $(\mathsf{pp}, x, w) \in \mathcal{R}$ *and let* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *be an arbitrary multi-round interactive proof. Let* $\mathsf{Coin}(\lambda)$ *be the set of coins used by the prover and* $\mathcal{P}_1(\cdot)$ *the function that outputs the first round message for* $\mathcal{P}$. *Consider the set*

$$\mathsf{msg}(\mathsf{pp}, x, w) = \{\Pi.\mathcal{P}_1(\mathsf{Setup}, x, w; \rho \xleftarrow{\$} \mathsf{Coin}(\lambda))\}$$

, i.e. the set of all possible first round messages associated to w. *The min-entropy function associated to* Π *is defined as* $\epsilon(\lambda) = \mathsf{min}_{(\mathsf{pp},x,w)}(-\log_2 \mu(\mathsf{pp}, x, w))$, *where the minimum is taken over all possible* (pp, x, w) *drawn from* $\mathcal{R}$, *and* $\mu(\mathsf{pp}, x, w)$ *is the maximum probability that a first round message takes on a particular value, i.e.,*

$$\mu(\mathsf{pp}, x, w) = \mathsf{max}_{m \in \mathsf{msg}(\mathsf{pp},x,w)} \left(\Pr\left[\Pi.\mathcal{P}_1(\mathsf{pp}, x, w; \rho) = m | \rho \xleftarrow{\$} \mathsf{Coin}\right]\right)$$

2.4 Non-Interactive Proofs via Fiat–Shamir

In this paper we focus on *non-interactive arguments* obtained via the Fiat–Shamir heuristic [FS87], which allows to transform an interactive proof into an non-interactive version in the random oracle model. Given an interactive proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ and random oracles $\mathsf{H} = \{\mathsf{H}_i\}_{i\in[1,r]}$ with $\mathsf{H}_i : \{0,1\}^* \to \mathsf{Ch}_i$, we define the corresponding non-interactive argument $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ as follows. We write $\mathcal{P}^{\mathsf{H}}_{\mathsf{FS}}$ and $\mathcal{V}^{\mathsf{H}}_{\mathsf{FS}}$ to denote that the prover and verifier have access to the random oracles H.

- We denote by $\pi \leftarrow \mathcal{P}^{\mathsf{H}}_{\mathsf{FS}}(\mathsf{pp}, x, w)$ the prover producing argument π on input pp, statement x, and witness w. For each round $i \in [1, r]$, $\mathcal{P}^{\mathsf{H}}_{\mathsf{FS}}$ invokes $\mathcal{P}(\mathsf{pp}, x, w, \mathsf{st}_{i-1}, c_{i-1})$ to get a_i and obtains the ith round challenge by computing $c_i \leftarrow \mathsf{H}_i(\mathsf{pp}, x, a_1, c_1, \ldots, c_{i-1}, a_i)$. Then $\mathcal{P}^{\mathsf{H}}_{\mathsf{FS}}$ outputs $\pi = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$ as a proof.
- We write $b \leftarrow \mathcal{V}^{\mathsf{H}}_{\mathsf{FS}}(\mathsf{pp}, x, \pi)$ for the decision bit of the verifier on input of pp, statement x, and argument π. $\mathcal{V}^{\mathsf{H}}_{\mathsf{FS}}$ outputs $b = 1$, meaning the verifier accepts the argument, iff $\mathcal{V}(\mathsf{pp}, x, \pi) = 1$ and $c_i = \mathsf{H}_i(\mathsf{pp}, x, a_1, c_1, \ldots, c_{i-1}, a_i)$ holds for all $i \in [1, r]$.

Definition 7 (Completeness). *An non-interactive argument* $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ *for relation* $\mathcal{R}$ *satisfies* perfect completeness *if for all* λ *and for all PPT adversaries* $\mathcal{A}$ *it holds that*

$$\Pr\left[1 \leftarrow \mathcal{V}^{\mathsf{H}}_{\mathsf{FS}}(\mathsf{pp}, x, \pi) \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \\ (x, w) \leftarrow \mathcal{A}(\mathsf{pp}) \text{ s.t. } (\mathsf{pp}, x, w) \in \mathcal{R} \\ \pi \leftarrow \mathcal{P}^{\mathsf{H}}_{\mathsf{FS}}(\mathsf{pp}, x, w) \end{array}\right] = 1.$$

2.5 The Algebraic Group Model

The algebraic group model was introduced in [FKL18]. An adversary $\mathcal{A}_{\mathsf{alg}}$ is called *algebraic* if every group element output by $\mathcal{A}_{\mathsf{alg}}$ is accompanied by a representation of that group element in terms of all the group elements that $\mathcal{A}_{\mathsf{alg}}$ has seen so far (input and output). Let $y_1, \ldots, y_k$ be all the group elements previously input and output by $\mathcal{A}_{\mathsf{alg}}$. Then, every group element y output by $\mathcal{A}_{\mathsf{alg}}$, is accompanied by its representation $(x_1, \ldots, x_k)$ such that $y = \prod_{i=1}^{k} y_i^{x_i}$. Following [FKL18], we write $[y]$ to denote a group element enhanced with its representation; $[y] = (y, x_1, \ldots, x_k)$.

2.6 Adaptive state-restoration witness extended emulation

Here we define an *adaptive* variant of *state-restoration witness extended emulation* (aSR-WEE) defined in [GT21]. Intuitively, *state-restoration witness extended emulation* says that having resettable access to the verifier (or "restoring its state", hence the name) should not help a malicious prover in producing a valid proof without knowing a witness for the statement. Formally, the definition consists of two games denoted as $\mathsf{aWEE\text{-}1}_{\Pi}^{\mathcal{P}_{\mathsf{alg}},\mathcal{D}}$ and $\mathsf{aWEE\text{-}0}_{\Pi,\mathcal{R}}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}$ described in Figure 3. The former captures the real game, lets the prover $\mathcal{P}_{\mathsf{alg}}$ interact with an oracle $\mathbf{O}_{\mathsf{ext}}^{1}$, which additionally stores all queried transcripts tr. The latter is finally given to a distinguisher $\mathcal{D}$ which outputs a decision bit. In contrast, the ideal game delegates the role of answering $\mathcal{P}_{\mathsf{alg}}$'s oracle queries to a (stateful) extractor $\mathcal{E}$. The extractor, at the end of the execution, also outputs a witness candidate w. Due to the adaptive nature of our variant, we also need to redefine the predicate $\mathsf{Acc}()$ so that it accepts a pair $(x^*, \mathcal{T}^*)$ output by the adversary at the end if and only if the pair exists in the execution paths and it gets accepted by the verifier. Formally, $\mathsf{Acc}(\mathsf{tr}, x^*, \mathcal{T}^*)$ now outputs 1 if $(x^*, \mathcal{T}^*) \in \mathsf{tr}$ and $\mathcal{V}(\mathsf{pp}, x^*, \mathcal{T}^*) = 1$, and outputs 0 otherwise. For an interactive proof $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ and an associated relation $\mathcal{R}$, non-uniform algebraic prover $\mathcal{P}_{\mathsf{alg}}$, a distinguisher $\mathcal{D}$, and an extractor $\mathcal{E}$ we define:

$$\mathbf{Adv}_{\Pi,\mathcal{R}}^{\text{aSR-WEE}}(\mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}, \lambda) := \left| \Pr\left[\mathsf{aWEE\text{-}1}_{\Pi}^{\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)\right] - \Pr\left[\mathsf{aWEE\text{-}0}_{\Pi,\mathcal{R}}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)\right] \right|. \tag{3}$$

Definition 8 (aSR-WEE security). *An interactive proof $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ is online aSR-WEE secure if there exists an efficient $\mathcal{E}$ such that for any (non-uniform algebraic) $\mathcal{P}_{\mathsf{alg}}$ and for any distinguisher $\mathcal{D}$, $\mathbf{Adv}_{\Pi,\mathcal{R}}^{\text{aSR-WEE}}(\mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}, \lambda)$ is negligible in λ.*

The main difference with the original definition in [GT21] is that we allow the adversary to change the statement associated with a transcript in every query, whereas [GT21] forces the adversary to commit to the fixed statement x in advance. We remark that their results about Bulletproofs still hold under this variant, because nowhere in the proof do they actually exploit the fact that the statement is fixed. Hence, the following is immediate from [GT21]. We provide more details on this in Appendix A for completeness.

Theorem 2 (Adapted from Theorem 6 of [GT21]). *The protocol* BP *is aSR-WEE secure.*

2.7 NIZK and Simulation Oracles

We define zero-knowledge for non-interactive arguments in the explicitly programmable random oracle model where the simulator can program the random oracle. The formalization below can be seen as that of [FKMV12] adapted to multi-round protocols. The zero-knowledge simulator $\mathcal{S}_{\mathsf{FS}}$ is defined as a stateful algorithm that operates in two modes. In the first mode, $(c_i, st') \leftarrow \mathcal{S}_{\mathsf{FS}}(1, st, t, i)$ takes care of random oracle calls to H_i on input t. In the second mode, $(\widetilde{\mathcal{T}}, st') \leftarrow \mathcal{S}_{\mathsf{FS}}(2, st, x)$ simulates the actual argument. For convenience we define three "wrapper" oracles. These oracles are stateful and share state.

- $\mathcal{S}_1(t, i)$ to denote the oracle that returns the first output of $\mathcal{S}_{\mathsf{FS}}(1, st, t, i)$;
- $\mathcal{S}_2(x, w)$ that returns the first output of $\mathcal{S}_{\mathsf{FS}}(2, st, x)$ if $(\mathsf{pp}, x, w) \in \mathcal{R}$ and $\perp$ otherwise;
- $\mathcal{S}_2'(x)$ that returns the first output of $\mathcal{S}_{\mathsf{FS}}(2, st, x)$.

Since NIZK is a security property that is only guaranteed for valid statements in the language, the definition below makes use of $\mathcal{S}_2$ as a proof simulation oracle. As we shall see later, *simulation-extractability* on the other hand is defined with respect to an oracle similar to $\mathcal{S}_2'$ following [FKMV12].

Definition 9 (Non-interactive Zero Knowledge). *Let $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}^{\mathsf{H}}, \mathcal{V}_{\mathsf{FS}}^{\mathsf{H}})$ be a non-interactive argument for relation $\mathcal{R}$. Π_{FS} is unbounded* non-interactive zero knowledge *(NIZK) in the random oracle model, if there exist a PPT simulator $\mathcal{S}_{\mathsf{FS}}$ with wrapper oracles $\mathcal{S}_1$ and $\mathcal{S}_2$ such that for all PPT distinguisher $\mathcal{D}$ there exist a negligible function $\mu(\lambda)$ it holds that*

$$\left| \Pr\left[\mathcal{D}^{\mathsf{H},\mathcal{P}_{\mathsf{FS}}^{\mathsf{H}}}(1^\lambda) = 1\right] - \Pr\left[\mathcal{D}^{\mathcal{S}_1,\mathcal{S}_2}(1^\lambda)\right] \right| \leq \mu(\lambda)$$

where both $\mathcal{P}_{\mathsf{FS}}^{\mathsf{H}}(\mathsf{pp}, x, w)$ and $\mathcal{S}_2$ return $\perp$ if $(\mathsf{pp}, x, w) \notin \mathcal{R}$.

Given a perfect HVZK simulator $\mathcal{S}$ for Π, we immediately obtain the following *canonical* NIZK simulator $\mathcal{S}_{\mathsf{FS}}$ for Π_{FS} by defining responses of each mode as follows.

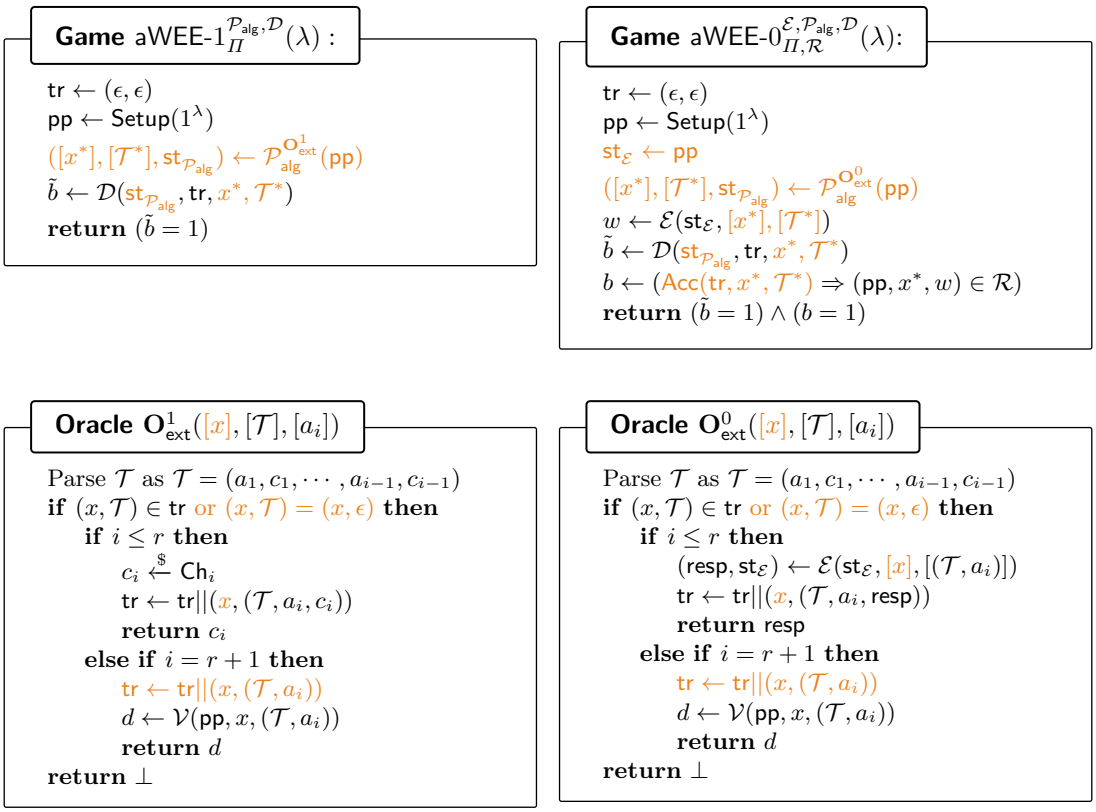

Fig. 3: Online aSR-WEE Security (adapted from [GT21], with differences highlighted in orange).

- To answer query (t, i) with mode 1, $\mathcal{S}_{\mathsf{FS}}(1, \mathsf{st}, t, i)$ lazily samples a lookup table $\mathcal{Q}_{1,i}$ kept in state st. It checks whether $\mathcal{Q}_{1,i}[t]$ is already defined. If this is the case, it returns the previously assigned value; otherwise it returns and sets a fresh random value c_i sampled from Ch_i.
- To answer query x with mode 2, $\mathcal{S}_{\mathsf{FS}}(2, \mathsf{st}, x)$ calls the perfect HVZK simulator $\mathcal{S}$ of Π to obtain a simulated proof $\pi = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$. Then, it programs the tables such that $\mathcal{Q}_{1,1}[x, a_1] := c_1, \ldots, \mathcal{Q}_{1,r}[x, a_1, c_1, \ldots, a_r] := c_r$. If any of the table entries has been already defined $\mathcal{S}_{\mathsf{FS}}$ aborts, which should happen with negligible probability assuming high min-entropy of a_1.

2.8 Online Extractability in the AGM

We introduce the definition of (adaptive) *online extractability* (FS-EXT) in the AGM. Unlike the usual online extraction scenario (e.g., [Pas03, Fis05, Unr15]), where an extractor is only given x^*, $\mathcal{T}^*$ and the random oracle query history as inputs and asked to extract the witness, our definition below requires the extractor to intercept/program the queries/answers to the RO for $\mathcal{P}_{\mathsf{alg}}$. We do so because some proofs in [GT21] (such as Theorem 2 and 3) relating state-restoration witness-extended emulation for Π and argument of knowledge for Π_{FS} do appear to exploit this extra power of the extractor, which to the best of our understanding appears necessary for their proofs to go through.

This modification in turn requires the existence of an extractor $(\mathcal{E}_0, \mathcal{E}_1)$ where $\mathcal{E}_1$ takes care of simulating the RO responses for $\mathcal{P}_{\mathsf{alg}}$ and then $\mathcal{E}_0$ produces a valid witness given an adversarial forgery. Our formalization therefore follows variants of extractability in the literature that explicitly introduce a distinguisher to guarantee the validity of simulation conducted by $\mathcal{E}_1$, e.g., [Unr17, Def. 11] for Fiat–Shamir NIZK or [Gro06] for CRS-based NIZK. On the other hand, we do not grant the extractor an oracle access to $\mathcal{P}_{\mathsf{alg}}$ to explicitly capture the "online" nature of extraction, i.e., no rewinding step is required.

Note also that the roles of $(\mathcal{E}_0, \mathcal{E}_1)$ and $\mathcal{D}$ below are also analogous to those of the extractor and the distinguisher in aSR-WEE. Thus, our definition allows smooth transition from aSR-WEE to FS-EXT.

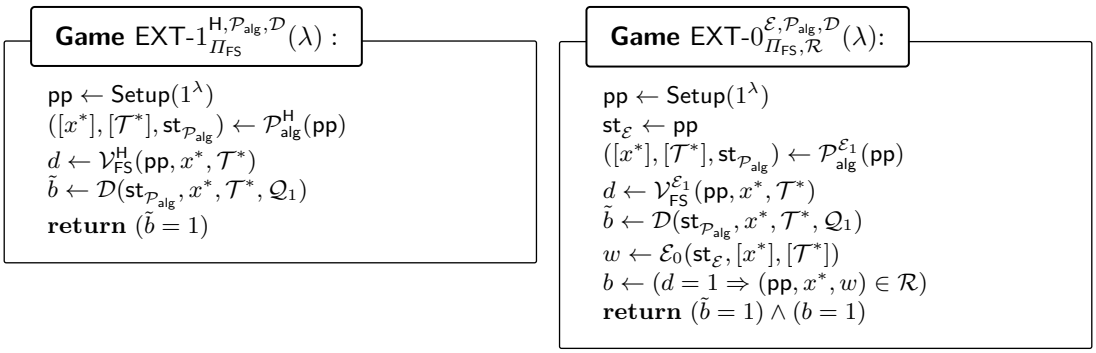

Fig. 4: Extractability games. Note that in the EXT-1 experiment, calling the verification algorithm $\mathcal{V}_{\mathsf{FS}}$ has an impact on the RO query set $\mathcal{Q}_1$. In particular, omitting this, there might be trivial distinguishing attacks due to the differences in $\mathcal{Q}_1$ between EXT-1 and EXT-0.

Definition 10 (FS-EXT security). *Let $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ be a NIZK scheme for language $\mathcal{L}$. Let H be a random oracle. Π_{FS} is* online extractable *(FS-EXT) in the AGM and the ROM if there exists an efficient extractor $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1)$ such that for every PPT algebraic adversary $\mathcal{P}_{\mathsf{alg}}$ and every distinguisher $\mathcal{D}$, the following probability is negligible in λ:*

$$\mathbf{Adv}^{\textit{FS-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathsf{H},\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D},\lambda) := \left|\Pr[\textit{EXT-1}^{\mathsf{H},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}}}(\lambda)] - \Pr[\textit{EXT-0}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\lambda)]\right|.$$

In Fig. 4, each of $\mathcal{Q}_1 = \{\mathcal{Q}_{1,i}\}_{i\in[1,r]}$ is a set of query response pairs corresponding to queries to H or $\mathcal{E}_1$ with random oracle index i.

We recall a relation between aSR-WEE and FS-EXT, because one of our claims (Lemma 3) uses FS-EXT as an assumption. Although Theorem 2 of [GT21] is for non-adaptive variants of these notions, the proof for the following theorem is almost identical except that we do not ask $\mathcal{P}^*_{\mathsf{alg}}$ to submit the statement x in the beginning, just like in Theorem 2.

Theorem 3. *Let $\mathcal{R}$ be a relation. Let Π be a r-challenge public coin interactive protocol for the relation $\mathcal{R}$ where the ith challenge is sampled from Ch_i for $i \in [1,r]$. Let $\mathcal{E}$ be an aSR-WEE extractor for Π. There exists an FS-EXT extractor $\mathcal{E}^* = (\mathcal{E}^*_0, \mathcal{E}^*_1)$ for Π_{FS} such that for every non-uniform algebraic prover $\mathcal{P}^*_{\mathsf{alg}}$ against Π_{FS} that makes q random oracle queries, and for every distinguisher $\mathcal{D}^*$, there exists a non-uniform algebraic prover $\mathcal{P}_{\mathsf{alg}}$ and a distinguisher $\mathcal{D}$ such that for all $\lambda \in \mathbb{N}^+$,*

$$\mathbf{Adv}^{\textit{FS-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathsf{H},\mathcal{E}^*,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*,\lambda) \leq \mathbf{Adv}^{\textit{aSR-WEE}}_{\Pi,\mathcal{R}}(\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D},\lambda) + (q+1)/|\mathsf{Ch}_{i_0}|$$

where $i_0 \in [1,r]$ is the round with the smallest challenge set Ch_{i_0}. Moreover, $\mathcal{P}_{\mathsf{alg}}$ makes at most q queries to its oracle and is nearly as efficient as $\mathcal{P}^*_{\mathsf{alg}}$. The extractor $\mathcal{E}^*$ is nearly as efficient as $\mathcal{E}$.

For completeness, we sketch the proof in Appendix B.

3 Simulation-Extractability from State-Restoration Unique Response

Our results make use of the concrete security proof of extractability for Bulletproofs given by [GT21] in the algebraic group model. Thus, the first step towards proving simulation-extractability for Bulletproofs is to provide a formal definition of simulation-extractability in the algebraic group model, which has not previously appeared in the literature.

3.1 Simulation-Extractability in the AGM

On a high-level, the simulation-extractability (SIM-EXT) property ensures that extractability holds even if the cheating adversary sees simulated proofs. Defining SIM-EXT in the AGM is a non-trivial task: because the algebraic adversary outputs group representation *with respect to all the group elements they have observed so far*, the format of representation gets complex as the adversary receives more simulated proofs, whose representation might not be w.r.t. generators present in pp. To make our analysis simpler, we introduce the notion of *algebraic simulator*.

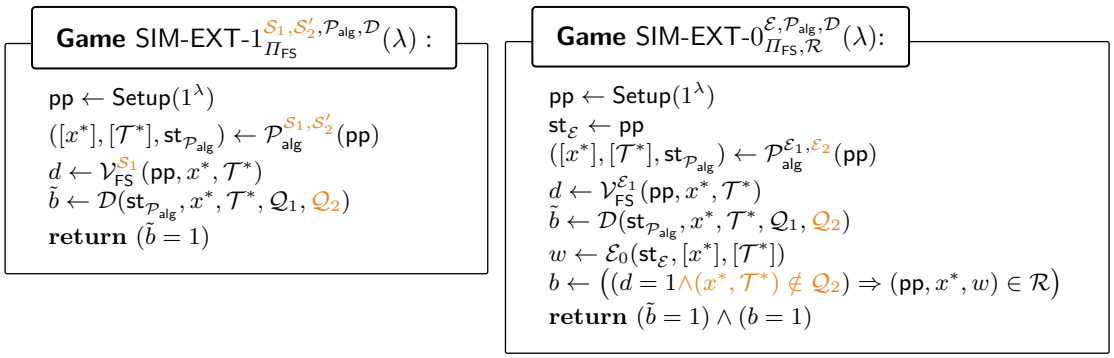

Fig. 5: Simulation extractability games. Like in Fig. 4, in the SIM-EXT-1 experiment, calling the verification algorithm $\mathcal{V}_{\mathsf{FS}}$ has an impact on the RO query set $\mathcal{Q}_1$.

Definition 11 (Algebraic simulator). *Consider a perfectly HVZK argument of knowledge* $(\mathsf{Setup},\mathcal{P},\mathcal{V})$ *with a PPT simulator* $\mathcal{S}$*. The simulator* $\mathcal{S}$ *is* algebraic *if on receiving a statement* x *and its group representation* $[\mathsf{x}]$ *as input, it outputs a proof* $\widetilde{\mathcal{T}}$ *and its group representation* $[\widetilde{\mathcal{T}}]$ *with respect to generators in* pp *and generators used for representing* x*. For an algebraic simulator* $\mathcal{S}$*, we denote* $[\widetilde{\mathcal{T}}] \leftarrow \mathcal{S}([\mathsf{x}])$*.*

Definition 12 (Algebraic simulator for NIZK). *Consider a non-interactive argument of knowledge* $(\mathsf{Setup},\mathcal{P}_{\mathsf{FS}},\mathcal{V}_{\mathsf{FS}})$ *with NIZK simulator* $\mathcal{S}_{\mathsf{FS}}$*. The simulator* $\mathcal{S}_{\mathsf{FS}}$ *is* algebraic *if on receiving a statement* x *and its group representation* $[\mathsf{x}]$ *as input, its second mode outputs proofs* $\widetilde{\mathcal{T}}$*, their group representations* $[\widetilde{\mathcal{T}}]$ *with respect to generators in* pp *and generators used for representing* x*. For an algebraic simulator* $\mathcal{S}_{\mathsf{FS}}$*, we denote* $([\widetilde{\mathcal{T}}],st') \leftarrow \mathcal{S}_{\mathsf{FS}}(2,st,[\mathsf{x}])$*.*

Remark 1. Our use of algebraic is similar in spirit to composability results in the AGM [ABK+21] where the environment is required to be algebraic as well, in addition to the adversary; in particular they require the simulator for proving security to be algebraic. Restricting the simulator to be algebraic does not seem to limit the class of protocols that we can analyze, since typical simulators for discrete-log-based protocols are already algebraic. Consider the simulator for the Schnorr protocol: given a statement $x \in \mathbb{G}$ and random challenge ρ the simulator outputs $(g^z x^{-\rho},\rho,z)$ where z is uniformly sampled from $\mathbb{Z}_q$. In the next section, we show that the simulator for Bulletproofs is also algebraic.

Remark 2. By construction, if we have an algebraic HVZK simulator $\mathcal{S}$ for Π, then the corresponding canonical NIZK simulator $\mathcal{S}_{\mathsf{FS}}$ for Π_{FS} (see the paragraph after Definition 9) fixed by $\mathcal{S}$ is also algebraic, since $\mathcal{S}_{\mathsf{FS}}$ internally invokes $\mathcal{S}$ to obtain a proof.

We now extend the definition of FS-EXT to simulation-extractability, by equipping the cheating algebraic prover with access to proof simulation oracles in addition to the random oracle. Formally, we define simulation-extractability *with respect to a specific NIZK simulator* $\mathcal{S}_{\mathsf{FS}}$ *and the corresponding wrapper oracles* $(\mathcal{S}_1,\mathcal{S}_2')$ *(see Section 2.7).* That is, $\mathcal{S}_1$ on input (t,i) returns the first output of $\mathcal{S}_{\mathsf{FS}}(1,st,t,i)$ (i.e., corresponding the random oracle H in FS-EXT) and $\mathcal{S}_2'$ on an input statement x returns the first output of $\mathcal{S}_{\mathsf{FS}}(2,st,x)$, respectively.

Following FS-EXT, we define a *simulator-extractor* $\mathcal{E}=(\mathcal{E}_0,\mathcal{E}_1,\mathcal{E}_2)$, where $\mathcal{E}_1$ receives a random oracle query of the form (t,i) (similar to the wrapper oracle $\mathcal{S}_1$) and returns a challenge from Ch_i; $\mathcal{E}_2$ receives a statement query x and returns a simulated proof (similar to the wrapper oracle $\mathcal{S}_2'$); $\mathcal{E}_0$ extracts a witness at the end. The differences with Definition 10 are highlighted in orange.

At a high-level, the security requirement of FS-SIM-EXT is two-fold: (1) $(\mathcal{E}_1,\mathcal{E}_2)$ in the game SIM-EXT-0 correctly simulates the adversary's view in SIM-EXT-1 (indicated by a bit $\tilde{b}$), and (2) the extractor $\mathcal{E}_0$ outputs a valid witness as long as an adversarial forgery $(x^*,\mathcal{T}^*)$ is accepting *and* non-trivial, i.e., not identical to what's obtained by querying a proof simulation oracle (indicated by a bit b).

Definition 13 (FS-SIM-EXT security). *Consider a NIZK scheme* $\Pi_{\mathsf{FS}}=(\mathsf{Setup},\mathcal{P}_{\mathsf{FS}},\mathcal{V}_{\mathsf{FS}})$ *for language* $\mathcal{L}$ *with an NIZK simulator* $\mathcal{S}_{\mathsf{FS}}$*. Let* $(\mathcal{S}_1,\mathcal{S}_2')$ *be wrapper oracles for* $\mathcal{S}_{\mathsf{FS}}$ *as defined in Section 2.7.* Π_{FS} *is* online simulation-extractable *(FS-SIM-EXT) with respect to* $\mathcal{S}_{\mathsf{FS}}$ *in the AGM and ROM, if there exists an*

efficient simulator-extractor $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2)$ *such that for every PPT algebraic adversary* $\mathcal{P}_{\mathsf{alg}}$ *and every distinguisher* $\mathcal{D}$*, the following probability is negligible in* λ*:*

$$\mathbf{Adv}^{\textsf{FS-SIM-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{S}_{\mathsf{FS}}, \mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}, \lambda) := \left|\Pr[\textit{SIM-EXT-1}^{\mathcal{S}_1,\mathcal{S}'_2,\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}}}(\lambda)] - \Pr[\textit{SIM-EXT-0}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\lambda)]\right|.$$

In Fig. 5, each of $\mathcal{Q}_1 = \{\mathcal{Q}_{1,i}\}_{i\in[1,r]}$ *is a set of query response pairs corresponding to queries to* $\mathcal{S}_1$ *or* $\mathcal{E}_1$ *with random oracle index* i*.* $\mathcal{Q}_2$ *is a set of statement-transcript pairs* $(x, \widetilde{T})$*, where* x *is a statement queried to the proof simulation oracle* $\mathcal{S}'_2$ *or* $\mathcal{E}_2$ *by* $\mathcal{P}_{\mathsf{alg}}$*, and* $\widetilde{T}$ *is the corresponding simulated proof, respectively.*

Comparison with previous SIM-EXT definitions Although we borrow the formalization of wrapper oracles $(\mathcal{S}_1, \mathcal{S}'_2)$ from [FKMV12], our definition of FS-SIM-EXT is different from their "weak" (Def. 6, an extractor requires rewinding access to the adversary) and "full" (Def. 7, an extractor is tasked with extracting a witness by only looking at an adversarial statement-proof pair) SIM-EXT. Indeed, neither of these is suitable in our setting. The former is too weak because we aim for an "online" way of extraction; the latter is too strong since the extractor used for showing reduction from FS-EXT to aSR-WEE (Theorem 3) already needs additional control over RO queries. To the best of our knowledge, there has been no previous work analyzing Fiat–Shamir NIZK under the latter notion even in the AGM.

Another difference with previous FS-SIM-EXT definitions is that ours explicitly handles indistinguishability of two games. This wasn't the case in [FKMV12] because their proof of weak SIM-EXT invokes the general forking lemma [BN06] that implicitly takes care of perfect indistinguishability of two runs. Our definition can essentially be seen as Def. 11 of Unruh [Unr17] extended with a proof simulation oracle, which however was considered "too strong" in that work as its focus is security in the QROM. In contrast, our main focus is analysis in the CROM and online extraction enabled by the AGM (following the previous FS-EXT analysis conducted by [GT21]). Thus, we believe ours is most suitable for formally analyzing SIM-EXT of Bulletproofs based on the state-of-the-art.

There also exist several SIM-EXT definitions for CRS-NIZK (e.g., [Sah99,DDO+01,Gro06,GO07,ARS20, BKSV20]) but the way they are formulated is naturally different since the plain extractability already varies and simulators for CRS-NIZK behave in a different fashion. Perhaps a variant of Groth [Gro06] is somewhat close to ours: the first part of the extractor handles simulation of CRS (so that it generates a trapdoor without the adversary noticing) and the second part takes care of witness extraction.

Remark 3. In the AGM, the representation submitted by the adversary is w.r.t. the group elements present in pp *and all the simulated proofs they have seen so far*. However, once we assume an algebraic simulator, it is always possible for $\mathcal{E}$ to convert such representation to the one w.r.t. pp and previously queried statements. As we shall see later, this will greatly simplify our security proof in the AGM because it will allow us to reuse the existing extractor analysis (where there is no simulation oracle).

Remark 4. Analogues of algebraic simulation appear in Fuchsbauer et al. [FPS20] and Kastner et al. [KLX22] in which they prove *one-more-unforgeability (OM-UF)* of DLog-based blind signatures in the AGM+ROM. Recall that relation between FS-EXT and FS-SIM-EXT of FS-NIZK is analogous to EUF-KOA and (strong) EUF-CMA of FS-*signatures*, i.e., proof simulation oracle in FS-SIM-EXT corresponds to the signing oracle in EUF-CMA; witness extraction corresponds to breaking the hard relation that EUF-KOA relies on. OM-UF security for blind signatures is essentially a variant of EUF-CMA adapted in the two-party setting. Although they do not explicitly define the notion of algebraic simulator, their reduction to DLog also relies on the ability to convert algebraic representations based on the knowledge of how the simulator composed responses to sign/random oracle queries.

3.2 State-restoration Unique Response and Weak Unique Response

In this part we introduce two new definitions for the *unique response* property: one for interactive and the other for non-interactive protocols. Our supporting claim (Lemma 2) below relates the two notions, which might be of independent interest.

State-restoration unique response. Our first definition considers the game $\textsf{SR-UR}^{\mathcal{A}_{\mathsf{alg}},\mathcal{S}}_{\Pi}(\lambda)$ in Figure 6. As the name indicates it has a flavor of aSR-WEE and it is therefore – compared to the the usual UR definition for interactive protocols – both stronger (in the sense that an adversary can rewind the verifier) and weaker (in the sense that an adversary is forced to use the simulated transcript to find a forgery).

Concretely, the prover initially generates an instance x on which it attempts to break the unique response property. Similar to aSR-WEE, we capture the power of the prover to rewind the verifier with

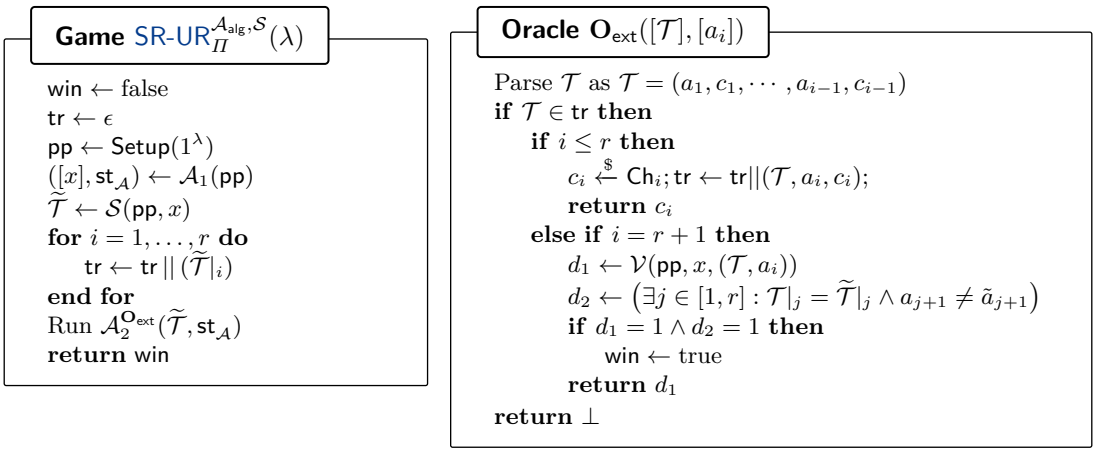

Fig. 6: State-restoration Unique Response.

an oracle $\mathbf{O}_{\mathsf{ext}}$. Roughly, the oracle allows the prover to build an execution tree, which is extended with each query to it by the prover. The prover succeeds if it comes up with another accepting transcript $\mathcal{T}$ that is part of the execution tree and have a prefix in common with the simulated transcript $\widetilde{\mathcal{T}}$. Let $\mathcal{T} = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$ denote a transcript. We write $\mathcal{T}|_i$ to denote a partial transcript consisting of the first $2i$ messages of $\mathcal{T}$, i.e., $\mathcal{T}|_i = (a_1, c_1, \ldots, a_i, c_i)$.

We also remark that, unlike aSR-WEE, our SR-UR is deliberately made non-adaptive to prove subsequent lemmas with a weaker assumption. Indeed, the reductions we present later will go through even though the resulting simulation-extractability claim has an adaptive flavor.

Definition 14 (SR-UR). *Consider a $(2r+1)$-round public-coin interactive proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ that has perfect HVZK simulator $\mathcal{S}$. Π is said to have* state-restoration unique response *(SR-UR) with respect to a simulator $\mathcal{S}$, if for all PPT algebraic adversaries $\mathcal{A}_{\mathsf{alg}} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage $\mathbf{Adv}_{\Pi}^{\text{SR-UR}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}) := \Pr[\text{SR-UR}_{\Pi}^{\mathcal{A}_{\mathsf{alg}},\mathcal{S}}(\lambda)]$ is $\mathsf{negl}(\lambda)$.*

Weak Unique Response We now present our *weak unique response* definition tailored to non-interactive protocols. While typical unique response properties in the literature are defined for interactive protocols, [KZ21, Definition 7] is in the non-interactive setting. Our definition below is strictly weaker than theirs, as we only need to guarantee the hardness of finding another accepting transcript forked from simulated (honest) one.

Definition 15 (FS-WUR). *Consider a $(2r+1)$-round public-coin interactive proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ and the resulting NIZK $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ via Fiat-Shamir transform. Let $\mathcal{S}_{\mathsf{FS}}$ be a perfect NIZK simulator for Π_{FS} (Definition 9) with wrapper oracles $(\mathcal{S}_1, \mathcal{S}_2')$ as defined in Section 2.7. Π_{FS} is said to have* weak unique responses *(FS-WUR) with respect to $\mathcal{S}_{\mathsf{FS}}$ if given a transcript $\widetilde{\mathcal{T}} = (\tilde{a}_1, \tilde{c}_1, \ldots, \tilde{a}_r, \tilde{c}_r, \tilde{a}_{r+1})$ simulated by $\mathcal{S}_{\mathsf{FS}}$, it is hard to find another accepting transcript $\mathcal{T} = (a_1, c_1, \ldots, a_r, c_r, a_{r+1})$ that both have a common prefix up to the ith challenge for an instance x. That is, for all PPT algebraic adversaries $\mathcal{A}_{\mathsf{alg}} = (\mathcal{A}_1, \mathcal{A}_2)$ the advantage $\mathbf{Adv}_{\Pi_{\mathsf{FS}}}^{\text{FS-WUR}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}})$ defined as the following probability is $\mathsf{negl}(\lambda)$:*

$$\Pr\left[\begin{array}{l} \mathcal{V}_{\mathsf{FS}}^{\mathcal{S}_1}(\mathsf{pp}, x, \mathcal{T}) = 1 \\ \wedge\ (\exists j \in [1, r] : \mathcal{T}|_j = \widetilde{\mathcal{T}}|_j \\ \quad \wedge\ a_{j+1} \neq \tilde{a}_{j+1}) \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda); \\ ([x], \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathcal{S}_1}(\mathsf{pp}); \\ \widetilde{\mathcal{T}} \leftarrow \mathcal{S}_2'(x); \\ [\mathcal{T}] \leftarrow \mathcal{A}_2^{\mathcal{S}_1}(\widetilde{\mathcal{T}}, \mathsf{st}); \end{array} \right].$$

We now show that FS-WUR of Π_{FS} reduces to SR-UR of the interactive proof system Π in the AGM. Informally, the lemma below guarantees that one can construct an adversary breaking unique response in the interactive setting, given an adversary breaking unique response in the non-interactive setting, as long as it makes RO queries for the accepting transcript in right order. As mentioned earlier, the reduction below does not crucially depend on the AGM: if a given protocol meets SR-UR without the AGM the proof holds almost verbatim without the AGM as well.

Lemma 2. *Consider a $(2r+1)$-round public-coin interactive proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ and the resulting NIZK $\Pi_{\mathsf{FS}} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{FS}}, \mathcal{V}_{\mathsf{FS}})$ via Fiat-Shamir transform. Let $\mathcal{S}$ be a perfect algebraic HVZK simulator for Π and $\mathcal{S}_{\mathsf{FS}}$ be the corresponding canonical NIZK simulator for Π_{FS}. If Π has SR-UR with respect to $\mathcal{S}$, then Π_{FS} has FS-WUR with respect to $\mathcal{S}_{\mathsf{FS}}$. That is, for every PPT adversary $\mathcal{A}$ against FS-WUR of Π_{FS} that makes q queries to $\mathcal{S}_1$, there exists a PPT adversary $\mathcal{B}$ against SR-UR of Π such that,*

$$\mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}}}(\mathcal{A}, \mathcal{S}_{\mathsf{FS}}) \leq \mathbf{Adv}^{\text{SR-UR}}_{\Pi}(\mathcal{B}, \mathcal{S}) + \frac{q+1}{|\mathsf{Ch}_{i_0}|}$$

where $i_0 \in [1, r]$ is the round with the smallest challenge set Ch_{i_0}. Moreover, $\mathcal{B}$ makes at most q queries to its oracle and is nearly as efficient as $\mathcal{A}$.

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be the NIZK prover against FS-WUR. We define an adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against SR-UR. Without loss of generality we assume that $\mathcal{A}$ does not repeat the same RO queries.

In the first stage, $\mathcal{B}_1$ will run $\mathcal{A}_1$ on pp. Whenever $\mathcal{A}_1$ makes RO queries $\mathcal{B}_1$ invokes $\mathcal{S}_1$ and keeps track of the query history in tables $\mathcal{Q}_1 = \{\mathcal{Q}_{1,1}, \ldots, \mathcal{Q}_{1,r}\}$. When $\mathcal{A}_1$ returns $([x], \mathsf{st}_{\mathcal{A}})$, $\mathcal{B}_1$ outputs $([x], \mathsf{st}_{\mathcal{B}})$ where $\mathsf{st}_{\mathcal{B}} = (\mathsf{pp}, \mathsf{st}_{\mathcal{A}}, [x], \mathcal{Q})$.

In the second stage, $\mathcal{B}_2$ is given $\mathsf{st}_{\mathcal{B}}$ and $\widetilde{\mathcal{T}}$, which is the output on invoking the HVZK simulator $\mathcal{S}$ on x. Then, $\mathcal{B}_2$ programs the RO tables such that $\mathcal{Q}_{1,1}[\mathsf{pp}, x, \tilde{a}_1] := \tilde{c}_1, \ldots, \mathcal{Q}_{1,r}[\mathsf{pp}, x, \tilde{a}_1, \tilde{c}_1, \ldots, \tilde{a}_r] := \tilde{c}_r$. Now, $\mathcal{B}_2$ invokes $\mathcal{A}_2$ on $\widetilde{\mathcal{T}}$ and $\mathsf{st}_{\mathcal{A}}$ and responds to the RO queries as follows.

- $\mathcal{B}_2$ initializes a set of partial transcripts tr as

$$\mathsf{tr} = \{(\tilde{a}_1, \tilde{c}_1), \ldots, (\tilde{a}_1, \tilde{c}_1, \ldots, \tilde{a}_r, \tilde{c}_r)\}.$$

- Whenever $\mathcal{B}_2$ receives from $\mathcal{A}_2$ a query to $\mathcal{S}_1$ of the form $((\mathsf{pp}, [x], [\mathcal{T}], [a_i]), i)$ where transcript $\mathcal{T} = (a_1, c_1, \ldots, a_{i-1}, c_{i-1})$, it checks if $\mathcal{T} \in \mathsf{tr}$. If that is the case, it queries $\mathbf{O}_{\mathsf{ext}}$ with $\mathcal{T}$ and a_i (with representation). On receiving challenge c_i, it updates tr such that $\mathsf{tr} \leftarrow \mathsf{tr}||(\mathcal{T}, a_i, c_i)$ and programs the RO table such that $\mathcal{Q}_{1,i}[\mathsf{pp}, x, \mathcal{T}, a_i] := c_i$. Note that the partial transcript set tr constructed as above is guaranteed to be identical to that of SR-UR.
- When $\mathcal{B}_2$ receives from $\mathcal{A}_2$ a forged transcript $\mathcal{T}$ at the end, it checks whether the winning condition is satisfied, i.e.,

$$\mathcal{V}^{\mathcal{S}_1}_{\mathsf{FS}}(\mathsf{pp}, x, \mathcal{T}) = 1 \wedge (\exists j \in [1, r] : \mathcal{T}|_j = \widetilde{\mathcal{T}}|_j \wedge a_{j+1} \neq \tilde{a}_{j+1}).$$

  If that is the case but $\mathcal{T}|_i \notin \mathsf{tr}$ for some $i \in [1, r]$, $\mathcal{B}_2$ aborts, since it implies $\mathcal{T}$ is not present in the partial transcript set maintained by $\mathbf{O}_{\mathsf{ext}}$ and therefore does not qualify as a winning transcript in the SR-UR game. Otherwise, it forwards $\mathcal{T}$ to $\mathbf{O}_{\mathsf{ext}}$ (with representation).

If $\mathcal{B}_2$ does not abort, $\mathcal{B}_2$ wins the SR-UR game because all the partial transcripts of $\mathcal{T}$ are recorded by $\mathbf{O}_{\mathsf{ext}}$ in the SR-UR game. We now bound the probability that $\mathcal{B}_2$ aborts. It aborts only if (1) the accepting transcript $\mathcal{T}$ was crafted by $\mathcal{A}_2$ without querying the RO in order, or (2) the accepting transcript $\mathcal{T}$ contains some challenge that was obtained without querying the RO at all.

The first case happens only if for some $i, j \in [1, r]$ such that $j < i$, a tuple $(\mathsf{pp}, x, \mathcal{T}|_{i-1}, a_i)$ was queried to $\mathcal{S}_1$ *before* $(\mathsf{pp}, x, \mathcal{T}|_{j-1}, a_j)$ was queried. In that case, when $(\mathsf{pp}, x, \mathcal{T}|_{j-1}, a_j)$ is queried later, $\mathcal{S}_1$ must return c_j already present in $\mathcal{T}|_{i-1}$ for $\mathcal{T}$ to be accepting. This happens with probability at most $q/|\mathsf{Ch}_j| \leq q/|\mathsf{Ch}_{i_0}|$ (where i_0 is the round with the smallest Ch_{i_0}) since the adversary can try to guess any challenge value for at most q times.

The second case happens only if for some $i \in [1, r]$, a tuple $(\mathsf{pp}, x, \mathcal{T}|_{i-1}, a_i)$ was never queried to $\mathcal{S}_1$ by $\mathcal{A}$. In that case, when $(\mathsf{pp}, x, \mathcal{T}|_{i-1}, a_i)$ is queried later, $\mathcal{S}_1$ must return c_i already present in $\mathcal{T}$ for $\mathcal{T}$ to be accepting. This happens with probability at most $1/|\mathsf{Ch}_i| \leq 1/|\mathsf{Ch}_{i_0}|$ since $\mathcal{V}_{\mathsf{FS}}$ queries $\mathcal{S}_1$ once for each challenge.

Overall, $\mathcal{B}$ wins whenever $\mathcal{A}$ wins except with probability at most $(q+1)/|\mathsf{Ch}_{i_0}|$.

□

3.3 From Weak Unique Response to Simulation-extractability

We now prove the simulation-extractability of a non-interactive protocol Π_{FS} assuming it comes with an algebraic NIZK simulator $\mathcal{S}_{\mathsf{FS}}$, it is extractable and has weak unique responses with respect to $\mathcal{S}_{\mathsf{FS}}$. On a high-level the proof works by constructing another adversary $\mathcal{P}_{\mathsf{alg}}$ that forwards the RO queries made by a

FS-SIM-EXT adversary $\mathcal{P}^*_{\mathsf{alg}}$ to the FS-EXT game, *except for the ones that have prefix in common with any of the simulated transcripts.* This will allow us to invoke the extractor $\mathcal{E}$ that is only guaranteed to work in the FS-EXT setting. On the other hand, thanks to the FS-WUR property we can argue that a cheating prover also has a hard time finding another transcript by reusing any prefix of a simulated transcript.

We stress that, as long as FS-WUR and FS-EXT are satisfied without the AGM the proof below holds almost verbatim without the AGM as well. Interestingly, proof in the AGM requires additional care about representation submitted by $\mathcal{P}_{\mathsf{alg}}$: whenever $\mathcal{P}_{\mathsf{alg}}$ forwards group elements with representation to external entities (i.e., H, $\mathcal{E}_1$, and $\mathcal{E}_0$), it must always convert representation to the one *only with respect to generators in* pp. This is made possible thanks to an *algebraic* simulator $\mathcal{S}_{\mathsf{FS}}$; by probing how $\mathcal{S}_{\mathsf{FS}}$ simulates a transcript with respect to the generators in pp, $\mathcal{P}_{\mathsf{alg}}$ can translate the group representation submitted by $\mathcal{P}^*_{\mathsf{alg}}$ even if it depends on previously simulated transcripts. This is crucial for invoking the extractor from FS-EXT, since a cheating prover against FS-EXT is only allowed to use the generators present in pp. We also remark that the additive security loss due to failure of RO programming by $\mathcal{S}'_2$ is not present in the bound since we use a canonical NIZK simulator as an assumption and such a loss already appears when showing NIZK from HVZK.

Lemma 3. *Consider a NIZK argument system Π_{FS} with an algebraic NIZK simulator $\mathcal{S}_{\mathsf{FS}}$. If Π_{FS} is FS-WUR with respect to $\mathcal{S}_{\mathsf{FS}}$ and online FS-EXT, then it is online FS-SIM-EXT with respect to $\mathcal{S}_{\mathsf{FS}}$.*

Concretely, let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1)$ be an FS-EXT extractor for Π_{FS}. There exists an efficient FS-SIM-EXT simulator-extractor $\mathcal{E}^* = (\mathcal{E}^*_0, \mathcal{E}^*_1, \mathcal{E}^*_2)$ for Π_{FS} such that for every algebraic prover $\mathcal{P}^*_{\mathsf{alg}}$ against Π_{FS} that makes q_1 random oracle queries (i.e., queries to $\mathcal{S}_1$ or $\mathcal{E}^*_1$), and q_2 simulation queries (i.e., queries to $\mathcal{S}'_2$ or $\mathcal{E}^*_2$), and for every distinguisher $\mathcal{D}^*$, there exists another algebraic prover $\mathcal{P}_{\mathsf{alg}}$, a distinguisher $\mathcal{D}$, and an FS-WUR adversary $\mathcal{A}_{\mathsf{alg}}$, such that for all $\lambda \in \mathbb{N}^+$,

$$\mathbf{Adv}^{\text{FS-SIM-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{S}_{\mathsf{FS}}, \mathcal{E}^*, \mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*) \leq \mathbf{Adv}^{\text{FS-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathsf{H}, \mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}) + q_2 \cdot \mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}}).$$

Moreover, $\mathcal{P}_{\mathsf{alg}}$ and $\mathcal{A}_{\mathsf{alg}}$ make at most q_1 queries to their oracle and is nearly as efficient as $\mathcal{P}^*_{\mathsf{alg}}$. The extractor $\mathcal{E}^*$ is nearly as efficient as $\mathcal{E}$.

Proof. Without loss of generality we assume $\mathcal{P}^*_{\mathsf{alg}}$ does not repeat the same RO queries. We first construct a cheating prover $\mathcal{P}_{\mathsf{alg}}$ against FS-EXT that internally uses the FS-SIM-EXT adversary $\mathcal{P}^*_{\mathsf{alg}}$ and simulates its view in FS-SIM-EXT.

We now describe the following simple hybrids.

G_0 This game is identical to $\mathsf{SIM\text{-}EXT\text{-}1}^{\mathcal{S}_1,\mathcal{S}'_2,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}}}(\lambda)$. We have

$$\Pr[\mathsf{G}_0(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] = \Pr[\mathsf{SIM\text{-}EXT\text{-}1}^{\mathcal{S}_1,\mathcal{S}'_2,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}}}(\lambda)].$$

G_1 This game is identical to G_0 except that it aborts if $d = 1$ (i.e., $(x^*, \mathcal{T}^*)$ is accepting) and $(x^*, \mathcal{T}^*) \notin \mathcal{Q}_2$, while there exists some $(x^*, \widetilde{\mathcal{T}}) \in \mathcal{Q}_2$ that has prefix in common with $\mathcal{T}^*$ but differs at the response right after that prefix, i.e., for some $j \leq r$ it holds that $\mathcal{T}^*|_j = \widetilde{\mathcal{T}}|_j$ *and* $a^*_{j+1} \neq \tilde{a}_{j+1}$. The abort event implies that there exists an efficient FS-WUR adversary $\mathcal{A}_{\mathsf{alg}}$ that internally uses $\mathcal{P}^*_{\mathsf{alg}}$. That is,

$$\left|\Pr[\mathsf{G}_0(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] - \Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)]\right| \leq \Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*) \text{ aborts}] \leq q_2 \cdot \mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}}). \quad (4)$$

We defer the reduction deriving (4) to later.

Constructing $\mathcal{P}_{\mathsf{alg}}$ and $\mathcal{D}$ for FS-EXT. We now construct a FS-EXT adversary $\mathcal{P}_{\mathsf{alg}}$ and a distinguisher $\mathcal{D}$. $\mathcal{P}_{\mathsf{alg}}$ plays an FS-EXT game while internally simulating the view of $\mathcal{P}^*_{\mathsf{alg}}$ in the game G_1 as follows.

- On receiving pp from $\mathsf{Setup}(1^\lambda)$, $\mathcal{P}_{\mathsf{alg}}$ forwards pp to $\mathcal{P}^*_{\mathsf{alg}}$.
- Whenever $\mathcal{P}^*_{\mathsf{alg}}$ makes a simulation query with input $[x]$, $\mathcal{P}_{\mathsf{alg}}$ internally invokes $\mathcal{S}_{\mathsf{FS}}(2, st, [x])$ to obtain $([\widetilde{\mathcal{T}}], st')$ and records a statement-proof pair $(x, \widetilde{\mathcal{T}})$ in the set $\mathcal{Q}_2$. $\mathcal{P}_{\mathsf{alg}}$ also separately keeps track of representation of every entry in $\mathcal{Q}_2$. Then it programs the RO tables $\mathcal{Q}_1$ for every challenge in $\widetilde{\mathcal{T}}$ as $\mathcal{S}_{\mathsf{FS}}(2, st, [x])$ would do.
- Whenever $\mathcal{P}^*_{\mathsf{alg}}$ (or $\mathcal{V}_{\mathsf{FS}}$ at the end) makes a random oracle query with input $((\mathsf{pp}, [x], [\mathcal{T}], [a_i]), i)$, where $\mathcal{T} = (a_1, c_1, \ldots, a_{i-1}, c_{i-1})$, $\mathcal{P}_{\mathsf{alg}}$ checks whether there exists some $(x, \widetilde{\mathcal{T}}) \in \mathcal{Q}_2$ that has prefix in common with $\mathcal{T}$, i.e., for some $j \leq i-1$ it holds that $\mathcal{T}|_j = \widetilde{\mathcal{T}}|_j$. If that is the case, it lazily samples c_i from Ch_i and updates $\mathcal{Q}_{1,i}$ accordingly, as $\mathcal{S}_{\mathsf{FS}}(1, st, (\mathsf{pp}, [x], [\mathcal{T}], [a_i]), i)$ would do. Otherwise, it forwards the query $((\mathsf{pp}, x, \mathcal{T}, a_i), i)$ to a FS-EXT game with converted group representation, receives $c_i \in \mathsf{Ch}_i$, and updates $\mathcal{Q}_{1,i}$ accordingly.

– When $\mathcal{P}^*_{\mathsf{alg}}$ outputs a forgery $([x^*], [\mathcal{T}^*])$, $\mathcal{P}_{\mathsf{alg}}$ first checks whether it causes aborts in the game G_1. If that is the case, $\mathcal{P}_{\mathsf{alg}}$ also aborts because it implies that the challenge values in $\mathcal{T}^*$ are not obtained by forwarding the corresponding queries to a FS-EXT game and therefore $(x^*, \mathcal{T}^*)$ is not accepting in the FS-EXT game.

– Otherwise, $\mathcal{P}_{\mathsf{alg}}$ outputs $(x^*, \mathcal{T}^*, \mathsf{st}_{\mathcal{P}_{\mathsf{alg}}})$ to a FS-EXT game with converted group representation, where $\mathsf{st}_{\mathcal{P}_{\mathsf{alg}}} = (\mathcal{Q}_1, \mathcal{Q}_2)$.

A FS-EXT distinguisher $\mathcal{D}$ internally invokes $\mathcal{D}^*$ on input $(\mathsf{st}_{\mathcal{P}_{\mathsf{alg}}}, x^*, \mathcal{T}^*, \mathcal{Q}_1, \mathcal{Q}_2)$ and outputs whatever $\mathcal{D}^*$ returns. By construction, we have

$$\Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] = \Pr[\mathsf{EXT\text{-}1}^{\mathsf{H}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}}_{\Pi_{\mathsf{FS}}}(\lambda)].$$

Constructing $\mathcal{E}^*$ for FS-SIM-EXT. We define a simulator-extractor $\mathcal{E}^* = (\mathcal{E}^*_0, \mathcal{E}^*_1, \mathcal{E}^*_2)$ using a FS-EXT extractor $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1)$. $\mathcal{E}^*_1$ answers the random oracle queries made by $\mathcal{P}^*_{\mathsf{alg}}$ as $\mathcal{P}_{\mathsf{alg}}$ would, by using the responses from $\mathcal{E}_1$. $\mathcal{E}^*_2$ answers the simulation queries made by $\mathcal{P}^*_{\mathsf{alg}}$ as $\mathcal{P}_{\mathsf{alg}}$ would, by internally invoking $\mathcal{S}_{\mathsf{FS}}$. $\mathcal{E}^*_0$ outputs whatever $\mathcal{E}_0$ returns on input $(\mathsf{st}_{\mathcal{E}}, [x^*], [\mathcal{T}^*])$. Note that, if $\mathcal{P}_{\mathsf{alg}}$ does not abort, $\mathcal{T}^*$ has no prefix in common with any of the previously simulated transcripts. In that case, thanks to the random oracle simulation conducted by $\mathcal{P}_{\mathsf{alg}}$ as above, for every $i \in [1, r]$, c^*_i has been obtained by querying the random oracle in a FS-EXT game with input $((\mathsf{pp}, x^*, \mathcal{T}^*|_{i-1}, a^*_i), i)$. Therefore, $(x^*, \mathcal{T}^*)$ gets accepted by $\mathcal{V}^{\mathcal{E}_1}_{\mathsf{FS}}$ whenever it gets accepted by $\mathcal{V}^{\mathcal{E}^*_1}_{\mathsf{FS}}$, $(x^*, \mathcal{T}^*) \notin \mathcal{Q}_2$, *and* $\mathcal{P}_{\mathsf{alg}}$ does not abort. By construction, $\mathcal{E}^*$ succeeds in extraction if and only if $\mathcal{E}$ does so in the game $\mathsf{EXT\text{-}0}^{\mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}}_{\Pi_{\mathsf{FS}}, \mathcal{R}}(\lambda)$. Thus we have

$$\Pr[\mathsf{SIM\text{-}EXT\text{-}0}^{\mathcal{E}^*, \mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*}_{\Pi_{\mathsf{FS}}, \mathcal{R}}(\lambda)] = \Pr[\mathsf{EXT\text{-}0}^{\mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}}_{\Pi_{\mathsf{FS}}, \mathcal{R}}(\lambda)].$$

Reduction to FS-WUR. We now bound the probability that the game G_1 aborts. We argue that, if there exists $(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)$ that causes $\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)$ to abort (or in other words, that causes $\mathcal{P}_{\mathsf{alg}}$ to abort), one can use $\mathcal{P}^*_{\mathsf{alg}}$ to construct another adversary $\mathcal{A}_{\mathsf{alg}} = (\mathcal{A}_1, \mathcal{A}_2)$ that breaks FS-WUR with respect to $\mathcal{S}_{\mathsf{FS}}$. The reduction goes as follows. The differences with $\mathcal{P}_{\mathsf{alg}}$ are highlighted in orange.

– $\mathcal{A}_1$ first picks a query index $k \in [1, q_2]$ uniformly at random.

– On receiving pp from $\mathsf{Setup}(1^\lambda)$, $\mathcal{A}_1$ forwards pp to $\mathcal{P}^*_{\mathsf{alg}}$.

– Whenever $\mathcal{P}^*_{\mathsf{alg}}$ makes a simulation query with input $[x]$, if this is the kth simulation query then it forwards x to $\mathcal{S}'_2$ in the FS-WUR game with converted group representation. We denote the statement-transcript pair of the kth query by $(x^k, \widetilde{\mathcal{T}}^k)$.[8] Otherwise, $\mathcal{A}_1$ internally invokes $\mathcal{S}_{\mathsf{FS}}(2, st, [x])$ to obtain $([\widetilde{\mathcal{T}}], st')$. It records a statement-proof pair $(x, \widetilde{\mathcal{T}})$ in the set $\mathcal{Q}_2$. $\mathcal{A}$ also separately keeps track of representation of every entry in $\mathcal{Q}_2$. Then it programs the RO tables $\mathcal{Q}_1$ for every challenge in $\widetilde{\mathcal{T}}$ as $\mathcal{S}_{\mathsf{FS}}(2, st, [x])$ would do. $\mathcal{A}_2$ also responds to simulation queries in the same way, except that it never forwards a statement to the FS-WUR game.

– Whenever $\mathcal{P}^*_{\mathsf{alg}}$ (or $\mathcal{V}_{\mathsf{FS}}$ at the end) makes a random oracle query with input $((\mathsf{pp}, [x], [\mathcal{T}], [a_i]), i)$, where $\mathcal{T} = (a_1, c_1, \ldots, a_{i-1}, c_{i-1})$, $\mathcal{A}_2$ checks whether $(x^k, \widetilde{\mathcal{T}}^k)$ has prefix in common with $\mathcal{T}$, i.e., for some $j \leq i-1$ it holds that $\mathcal{T}|_j = \widetilde{\mathcal{T}}^k|_j$. If that is the case, it forwards the query $((\mathsf{pp}, x, \mathcal{T}, a_i), i)$ to $\mathcal{S}_1$ in the FS-WUR game with converted group representation, receives $c_i \in \mathsf{Ch}_i$, and updates $\mathcal{Q}_{1,i}$ accordingly. Otherwise, it lazily samples c_i from Ch_i and updates $\mathcal{Q}_{1,i}$ accordingly, as $\mathcal{S}_{\mathsf{FS}}(1, st, (\mathsf{pp}, [x], [\mathcal{T}], [a_i]), i)$ would do. $\mathcal{A}_1$ also responds to random oracle queries in the same way, except that it never forwards queries to the FS-WUR game.

– When $\mathcal{P}^*_{\mathsf{alg}}$ outputs a forgery $([x^*], [\mathcal{T}^*])$, $\mathcal{A}_{\mathsf{alg}}$ first checks whether it causes aborts in the game G_1. If that is the case, $\mathcal{A}_2$ forwards $\mathcal{T}^*$ to the FS-WUR game as a forgery with converted group representation.

The above procedure perfectly simulates $\mathcal{P}^*_{\mathsf{alg}}$'s view in the game G_1. By construction $\mathcal{A}_{\mathsf{alg}}$ breaks FS-WUR with respect to $\mathcal{S}_{\mathsf{FS}}$ if G_1 aborts *and* ($x^* = x^k \wedge \mathcal{T}^*$ has some prefix in common with $\widetilde{\mathcal{T}}^k$), because then it is guaranteed that for every $i \in [1, r]$, c^*_i has been obtained by querying the oracles $(\mathcal{S}_1, \mathcal{S}'_2)$ in the FS-WUR game. Therefore, $\mathcal{T}^*$ does qualify as a valid forgery in the FS-WUR game. Conditioned on the event that G_1 aborts, the probability that $\mathcal{A}_{\mathsf{alg}}$ wins is at least $1/q_2$. Therefore, we have

[8] We note that $\mathcal{A}_{\mathsf{alg}}$ does not get to know the representation of $\widetilde{\mathcal{T}}^k$ unlike other simulated transcripts, as that particular one comes from the FS-WUR game and its representation is not disclosed to the adversary. Therefore, all the subsequent outputs from $\mathcal{P}^*_{\mathsf{alg}}$ are with respect to pp and $\widetilde{\mathcal{T}}^k$. This, however, does not prevent us from showing reduction because outputting representation w.r.t. pp and $\widetilde{\mathcal{T}}^k$ is indeed allowed in the FS-WUR game.

$$\frac{1}{q_2} \cdot \Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*) \text{ aborts}] \leq \mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}}) \tag{5}$$

which derives (4). Putting together, we obtain

$$\begin{aligned}
&\left|\Pr[\text{SIM-EXT-1}^{\mathcal{S}_1,\mathcal{S}'_2,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}}}(\lambda)] - \Pr[\text{SIM-EXT-0}^{\mathcal{E}^*,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\lambda)]\right| \\
&\leq \left|\Pr[\text{EXT-1}^{\mathsf{H},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}}}(\lambda)] - \Pr[\text{EXT-0}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\lambda)]\right| + q_2 \cdot \mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}}) \\
&\leq \mathbf{Adv}^{\text{FS-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathsf{H}, \mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}) + q_2 \cdot \mathbf{Adv}^{\text{FS-WUR}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}_{\mathsf{FS}}).
\end{aligned}$$

□

3.4 Generic Result on Simulation-extractability

Theorem 4. *Let $\mathcal{R}$ be a relation. Let Π be a r-challenge public coin interactive protocol for the relation $\mathcal{R}$ where the ith challenge is sampled from Ch_i for $i \in [1, r]$. Suppose Π satisfies: aSR-WEE, perfect HVZK with algebraic simulator $\mathcal{S}$, and SR-UR with respect to $\mathcal{S}$. Let $\mathcal{S}_{\mathsf{FS}}$ be the corresponding canonical NIZK simulator for $\mathcal{S}_{\mathsf{FS}}$ fixed by $\mathcal{S}$. Then Π_{FS} is FS-SIM-EXT with respect to $\mathcal{S}_{\mathsf{FS}}$.*

Concretely, let $\mathcal{E}$ be an aSR-WEE extractor for Π. There exists an efficient FS-SIM-EXT simulator-extractor $\mathcal{E}^*$ for Π_{FS} such that for every non-uniform algebraic prover $\mathcal{P}^*_{\mathsf{alg}}$ against Π_{FS} that makes q_1 random oracle queries, and q_2 simulation queries, and for every distinguisher $\mathcal{D}^*$, there exists a non-uniform algebraic prover $\mathcal{P}_{\mathsf{alg}}$, an SR-UR adversary $\mathcal{A}_{\mathsf{alg}}$, and a distinguisher $\mathcal{D}$ such that for all $\lambda \in \mathbb{N}^+$,

$$\begin{aligned}
&\mathbf{Adv}^{\text{FS-SIM-EXT}}_{\Pi_{\mathsf{FS}},\mathcal{R}}(\mathcal{S}_{\mathsf{FS}}, \mathcal{E}^*, \mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*, \lambda) \\
\leq &\mathbf{Adv}^{\text{aSR-WEE}}_{\Pi,\mathcal{R}}(\mathcal{E}, \mathcal{P}_{\mathsf{alg}}, \mathcal{D}, \lambda) + q_2 \cdot \mathbf{Adv}^{\text{SR-UR}}_{\Pi,\mathcal{R}}(\mathcal{A}_{\mathsf{alg}}, \mathcal{S}, \lambda) + \frac{(q_2+1)(q_1+1)}{|\mathsf{Ch}_{i_0}|}
\end{aligned}$$

where $i_0 \in [1, r]$ is the round with the smallest challenge set Ch_{i_0}.

Proof. From Theorem 3, aSR-WEE of Π implies FS-EXT security of Π_{FS}. From Lemma 2, SR-UR and HVZK of Π implies FS-WUR security of Π_{FS}. Finally, from Lemma 3, FS-EXT and FS-WUR imply FS-SIM-EXT security of Π_{FS}. Putting together all the concrete bounds, we obtain the result. □

4 Non-Malleability of Bulletproofs – Arithmetic Circuits

The protocol for arithmetic circuit satisfiability (see relation 2) as it appears in Bulletproofs (henceforth referred as BP) [BBB+17] is formally described in Protocol 1 and proceeds as follows: In the first round, the prover commits to values on the wire of the circuit (i.e. $\mathbf{a}_L, \mathbf{a}_R$ and $\mathbf{a}_O$), and the blinding vectors $(\mathbf{s}_L, \mathbf{s}_R)$. It receives challenges y, z from the verifier. Based on these challenges, the prover defines three polynomials, l, r and t, where $t(X) = \langle l(X), r(X)\rangle$, and commits to the coefficients of the polynomial t in the third round, i.e. commitments T_1, T_3, T_4, T_5, and, T_6[9]. On receiving a challenge x from the verifier, the prover evaluates polynomials l, r on this challenge point, computes $\hat{t} = \langle l(x), r(x)\rangle$, and values β_x, μ, and sends $\beta_x, \mu, \hat{t}, \mathbf{l} = l(x)$ and $\mathbf{r} = r(x)$ in the fifth round. The verifier accepts if: the commitments $\{T_i\}_{i=\mathcal{S}}$ (for $\mathcal{S} = \{1, 3, 4, 5, 6\}$) are to the correct polynomial t and if $\hat{t} = \langle \mathbf{l}, \mathbf{r}\rangle$. To get logarithmic proof size, the prover and verifier define an instance of the inner dot product for checking the condition $\hat{t} = \langle \mathbf{l}, \mathbf{r}\rangle$, instead of sending vectors $\mathbf{l}, \mathbf{r}$ in clear.

The inner product subroutine is presented in Appendix C.

Protocol 1: BP

This is a protocol for the following relation

$$\mathcal{R}_{\mathsf{ACSPf}} = \left\{ \begin{array}{l} ((n, Q, g, h, u, \mathbf{g}, \mathbf{h}), (\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O, \mathbf{c}), (\mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O)) \mid \\ \mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O \wedge \mathbf{W}_L \cdot \mathbf{a}_L + \mathbf{W}_R \cdot \mathbf{a}_R + \mathbf{W}_O \cdot \mathbf{a}_R = \mathbf{c} \end{array} \right\}.$$

Prover $\mathcal{P}$ and verifier $\mathcal{V}$ interact as follows:

[9] The degree two term is independent of the witness and can be computed by the verifier, therefore there is no T_2 commitment.

1. $\mathcal{P}$ samples $\mathbf{s}_L \xleftarrow{\$} \mathbb{Z}_p^n, \mathbf{s}_R \xleftarrow{\$} \mathbb{Z}_p^n, \alpha, \beta, \rho \xleftarrow{\$} \mathbb{Z}_p$. $\mathcal{P}$ sends

$$S = h^{\rho}\mathbf{g}^{\mathbf{s}_L}\mathbf{h}^{\mathbf{s}_R},\ A_I = h^{\alpha}\mathbf{g}^{\mathbf{a}_L}\mathbf{h}^{\mathbf{a}_R},\ A_O = h^{\beta}\mathbf{g}^{\mathbf{a}_O}.$$

2. $\mathcal{V}$ sends challenge $y, z \xleftarrow{\$} \mathbb{Z}_p^*$.
3. $\mathcal{P}$ samples $\beta_i \xleftarrow{\$} \mathbb{Z}_p, \forall i \in \mathcal{S}$, computes polynomials

$$\begin{aligned}
l(X) &= \mathbf{a}_L X + \mathbf{a}_O X^2 + \mathbf{y}^{-n} \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R) \cdot X + \mathbf{s}_L X^3, \\
r(X) &= \mathbf{y}^n \circ \mathbf{a}_R \cdot X - \mathbf{y}^n + \mathbf{z}_{[1:]}^{Q+1} \cdot (\mathbf{W}_L \cdot X + \mathbf{W}_O) + \mathbf{y}^n \circ \mathbf{s}_R \cdot X^3, \\
t(X) &= \langle l(X), r(X) \rangle = \sum_{i=1}^{6} t_i X^i.
\end{aligned}$$

   $\mathcal{P}$ computes $\forall i \in \mathcal{S} : T_i = g^{t_i} h^{\beta_i}$ and sends $\{T_i\}_{i=\mathcal{S}}$.
4. $\mathcal{V}$ sends challenge $x \xleftarrow{\$} \mathbb{Z}_p^*$.
5. $\mathcal{P}$ computes

$$\begin{aligned}
&\mathbf{l} = l(x),\ \mathbf{r} = r(x),\ \hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle, \\
&\beta_x = \beta_1 \cdot x + \sum_{i=3}^{6} \beta_i \cdot x^i, \mu = \alpha \cdot x + \beta \cdot x^2 + \rho \cdot x^3
\end{aligned}$$

   and sends $\hat{t}, \beta_x, \mu$.
6. $\mathcal{V}$ sends challenge $w \in \mathbb{Z}_q$.
7. $\mathcal{P}, \mathcal{V}$ both compute

$$\begin{aligned}
&\mathbf{h}' = \mathbf{h}^{\mathbf{y}^{-n}},\ u' = u^w, \\
&W_L = \mathbf{h}'^{\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L},\ W_R = \mathbf{g}^{\mathbf{y}^{-n} \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R)},\ W_O = \mathbf{h}'^{\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O}, \\
&P = A_I^x \cdot A_O^{x^2} \cdot \mathbf{h}'^{-\mathbf{y}^n} \cdot W_L^x \cdot W_R^x \cdot W_O \cdot S^{x^3}, \\
&P' = h^{-\mu} P (u')^{\hat{t}}
\end{aligned}$$

   and jointly execute the inner product argument protocol as

$$\langle \mathsf{InPrd}.\mathcal{P}((\mathbf{g}, \mathbf{h}', u', P'), (\mathbf{l}, \mathbf{r})), \mathsf{InPrd}.\mathcal{V}(\mathbf{g}, \mathbf{h}', u', P') \rangle.$$

8. $\mathcal{V}$ computes

$$\begin{aligned}
&\delta(y, z) = \left\langle \mathbf{y}^{-n} \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R), \mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L \right\rangle, \\
&R = g^{x^2(\delta(y,z) + \left\langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \right\rangle)} \cdot T_1^x \cdot \prod_{i=3}^{6} T_i^{x^i}, \\
&\mathsf{InPrd}.\mathcal{V}(\mathbf{g}, \mathbf{h}', u', P') \to b
\end{aligned}$$

   and returns 1 iff $b = 1$ and $g^{\hat{t}} h^{\beta_x} = R$.

Simulator 1: $\mathcal{S}_{\mathsf{BP}}$

The algebraic simulator $\mathcal{S}_{\mathsf{BP}}$ is given as input:

$$\mathsf{pp} = (n, Q, g, h, u, \mathbf{g}, \mathbf{h}),\ \mathsf{x} = (\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O, \mathbf{c})$$

The transcript is simulated as follows where the difference with the original simulator is marked in orange:

1. $x, y, w, z \xleftarrow{\$} \mathbb{Z}_p$
2. $\beta_x, \mu \xleftarrow{\$} \mathbb{Z}_p$
3. $\mathbf{l}, \mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^n$
4. $\hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle$
5. $\rho_I, \rho_O, t_3, t_4, t_5, t_6, \beta_3, \beta_4, \beta_5, \beta_6 \xleftarrow{\$} \mathbb{Z}_p$
6. $A_I = g^{\rho_I}, A_O = u^{\rho_O}$
7. $T_i = g^{t_i} h^{\beta_i}$ for $i \in \{3, 4, 5, 6\}$
8. $\mathbf{h}' = \mathbf{h}^{\mathbf{y}^{-n}}, \ u' = u^w$
9. $W_L = \mathbf{h}'^{\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_L}, \ W_R = \mathbf{g}^{\mathbf{y}^{-n} \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_R)}, \ W_O = \mathbf{h}^{\mathbf{y}^{-n} \circ (\mathbf{z}_{[1:]}^{Q+1} \cdot \mathbf{W}_O)}$
10. $S = \left(A_I^x \cdot A_O^{x^2} \cdot \mathbf{g}^{\mathbf{l}} \cdot (\mathbf{h}')^{-\mathbf{y}^n - \mathbf{r}} \cdot W_L^x \cdot W_R^x \cdot W_O \cdot h^{-\mu} \right)^{-x^{-3}}$
11. $T_1 = \left(h^{-\beta_x} \cdot g^{x^2 \cdot (\delta(y,z) + \langle \mathbf{z}_{[1:]}^{Q+1}, \mathbf{c} \rangle) - \hat{t}} \cdot \prod_{i=3}^{6} T_i^{x^i} \right)^{-x^{-1}}$
12. $\mathcal{T} = (S, A_I, A_O; y, z; \{T_i\}_{i \in \mathcal{S}}; x; \hat{t}, \beta_x, \mu; w; \mathbf{l}, \mathbf{r})$
13. Output $[\mathcal{T}]$

4.1 Algebraic Simulation

In Simulator 1 we define an algebraic simulator $\mathcal{S}_{\mathsf{BP}}$ for BP which is going to be used in both the proof of HVZK and SR-UR. The simulator $\mathcal{S}_{\mathsf{BP}}$ essentially works as the simulator from [BBB+18], except that, since it needs to explicitly output group representation for each simulated element, it will generate A_I, A_O as well as the T_i's by learning their discrete logarithm in bases g, h, u instead of generically sampling random group elements like in the original proof. This makes no difference for the ZK claim and makes the proof of SR-UR simpler. Note that, since the simulator picks all the challenges at random in the first step, the simulator can easily be changed to satisfy the stronger *special* HVZK. However, by defining the simulator like this we can reuse it in both of the following claims. Note also that while the output of the simulator does not explicitly contain the group representation (t_1, β_1) of T_1 w.r.t base (g, h), it is possible to compute these values from the output of the simulator.

Remark 5. The simulator for the recursive version of Bulletproof e.g., the one that calls InPrd instead of sending $\mathbf{l}, \mathbf{r}$ directly, can easily be constructed from the simulator above by running the InPrd protocol on $\mathbf{l}, \mathbf{r}$. The algebraic simulator also outputs the representation for the elements L_i, R_i generated during this protocol and this representation will be used explicitly in the proof later.

Claim 1. *The protocol* BP *(Protocol 1) is perfect HVZK with algebraic simulator* $\mathcal{S}_{\mathsf{BP}}$ *(Simulator 1) .*

Proof. The claim follows directly from the proof of HVZK in [BBB+18] by observing that the way $A_I, A_O, T_3, T_4, T_5, T_6$ are generated in our and their simulator produces the exact same distribution (in their case they are sampled as random elements from the group; in ours, we generate them by raising generators to random exponents, and those are not re-used anywhere else).

4.2 State-restoration Unique Responses

The following claim is crucial for invoking our generic result from Theorem 4. We remind the reader that proving uniqueness of the randomized commitments T_i's is made possible thanks to our relaxed definition: if the adversary was allowed to control both transcripts, it would be trivial to break the (strong) unique response by honestly executing the prover algorithm twice with known witness and by committing to t_i using distinct randomnesses β_i and β_i'. Our proof below on the other hand argues that a cheating prover in SR-UR has a hard time forging T_i once one of the transcripts has been fixed by a simulator. In other words, they cannot reuse parts of simulated proofs without knowing how the simulated messages were generated. This is true even for true statements where the prover might know the witness.

Claim 2. *Protocol* BP *(Protocol 1) satisfies state-restoration unique response (SR-UR) with respect to* $\mathcal{S}_{\mathsf{BP}}$ *(Simulator 1) in the AGM, under the assumption that solving the discrete-log relation is hard. That is, for every PPT adversary* $\mathcal{A}_{ur}$ *against SR-UR of* BP *that makes* q *queries to* $\mathbf{O}_{\mathsf{ext}}$ *(Fig. 6), there exists a PPT adversary* $\mathcal{A}$ *against DL-REL such that,*

$$\mathbf{Adv}^{\mathsf{SR\text{-}UR}}_{\mathsf{BP}}(\mathcal{A}_{ur}, \mathcal{S}_{\mathsf{BP}}) \leq \mathbf{Adv}^{\mathsf{DL\text{-}REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q}{(p-1)}.$$

Proof. Given an algebraic adversary for SR-UR-game $\mathcal{A}_{ur} = (\mathcal{A}_1, \mathcal{A}_2)$ for protocol BP (Fig. 6), we construct an adversary, $\mathcal{A}$, who breaks the discrete-log relation.

$\mathcal{A}$, upon receiving a discrete-log relation challenge interacts with $\mathcal{A}_{ur}$ as follows: It first runs $\mathcal{A}_1(\mathsf{pp})$ (where pp includes all the generators from the discrete-log relation assumption) to receive an instance $[x]$ and st. $\mathcal{A}$ then invokes the simulator $\mathcal{S}_{\mathsf{BP}}$ on $[x]$ to receive a transcript $\widetilde{\mathcal{T}}$. $\mathcal{A}$ then runs $\mathcal{A}_2$ on $\widetilde{\mathcal{T}}$ and st. Queries to the SR-UR-oracle $\mathbf{O}_{\mathsf{ext}}$ are handled by $\mathcal{A}$ locally as in the SR-UR game, by sampling random challenges and forwarding to $\mathcal{A}_2$. $\mathcal{A}$ locally records the tree of transcripts. Note that when $\mathcal{A}_{ur}$ queries $\mathbf{O}_{\mathsf{ext}}$, it also submits the group representation in terms of all groups elements seen so far. Moreover, the simulator $\mathcal{S}_{\mathsf{BP}}$ is algebraic, and therefore $\mathcal{A}$ can efficiently recover all representation for elements in $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ into an equivalent representation purely in terms of $\mathbf{g}, \mathbf{h}, g, h, u$ which will be used to break the discrete-logarithm assumption.

Since $\mathcal{A}_{ur}$ wins the SR-UR game $[\mathcal{T}]$ is an accepting transcript for statement $[x]$ which is different from $[\widetilde{\mathcal{T}}]$, but has a common prefix. Therefore, at least the first two messages must be equal. In particular, $\mathcal{S}_{\mathsf{BP}}$ outputs transcript of the form

$$\widetilde{\mathcal{T}} = \left(\tilde{A}_I, \tilde{A}_O, \tilde{S}; \tilde{y}, \tilde{z}; (\tilde{T}_i)_{i\in\mathcal{S}}; \tilde{x}, \tilde{\beta}_x, \tilde{\mu}, \tilde{\hat{t}}, \tilde{w}, \tilde{L}_1, \tilde{R}_1, \tilde{x}_1, \ldots, \tilde{L}_m, \tilde{R}_m, \tilde{x}_m, \tilde{a}, \tilde{b}\right)$$

and $\mathcal{A}_{ur}$ outputs transcripts of the form

$$\mathcal{T} = (\tilde{A}_I, \tilde{A}_O, \tilde{S}; \tilde{y}, \tilde{z}; (T_i)_{i\in\mathcal{S}}; x, \beta_x, \mu, \hat{t}, w, L_1, R_1, x_1, \ldots, L_m, R_m, x_m, a, b)$$

where we denote $m = \log(n)$.

We now proceed with a case by case analysis based on the first message in $\mathcal{T}$ which is different from $\widetilde{\mathcal{T}}$.

If $\tilde{T}_i \neq T_i$ **for some** $i \in \mathcal{S}$, then the verification equation satisfied by $\mathcal{T}$ is

$$\begin{aligned}
&(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} \\
&\quad = \left(\prod_{i=1}^{m} L_i^{x_i^2}\right) \cdot \left(\prod_{i=1}^{m} R_i^{x_i^{-2}}\right) \cdot h^{-\mu} \cdot \tilde{A}_I^{x} \cdot \tilde{A}_O^{x^2} \cdot (\tilde{\mathbf{h}}')^{-\tilde{\mathbf{y}}^n} \cdot \tilde{W}_L^{x} \cdot \tilde{W}_R^{x} \cdot \tilde{W}_O \cdot \tilde{S}^{x^3} \cdot (u')^{\hat{t}}.
\end{aligned}$$

(The values $\tilde{W}_{(\cdot)}$ and $\tilde{\mathbf{h}'}$ are also marked as $\tilde{(\cdot)}$ to remind the reader that they are the same in both $\mathcal{T}$ and $\widetilde{\mathcal{T}}$. Remember that $\mathbf{g}^{(m)}, \mathbf{h}^{(m)}$ are different in the two transcripts and they are generated as part of the InPrd). Dividing it by the verification equation for the simulated transcript, we get

$$\begin{aligned}
&(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} \cdot (\tilde{\mathbf{g}}^{(m)})^{-\tilde{a}} (\tilde{\mathbf{h}}^{(m)})^{-\tilde{b}} (\tilde{u}')^{-\tilde{a}\tilde{b}} \qquad (6)\\
&\quad = \left(\prod_{i=1}^{m} L_i^{x_i^2}\right) \left(\prod_{i=1}^{m} \tilde{L}_i^{-\tilde{x}_i^2}\right) \left(\prod_{i=1}^{m} R_i^{x_i^{-2}}\right) \left(\prod_{i=1}^{m} \tilde{R}_i^{-\tilde{x}_i^{-2}}\right) \\
&\qquad \cdot h^{-(\mu-\tilde{\mu})} \cdot \tilde{A}_I^{x-\tilde{x}} \cdot \tilde{A}_O^{x^2-\tilde{x}^2} \cdot \tilde{W}_L^{x-\tilde{x}} \cdot \tilde{W}_R^{x-\tilde{x}} \cdot \tilde{S}^{x^3-\tilde{x}^3} \cdot (u')^{\hat{t}} \cdot (\tilde{u}')^{-\hat{t}}. \qquad (7)
\end{aligned}$$

We rearrange the exponents w.r.t. the generators $(g, h, \mathbf{g}, \mathbf{h}, u)$. Let us focus on the exponent of g. The only elements with a non-zero component for g are: the simulated $\tilde{A}_I$ and $\tilde{S}$ that have ρ_I and $-\rho_I \tilde{x}^{-2}$ in the exponents of g, respectively; and L_i (resp. R_i) with g-component $l_{i,g}$ (resp. $r_{i,g}$) submitted by the adversary during the oracle queries. Then the exponent of g in (7) is

$$\sum_{i=1}^{m} l_{i,g} x_i^2 + \sum_{i=1}^{m} r_{i,g} x_i^{-2} - \rho_I \tilde{x}^{-2} x^3 + \rho_I x. \qquad (8)$$

If (8) is non-zero then we find a non-trivial DL solution since the left-hand side of 7 has g-component 0. Now we argue that (8) vanishes with negligible probability. Since the state-restoration adversary makes queries to $\mathbf{O}_{\mathsf{ext}}$ in order (e.g., it cannot query a transcript whose prefix has not been queried yet), the

challenges $x, x_1, \ldots, x_m$ are also assigned in order. Suppose the first m variables are fixed to $x, x_1, \ldots, x_{m-1}$ and regard (8) as a univariate polynomial with indeterminate X_m. Define

$$e_g^{(m)}(X_m) = l_{m,g} X_m^2 + r_{m,g} X_m^{-2} + \sum_{i=1}^{m-1} l_{i,g} x_i^2 + \sum_{i=1}^{m-1} r_{i,g} x_i^{-2} - \rho_I \tilde{x}^{-2} x^3 + \rho_I x.$$

Then, by the Schwartz–Zippel Lemma (Lemma 1), if the polynomial $e_g^{(m)}(X_m)$ is non-zero, $e_g^{(m)}(x_m)$ vanishes with probability at most $4/(p-1)$ over the random choice of $x_m \in \mathbb{Z}_p$; if it is a zero-polynomial, it must be that the constant term of $e_g^{(m)}$ is 0. Hence, if the polynomial

$$e_g^{(m-1)}(X_{m-1}) = l_{m-1,g} X_{m-1}^2 + r_{m-1,g} X_{m-1}^{-2} + \sum_{i=1}^{m-2} l_{i,g} x_i^2 + \sum_{i=1}^{m-2} r_{i,g} x_i^{-2} - \rho_I \tilde{x}^{-2} x^3 + \rho_I x$$

is non-zero, $e_g^{(m-1)}(x_{m-1})$ vanishes with probability at most $4/(p-1)$ over the random choice of $x_{m-1} \in \mathbb{Z}_p$. Iterating the same argument, we are eventually tasked with showing $e_g^{(0)}(x) = -\rho_I \tilde{x}^{-2} x^3 + \rho_I x = 0$ with negligible probability. This only happens if (1) $\rho_I = 0$, i.e., $e_g^{(0)}(X)$ is a zero-polynomial, or (2) $e_g^{(0)}(x) = 0$ over the random choice of $x \in \mathbb{Z}_p$. The former happens with probability $1/(p-1)$ because ρ_I are uniformly chosen by the simulator; the latter happens with probability at most $3/(p-1)$.

If $\beta_x \neq \tilde{\beta}_x$ or $\hat{t} \neq \tilde{\hat{t}}$, then we have another transcript

$$\mathcal{T}_{\mathsf{BP}} = (\tilde{A}_I, \tilde{A}_O, \tilde{S}; \tilde{y}, \tilde{z}; (\tilde{T}_i)_{i \in \mathcal{S}}; \tilde{x}, \beta_x, \mu, \hat{t}, w, L_1, R_1, x_1, \ldots, L_m, R_m, x_m, a, b).$$

Since both simulated and adversarial transcripts satisfy the verification equation w.r.t. the same R, we have

$$g^{\hat{t}} h^{\beta_x} = R = g^{\tilde{\hat{t}}} h^{\tilde{\beta}_x}$$

which leads to a non-trivial DL relation.

If $\mu \neq \tilde{\mu}$, the analysis is similar to the case where $\tilde{T}_i \neq T_i$. The verification equation satisfied by $\mathcal{T}_{\mathsf{BP}}$ is

$$\begin{aligned}
&(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} \\
&\quad = \left(\prod_{i=1}^{m} L_i^{x_i^2} \right) \cdot \left(\prod_{i=1}^{m} R_i^{x_i^{-2}} \right) \cdot h^{-\mu} \cdot \tilde{A}_I^{\tilde{x}} \cdot \tilde{A}_O^{\tilde{x}^2} \cdot (\tilde{\mathbf{h}}')^{-\tilde{\mathbf{y}}^n} \cdot \tilde{W}_L^{\tilde{x}} \cdot \tilde{W}_R^{\tilde{x}} \cdot \tilde{W}_O \cdot \tilde{S}^{\tilde{x}^3} \cdot (u')^{\hat{t}}.
\end{aligned}$$

Dividing it by the verification equation for the simulated transcript, we get

$$\begin{aligned}
&(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} \cdot (\tilde{\mathbf{g}}^{(m)})^{-\tilde{a}} (\tilde{\mathbf{h}}^{(m)})^{-\tilde{b}} (\tilde{u}')^{-\tilde{a}\tilde{b}} \\
&= \left(\prod_{i=1}^{m} L_i^{x_i^2} \right) \left(\prod_{i=1}^{m} \tilde{L}_i^{-\tilde{x}_i^2} \right) \left(\prod_{i=1}^{m} R_i^{x_i^{-2}} \right) \left(\prod_{i=1}^{m} \tilde{R}_i^{-\tilde{x}_i^{-2}} \right) \cdot h^{-(\mu - \tilde{\mu})} \cdot (u')^{\hat{t}} \cdot (\tilde{u}')^{-\hat{t}}. \qquad (9)
\end{aligned}$$

We rearrange the exponents w.r.t. the generators $(g, h, \mathbf{g}, \mathbf{h}, u)$. Let us focus on the exponent of h. Then the exponent of h in (9) is

$$\sum_{i=1}^{m} l_{i,g} x_i^2 + \sum_{i=1}^{m} r_{i,g} x_i^{-2} - (\mu - \tilde{\mu}) \qquad (10)$$

where $l_{i,h}$ (resp. $r_{i,h}$) is the exponent of h available as group representation of L_i (resp. R_i) submitted by the adversary. Using the same argument as before, since the h-component in the left-hand side of 9 is 0, if $\mu \neq \tilde{\mu}$ we obtain non-trivial DL relation except with negligible probability.

If $L_i \neq \tilde{L}_i$ or $R_i \neq \tilde{R}_i$ This part of the proof uses similar techniques as the ones for Lemma 8 in [GT21], with the main difference that we explicitly show the equalities and constraints that must hold for all exponents of parameters $\mathbf{g}, \mathbf{h}, g, h, u$. For instance, we introduce polynomials $\ell^{\mathbf{g}}$ and $\ell^{\mathbf{h}}$ which are essential for the full analysis, but were absent from proof in [GT21].

Let the representations output by the adversary for L_i, R_i be

$$L_i = \prod_{j=1}^{n} \left(g_j^{l_{ig_j}} h_j^{l_{ih_j}} \right) g^{l_g} h^{l_h} u^{l_u} \text{ and } R_i = \prod_{j=1}^{n} \left(g_j^{r_{ig_j}} h_j^{r_{ih_j}} \right) g^{r_g} h^{r_h} u^{r_u}$$

and let $P' = \prod_{j=1}^{n} \left(g_j^{p'_{g_j}} h_j^{p'_{h_j}} \right) g^{p'_g} h^{p'_h} u^{p'_u}$ be the representation of P' which is same in both the transcript of the simulator and the one of the adversary. In what follows we prove that the exponents of L_i (resp. R_i) match those of $\tilde{L}_i$ (resp. $\tilde{R}_i$) for $i = 1, \ldots, m$ except with negligible probability and otherwise one can find non-trivial discrete-log relation. Let $\mathsf{bit}(k, i, t)$ be the function that return the bit k_i where $(k_1, \cdots, k_t)$ is the t-bit representation of k.

Since $\mathcal{T}$ is accepting, the outcome of $\mathsf{InPrd.V}$ should be 1, and therefore, the following must hold:

$$(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} = \left(\prod_{i=1}^{m} L_i^{x_i^2} \right) P' \left(\prod_{i=1}^{m} R_i^{x_i^{-2}} \right), \tag{11}$$

where $\mathbf{g}^{(m)}, \mathbf{h}^{(m)}$ are parameters for the last round, and a, b are the last round messages. All terms in this equality can be expressed in terms of $\mathbf{g}, \mathbf{h}, g, h, u$ and we can compute the tuple

$$(e_{\mathbf{g}}^{(2)}, e_{\mathbf{h}}^{(2)}, e_g^{(2)}, e_h^{(2)}, e_u^{(2)})$$

such that $\mathbf{g}^{e_{\mathbf{g}}^{(2)}} \mathbf{h}^{e_{\mathbf{h}}^{(2)}} g^{e_g^{(2)}} h^{e_g^{(2)}} u^{e_u^{(2)}} = 1$. We compute $e_{\mathbf{g}}^{(2)}, e_{\mathbf{h}}^{(2)}, e_g^{(2)}, e_h^{(2)}, e_u^{(2)}$ as in Equations 12 to 16. Note that if $\mathcal{T}$ is accepting, $(e_{\mathbf{g}}^{(2)}, e_{\mathbf{h}}^{(2)}, e_g^{(2)}, e_h^{(2)}, e_u^{(2)}) = (\mathbf{0}, \mathbf{0}, 0, 0, 0)$, otherwise we get a non-trivial discrete-log relation.

For k=0 to $n-1$:

$$e_{g_{k+1}}^{(2)} = 0 = \left(\sum_{i=1}^{m} (l_{ig_{1+k}} x_i^2 + r_{ig_{1+k}} x_i^{-2}) + p'_{g_{1+k}} \right) - a \cdot \left(\prod_{i=1}^{m} x_i^{(-1)^{1-\mathsf{bit}(k,i,m)}} \right) \tag{12}$$

$$e_{h_{k+1}}^{(2)} = 0 = \left(\sum_{i=1}^{m} (l_{ih_{1+k}} x_i^2 + r_{ih_{1+k}} x_i^{-2}) + p'_{h_{1+k}} \right) - by^{(-(k))} \cdot \left(\prod_{i=1}^{m} x_i^{(-1)^{\mathsf{bit}(k,i,m)}} \right) \tag{13}$$

$$e_u^{(2)} = 0 = \left(\sum_{i=1}^{m} (l_{iu} x_i^2 + r_{iu} x_i^{-2}) + p'_u \right) - w \cdot ab \tag{14}$$

$$e_g^{(2)} = 0 = \left(\sum_{i=1}^{m} (l_{ig} x_i^2 + r_{ig} x_i^{-2}) + p'_g \right) \tag{15}$$

$$e_h^{(2)} = 0 = \left(\sum_{i=1}^{m} (l_{ih} x_i^2 + r_{ih} x_i^{-2}) + p'_h \right) \tag{16}$$

In order to derive relation between values $l_{ig_j}, r_{ig_j}, l_{ih_j}, r_{ih_j}, u_i$, and the group representation of statement P', we will invoke Schwartz-Zippel lemma in a recursive way. It is convenient to define the following polynomials to invoke the lemma recursively. For all $t \in \{1, \ldots, m\}$, for all $j \in \{0, \ldots, n-1\}$,

$$f_{t,j}^{\mathbf{g}}(X) = l_{t,g_{1+j}} X^2 + r_{t,g_{1+j}} X^{-2} + p'_{g_{1+j}} + \sum_{i=1}^{k-1} \left(l_{i,g_{1+j}} x_i^2 + r_{i,g_{1+j}} x_i^{-2} \right),$$

$$f_{t,j}^{\mathbf{h}}(X) = l_{t,h_{1+j}} X^2 + r_{t,h_{1+j}} X^{-2} + p'_{h_{1+j}} + \sum_{i=1}^{k-1} \left(l_{i,h_{1+j}} x_i^2 + r_{i,h_{1+j}} x_i^{-2} \right),$$

and

$$f_t^u(X) = l_{t,u} X^2 + r_{t,u} X^{-2} + p'_u + \sum_{i=1}^{t-1} \left(l_{i,u} x_i^2 + r_{i,u} x_i^{-2} \right).$$

Combining different polynomials, one can eliminate a (and b) from Equation (12) (and similarly from (13)) and rewrite the resultant equation in terms of polynomial $f_{t,j}^{\mathbf{g}}$ (similarly, $f_{t,j}^{\mathbf{h}}$) to get: For $t \in \{1, \ldots, m\}$, $j \in \{0, \ldots, n/2^t - 1\}$,

$$f_{t,j}^{\mathbf{g}}(x_t) \cdot x_t^2 - f_{t,j+n/2^t}^{\mathbf{g}}(x_t) = 0 \tag{17}$$

and

$$f_{\log(n)}^{u}(x_{\log(n)}) - w \cdot f_{\log(n),j}^{\mathbf{g}}(x_{\log(n)}) \cdot f_{\log(n),j}^{\mathbf{h}}(x_{\log(n)}) = 0. \tag{18}$$

Since all the challenges are in order, we rewrite (17) as a univariate polynomial in terms of variable X_t:

$$f^{\mathbf{g}}_{t,j}(X_t) \cdot X_t^2 - f^{\mathbf{g}}_{t,j+n/2^t}(X_t) = 0. \tag{19}$$

(19) vanishes with probability at most $6/(p-1)$, and otherwise it is a zero polynomial. Equating each coefficient term to 0, we get

$$r_{t,g_{1+j}} = f^{\mathbf{g}}_{t-1,j+n/2^t}(x_{t-1}),\ l_{t,g_{1+j}} = 0,\ r_{t,g_{j+n/2^t}} = 0, \tag{20}$$

$$l_{t,g_{j+n/2^t}} = p'_{g_{1+j}} + \sum_{i=1}^{t-1}(l_{i,g_{1+j}}x_i^2 + r_{i,g_{1+j}}x_i^{-2}) = \ell^{\mathbf{g}}_{t-1,j}(x_{t-1}) \tag{21}$$

where the last term in (21) can be rewritten as a univariate polynomial:

$$\ell^{\mathbf{g}}_{t-1,j}(X) = l_{t-1,g_{1+j}}X^2 + r_{t-1,g_{1+j}}X^{-2} + p'_{g_{1+j}} + \sum_{i=1}^{t-2}(l_{ig_{1+j}}x_i^2 + r_{ig_{1+j}}x_i^{-2}).$$

Iterating a similar argument for all rounds, for $t = 1$ we get, $r_{1,g_{1+j}} = p'_{g_{1+j+n/2}}$ and $l_{1,g_{j+n/2}} = p'_{g_{1+j}}$. Similarly, arguing for polynomial $f^{\mathbf{h}}_{k,j}$, we get the condition:

$$f^{\mathbf{h}}_{t-1,j}(X_t) \cdot X_t^{-2} - f^{\mathbf{h}}_{t,j+n/2^t}(X_t) = 0. \tag{22}$$

Analogous to polynomial $\ell^{\mathbf{g}}_{t,j}$, we define

$$\ell^{\mathbf{h}}_{t,j}(X) = l_{t-1,h_{1+j}}X^2 + r_{t-1,h_{1+j}}X^{-2} + p'_{h_{1+j}} + \sum_{i=1}^{t-2}(l_{ih_{1+j}}x_i^2 + r_{ih_{1+j}}x_i^{-2}).$$

Equalities 19,22 gives following constraints:
For all $t \in \{2,\ldots,m\}$, for all $j \in \{0,\ldots,n/2-1\}$:

$$\begin{aligned}
&r_{tg_{1+j}} = f^{\mathbf{g}}_{t-1,j+n/2^t}(x_{t-1}),\ l_{tg_{1+j}} = 0,\ r_{t,g_{j+n/2^t}} = 0,\\
&l_{t,g_{j+n/2^t}} = \ell^{\mathbf{g}}_{t,j}(x_{t-1}),\ r_{th_{1+j}} = 0,\ l_{th_{1+j}} = f^{\mathbf{h}}_{t-1,j+n/2^t}(x_{t-1}) \cdot y^{n/2^t},\\
&l_{t,h_{1+j+n/2^t}} = 0,\ r_{t,h_{1+j+n/2^t}} = \ell^{\mathbf{h}}_{t,j}(x_{t-1})
\end{aligned} \tag{23}$$

For $t = 1$, for all $j \in \{0,\ldots,n/2-1\}$:

$$\begin{aligned}
&r_{1g_{1+j}} = p'_{g_{1+n/2}},\ l_{1g_{1+j}} = 0,\ r_{1,g_{j+n/2}} = 0,\\
&l_{1,g_{j+n/2}} = p'_{g_{1+j}},\ r_{1h_{1+j}} = 0, l_{1h_{1+j}} = p'_{h_{1+j+n/2}} \cdot y^{n/2},\\
&l_{1,h_{1+j+n/2}} = 0,\ r_{1,h_{1+j+n/2}} = p'_{h_{1+j}} \cdot y^{n/2}
\end{aligned} \tag{24}$$

Note that the output of polynomials $f^{\mathbf{g}}_{k,j}, f^{\mathbf{h}}_{k,j}, \ell^{\mathbf{g}}_{k,j}, \ell^{\mathbf{h}}_{k,j}$ are deterministic given challenges $(x_1,\ldots,x_k)$. Also note, values $p'_{\mathbf{g}},\ldots,p'_u$ are fixed as they are equal to the representation output by the simulator. Hence, values for $r_{i,g_{1+j}}, r_{i,h_{1+j}}$, $l_{i,g_{1+j}}$ and $l_{i,h_{1+j}}$ (in Equation 24) are fixed given previous round challenges.

Now, consider exponents for generators g, h and u. Since equations (14,16,15) hold, using Schwartz-Zippel lemma recursively, it can be shown that $l_{i,u}, r_{i,u} = 0, l_{i,g}, r_{i,g}, l_{i,h} = r_{i,h} = 0$.

Note that, for a honest execution of InPrd, the exponents for L_i, R_i are derived using constraints in (24). Thus, L_i, R_i cannot differ from $\tilde{L}_i, \tilde{R}_i$.

Concrete advantage of the adversary. This analysis comes directly from the Bad Challenge analysis for ACSPf in [GT21]. For the case $T_i \neq \tilde{T}_i$, the adversary succeeds in forging if any one of the polynomials $e_g^{(0)},\ldots,e_g^{(m)}$ vanishes. Using union bound, this happens with probability $4(m+1)/(p-1)$. Similarly, for the case $\mu \neq \tilde{\mu}$, we break discrete-log relation except with probability: $4m+1/(p-1)$. Now, consider the case, $L_i \neq \tilde{L}_i$. The adversary succeeds in forging a proof for a false statement if they were lucky enough to get a challenge x_i such that equations 19, 22 and 18 vanish at x_i. This means, for round $t \in \{1,\ldots,m = \log(n)\}$, if any of the $\sum_{i=1}^{t-1} 2n/2^t$ polynomials of degree at most 4, $2n/2^t$ polynomials of degree at most 6, and one polynomial of degree at most 8, vanish, i.e., adversary succeeding in forging a proof, which turns out to be at most $(14n+8)/(p-1)$. Note that the adversary can query $\mathbf{O}_{\mathsf{ext}}$ for SR-UR q times. It is enough to take max of all case-by-case probabilities to get an upper bound for the probability of the adversary succeeding in forging a proof. This is because all the cases are sequential and the adversary succeeds in forging unless we break discrete-log relation for the very first case that the adversary exploits. Thus, adversary succeeds in forging a proof with probability at most $(14n+8)q/(p-1)$. □

Combining the results from Theorem 4 and Claim 2, we get the following corollary.

Corollary 1. *Fiat-Shamir transform of* BP *satisfies FS-SIM-EXT with respect to a canonical simulator* $\mathcal{S}_{\mathsf{FS\text{-}BP}}$ *corresponding to the algebraic simulator* $\mathcal{S}_{\mathsf{BP}}$*. Concretely, there exists an efficient FS-SIM-EXT extractor* $\mathcal{E}^*$ *for* FS-BP *such that for every non-uniform algebraic prover* $\mathcal{P}^*_{\mathsf{alg}}$ *against* FS-BP *that makes* q_1 *random oracle queries and* q_2 *simulation queries, and for every distinguisher* $\mathcal{D}^*$*, there exists a non-uniform adversary* $\mathcal{A}$ *against DL-REL with the property that for all* $\lambda \in \mathbb{N}^+$*,*

$$\begin{aligned}&\mathbf{Adv}^{\textit{FS-SIM-EXT}}_{\mathsf{FS\text{-}BP},\mathcal{R}}(\mathcal{S}_{\mathsf{FS\text{-}BP}}, \mathcal{E}^*, \mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*, \lambda)\\ &\leq \left(\mathbf{Adv}^{\textit{DL-REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q_1}{(p-1)}\right) + q_2 \cdot \left(\mathbf{Adv}^{\textit{DL-REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q_2}{(p-1)}\right) + \frac{(q_2+1)(q_1+1)}{|\mathsf{Ch}_{i_0}|}\end{aligned}$$

where $i_0 \in [1, r]$ *is the round with the smallest challenge set* Ch_{i_0}*.*

Acknowledgment

The authors are grateful to Thomas Attema, Matteo Campanelli, Jelle Don, Serge Fehr, Ashrujit Ghoshal, Christian Majenz, Stefano Tessaro, and anonymous reviewers of EUROCRYPT 2022 for helpful comments and insightful discussions. This research was supported by: the Concordium Blockchain Research Center, Aarhus University, Denmark; the Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM); the European Research Council (ERC) under the European Unions's Horizon 2020 research and innovation programme under grant agreement No 803096 (SPEC); Core Research Grant CRG/2020/004488, SERB, Department of Science and Technology.

References

ABK+21. M. Abdalla, M. Barbosa, J. Katz, J. Loss, and J. Xu. Algebraic adversaries in the universal composability framework. In *ASIACRYPT 2021*, vol. 13092 of *LNCS*, pp. 311–341. Springer, 2021.

AFK21. T. Attema, S. Fehr, and M. Klooß. Fiat-shamir transformation of multi-round interactive proofs. Cryptology ePrint Archive, Report 2021/1377, 2021. https://ia.cr/2021/1377.

ARS20. B. Abdolmaleki, S. Ramacher, and D. Slamanig. Lift-and-shift: Obtaining simulation extractable subversion and updatable SNARKs generically. In *ACM CCS 2020*, pp. 1987–2005. ACM Press, 2020.

BBB+17. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Report 2017/1066, 2017. https://eprint.iacr.org/2017/1066.

BBB+18. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pp. 315–334. IEEE Computer Society Press, 2018.

BCCJG16. J. Bootle, A. Cerulli, P. Chaidos, and C. P. Jens Groth. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. Cryptology ePrint Archive, Report 2016/263, 2016. https://eprint.iacr.org/2016/263.

BCS16. E. Ben-Sasson, A. Chiesa, and N. Spooner. Interactive oracle proofs. In *TCC 2016-B, Part II*, vol. 9986 of *LNCS*, pp. 31–60. Springer, Heidelberg, 2016.

BKSV20. K. Baghery, M. Kohlweiss, J. Siim, and M. Volkhov. Another look at extraction and randomization of groth's zk-snark. Cryptology ePrint Archive, Report 2020/811, 2020. https://ia.cr/2020/811.

BMM+19. B. Bünz, M. Maller, P. Mishra, N. Tyagi, and P. Vesely. Proofs for inner pairing products and applications. Cryptology ePrint Archive, Report 2019/1177, 2019. https://eprint.iacr.org/2019/1177.

BN06. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS 2006*, pp. 390–399. ACM Press, 2006.

CCH+18. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, and R. D. Rothblum. Fiat-shamir from simpler assumptions. Cryptology ePrint Archive, Report 2018/1004, 2018. https://eprint.iacr.org/2018/1004.

CCH+19. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs. Fiat-Shamir: from practice to theory. In *51st ACM STOC*, pp. 1082–1090. ACM Press, 2019.

CDS94. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO'94*, vol. 839 of *LNCS*, pp. 174–187. Springer, Heidelberg, 1994.

DDN91. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography (extended abstract). In *23rd ACM STOC*, pp. 542–552. ACM Press, 1991.

DDO+01. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero knowledge. In *CRYPTO 2001*, vol. 2139 of *LNCS*, pp. 566–598. Springer, Heidelberg, 2001.

DFM20. J. Don, S. Fehr, and C. Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In *CRYPTO 2020, Part III*, vol. 12172 of *LNCS*, pp. 602–631. Springer, Heidelberg, 2020.

DW14. C. Decker and R. Wattenhofer. Bitcoin transaction malleability and MtGox. In *ESORICS 2014, Part II*, vol. 8713 of *LNCS*, pp. 313–326. Springer, Heidelberg, 2014.

Fis05. M. Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO 2005*, vol. 3621 of *LNCS*, pp. 152–168. Springer, Heidelberg, 2005.

FKL18. G. Fuchsbauer, E. Kiltz, and J. Loss. The algebraic group model and its applications. In *CRYPTO 2018, Part II*, vol. 10992 of *LNCS*, pp. 33–62. Springer, Heidelberg, 2018.

FKMV12. S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the Fiat-Shamir transform. In *INDOCRYPT 2012*, vol. 7668 of *LNCS*, pp. 60–79. Springer, Heidelberg, 2012.

FPS20. G. Fuchsbauer, A. Plouviez, and Y. Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In *EUROCRYPT 2020, Part II*, vol. 12106 of *LNCS*, pp. 63–95. Springer, Heidelberg, 2020.

FS87. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86*, vol. 263 of *LNCS*, pp. 186–194. Springer, Heidelberg, 1987.

GK03. S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pp. 102–115. IEEE Computer Society Press, 2003.

GM17. J. Groth and M. Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In *CRYPTO 2017, Part II*, vol. 10402 of *LNCS*, pp. 581–612. Springer, Heidelberg, 2017.

GMR85. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pp. 291–304. ACM Press, 1985.

GO07. J. Groth and R. Ostrovsky. Cryptography in the multi-string model. In *CRYPTO 2007*, vol. 4622 of *LNCS*, pp. 323–341. Springer, Heidelberg, 2007.

Gro06. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT 2006*, vol. 4284 of *LNCS*, pp. 444–459. Springer, Heidelberg, 2006.

Gro16. J. Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT 2016, Part II*, vol. 9666 of *LNCS*, pp. 305–326. Springer, Heidelberg, 2016.

GT21. A. Ghoshal and S. Tessaro. Tight state-restoration soundness in the algebraic group model. In *CRYPTO 2021, Part III*, vol. 12827 of *LNCS*, pp. 64–93, Virtual Event, 2021. Springer, Heidelberg.

GWC19. A. Gabizon, Z. J. Williamson, and O. Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. https://eprint.iacr.org/2019/953.

Hol19. J. Holmgren. On round-by-round soundness and state restoration attacks. Cryptology ePrint Archive, Report 2019/1261, 2019. https://eprint.iacr.org/2019/1261.

KLX22. J. Kastner, J. Loss, and J. Xu. On pairing-free blind signature schemes in the algebraic group model. In *PKC 2022*, vol. 13178 of *LNCS*, pp. 468–497. Springer, 2022.

KZ21. M. Kohlweiss and M. Zając. On simulation-extractability of universal zksnarks. Cryptology ePrint Archive, Report 2021/511, 2021. https://eprint.iacr.org/2021/511.

MBKM19. M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In *ACM CCS 2019*, pp. 2111–2128. ACM Press, 2019.

Pas03. R. Pass. On deniability in the common reference string and random oracle model. In *CRYPTO 2003*, vol. 2729 of *LNCS*, pp. 316–337. Springer, Heidelberg, 2003.

Sah99. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pp. 543–553. IEEE Computer Society Press, 1999.

Unr15. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015, Part II*, vol. 9057 of *LNCS*, pp. 755–784. Springer, Heidelberg, 2015.

Unr17. D. Unruh. Post-quantum security of Fiat-Shamir. In *ASIACRYPT 2017, Part I*, vol. 10624 of *LNCS*, pp. 65–95. Springer, Heidelberg, 2017.

Wik21. D. Wikström. Special soundness in the random oracle model. Cryptology ePrint Archive, Report 2021/1265, 2021. https://ia.cr/2021/1265.

A Generalizing the results from [GT21] in the adaptive setting

As pointed out, our definitions of FS-EXT and aSR-WEE are the adaptive version of definitions that appear in [GT21], i.e., FS-EXT-1 and SR-WEE. The results in [GT21], namely the proof for SR-WEE security of BP (Theorem 6), and SR-WEE security implies FS-EXT-1 (Theorem 2), are presented for the case where the adversary submits the statement beforehand. We notice that these proofs are applicable even for the adaptive case where the adversary need not submit a statement beforehand. This observation will imply Theorem 2 and Theorem 3 stated in preliminaries.

To see this, notice the proof strategy for Theorem 6 in [GT21]. It uses two functions e (which is the extractor) and h (which solves discrete-log relation between the generators). The central claim in the proof is that, given an accepting transcript, either e outputs a valid witness or h outputs a non-trivial discrete-log relation. Both these functions take as input the statement and the transcript. Their description does not depend on the statement being fixed beforehand. Theorem 2 is proved by constructing an adversary for SR-WEE security of BP given an adversary for FS-EXT-1. Here, the idea is to forward all random oracle queries to $\mathbf{O}_{\mathsf{ext}}$ in the SR-WEE definition. First, the adversary for FS-EXT-1 must send a statement that is forwarded to SR-WEE and is used for intialization, i.e. the step $([x], \mathsf{st}_{\mathcal{P}}) \leftarrow \mathcal{P}^*_{\mathsf{alg}}(\mathsf{pp})$ in definition of SR-WEE (Fig.3) in [GT21]. However, this proof strategy does not depend on the statement being fixed. We can extend the result for adaptive adversaries by first changing the SR-WEE definition to be adaptive and then changing the proof to allow $\mathcal{P}^*_{\mathsf{alg}}$ to query the random oracle for statement and transcript pairs. In fact, this argument can be seen as an alternative to their Theorem 3, which directly bounds the advantage of FS-EXT-2 (equivalent to Definition 10) without assuming aSR-WEE.

B Proof for Theorem 3

Proof. We prove via standard hybrid arguments.

G_0 This game is identical to $\mathsf{EXT\text{-}1}^{\mathsf{H},\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}}}(\lambda)$. We have

$$\Pr[\mathsf{G}_0(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] = \Pr[\mathsf{EXT\text{-}1}^{\mathsf{H},\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}_{\Pi_{\mathsf{FS}}}(\lambda)]$$

G_1 This game is identical to G_0 except that it aborts if $d = 1$ (i.e., $(x^*, \mathcal{T}^*)$ is accepting) while any of the challenges in $\mathcal{T}^*$ have been queried in wrong order or have never been queried by $\mathcal{P}^*_{\mathsf{alg}}$ at all. According to the proof for Theorem 2 of [GT21], this happens with probability at most $(q+1)/|\mathsf{Ch}_{i_0}|$. Thus we have

$$\left|\Pr[\mathsf{G}_0(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] - \Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)]\right| \leq (q+1)/|\mathsf{Ch}_{i_0}|.$$

Constructing $\mathcal{P}_{\mathsf{alg}}$ and $\mathcal{D}$ for aSR-WEE. We now construct an aSR-WEE adversary $\mathcal{P}_{\mathsf{alg}}$ and a distinguisher $\mathcal{D}$. $\mathcal{P}_{\mathsf{alg}}$ plays an aSR-WEE game while internally simulating the view of $\mathcal{P}^*_{\mathsf{alg}}$ in the game G_1 as follows.

- On receiving pp in the aSR-WEE games, it $\mathcal{P}_{\mathsf{alg}}$ forwards pp to $\mathcal{P}^*_{\mathsf{alg}}$.
- On receiving a random oracle query from $\mathcal{P}^*_{\mathsf{alg}}$, if all of its prefix have been queried in order $\mathcal{P}_{\mathsf{alg}}$ forwards that query to the oracle $\mathbf{O}_{\mathsf{ext}}$ in a aSR-WEE game to receive challenge and returns it to $\mathcal{P}^*_{\mathsf{alg}}$. Otherwise it lazily samples a challenge for $\mathcal{P}^*_{\mathsf{alg}}$.
- On receiving $([x^*], [\mathcal{T}^*])$ from $\mathcal{P}^*_{\mathsf{alg}}$, it checks whether it causes aborts in the game G_1. If that is the case $\mathcal{P}_{\mathsf{alg}}$ also aborts because it implies that the challenge values in $\mathcal{T}^*$ were not obtained by forwarding the corresponding queries to a aSR-WEE game and therefore $(x^*, \mathcal{T}^*)$ is not accepting in the aSR-WEE game.
- Otherwise, $\mathcal{P}_{\mathsf{alg}}$ outputs $(x^*, \mathcal{T}^*, \mathsf{st}_{\mathcal{P}_{\mathsf{alg}}})$ in the aSR-WEE game (together with group representations), where $\mathsf{st}_{\mathcal{P}_{\mathsf{alg}}} = \mathcal{Q}_1$.

The aSR-WEE distinguisher $\mathcal{D}$ internally invokes $\mathcal{D}^*$ on input $(\mathsf{st}_{\mathcal{P}^*_{\mathsf{alg}}}, x^*, \mathcal{T}^*, \mathcal{Q}_1)$ and outputs whatever $\mathcal{D}^*$ returns.

By construction we have that

$$\Pr[\mathsf{G}_1(\mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*)] = \Pr[\mathsf{aWEE\text{-}1}^{\mathcal{P}_{\mathsf{alg}},\mathcal{D}}_{\Pi}(\lambda)]$$

Constructing $\mathcal{E}^*$ for FS-EXT. We define an extractor $\mathcal{E}^* = (\mathcal{E}^*_0, \mathcal{E}^*_1)$ using an aSR-WEE extractor $\mathcal{E}$. $\mathcal{E}^*_1$ answers the random oracle queries made by $\mathcal{P}^*_{\mathsf{alg}}$ as $\mathcal{P}_{\mathsf{alg}}$ would, by using the responses from $\mathcal{E}$. $\mathcal{E}^*_0$ outputs

whatever $\mathcal{E}$ returns on input $(\mathsf{st}_{\mathcal{E}}, [x^*], [\mathcal{T}^*])$. By construction, $\mathcal{E}^*$ succeeds in extraction if and only if $\mathcal{E}$ does so in the game $\mathsf{aWEE\text{-}0}_{\Pi}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)$. Thus we have

$$\Pr[\mathsf{EXT\text{-}0}_{\Pi_{\mathsf{FS}},\mathcal{R}}^{\mathcal{E}^*,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}(\lambda)] = \Pr[\mathsf{aWEE\text{-}0}_{\Pi,\mathcal{R}}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)]$$

Putting together we obtain

$$\begin{aligned}
&\left|\Pr[\mathsf{EXT\text{-}1}_{\Pi_{\mathsf{FS}}}^{\mathsf{H},\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}(\lambda)] - \Pr[\mathsf{EXT\text{-}0}_{\Pi_{\mathsf{FS}},\mathcal{R}}^{\mathcal{E}^*,\mathcal{P}^*_{\mathsf{alg}},\mathcal{D}^*}(\lambda)]\right| \\
\leq &\left|\Pr[\mathsf{aWEE\text{-}1}_{\Pi}^{\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)] - \Pr[\mathsf{aWEE\text{-}0}_{\Pi,\mathcal{R}}^{\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D}}(\lambda)]\right| + (q+1)/|\mathsf{Ch}_{i_0}| \\
= &\mathbf{Adv}_{\Pi,\mathcal{R}}^{\mathsf{aSR\text{-}WEE}}(\mathcal{E},\mathcal{P}_{\mathsf{alg}},\mathcal{D},\lambda) + (q+1)/|\mathsf{Ch}_{i_0}|
\end{aligned}$$

□

C Inner Product Argument

In this section we recall the inner product argument protocol from BP [BBB+17].

Protocol 2: InPRd

$$\mathcal{R}_{\mathsf{InPrd}} = \left\{((n,\mathbf{g},\mathbf{h},u),(P),(\mathbf{a},\mathbf{b})) \mid P = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}}u^c \wedge c = \langle \mathbf{a},\mathbf{b}\rangle\right\}$$

1. $\mathcal{P}$ and $\mathcal{V}$ initialize $\mathbf{g}^{(0)} = \mathbf{g}$, $\mathbf{h}^{(0)} = \mathbf{h}$, $n_0 = n$, $P_0 = P$. Additionally, $\mathcal{P}$ initializes $\mathbf{a}^{(0)} = \mathbf{a}$ and $\mathbf{b}^{(0)} = \mathbf{b}$.
2. For $i = \{1, \cdots, \log(n)\}$:
   (a) $\mathcal{P}, \mathcal{V}$ compute:
   $$n_i = n_{i-1}/2$$
   (b) $\mathcal{P}$ computes:
   $$\begin{aligned}
   c_L &= \left\langle \mathbf{a}^{(i)}_{[:n_i]}, \mathbf{b}^{(i)}{}_{[n_i:]} \right\rangle, c_R = \left\langle \mathbf{a}^{(i)}{}_{[n_i:]}, \mathbf{b}^{(i)}_{[:n_i]} \right\rangle \\
   L_i &= \left(\mathbf{g}^{(i-1)}_{[n_i:]}\right)^{\mathbf{a}^{(i)}[:n_i]} \left(\mathbf{h}^{(i-1)}_{[:n_i]}\right)^{\mathbf{b}^{(i)}[n_i:]} u^{c_L} \\
   R_i &= \left(\mathbf{g}^{(i-1)}_{[:n_i]}\right)^{\mathbf{a}^{(i)}[n_i:]} \left(\mathbf{h}^{(i-1)}_{[n_i:]}\right)^{\mathbf{b}^{(i)}[:n_i]} u^{c_R}
   \end{aligned}$$
   (c) $\mathcal{P}$ sends: L_i, R_i to $\mathcal{V}$.
   (d) $\mathcal{V}$ sends challenge $x_i \xleftarrow{\$} \mathbb{Z}_p^*$.
   (e) $\mathcal{P}, \mathcal{V}$ compute:
   $$\begin{aligned}
   \mathbf{g}^{(i)} &= \left(\mathbf{g}^{(i-1)}_{[:n_i]}\right)^{x_i^{-1}} \circ \left(\mathbf{g}^{(i-1)}_{[n_i:]}\right)^{x_i} \\
   \mathbf{h}^{(i)} &= \left(\mathbf{h}^{(i-1)}_{[:n_i]}\right)^{x_i} \circ \left(\mathbf{h}^{(i-1)}_{[n_i:]}\right)^{x_i^{-1}} \\
   P_i &= L_i^{x_i^2} P_{i-1} R_i^{x_i^{-2}}
   \end{aligned}$$
   (f) $\mathcal{P}$ computes:
   $$\begin{aligned}
   \mathbf{a}^{(i)} &= \mathbf{a}^{(i-1)}[:n_i]x_i + \mathbf{a}^{(i-1)}[n_i:]x_i^{-1} \\
   \mathbf{b}^{(i)} &= \mathbf{b}^{(i-1)}[:n_i]x_i^{-1} + \mathbf{b}^{(i-1)}[n_i:]x_i
   \end{aligned}$$
3. Let
   $$\begin{aligned}
   g &\leftarrow \mathbf{g}^{(\log(n))}, \quad h \leftarrow \mathbf{h}^{(\log(n))} \\
   a &\leftarrow \mathbf{a}^{(\log(n))}, \quad b \leftarrow \mathbf{b}^{(\log(n))}
   \end{aligned}$$

$\mathcal{P}$ sends: a, b to $\mathcal{V}$.

4. $\mathcal{V}$ checks:

$$P_{\log(n)} \stackrel{?}{=} g^a h^b u^{ab}$$

D Non-Malleability of Bulletproofs – Range Proofs

In this section we recall the range proof protocol from BP [BBB+17] and prove that they satisfy the SR-UR notion. The arguments follow closely the ones for the proofs for arithemtic circuits.

D.1 Range Proof Protocol

Protocol 3: RngPf

$$\mathcal{R}_{\mathsf{RngPf}} = \{((n, \mathbf{g}, \mathbf{h}, g, h, u), (V), (v, \gamma)) \mid V = g^v h^\gamma \wedge v \in [0, 2^n - 1]\}$$

1. $\mathcal{P}$ computes $\mathbf{a}_L \in \{0,1\}^n$ such that $\langle \mathbf{a}_L, \mathbf{2}^n \rangle = v$, and $\mathbf{a}_R = \mathbf{a}_L - \mathbf{1}^n$.
2. $\mathcal{P}$ samples $\alpha, \rho \xleftarrow{\$} \mathbb{Z}_p, \mathbf{s}_L, \mathbf{s}_R \xleftarrow{\$} \mathbb{Z}_p^n$, and sends to $\mathcal{V}$:

$$A = h^\alpha \mathbf{g}^{\mathbf{a}_L} \mathbf{h}^{\mathbf{a}_R}, \quad S = h^\rho \mathbf{g}^{\mathbf{s}_L} \mathbf{h}^{\mathbf{s}_R}$$

3. $\mathcal{V}$ sends challenges $y, z \xleftarrow{\$} \mathbb{Z}_p^*$.
4. $\mathcal{P}$ computes:

$$\begin{aligned} l(X) &= (\mathbf{a}_L - z \cdot \mathbf{1}^n) + \mathbf{s}_L \cdot X \\ r(X) &= \mathbf{y}^n \circ (\mathbf{a}_R + z \cdot \mathbf{1}^n + \mathbf{s}_R \cdot X) + z^2 \cdot \mathbf{2}^n \\ t(X) &= \langle l(X), r(X) \rangle = t_0 + t_1 \cdot X + t_2 \cdot X \end{aligned}$$

5. $\mathcal{P}$ samples $\beta_1, \beta_2 \xleftarrow{\$} \mathbb{Z}_p$, and sends to $\mathcal{V}$:

$$T_i = g^{t_i} h^{\beta_i}, i \in \{1, 2\}$$

6. $\mathcal{V}$ sends challenge $x \xleftarrow{\$} \mathbb{Z}_p^*$.
7. $\mathcal{P}$ computes:

$$\begin{aligned} &\mathbf{l} = l(x), \quad \mathbf{r} = r(x), \quad \hat{t} = \langle \mathbf{l}, \mathbf{r} \rangle \\ &\beta_x = \beta_2 \cdot x^2 + \beta_1 \cdot x + z^2 \gamma, \quad \mu = \alpha + \rho \cdot x \end{aligned}$$

8. $\mathcal{P}$ sends to $\mathcal{V}$:

$$\beta_x, \mu, \hat{t}$$

9. $\mathcal{V}$ sends challenge $w \xleftarrow{\$} \mathbb{Z}_p^*$.
10. $\mathcal{P}$ and $\mathcal{V}$ compute:

$$\begin{aligned} &\mathbf{h}' = \mathbf{h}^{\mathbf{y}^{-n}}, \quad u' = u^w \\ &P = A \cdot S^x \cdot \mathbf{g}^{-z \cdot \mathbf{1}^n} \cdot \mathbf{h}'^{z \cdot \mathbf{y}^n + z^2 \cdot \mathbf{2}^n}, P' = h^{-\mu} P (u')^{\hat{t}} \end{aligned}$$

11. $\mathcal{P}, \mathcal{V}$ execute inner product argument protocol as $\langle \mathsf{InPrd}.\mathcal{P}((\mathbf{g}, \mathbf{h}', u', P'), (\mathbf{l}, \mathbf{r})), \mathsf{InPrd}.\mathcal{V}(\mathbf{g}, \mathbf{h}', u', P') \rangle$.

12. $\mathcal{V}$ computes:

$$\delta(y,z) = (z-z^2)\cdot\langle \mathbf{1}^n, \mathbf{y}^n\rangle + z^2\cdot \mathbf{2}^n$$
$$R = V^{z^2} g^{\delta(y,z)} T_1^x T_2^{x^2}$$
$$\mathsf{InPrd}.\mathcal{V}(\mathbf{g}, \mathbf{h}', u', P') \to b$$

$\mathcal{V}$ returns 1 if $b = 1$ and $g^{\hat{t}} h^{\beta_x} = R$.

D.2 Algebraic Simulator

In this section we present an algebraic simulator $\mathcal{S}_{\mathsf{RngPf}}$ for RngPf. The simulator works similarly to [BBB+18] but now we explicitly mention how to generate elements A and T_2. This helps in simplifying the proof of SR-UR for RngPf.

Protocol 4: $\mathcal{S}_{\mathsf{RngPf}}$

The algebraic simulator $\mathcal{S}_{\mathsf{BP}}$ is given as input:

$$\mathsf{pp} = (n, Q, g, h, u, \mathbf{g}, \mathbf{h}), \quad \mathsf{x} = (V)$$

The transcript is simulated as follows:

1. $x, y, w, z \xleftarrow{\$} \mathbb{Z}_p$;
2. $\beta_x, \mu \xleftarrow{\$} \mathbb{Z}_p$;
3. $\mathbf{l}, \mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^n$;
4. $\hat{t} = \langle \mathbf{l}, \mathbf{r}\rangle$;
5. $\rho_A, t_2, \beta_2 \xleftarrow{\$} \mathbb{Z}_p$;[a]
6. $A = g^{\rho_A}, T_2 = g^{t_2} h^{\beta_2}$
7. $\mathbf{h}' = \mathbf{h}^{\mathbf{y}^{-n}}; u' = u^w$;
8. $S = \left(A \cdot \mathbf{g}^{-z\cdot\mathbf{1}^n - \mathbf{l}} \cdot (\mathbf{h}')^{z\cdot\mathbf{y}^n + z^2\cdot\mathbf{2}^n - \mathbf{r}} \cdot h^{-\mu}\right)^{-x^{-1}}$
9. $T_1 = \left(h^{-\beta_x} \cdot g^{(\delta(y,z)-\hat{t})} \cdot V^{z^2} \cdot T_2^{x^2}\right)^{-x^{-1}}$
10. $\mathcal{T} = (S, A; y, z; T_1, T_2; x; \hat{t}, \beta_x, \mu; w; \mathbf{l}, \mathbf{r})$;
11. Output $[\mathcal{T}]$.

[a] Steps 5,6 are the difference with the original simulator.

Claim 3. *The protocol* RngPf *is perfect* HVZK *(Definition 5) with algebraic simulator* $\mathcal{S}_{\mathsf{RngPf}}$ *(Protocol 4).*

Proof. Observer that elements A, and T_2 are distributed exactly as in the HVZK simulator for Range proofs in [BBB+18]. Thus, this claim follows from the HVZK proof in [BBB+18].

D.3 Unique Response for Range Proof

In this section we show that RngPf is SR-UR with respect to simulator $\mathcal{S}_{\mathsf{RngPf}}$.

Claim 4. *Protocol* RngPf *has state-restoration unique responses with respect to* $\mathcal{S}_{\mathsf{RngPf}}$ *in the AGM as in Definition 14, under the assumption that solving the discrete-log relation is hard. That is, for every PPT adversary* $\mathcal{A}_{ur}$ *against* SR-UR *of* RngPf *that makes* q *queries to* $\mathbf{O}_{\mathsf{ext}}$ *(Fig. 6), there exists a PPT adversary* $\mathcal{A}$ *against* DL-REL *such that,*

$$\mathbf{Adv}^{\text{SR-UR}}_{\mathsf{RngPf}}(\mathcal{A}_{ur}, \mathcal{S}_{\mathsf{RngPf}}) \le \mathbf{Adv}^{\text{DL-REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q}{(p-1)}$$

Proof. Given an algebraic adversary for SR-UR-game $\mathcal{A}_{ur} = (\mathcal{A}_1, \mathcal{A}_2)$ for protocol RngPf (Fig. 6), we construct an adversary, $\mathcal{A}$, who breaks the discrete-log relation.

$\mathcal{A}$, upon receiving a discrete-log relation challenge interacts with $\mathcal{A}_{ur}$ as follows: It runs $\mathcal{A}_1(\mathsf{pp})$ to receive an instance $[x]$ and st, and invokes the simulator $\mathcal{S}_{\mathsf{RngPf}}$ on $[x]$ to receive a transcript $\widetilde{\mathcal{T}}$. $\mathcal{A}$ then runs $\mathcal{A}_2$ on $\widetilde{\mathcal{T}}$ and st. Queries to the SR-UR-oracle $\mathbf{O}_{\mathsf{ext}}$ are handled by $\mathcal{A}$ locally as in the SR-UR game, by sampling random challenges and forwarding to $\mathcal{A}_2$. $\mathcal{A}$ locally records the tree of transcripts. Note that when $\mathcal{A}_{ur}$ queries $\mathbf{O}_{\mathsf{ext}}$, it also submits the group representation in terms of all groups elements seen so far, which can represented purely in terms of $\mathbf{g}, \mathbf{h}, g, h, u$ and will be used to break the discrete-logarithm assumption.

$\mathcal{S}_{\mathsf{RngPf}}$ outputs transcript of the form:

$$\widetilde{\mathcal{T}} = (\tilde{A}, \tilde{S}; \tilde{y}, \tilde{z}; \tilde{T}_1, \tilde{T}_2; \tilde{x}, \tilde{\beta}_x, \tilde{\mu}, \tilde{\hat{t}}, \tilde{w}, \tilde{L}_1, \tilde{R}_1, \tilde{x}_1, \ldots, \tilde{L}_m, \tilde{R}_m, \tilde{x}_m, \tilde{a}, \tilde{b})$$

and $\mathcal{A}_{ur}$ outputs transcripts of the form:

$$\mathcal{T} = (\tilde{A}, \tilde{S}; \tilde{y}, \tilde{z}; T_1, T_2; x, \beta_x, \mu, \hat{t}, w, L_1, R_1, x_1, \ldots, L_m, R_m, x_m, a, b).$$

Note that at least the first two round messages are the same in both $\mathcal{T}$ and $\widetilde{\mathcal{T}}$. In the following, we do a case by case analysis of $\mathcal{T}$ and $\widetilde{\mathcal{T}}$.

If $\tilde{T}_i \neq T_i$ for some $i \in \{1, 2\}$. Since transcript $\mathcal{T}$ is accepting, the InPrd sub-protocol returns $b = 1$.

$$\begin{aligned}(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} &= \left(\prod_{i=1}^{m} L_i^{x_i^2}\right) \cdot P' \cdot \left(\prod_{i=1}^{m} R_i^{x_i^{-2}}\right). \\ &= \left(\prod_{i=1}^{m} L_i^{x_i^2}\right) h^{-\mu} \cdot A \cdot S^x \cdot \mathbf{g}^{-z \cdot \mathbf{1}^n} \cdot \mathbf{h}'^{z \cdot \mathbf{y}^n + z^2 \cdot \mathbf{2}^n} \cdot (u')^{\hat{t}} \left(\prod_{i=1}^{m} R_i^{x_i^{-2}}\right)\end{aligned}$$

Let us focus on the exponent of generator g. We get $l_{i,g}$ and $r_{i,g}$ from L_i, R_i, respectively, ρ_A from A, and $\rho \cdot (-\tilde{x}^{-1})$ from S.

$$\sum_{i=1}^{m} l_{i,g} x_i^2 + \sum_{i=1}^{m} r_{i,g} x_i^{-2} - (\rho_A \cdot \tilde{x}^{-1}) \cdot x + \rho_A \tag{25}$$

If (25) is non-zero then we find a non-trivial discrete-log relation. Otherwise, it must be a zero polynomial. However, we show that (25) vanishes with negligible probability.

Note that challenges $x, x_1, \ldots, x_m$ are assigned in order. Let (25) be defined in terms of na univariate polynomial with indeterminate X_m:

$$e_g^{(m)}(X_m) = l_{m,g} X_m^2 + r_{m,g} X_m^{-2} + \sum_{i=1}^{m-1} l_{i,g} x_i^2 + \sum_{i=1}^{m-1} r_{i,g} x_i^{-2} - (\rho_A \cdot \tilde{x}^{-1}) \cdot x + \rho_A.$$

Iterating argument similar to the case: "**If $\tilde{T}_i \neq T_i$ for some $i \in \mathcal{S}$**" in Claim 2, we are eventually tasked with showing $e_g^{(0)}(x) = -(\rho_A \cdot \tilde{x}^{-1}) \cdot x + \rho_A = 0$. This only happens if $\rho_A = 0$, or $e_g^{(0)}(x) = 0$ over the random choice of $x \in \mathbb{Z}_p$. The former happens with probability $1/(p-1)$ because ρ_A is uniformly chosen by the simulator; the latter happens with probability at most $1/(p-1)$.

If $\beta_x \neq \tilde{\beta}_x$ or $\hat{t} \neq \tilde{\hat{t}}$. This case is the same as the analogous case in Claim 2.

If $\mu \neq \tilde{\mu}$. Similar to (Eq. (9)) in Claim 2, dividing the InPrd verification equation for $\mathcal{T}$ and $\tilde{\mathcal{T}}$, we get:

$$\begin{aligned}&(\mathbf{g}^{(m)})^a (\mathbf{h}^{(m)})^b (u')^{ab} \cdot (\tilde{\mathbf{g}}^{(m)})^{-\tilde{a}} (\tilde{\mathbf{h}}^{(m)})^{-\tilde{b}} (\tilde{u}')^{-\tilde{a}\tilde{b}} \\ &= \left(\prod_{i=1}^{m} L_i^{x_i^2}\right) \left(\prod_{i=1}^{m} \tilde{L}_i^{-\tilde{x}_i^2}\right) \left(\prod_{i=1}^{m} R_i^{x_i^{-2}}\right) \left(\prod_{i=1}^{m} \tilde{R}_i^{-\tilde{x}_i^{-2}}\right) \cdot h^{-(\mu - \tilde{\mu})} \cdot (u')^{\hat{t}} \cdot (\tilde{u}')^{-\hat{t}}.\end{aligned}$$

The rest of the analysis is same as in Claim 2.

If $L_i \neq \tilde{L}_i$ or $R_i \neq \tilde{R}_i$. This case is the same as the analogous case in Claim 2.

Concrete advantage of the adversary. For the case $T_i \neq \tilde{T}_i$, the adversary succeeds in forging if any one of the polynomials $e_g^{(0)}, \ldots, e_g^{(m)}$ vanishes. Using union bound, this happens with probability $4(m) + 1/(p-1)$. For $\mu \neq \tilde{\mu}$, we break discrete-log relation except with probability: $4m + 1/(p-1)$. Finally, for $L_i \neq \tilde{L}_i$, the adversary succeeds with probability $(14n+8)/(p-1)$. Thus, we upper bound the probability of the adversary succeeding in forging a proof, which turns out to be at most $(14n+8)q/(p-1)$.

Combining the results from Theorem 4 and Claim 4, we get the following corollary.

Corollary 2. *Fiat-Shamir transform of* RngPf *satisfies* FS-SIM-EXT *with respect to a canonical simulator* $\mathcal{S}_{\mathsf{FS\text{-}RngPf}}$ *corresponding to the algebraic simulator* $\mathcal{S}_{\mathsf{RngPf}}$*. Concretely, there exists an efficient* FS-SIM-EXT *extractor* $\mathcal{E}^*$ *for* FS-RngPf *such that for every non-uniform algebraic prover* $\mathcal{P}^*_{\mathsf{alg}}$ *against* FS-RngPf *that makes* q_1 *random oracle queries and* q_2 *simulation queries, and for every distinguisher* $\mathcal{D}^*$*, there exists a non-uniform adversary* $\mathcal{A}$ *against* DL-REL *with the property that for all* $\lambda \in \mathbb{N}^+$*,*

$$\mathbf{Adv}^{\text{FS-SIM-EXT}}_{\mathsf{FS\text{-}RngPf},\mathcal{R}}(\mathcal{S}_{\mathsf{FS\text{-}RngPf}}, \mathcal{E}^*, \mathcal{P}^*_{\mathsf{alg}}, \mathcal{D}^*, \lambda) \leq \left(\mathbf{Adv}^{\text{DL-REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q_1}{(p-1)}\right)$$
$$+q_2 \cdot \left(\mathbf{Adv}^{\text{DL-REL}}(\mathbb{G}_\lambda, \mathcal{A}_\lambda) + \frac{(14n+8)q_2}{(p-1)}\right) + \frac{(q_2+1)(q_1+1)}{|\mathsf{Ch}_{i_0}|}$$

where $i_0 \in [1, r]$ *is the round with the smallest challenge set* Ch_{i_0}*.*