Gentry-Wichs Is Tight: A Falsifiable Non-Adaptively Sound SNARG

Helger Lipmaa and Kateryna Pavlyk

Simula UiB, Bergen, Norway

Abstract. By the impossibility result of Gentry and Wichs, non-falsifiable assumptions are needed to construct (even non-zero-knowledge) adaptively sound succinct non-interactive arguments (SNARGs) for hard languages. It is important to understand whether this impossibility result is tight. While it is known how to construct adaptively sound non-succinct non-interactive arguments for NP from falsifiable assumptions, adaptively sound SNARGs for NP from non-falsifiable assumptions, and adaptively sound SNARGs for P from falsifiable assumptions, there are no known non-adaptively sound SNARGs for NP from falsifiable assumptions. We show that Gentry-Wichs is tight by constructing the latter. In addition, we prove it is non-adaptively knowledge-sound in the algebraic group model and Sub-ZK (i.e., zero-knowledge even if the CRS is subverted) under a non-falsifiable assumption.

Keywords: Falsifiable assumptions, Gentry-Wichs, non-adaptive soundness, SNARG, SNARK, Sub-ZK

1 Introduction

Due to excellent efficiency properties, zk-SNARKs (zero-knowledge succinct non-interactive arguments of knowledge, [22]) are currently the most popular argument systems for NP. Zk-SNARKs are usually defined in the CRS model, where a universally trusted third party generates a CRS used by both the prover and the verifier. A more realistic model is subversion zero-knowledge (Sub-ZK, [5,1,14,3]); a Sub-ZK SNARK is zero-knowledge even if the CRS was subverted. Zk-SNARGs are zero-knowledge succinct non-interactive argument systems that are not necessarily knowledge-sound. NIZKs are non-interactive zero-knowledge argument systems that are not necessarily succinct.

Unfortunately, known SNARKs for NP are based on non-falsifiable assumptions. Gentry and Wichs [17] showed that this is (in a quite precise sense) unavoidable. Their impossibility result balances four aspects of efficient NIZKs: succinctness, adaptive soundness, reliance on falsifiable assumptions, and hardness of the languages. All four aspects are highly desirable:

(1) Succinctness plays a crucial role in the practical adaptation since non-succinct NIZKs are not efficient enough for applications like cryptocurrencies.

Table 1. Some known (im)possibility results. Impossibility results mean that one cannot achieve all ○'s at the same time. Possibility results achieve ✓'s but do not achieve ✗'s. AS = adaptive soundness, s = succinctness, HL = hard languages, FA = falsifiable assumptions, PZK = perfect zero-knowledge, BBR = black-box reduction.

AS	s	HL	FA	PZK	Some papers
					Impossibility results
○	○	○	○		Gentry-Wichs [17] (BBRs), [7] (nonuniform BBRs)
○		○	○	○	[4] (direct BBRs), [38] (BBRs), [7] (non-uniform BBRs)
					Possibility results: the tightness of Gentry-Wichs
✓	✗	✓	✓	✗	Feige-Lapidot-Shamir [13]
✓	✓	✓	✗	✓	SNARKs [22]
✓	✓	✗	✓		Delegation schemes [29,21] (ZK is irrelevant)
✗	✓	✓	✓	✓	The current paper
					Additional possibility result: tightness of [38]
✗	✗	✓	✓	✓	Groth-Ostrovsky-Sahai [25]

(2) A falsifiable assumption is an assumption where a challenger can efficiently decide whether the adversary broke it. Non-falsifiable assumptions are controversial in general [36].

(3) Adaptive soundness guarantees that the SNARK stays sound even if the malicious prover can choose the input $\mathbb{x}$ after seeing the CRS. Non-adaptive soundness guarantees soundness only if $\mathbb{x}$ is chosen before the CRS is fixed.

(4) Most of the applications need SNARKs for hard languages (i.e., languages with hard subset membership problem) like circuit satisfiability; SNARKs for easy languages have their uses, but they are limited.

Gentry and Wichs [17] proved that non-falsifiable assumptions are needed to construct (even non-zero-knowledge) adaptively sound succinct non-interactive arguments (SNARGs) for hard languages under black-box reductions. Assuming black-box reductions (or stronger non-uniform black-box-reductions, [7]), Gentry-Wichs is known to be tight in three aspects, see Table 1. First, non-succinct adaptively sound falsifiable NIZKs are known for NP [13]. Second, adaptively sound falsifiable SNARGs are known for P [29,21] (note that in this case, zero-knowledge is not important). Third, adaptively sound non-falsifiable SNARGs are known for NP [22,31,16,37,32,23]. However, it is a major open problem *whether Gentry-Wichs is tight in the fourth aspect; i.e., whether non-adaptively sound falsifiable SNARGs for hard languages are possible.*[1]

Our Contributions. We construct the first **fa**lsifiable **n**on-**a**daptively sound SNARG FANA for NP. Thus, Gentry-Wichs is tight. We also prove that FANA is both non-adaptively knowledge-sound and Sub-ZK (zero-knowledge, even if the CRS is maliciously generated, [5,1,14,3]). While the last two properties are not related to Gentry-Wichs, they are important in applications.

[1] Note that even non-succinct falsifiable adaptively sound NIZKs for NP do not exist when one aims to obtain perfect zero-knowledge, [38]. The impossibility result of [38] is known to be tight, see Table 1. Thus, we will focus on [17].

FANA is inspired by [10,12] who proposed two NIZKs (DGPRS and FLPS) for well-known constraint systems SSP [9] and SAP [23], correspondingly. We emphasize that DGPRS and FLPS do not seem to be good starting points for our goal:

(a) They are quasi-adaptive SNARGs (QA-SNARGs [27]). (We use the term QA-SNARG instead of the common QA-NIZK to emphasize the succinctness property.) In QA-SNARGs, the NP language is parameterized by a language parameter lpar. Both the quasi-adaptive soundness and zero-knowledge properties hold only if lpar is honestly generated. Since the latter is an undesirable trust assumption, we aim to avoid it by constructing a SNARG and not a QA-SNARG.

(b) They are quasi-adaptively sound [27] (which means the argument system is sound against an adversary who chooses the input $\mathbb{x}$ after seeing lpar and crs), and thus they do not seem to be candidates for *non-adaptive* NIZKs.

(c) They are commit-and-prove argument systems, having a non-succinct perfectly-binding commitment and are thus not succinct.

(d) They are for the SSP [9] and the SAP [23], which are less standard and less powerful constraint systems compared to the QAP [16].

(e) They are not known to be knowledge-sound.

(f) They are not known to be Sub-ZK.

We solve Items a to c by carefully modifying the construction and the soundness proof of [10,12]. In DGPRS and FLPS, the prover commits to the input $\mathbb{x}$ and the witness $\mathbb{w}$ by using a perfectly-binding and several succinct commitment schemes, including a *functional SSB commitment scheme* [12]. Functional SSB commitment schemes satisfy the following helpful property: for a small locality parameter q ($q < 10$ in DGPRS and FLPS), one can reprogram its commitment key ck during the security proof so that the reduction will obtain q linear combinations of the input and witness coordinates; moreover, in existing schemes, the commitment length is $q + 1$ group elements.

The quasi-adaptive soundness proof of [10,12] consists of several games. Assume that $\mathcal{A}$ is a successful soundness adversary. The first game is the classic (quasi-adaptive) soundness game. In the second game, one picks a random J, which is a guess for the SSP/SAP/QAP constraint that is not satisfied. One aborts if the guess was wrong. (This results in n-time security loss where n is the number of constraints.) Crucially, one uses the perfectly binding commitment scheme to extract values required to do this check. In the third game, one additionally modifies the commitment key of the functional SSB scheme to be a function of J. One can do so due to the “function-set hiding” property [12] of the functional SSB scheme. One then shows that the last game is secure by constructing two different reductions to two different security assumptions.

In comparison, we check whether the reduction guessed a non-satisfied constraint correctly by using the succinct functional SSB commitment. Thus, we do not need the perfectly binding commitment at all, solving Item c. Hence, we have a succinct NIZK, i.e., a SNARG. Moreover, since the language parameter lpar in DGPRS and FLPS is the commitment key of the perfectly-binding commitment scheme and we will not use the latter at all, FANA will not have lpar. Importantly, it means that FANA is not a QA-NIZK but a usual NIZK. This solves Item a. Moreover, since FANA is secure under a variant of the security assumptions of [10,12], we have a falsifiable SNARG.

At this moment, it might seem that we have breached the Gentry-Wichs impossibility result since DGPRS and FLPS are quasi-adaptively sound. However, this is not the case. Namely, since we use the functional SSB commitment to check whether the Jth constraint is satisfied, we cannot do a check (and a conditional abort) before changing the commitment key. In the case of (quasi-)adaptive soundness, $\mathbb{x}$ (and thus also the unsatisfied constraint's number) can depend on ck, where the latter depends on J. A malicious prover can thus, after seeing ck, choose $\mathbb{x}$ so that the Jth constraint is satisfied.

We solve this seeming contradiction by resorting to non-adaptive soundness, i.e., we ask $\mathcal{A}$ to output $\mathbb{x}$ before seeing ck so that it cannot depend on J that is embedded in ck. In this case, the security proof follows. This solves Item b. Since we now have a non-adaptively sound SNARG for NP under falsifiable assumptions, we have also shown that Gentry-Wichs is tight. We emphasize that while this change to [10,12] may sound simple, it is pretty surprising: as we already argued, DGPRS and FLPS do not seem to be suitable starting points for our endeavor. It also results in a multiple changes to the construction of the SNARG, including the omission of perfectly-binding commitment and lpar.

Additional Features. While we have already solved our main open problem, to make FANA more attractive in practice, we will also tackle Items d to f. In addition, we will base FANA on an—arguably—better falsifiable assumption, which also results in slight efficiency gain. Due to this, FANA's argument length and verifier's complexity are almost the same as in FLPS.

Finally, FANA relies on the González-Hevia-Ràfols bilateral subspace QA-NIZK BLS from [19]. For FANA to be non-adaptively sound, non-adaptively knowledge-sound, and Sub-ZK, BLS has to satisfy quasi-adaptive σ-strong soundness, adaptive knowledge-soundness, and Sub-ZK. Here, quasi-adaptive σ-strong soundness is a new security property of QA-SNARGs that lies between quasi-adaptive soundness and quasi-adaptive strong soundness [28]. We prove that BLS satisfies all three properties. Since bilateral subspace QA-NIZKs have many independent applications, this constitutes a contribution of independent interest.

QAP (Item d). DGPRS is for SSP (Square Span Program, [9]), a constraint system that has an efficient reduction to Boolean circuit satisfiability. In many applications, it is desirable to construct a (QA-)SNARG for arithmetic circuits. FLPS is for SAP (Square Arithmetic Program, [23,24]), a constraint system that has an efficient reduction to arithmetic circuit satisfiability for circuits that consist of addition and square gates. The use of square gates instead of general multiplication gates results in a factor of two overhead.

The constraint system QAP (Quadratic Arithmetic Program, [16]) models efficiently arithmetic circuits with general multiplication gates. FANA is directly for QAP. In the pairing-based setting, SNARKs for QAP have one complication compared to SNARKs for SSP and SAP: namely, in the former, the prover outputs an element in both source groups. Hence, differently from DGPRS and FLPS, we use functional SSB commitments in both source groups $\mathbb{G}_1$ and $\mathbb{G}_2$. In the soundness proof, this means adding one more game to change the functional SSB ck in both groups. Adding another commitment means that, at least when using the same approach as DGPRS and FLPS, SNARKs for QAP are necessarily less efficient. We mitigate it by using a different assumption.

Better Assumption. The q-type assumptions S-TSDH (Square Target Strong Diffie-Hellman) and SA-TSDH (Square Arithmetic Target Strong Diffie-Hellman) used in [10] and [12] respectively, look quite complicated.[2] To argue that such assumptions are sensible, one can prove that they hold in the *generic group model* (GGM). In a GGM proof, one considers a generic adversary that is only allowed to (i) execute group operations in the source and target groups, (ii) perform the pairing operation, and (iii) check for equality of two group elements. GGM is a very restrictive model. One of the many criticisms against GGM is that the target group $\mathbb{G}_T$ is a subgroup of the finite field, and thus it is questionable whether it can be modeled as a generic group, [26]. Indeed, one can use the finite field structure to operate on the elements of the $\mathbb{G}_T$. To address this issue, [26] defined the *semi-GGM*, where one assumes that only the source groups are generic. A significant drawback of S-TSDH and SA-TSDH is that, in their definition, the adversary can output a value in the target group. Thus, they are not (known to be) secure in the semi-GGM.

Moreover, the adversary of the $\{*\}$TSDH assumptions is required to output some elements together with their "knowledge components" [8]. To prove soundness under $\{*\}$TSDH assumptions, the prover of the SNARG must also output the knowledge components. Due to this, $\{*\}$TSDH assumptions "force" one to design SNARGs that might not be optimal.

Instead of $\{*\}$TSDH assumptions, we introduce a very different-looking assumption QA-LINRES. QA-LINRES (see Definition 2) holds in the algebraic group model (AGM, [15]).[3] Since the QA-LINRES adversary does not have to output "knowledge components", QA-LINRES allows to design more efficient SNARGs. Even without counting the cost of perfectly-binding commitment in DGPRS and FLPS, FANA is efficiency-wise competitive with DGPRS and FLPS despite being for QAP and thus involving one more functional SSB commitment.

Knowledge-Soundness (Item e). In many applications, knowledge-soundness is desirable. It is especially important in the case of succinct NIZKs, where the verifier only has access to a succinct commitment to the witness. Such commitments can be information-theoretically opened to an exponential number of witnesses, and it is important to know which witness was used by the adversary. Unfortunately, neither DGPRS nor FLPS is known to be knowledge-sound.

[2] DGPRS, FLPS, and FANA also rely on two standard assumptions SKerMDH [19] and DDH. We focus on the least standard assumptions, S-TSDH and SA-TSDH.

[3] We recall that the AGM is a modern, somewhat more realistic alternative to the GGM. In particular, like the semi-GGM, the AGM of [15] considers only the source groups to be "algebraic". Thus, QA-LINRES also holds in the semi-GGM.

Sub-ZK (Item f). DGPRS and FLPS are proven to be sound and zero-knowledge, assuming that both lpar and crs are *trusted*. Since in many applications, it is crucial to avoid trust assumptions (like crs's correctness), this situation is not satisfactory. Instead, one should aim to prove Sub-ZK [5]. It is known that the most efficient zk-SNARK [23] is also Sub-ZK [1,14,3] under non-falsifiable assumptions. As noted in [2], non-falsifiable assumptions are also needed due to the well-known impossibility result of [18]. In Theorem 2, we prove that FANA is Sub-ZK assuming that BLS is Sub-ZK.

Efficiency. The FANA argument π is succinct, consisting of 9 elements of $\mathbb{G}_1$ and 5 elements of $\mathbb{G}_2$.

The Bilateral Subspace Argument. FANA uses a bilinear subspace argument system that, in particular, allows one to prove that different commitments in both $\mathbb{G}_1$ and $\mathbb{G}_2$ commit to the same message. As a contribution of independent interest, we study the quasi-adaptively strongly sound and perfectly zero-knowledge González-Hevia-Ràfols bilateral subspace argument system BLS [19].

Let σ be an efficiently computable function. We define a new soundness notion for QA-SNARGs, σ-strong soundness, that lies between soundness and strong soundness [28]. Since BLS is quasi-adaptively strongly sound, it is also quasi-adaptively σ-strongly sound for any efficiently computable σ. While quasi-adaptive strong soundness of BLS is known to be sufficient for the non-adaptive (knowledge-) soundness of FANA, we show that it suffices that BLS is σ_x-strongly sound for a particular function σ_x. There are two primary motivations for introducing the new security notion. First, it allows one to capture the exact security property of BLS needed by FANA. Second, it may be possible (though we leave it for future work) to construct more efficient bilinear subspace argument systems that are σ_x-strongly sound but not strongly sound.

In Theorem 1, we prove BLS is adaptively sound under the non-falsifiable assumption $\mathrm{SKerMDH}^{\mathrm{dl}}$ from [2]. We prove that BLS is adaptively knowledge-sound in the AGM under the $\mathrm{SDL}^{\mathrm{dl}}$ assumption from [2]. (See Theorem 1.) Both $\mathrm{SKerMDH}^{\mathrm{dl}}$ and $\mathrm{SDL}^{\mathrm{dl}}$ belong to the family of non-adaptive oracle assumptions, where the adversary is initially given access to the oracle who solves the discrete logarithm assumption. After that, the adversary has to break either the SKerMDH or the SDL [6] assumption on a fresh instance. We believe such assumptions are significantly more realistic than knowledge assumptions underlying efficient zk-SNARKs.

As shown in [2], to prove that a QA-SNARG is Sub-ZK, one must prove that the QA-SNARG is both black-box zero-knowledge (that is, zero-knowledge, if lpar and crs are trusted) and non-black-box persistent zero-knowledge (that is, zero-knowledge, if lpar and crs are not trusted; this notion was defined in [2]). In the latter case, one assumes that one can extract the simulation trapdoor from a malicious crs. Zero-knowledge does not follow from persistent zero-knowledge since the former is black-box and the latter is non-black-box, [2]. In Theorem 1, we prove that (1) BLS is perfectly zero-knowledge, and (2) BLS is persistent zero-knowledge under a novel knowledge-assumption GHR-KE, similar to the KW-KE assumption [2].

Since bilateral subspace argument systems have many more applications, the BLS section constitutes a significant independent contribution.

Summary of Security Results. To not overwhelm the reader, we did not describe all security results in the introduction. As a corollary of various theorems of the current paper, we can informally state the following result.

Corollary 1 (Informal). FANA *is a SNARG that is non-adaptively sound under the falsifiable* SKerMDH*,* DDH*, and* QA-LINRES *assumptions (where the latter is a new falsifiable assumption that holds under the PDL assumption in the AGM). It is non-adaptively knowledge-sound in the AGM if additionally the non-falsifiable assumptions* $\mathrm{SKerMDH}^{\mathrm{dl}}$ *and* $\mathrm{SDL}^{\mathrm{dl}}$ *[2] hold. It is Sub-ZK under the* DDH *and the non-falsifiable* GHR-KE *assumption (where the latter is a new knowledge assumption that holds in the AGM).*

Full Version. Due to the lack of space, we postpone most of the security proofs and several additional results to the full version, [35].

Open Problems. To be precise, we showed that [17] is tight with respect to black-box reductions [17] and non-uniform black-box reductions [7]. We leave the study of general non-black-box reductions as an interesting open problem.

2 Preliminaries

For a matrix $\boldsymbol{A} = (A_{ij})$, $\boldsymbol{A}_i$ denotes its ith row and $\boldsymbol{A}^{(j)}$ denotes its jth column. The cokernel of $\boldsymbol{A}$ is defined as $\mathrm{coker}(\boldsymbol{A}) = \{\boldsymbol{a} : \boldsymbol{a}^\top \boldsymbol{A} = \boldsymbol{0}\}$. Let $\mathrm{colspace}(\boldsymbol{A})$ be the column space of $\boldsymbol{A}$. For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, denote $\boldsymbol{A}/\!/\boldsymbol{B} := \left(\begin{smallmatrix}\boldsymbol{A}\\ \boldsymbol{B}\end{smallmatrix}\right)$.

Assume n is a power of two. Let ω be the nth primitive root of unity modulo p (ω exists, given that $n \mid (p-1)$.) Then,

- $Z(X) := \prod_{i=1}^{n}(X - \omega^{i-1}) = X^n - 1$ is the unique degree n monic polynomial, such that $Z(\omega^{i-1}) = 0$ for all $i \in [1, n]$.
- For $i \in [1, n]$, let $\ell_i(X)$ be the *ith Lagrange polynomial*, i.e., the unique degree $n-1$ polynomial, such that $\ell_i(\omega^{i-1}) = 1$ and $\ell_i(\omega^{j-1}) = 0$ for $i \neq j$. Let $Z'(X) = dZ(X)/dX = nX^{n-1}$. It is well known that

$$\ell_i(X) := \tfrac{Z(X)}{Z'(\omega^{i-1})(X-\omega^{i-1})} = \tfrac{(X^n-1)\omega^{i-1}}{n(X-\omega^{i-1})} \quad \text{for } X \neq \omega^{i-1} \enspace .$$

Given $X \in \mathbb{Z}_p$, one can efficiently compute $\{\ell_i(X)\}_{i=1}^n$. $L_{\mathbb{z}}(X) := \sum_{i=1}^n \mathbb{z}_i \ell_i(X)$ is the interpolating polynomial of the vector $\mathbb{z} \in \mathbb{Z}_p^n$ at points ω^{i-1}.

We denote assignment by $\leftarrow$ and (uniformly random) sampling by $\leftarrow_{\$}$. PPT denotes probabilistic polynomial-time; $\lambda \in \mathbb{N}$ is the security parameter. We assume all adversaries are stateful, i.e., keep up a state between different executions. For an algorithm $\mathcal{A}$, $\mathrm{range}(\mathcal{A})$ is the range of $\mathcal{A}$, i.e., the set of of valid outputs of $\mathcal{A}$, $\mathsf{RND}_\lambda(\mathcal{A})$ denotes the random tape of $\mathcal{A}$ (for given λ), and $r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A})$ denotes the uniformly random choice of the randomizer r from $\mathsf{RND}_\lambda(\mathcal{A})$. By $y \leftarrow \mathcal{A}(x; r)$ we denote the fact that $\mathcal{A}$, given an input x and a randomizer r, outputs y. Let $\mathsf{negl}(\lambda)$ be an arbitrary negligible function, and $\mathsf{poly}(\lambda)$ be an arbitrary polynomial function. We write $a \approx_\lambda b$ if $|a - b| \leq \mathsf{negl}(\lambda)$.

Bilinear Groups. A bilinear group generator $\mathsf{Pgen}(1^\lambda)$ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic groups of prime order p, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear pairing, and $[1]_\iota$ is a fixed generator of $\mathbb{G}_\iota$. While $[1]_\iota$ is a part of p, for the sake of clarity, we often give it as an explicit input to different algorithms. We assume $n \mid (p - 1)$, where n is a large deterministically fixed upper bound on the size of the statements that one handles in this bilinear group. As in [5], we assume that Pgen is deterministic and cannot be subverted. The bilinear pairing is of Type-3, i.e., there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the by-now standard bracket notation, i.e., for $\iota \in \{1, 2, T\}$, we write $[a]_\iota$ to denote $a[1]_\iota$. We denote $\hat{e}([a]_1, [b]_2)$ by $[a]_1 \bullet [b]_2$. We use freely the bracket notation together with matrix notation, e.g., $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{C}$ iff $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$. For an integer (vector) a, we denote $[a]_* := ([a_1]_1, [a_2]_2)$.

Assumptions. Let $\kappa^*, \kappa \in \mathbb{N}$, with $\kappa^* \geq \kappa$, be small constants. Let p be a large prime. A PPT-sampleable distribution $\mathcal{D}_{\kappa^*,\kappa}$ is a *matrix distribution* [11] if it samples matrices $\boldsymbol{A} \in \mathbb{Z}_p^{\kappa^* \times \kappa}$ of full rank κ. $\mathcal{D}_{\kappa^*,\kappa}$ is *robust* [27] if it samples matrices $\boldsymbol{A}$ whose upper $\kappa \times \kappa$ submatrix $\bar{\boldsymbol{A}}$ is invertible. Denote the lower $(\kappa^* - \kappa) \times \kappa$ submatrix of $\boldsymbol{A}$ by $\underline{\boldsymbol{A}}$. Denote $\mathcal{D}_\kappa = \mathcal{D}_{\kappa+1,\kappa}$. We denote $\mathcal{D}_{\kappa+1,\kappa}$ by $\mathcal{D}_\kappa$. In the full version [35], we define five common distributions [11]: $\mathcal{U}_\kappa$ (uniform), $\mathcal{L}_\kappa$ (linear), $\mathcal{IL}_\kappa$ (incremental linear), $\mathcal{C}_\kappa$ (cascade), $\mathcal{SC}_\kappa$ (symmetric cascade). All mentioned distributions can be made robust with minimal changes.

Let $d_1(n), d_2(n) \in \mathsf{poly}(\lambda)$. $(d_1(n), d_2(n))$*-PDL (Power Discrete Logarithm, [39,31])* holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{pdl}}_{\mathsf{Pgen},d_1,d_2,\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), x \leftarrow_{\$} \mathbb{Z}_p^* : \\ \mathcal{A}(\mathsf{p}, [(x^i)_{i=0}^{d_1(n)}]_1, [(x^i)_{i=0}^{d_2(n)}]_2) = x\end{array}\right] \approx_\lambda 0 \enspace .$$

The q-PDL assumption in $\mathbb{G}_1$ (resp., $\mathbb{G}_2$) is equal to the $(q, 0)$-PDL (resp., $(0, q)$-PDL) assumption. The *symmetric discrete logarithm* (SDL [6]) assumption is equal to the $(1, 1)$-PDL assumption.

Let $\iota \in \{1, 2\}$. *DDH*$_{\mathbb{G}_\iota}$ *(Decisional Diffie-Hellman)* holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{ddh}}_{\mathsf{Pgen},\mathbb{G}_\iota,\mathcal{A}}(\lambda) := |\varepsilon^0_{\mathcal{A}}(\lambda) - \varepsilon^1_{\mathcal{A}}(\lambda)| \approx_\lambda 0$, where

$$\varepsilon^\beta_{\mathcal{A}}(\lambda) := \Pr[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); x, y, z \leftarrow_{\$} \mathbb{Z}_p : \mathcal{A}(\mathsf{p}, [x, y, xy + \beta z]_\iota) = 1] \enspace .$$

Let $\iota \in \{1, 2\}$. $\mathcal{D}_{\kappa^*,\kappa}$*-KerMDH*$_{\mathbb{G}_\iota}$ (Kernel Diffie-Hellman) holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{kermdh}}_{\mathsf{Pgen},\mathbb{G}_\iota,\mathcal{D}_{\kappa^*,\kappa},\mathcal{A}}(\lambda) :=$

$$\Pr[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\kappa^*,\kappa}; [\boldsymbol{c}]_{3-\iota} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_\iota) : \boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_\kappa \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell] \approx_\lambda 0 \enspace .$$

$\mathcal{D}_{\kappa^*,\kappa}$*-SKerMDH* (Split Kernel Diffie-Hellman, [19]) holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{skermdh}}_{\mathsf{Pgen},\mathbb{G}_\iota,\mathcal{D}_{\kappa^*,\kappa},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\kappa^*,\kappa}; ([\boldsymbol{c}_1]_1, [\boldsymbol{c}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1, [\boldsymbol{A}]_2) : \\ \boldsymbol{A}^\top(\boldsymbol{c}_1 - \boldsymbol{c}_2) = \boldsymbol{0}_\kappa \wedge \boldsymbol{c}_1 - \boldsymbol{c}_2 \neq \boldsymbol{0}_{\kappa^*}\end{array}\right] \approx_\lambda 0 \enspace .$$

According to Lemma 1 of [19], if $\mathcal{D}_{\kappa^*,\kappa}$-KerMDH holds in generic symmetric bilinear groups, then $\mathcal{D}_{\kappa^*,\kappa}$-SKerMDH holds in generic asymmetric bilinear groups. The KerMDH assumption holds also for Type-1 pairings, where $\mathbb{G}_1 = \mathbb{G}_2$, but then one needs $\kappa \geq 2$, which affects efficiency.

Algebraic Group Model (AGM). The AGM is a new model [15] used to prove the security of a cryptographic assumption, protocol, or a primitive. Essentially, in the AGM, one assumes that each PPT algorithm $\mathcal{A}$ is algebraic in the following sense. Assume $\mathcal{A}$'s input includes $[\boldsymbol{x}_\iota]_\iota$ and no other elements from the group $\mathbb{G}_\iota$. We consider a less restrictive version of the AGM that gives the adversary additional access to random oracles. More precisely, assume $\mathcal{A}$ has an access to oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. For $\iota \in \{1, 2\}$, $\mathcal{O}_\iota$ samples and outputs a random element $[q_{\iota k}]_\iota$ from $\mathbb{G}_\iota$. The oracle access models the ability of $\mathcal{A}$ to create random group elements without knowing their discrete logarithms.

We assume that if $\mathcal{A}$ outputs group elements $[\boldsymbol{y}_\iota]_\iota$, then $\mathcal{A}$ knows matrices $\boldsymbol{N}_\iota$, such that $\boldsymbol{y}_\iota = \boldsymbol{N}_\iota \left(\begin{smallmatrix} \boldsymbol{x}_\iota \\ \boldsymbol{q}_\iota \end{smallmatrix}\right)$. Formally, a PPT algorithm $\mathcal{A}$ is *(*Pgen*-)algebraic* if there exists an efficient extractor $\mathsf{Ext}_\mathcal{A}$, such that for any PPT-sampleable distribution $\mathcal{D}$, $\mathsf{Adv}^{\mathrm{agm}}_{\mathsf{Pgen},\mathcal{D},\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} \mathsf{p} \leftarrow_\$ \mathsf{Pgen}(1^\lambda); \mathbb{x} = ([\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_2) \leftarrow_\$ \mathcal{D}; r \leftarrow_\$ \mathsf{RND}_\lambda(\mathcal{A}); \\ ([\boldsymbol{y}_1]_1, [\boldsymbol{y}_2]_2) \leftarrow_\$ \mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2}(\mathbb{x}; r); (\boldsymbol{N}_1, \boldsymbol{N}_2) \leftarrow \mathsf{Ext}_\mathcal{A}(\mathbb{x}; r) : \\ \boldsymbol{y}_1 \neq \boldsymbol{N}_1 \left(\begin{smallmatrix} \boldsymbol{x}_1 \\ \boldsymbol{q}_1 \end{smallmatrix}\right) \vee \boldsymbol{y}_2 \neq \boldsymbol{N}_2 \left(\begin{smallmatrix} \boldsymbol{x}_2 \\ \boldsymbol{q}_2 \end{smallmatrix}\right) \end{array}\right] = \mathsf{negl}(\lambda) \enspace .$$

For $\iota \in \{1, 2\}$, $\mathcal{O}_\iota$ is an oracle that samples and returns a random element from $\mathbb{G}_\iota$. $[\boldsymbol{q}_\iota]_\iota$ is the list of all elements output by $\mathcal{O}_\iota$. The AGM states that for any PPT-sampleable $\mathcal{D}$ and PPT $\mathcal{A}$, there exists a PPT $\mathsf{Ext}_\mathcal{A}$, such that $\mathsf{Adv}^{\mathrm{agm}}_{\mathsf{Pgen},\mathcal{D},\mathcal{A},\mathsf{Ext}_\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

Quadratic Arithmetic Program (QAP). QAP was introduced in [16] as a relation $\mathbf{R}$ where for an input $\mathbb{x}$ and a witness $\mathbb{w}$, $(\mathbb{x}, \mathbb{w}) \in \mathbf{R}$ can be verified by using a parallel quadratic check. QAP has an efficient reduction from the (either Boolean or Arithmetic) Circuit-SAT. Thus, an efficient zk-SNARK for QAP results in an efficient zk-SNARK for Circuit-SAT.

In QAP, one considers arithmetic circuits that consist only of fan-in-2 multiplication gates, but either input of each multiplication gate can be any weighted sum of wire values [16]. In arithmetic circuits, n is the number of multiplication gates, m is the number of wires, and $m_0 < m$ is the number of public inputs.

For the sake of efficiency, we require the existence of the n-th primitive root of unity modulo p, denoted by ω. (However, this is not needed for the new protocols to work.) Let $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W} \in \mathbb{Z}_p^{n \times m}$ be instance-dependent matrices and let $\mathbb{z} \in \mathbb{Z}_p^m$ be a witness. A QAP is characterized by the constraint $\boldsymbol{U}\mathbb{z} \circ \boldsymbol{V}\mathbb{z} = \boldsymbol{W}\mathbb{z}$, where $\circ$ denotes the entrywise product of two vectors and $\mathbb{z} = \left(\begin{smallmatrix} \mathbb{x} \\ \mathbb{w} \end{smallmatrix}\right)$. For $j \in [1, m]$, define $u_j(X) := L_{\boldsymbol{U}^{(j)}}(X)$, $v_j(X) := L_{\boldsymbol{V}^{(j)}}(X)$, and $w_j(X) := L_{\boldsymbol{W}^{(j)}}(X)$ to be interpolating polynomials of the jth columns of the corresponding matrices. Thus, $u_j, v_j, w_j \in \mathbb{Z}_p^{(\leq n-1)}[X]$. Let $u(X) = \sum_{j=1}^m \mathbb{z}_j u_j(X)$, $v(X) = \sum_{j=1}^m \mathbb{z}_j v_j(X)$, and $w(X) = \sum_{j=1}^m \mathbb{z}_j w_j(X)$. Then $\boldsymbol{U}\mathbb{z} \circ \boldsymbol{V}\mathbb{z} = \boldsymbol{W}\mathbb{z}$ iff $Z(X) \mid (u(X)v(X) - w(X))$

iff $u(X)v(X) \equiv w(X) \pmod{Z(X)}$ iff there exists a polynomial $h(X)$ such that $u(X)v(X) - w(X) = h(X)Z(X)$.

An QAP instance $\mathcal{I}_{\mathsf{qap}}$ is equal to $(\mathbb{Z}_p, m_0, \{u_j, v_j, w_j\}_{j=1}^{m})$. $\mathcal{I}_{\mathsf{qap}}$ defines the following relation:

$$\mathbf{R}_{\mathcal{I}_{\mathsf{qap}}} = \left\{ \begin{matrix} (\mathbb{x}, \mathbb{w}) \colon \mathbb{x} = (\mathbb{z}_1, \ldots, \mathbb{z}_{m_0})^\top \wedge \mathbb{w} = (\mathbb{z}_{m_0+1}, \ldots, \mathbb{z}_m)^\top \wedge \\ u(X)v(X) \equiv w(X) \pmod{Z(X)} \end{matrix} \right\} \tag{1}$$

where $u(X)$, $v(X)$, and $w(X)$ are defined as above. Alternatively, $(\mathbb{x}, \mathbb{w}) \in \mathbf{R} = \mathbf{R}_{\mathcal{I}_{\mathsf{qap}}}$ if there exists a (degree $\leq n-2$) polynomial $h(X)$, such that the following key equation holds:

$$\chi(X) := u(X)v(X) - w(X) - h(X)Z(X) = 0 \ , \tag{2}$$

On top of checking Eq. (2), the verifier also needs to check that $u(X)$, $v(X)$, and $w(X)$ are correctly computed: that is, (i) the first m_0 coefficients $\mathbb{z}_j$ in $u(X)$ are equal to the public inputs, and (ii) $u(X)$, $v(X)$, and $w(X)$ are all computed by using the same coefficients $\mathbb{z}_j$ for $j \in [1, m]$.

SAP and SSP. Square arithmetic programs (SAPs, [23]) are QAPs with the extra condition $\boldsymbol{U} = \boldsymbol{V}$; thus, all multiplication gates in the arithmetic circuit have equal inputs, i.e., they are square gates. Square span program (SSP, [9]) are QAPs with the restriction that $\boldsymbol{U} = \boldsymbol{V} = \boldsymbol{W}$; see [34]. There is an efficient relation between the arithmetic circuit evaluation problem and QAP/SAP and another one between the Boolean circuit evaluation problem and SSP. SSP is useful when the concrete zero-knowledge language is related to Boolean circuits.

2.1 Underlying Commitment Schemes

We will use several different commitment schemes that are all specific cases of the Multi-Pedersen commitment scheme.

EMP **Commitment.** Let $\iota \in \{1, 2\}$. Let q (the locality parameter) and n (the plaintext length) be two integers. Let $\mathcal{D}$ be a (matrix) distribution on $q \times (m+1)$ matrices. In the $(q, \mathcal{D})$-*Extended Multi-Pedersen commitment scheme* EMP [20,12], the commitment key is $\mathsf{ck} = [\boldsymbol{G}]_\iota$, where $\boldsymbol{G} \leftarrow_\$ \mathcal{D}$. The commitment $\mathsf{EMP.Com}(\mathsf{ck}; \boldsymbol{a}; r)$, where $\boldsymbol{a} \in \mathbb{Z}_p^m$ and $r \leftarrow_\$ \mathbb{Z}_p$, is defined as $[\boldsymbol{G}]_\iota \binom{\boldsymbol{a}}{r}$. The *interpolation commitment scheme* [33] is a perfectly-hiding EMP commitment scheme, with $\mathsf{ck} := [\ell_1(x), \ldots, \ell_m(x), Z(x)]_\iota \in \mathbb{G}_\iota^{1 \times (m+1)}$ for a random trapdoor $x \leftarrow_\$ \mathbb{Z}_p$.

Functional SSB Commitment [12]. Let F be a fixed function. In general, F may depend on p, but we will not emphasize it for notational simplicity. In our applications, $F : a \mapsto [a]_\iota$ for $\iota \in \{1, 2\}$. Let $\mathcal{F}$ be a function family, where $f \in \mathcal{F}$ inputs a vector $\boldsymbol{x}$ and outputs an element from the domain of F. An F-*extractable functional somewhere statistically-binding (SSB) commitment scheme* [12] $\varGamma = (\mathsf{Pgen}, \mathsf{KC}, \mathsf{Com}, \mathsf{LExt}_F)$ for a function family $\mathcal{F}$ makes it possible to commit to a vector $\boldsymbol{x}$, such that the following properties hold. (1) The commitment key ck is chosen depending on the description of a function tuple

$f_1, \ldots, f_q \in \mathcal{F}$, (2) commitment keys corresponding to different function tuples are computationally indistinguishable, and (3) given the extraction key, one can extract from the commitment the vector $(F(f_1(\boldsymbol{x})), \ldots, F(f_q(\boldsymbol{x})))$.

More precisely, an *F-extractable functional SSB commitment scheme $\mathit{\Gamma}$* $= (\mathsf{Pgen}, \mathsf{KC}, \mathsf{Com}, \mathsf{LExt}_F)$ for a function family $\mathcal{F}$ consists of the following polynomial-time algorithms. We will omit algorithms (like trapdoor opening) and properties not needed in the current paper.

Parameter generation: $\mathsf{Pgen}(1^\lambda)$ returns parameters p (for example, group description). Recall that F depends on p.

Commitment key generation: for parameters p, a positive integer $n \in \mathsf{poly}(\lambda)$, a locality parameter $q \in [1, n]$, and a tuple $\mathcal{S} = (f_1, \ldots, f_{|\mathcal{S}|}) \subseteq \mathcal{F}$ with $|\mathcal{S}| \leq q$, $\mathsf{KC}(\mathsf{p}, n, q, \mathcal{S})$ outputs a commitment key ck and a trapdoor $\mathsf{td} = (\mathsf{ek}, \mathsf{tk})$. Here, ek is the *extraction key*, and tk is the *trapdoor key*. ck, ek, and tk implicitly specify p, the message space $\mathcal{M}$, the randomizer space $\mathcal{R}$, and the commitment space $\mathcal{C}$, s.t. $F(\mathcal{M}) \subseteq \mathcal{C}$. For any other input, KC outputs $(\mathsf{ck}, \mathsf{td}) = (\bot, \bot)$.

Commitment: for a commitment key $\mathsf{ck} \neq \bot$, a message $\boldsymbol{x} \in \mathcal{M}^n$, and a randomizer $r \in \mathcal{R}$, $\mathsf{Com}(\mathsf{ck}; \boldsymbol{x}; r)$ outputs a commitment $c \in \mathcal{C}$.

Local extraction: for $\mathsf{p} \in \mathsf{Pgen}(1^\lambda)$, a positive integer $n \in \mathsf{poly}(\lambda)$, a locality parameter $q \in [1, n]$, a tuple $\mathcal{S} = (f_1, \ldots, f_{|\mathcal{S}|}) \subseteq \mathcal{F}$ with $1 \leq |\mathcal{S}| \leq q$, $(\mathsf{ck}, (\mathsf{ek}, \mathsf{tk})) \in \mathsf{KC}(\mathsf{p}, n, q, \mathcal{S})$, and $c \in \mathcal{C}$, $\mathsf{LExt}_F(\mathsf{ek}; c)$ returns a tuple $\big(F(f_1(x)), \ldots, F(f_{|\mathcal{S}|}(x))\big) \in \mathcal{M}^{|\mathcal{S}|}$;

For $\{f_i\}_{i=1}^q \subseteq \mathcal{F}$ and vector $\boldsymbol{x}$ let us denote $\boldsymbol{x}_\mathcal{S} = (f_1(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x}))$.

An *F-extractable functional SSB commitment scheme $\mathit{\Gamma}$* for function family $\mathcal{F}$ can satisfy the following security requirements.

Function-Set Hiding: $\forall \lambda$, PPT $\mathcal{A}$, $n \in \mathsf{poly}(\lambda)$, $q \in [1, n]$, $\mathsf{Adv}^{\mathsf{fsh}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) := 2 \cdot |\varepsilon^{\mathsf{fsh}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) - 1/2| \approx_\lambda 0$, where $\varepsilon^{\mathsf{fsh}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathcal{S}_0, \mathcal{S}_1) \leftarrow \mathcal{A}(\mathsf{p}, n, q) \text{ s.t. } \forall i \in \{0,1\}. \mathcal{S}_i \subseteq \mathcal{F} \wedge |\mathcal{S}_i| \leq q; \\ \beta \leftarrow_\$ \{0,1\}; (\mathsf{ck}_\beta, \mathsf{td}_\beta) \leftarrow \mathsf{KC}(\mathsf{p}, n, q, \mathcal{S}_\beta) : \mathcal{A}(\mathsf{ck}_\beta) = \beta\end{array}\right] .$$

Intuitively, ck reveals computationally no information about $\mathcal{S}$.

Almost Everywhere Perfectly Hiding: $\forall \lambda$, unbounded $\mathcal{A}$, $n \in \mathsf{poly}(\lambda)$, $q \in [1, n]$, $\mathsf{Adv}^{\mathsf{aeph}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) := 2 \cdot |\varepsilon^{\mathsf{aeph}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) - 1/2| = 0$, where $\varepsilon^{\mathsf{aeph}}_{\mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathcal{S} \leftarrow \mathcal{A}(\mathsf{p}, n, q) \text{ s.t. } \mathcal{S} \subseteq \mathcal{F} \wedge |\mathcal{S}| \leq q; (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{KC}(\mathsf{p}, n, q, \mathcal{S}); \\ (\boldsymbol{x}_0, \boldsymbol{x}_1) \leftarrow \mathcal{A}(\mathsf{ck}) \text{ s.t. } \boldsymbol{x}_{0\mathcal{S}} = \boldsymbol{x}_{1\mathcal{S}}; \beta \leftarrow_\$ \{0,1\}; r \leftarrow_\$ \mathcal{R} : \mathcal{A}(\mathsf{Com}(\mathsf{ck}; \boldsymbol{x}_\beta; r)) = \beta\end{array}\right] .$$

Intuitively, given ck, that depends on $\mathcal{S}$, the commitment hides perfectly the values of x_i for $i \notin \mathcal{S}$.

Local F-Extractability: $\forall \lambda$, $\mathsf{p} \in \mathsf{Pgen}(1^\lambda)$, $n \in \mathsf{poly}(\lambda)$, $q \in [1, n]$, $\mathcal{S} = (f_1, \ldots, f_{|\mathcal{S}|}) \subseteq \mathcal{F}$ with $|\mathcal{S}| \leq q$, $(\mathsf{ck}, (\mathsf{ek}, \mathsf{tk})) \leftarrow \mathsf{KC}(\mathsf{p}, n, q, \mathcal{S})$, and PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{lext}}_{F, \mathit{\Gamma}, n, q, \mathcal{A}}(\lambda) :=$

$$\Pr[\boldsymbol{x}, r \leftarrow \mathcal{A}(\mathsf{ck}) : \mathsf{LExt}_F(\mathsf{ek}; \mathsf{Com}(\mathsf{ck}; \boldsymbol{x}; r)) \neq \big(F(f_1(\boldsymbol{x})), \ldots, F(f_{|\mathcal{S}|}(\boldsymbol{x}))\big)] = 0 .$$

```
$\mathsf{KC}(\mathsf{p}, n, q, [\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{q \times n})$:
    Set implicitly $\mathcal{M} = \mathcal{R} = \mathbb{Z}_p^n$ and $\mathcal{C} = \mathbb{G}_\iota^{q+1}$;
    Sample $\boldsymbol{R} \leftarrow_\$ \mathbb{Z}_p^{(q+1)\times(q+1)}$ so that it has full rank; Sample $\boldsymbol{\varrho} \leftarrow_\$ \mathbb{Z}_p^n$;
    Set $[\boldsymbol{M}']_\iota \leftarrow \left[ \begin{smallmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{\varrho}^\intercal & 1 \end{smallmatrix} \right]_\iota \in \mathbb{Z}_p^{(q+1)\times(n+1)}$;
    Set $\mathsf{ck} \leftarrow \boldsymbol{R}[\boldsymbol{M}']_\iota \in \mathbb{G}_\iota^{(q+1)\times(n+1)}, \mathsf{td} \leftarrow (\mathsf{ek} \leftarrow \boldsymbol{R}^{-1}, \mathsf{tk} \leftarrow \boldsymbol{\varrho})$;
    return $(\mathsf{ck}, \mathsf{td})$;

$\mathsf{Com}(\mathsf{ck}; \boldsymbol{x} \in \mathbb{Z}_p^n; r \in \mathbb{Z}_p)$
    return $\mathsf{ck}\left( \begin{smallmatrix} \boldsymbol{x} \\ r \end{smallmatrix} \right)$;

$\mathsf{LExt}(\mathsf{ek}; [\mathsf{c}]_\iota)$
    return $\mathsf{ek}[\boldsymbol{c}]_\iota$ without the last element;
```

Fig. 1. Functional SSB commitment scheme FSSB_ι for linear functions in $\mathbb{G}_\iota$.

Intuitively, given ck, that depends on $\mathcal{S}$, and an extraction key, one can extract $F(\boldsymbol{x}_\mathcal{S})$. (This property was called somewhere perfect F-extractability in [12].)

Computational Hiding: $\forall$ PPT $\mathcal{A}$, $n \in \mathsf{poly}(\lambda)$, $q \in [1, n]$, $\mathsf{Adv}^{\mathsf{ch}}_{\Gamma,n,q,\mathcal{A}}(\lambda) := 2 \cdot |\varepsilon^{\mathsf{ch}}_{\Gamma,n,q,\mathcal{A}}(\lambda) - 1/2| = \mathsf{negl}(\lambda)$, where $\varepsilon^{\mathsf{ch}}_{\Gamma,n,q,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathcal{S} \leftarrow \mathcal{A}(\mathsf{p}, n, q) \text{ s.t. } \mathcal{S} \subseteq \mathcal{F} \wedge |\mathcal{S}| \leq q; \\ (\mathsf{ck}, \mathsf{td}) \leftarrow \mathsf{KC}(\mathsf{p}, n, q, \mathcal{S}); (\boldsymbol{x}_0, \boldsymbol{x}_1) \leftarrow \mathcal{A}(\mathsf{ck}); \beta \leftarrow_\$ \{0,1\}; r \leftarrow_\$ \mathcal{R} : \\ \mathcal{A}(\mathsf{Com}(\mathsf{ck}; \boldsymbol{x}_\beta; r)) = \beta \end{array}\right] .$$

Intuitively, given ck, that can depend on any $\mathcal{S}$, the commitment hides computationally the vector $\boldsymbol{x}$.

Construction. [12] constructed a functional SSB scheme for the family of all linear functions, see Fig. 1. It represents q linear functions by a matrix $[\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{q \times n}$, where each row contains coefficients of one function. Clearly, the commitment computes $[\mathsf{c}]_\iota \leftarrow \mathsf{Com}(\mathsf{ck}; \boldsymbol{x}; r) = \mathsf{ck}\left(\begin{smallmatrix} \boldsymbol{x} \\ r \end{smallmatrix} \right) = \boldsymbol{R}[\boldsymbol{M}']_\iota \left(\begin{smallmatrix} \boldsymbol{x} \\ r \end{smallmatrix} \right) = \left[\begin{smallmatrix} \boldsymbol{R}\boldsymbol{M}\boldsymbol{x} \\ \boldsymbol{R}(\boldsymbol{\varrho}^\top \boldsymbol{x} + r) \end{smallmatrix} \right]_\iota$, while $\mathsf{LExt}(\mathsf{ek}; [\mathsf{c}]_\iota)$ computes $\mathsf{ek} \cdot [\mathsf{c}]_\iota = \boldsymbol{R}^{-1}[\boldsymbol{R}\boldsymbol{M}'\left(\begin{smallmatrix} \boldsymbol{x} \\ r \end{smallmatrix} \right)]_\iota = [\boldsymbol{M}'\left(\begin{smallmatrix} \boldsymbol{x} \\ r \end{smallmatrix} \right)]_\iota = \left[\begin{smallmatrix} \boldsymbol{M}\boldsymbol{x} \\ \boldsymbol{\varrho}^\top \boldsymbol{x} + r \end{smallmatrix} \right]_\iota$, and returns $[\boldsymbol{M}\boldsymbol{x}]_\iota$.

Proposition 1 ([12]). *Let* Pgen *be a bilinear group generator. Fix data size* n *and locality parameter* q. *The commitment scheme in Fig. 1 is (i) function-set hiding relative to* Pgen *under the* $DDH_{\mathbb{G}_\iota}$ *assumption: for each PPT* $\mathcal{A}$*, there exists a PPT* $\mathcal{B}$*, such that* $\mathsf{Adv}^{\mathrm{fsh}}_{\Gamma,n,q,\mathcal{A}}(\lambda) \leq \lceil \log_2(q+1) \rceil \cdot \mathsf{Adv}^{\mathrm{ddh}}_{\mathbb{G}_\iota,\mathsf{Pgen},\mathcal{B}}(\lambda)$. *(ii) locally* F*-extractable for* $F = [\cdot]_\iota$ *(thus,* F *depends on* p*), (iii) almost everywhere perfectly-hiding, (iv) computationally-hiding. More precisely, for all PPT* $\mathcal{A}$*, there exist PPT* $\mathcal{B}_1$ *and unbounded* $\mathcal{B}_2$*, such that* $\mathsf{Adv}^{\mathrm{ch}}_{\Gamma,n,q,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{fsh}}_{\Gamma,n,q,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathrm{aeph}}_{\Gamma,n,q,\mathcal{B}_2}(\lambda)$.

Due to (iv), computational hiding does not have to be proven separately since it always follows from function-set hiding and almost everywhere perfect hiding.

2.2 Sub-ZK NIZK And QA-NIZK

In the current paper, we use both NIZKs and quasi-adaptive NIZKs [27]. To save space, we first give a complete description of QA-NIZKs (both since QA-NIZKs are less known and their security definitions subsume those of NIZKs) and then point out the differences in the case of NIZKs. We postpone the formal definitions of non-QA NIZKs to the full version [35].

A QA-NIZK argument system in the CRS model proves membership in the language $\mathbf{L}_{\mathsf{lpar}}$ defined by a relation $\mathbf{R}_{\mathsf{lpar}} = \{(\mathbb{x}, \mathbb{w})\}$, where both are determined by a language parameter lpar. In the honest case, lpar is sampled from a distribution $\mathcal{D}_{\mathsf{p}}$; let $\mathsf{setup.lpar}$ be the PPT algorithm that does this sampling. We assume that lpar contains p, and thus, we do not include p as an argument to algorithms that also input lpar; recall that we assumed that p cannot be subverted. A distribution $\mathcal{D}_{\mathsf{p}}$ is *witness-sampleable* if there exists a PPT algorithm $\mathsf{setup.ltrap}$ that samples $(\mathsf{lpar}, \mathsf{ltrap})$ such that lpar is distributed according to $\mathcal{D}_{\mathsf{p}}$, and the membership of lpar in $\mathbf{L}_{\mathsf{p}}$ can be efficiently verified given ltrap. The CRS crs can depend on lpar, but the simulator has to be a single algorithm that works for the whole collection of relations $\mathbf{R}_{\mathsf{p}} = \{\mathbf{R}_{\mathsf{lpar}}\}_{\mathsf{lpar} \in \mathrm{image}(\mathcal{D}_{\mathsf{p}})}$. We will assume that crs contains lpar implicitly.

The zero-knowledge simulator is usually required to be a single (non-black-box) PPT algorithm that works for the whole collection of relations $\mathbf{R}_{\mathsf{p}} = \{\mathbf{R}_{\mathsf{lpar}}\}_{\mathsf{lpar} \in \mathrm{image}(\mathcal{D}_{\mathsf{p}})}$; that is, one requires *uniform simulation* (see [27]). Following [1,14,3], we accompany the universal simulator Sim with an adversary-dependent extractor. We assume Sim also works when one cannot efficiently establish whether $\mathsf{lpar} \in \mathrm{image}(\mathcal{D}_{\mathsf{p}})$. The simulator is not allowed to create new lpar or crs but has to operate with one given to it as an input.

A *Sub-ZK QA-NIZK argument system in the CRS model* for a set of witness-relations $\mathbf{R}_{\mathsf{p}} = \{\mathbf{R}_{\mathsf{lpar}}\}_{\mathsf{lpar} \in \mathrm{image}(\mathcal{D}_{\mathsf{p}})}$ is a tuple of PPT algorithms $\Pi = (\mathsf{Pgen}, \mathsf{setup.lpar}, \mathsf{K}_{\mathsf{crs}}, \mathsf{PARV}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$. In the case of witness-sampleable languages, $\mathsf{setup.lpar}$ is replaced by $\mathsf{setup.ltrap}$. Here, Pgen is the parameter generation algorithm, $\mathsf{setup.lpar}$ is the language parameter generation algorithm, $\mathsf{setup.ltrap}$ is the corresponding $\mathsf{lpar}/\mathsf{ltrap}$ generation algorithm in the witness-sampleable case, $\mathsf{K}_{\mathsf{crs}}$ is the CRS generation algorithm, PARV is the lpar-verification algorithm, CV is the CRS verification algorithm, P is the prover, V is the verifier, and Sim is the simulator.

Π can satisfy the following security notions. Intuitively, quasi-adaptive soundness is soundness in the case when lpar is honestly generated. Quasi-adaptive strong soundness is soundness when lpar is honestly generated from a witness-sampleable distribution, and the adversary additionally gets access to ltrap. Adaptive soundness is soundness in the case of maliciously generated lpar. In all previous cases, the adversary sees crs before creating the input $\mathbb{x}$. Non-adaptive soundness is soundness in the case of maliciously generated lpar when the adversary has to fix $\mathbb{x}$ before seeing crs. Similar intuition holds in the case of various knowledge-soundness notions. Quasi-adaptive (knowledge)-soundness follows from adaptive (knowledge-)soundness. (Quasi-)adaptive soundness follows from (quasi-)adaptive knowledge-soundness.

Perfect Completeness: $\forall$ λ, PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathsf{lpar} \leftarrow \mathsf{setup.lpar}(\mathsf{p}); (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar});\\ (\mathbb{x}, \mathbb{w}) \leftarrow \mathcal{A}(\mathsf{crs}); \pi \leftarrow \mathsf{P}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \mathbb{w}) : \mathsf{PARV}(\mathsf{lpar}) = 1 \wedge \\ \mathsf{CV}(\mathsf{lpar}, \mathsf{crs}) = 1 \wedge ((\mathbb{x}, \mathbb{w}) \not\in \mathbf{R}_{\mathsf{lpar}} \vee \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1)\end{array}\right] = 1 \enspace .$$

Computational Quasi-Adaptive Strong Soundness: defined if lpar is witness-sampleable. For any stateful PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{strsound}}_{\mathsf{Pgen},\Pi,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathsf{lpar}, \mathsf{ltrap}) \leftarrow \mathsf{setup.ltrap}(\mathsf{p}); (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar});\\ (\mathbb{x}, \pi) \leftarrow \mathcal{A}(\mathsf{lpar}, \mathsf{ltrap}, \mathsf{crs}) : \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1 \wedge \neg(\exists \mathbb{w}. \mathbf{R}_{\mathsf{lpar}}(\mathbb{x}, \mathbb{w}) = 1)\end{array}\right] \approx_\lambda 0 \enspace .$$

In the definition of computational quasi-adaptive soundness (also defined in the non-witness-sampleable case), the only difference is that one samples $\mathsf{lpar} \leftarrow \mathsf{setup.lpar}(\mathsf{p})$, and the adversary does not get ltrap as an input.

Computational Non-Adaptive Soundness: $\forall$ stateful PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{nas}}_{\mathsf{Pgen},\Pi,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathsf{lpar}, \mathbb{x}) \leftarrow \mathcal{A}(\mathsf{p}); (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar}); \pi \leftarrow \mathcal{A}(\mathsf{crs}) :\\ \mathsf{PARV}(\mathsf{lpar}) = 1 \wedge \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1 \wedge \neg(\exists \mathbb{w}. \mathbf{R}_{\mathsf{lpar}}(\mathbb{x}, \mathbb{w}) = 1)\end{array}\right] \approx_\lambda 0 \enspace .$$

Computational Adaptive Soundness: $\forall$ stateful PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{as}}_{\mathsf{Pgen},\Pi,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathsf{lpar} \leftarrow \mathcal{A}(\mathsf{p}); (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar}); (\mathbb{x}, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) :\\ \mathsf{PARV}(\mathsf{lpar}) = 1 \wedge \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1 \wedge \neg(\exists \mathbb{w}. \mathbf{R}_{\mathsf{lpar}}(\mathbb{x}, \mathbb{w}) = 1)\end{array}\right] \approx_\lambda 0 \enspace .$$

Computational Adaptive Knowledge-Soundness: $\forall$ PPT stateful adversary $\mathcal{A}$, there exist a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, s.t. $\mathsf{Adv}^{\mathrm{aks}}_{\mathsf{Pgen},\Pi,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A}); \mathsf{lpar} \leftarrow \mathcal{A}(\mathsf{p}, r);\\ (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar}); (\mathbb{x}, \pi) \leftarrow \mathcal{A}(\mathsf{crs}; r); \mathbb{w} \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{p}, \mathsf{crs}; r) :\\ \mathsf{PARV}(\mathsf{lpar}) = 1 \wedge \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1 \wedge \mathbf{R}_{\mathsf{lpar}}(\mathbb{x}, \mathbb{w}) = 0\end{array}\right] \approx_\lambda 0 \enspace .$$

A knowledge-sound argument system is called an *argument of knowledge.*

Computational (resp., Perfect) Zero Knowledge: $\forall$ PPT (resp., unbounded) adversary $\mathcal{A}$, $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| \approx_\lambda 0$ (resp., $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| = 0$), where $\varepsilon_b^{zk} :=$

$$\Pr[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathsf{lpar} \leftarrow \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar}) : \mathcal{A}^{\mathcal{O}_b(\cdot,\cdot)}(\mathsf{lpar}, \mathsf{crs}) = 1] \enspace .$$

That is, $\mathcal{A}$ is given an oracle access to $\mathcal{O}_b(\cdot, \cdot)$, where $\mathcal{O}_0(\mathbb{x}, \mathbb{w})$ returns $\perp$ (reject) if $(\mathbb{x}, \mathbb{w}) \not\in \mathbf{R}_{\mathsf{lpar}}$, and otherwise it returns $\mathsf{P}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \mathbb{w})$. Similarly, $\mathcal{O}_1(\mathbb{x}, \mathbb{w})$ returns $\perp$ (reject) if $(\mathbb{x}, \mathbb{w}) \not\in \mathbf{R}_{\mathsf{lpar}}$, and otherwise it returns $\mathsf{Sim}(\mathsf{lpar}, \mathsf{crs}, \mathsf{td}, \mathbb{x})$.

Intuitively, zero knowledge in this sense corresponds to black-box zero-knowledge in the case when lpar and crs are trusted.

Computational (resp., Perfect) Persistent Zero Knowledge: $\forall$ PPT subverter $\mathcal{Z}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{Z}}$, s.t. $\forall$ PPT (resp., unbounded) adversary $\mathcal{A}$, $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| \approx_\lambda 0$ (resp., $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| = 0$), where $\varepsilon_b^{zk} :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{Z}); (\mathsf{lpar}, \mathsf{crs}, \mathsf{aux}) \leftarrow \mathcal{Z}(\mathsf{p}, r); \mathsf{td} \leftarrow \mathsf{Ext}_{\mathcal{Z}}(\mathsf{p}, r) :\\ \mathsf{PARV}(\mathsf{lpar}) = 1 \wedge \mathsf{CV}(\mathsf{lpar}, \mathsf{crs}) = 1 \wedge \mathcal{A}^{\mathcal{O}_b(\cdot,\cdot)}(\mathsf{lpar}, \mathsf{crs}, \mathsf{aux}) = 1\end{array}\right] \enspace .$$

The oracles are as above. Persistent zero-knowledge corresponds to non-black-box zero-knowledge in the case when lpar and crs are not trusted.

Π is *Sub-ZK* if it is both perfectly ZK and perfectly persistent zero-knowledge. ZK does not follow from persistent zero-knowledge in the case of QA-NIZKs [2] and thus, one has to prove both properties separately.

NIZKs. In the case of a (non-QA) NIZK, there is no language parameter and thus, no algorithms $\mathsf{setup.lpar}$ and PARV; other algorithms (including the adversary) do not take lpar as an argument or output it. Thus, $\Pi = (\mathsf{Pgen}, \mathsf{K}_{\mathsf{crs}}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$. Moreover, one deals with a single non-parametrized language $\mathbf{L}$. Otherwise, all properties of QA-NIZKs carry over but in a simplified form. Note that (1) one is not interested in quasi-adaptive (strong) soundness and (2) Sub-ZK and persistent zero-knowledge coincide. We postpone the formal definitions of non-QA NIZKs to the full version [35].

SNARKs. A (QA-)NIZK is *succinct* ((QA-)SNARG) if the argument π has a sublinear (desirably, logarithmic) length in $\mathsf{poly}(\lambda)(|\mathbb{x}| + |\mathbb{w}|)$. A *(QA-)SNARK* is a (QA-)SNARG that is additionally knowledge-sound.

Gentry-Wichs Impossibility Result. Gentry and Wichs [17] proved that if an NP language $\mathbf{L}$ has a sub-exponentially (resp., exponentially) hard subset-membership proof and Π is a complete SNARG in the CRS model with $|\pi| = \mathsf{poly}(\lambda)(|\mathbb{x}| + |\mathbb{w}|)^{o(1)}$ (resp., $|\pi| = \mathsf{poly}(\lambda)(|\mathbb{x}| + |\mathbb{w}|)^c + o(|\mathbb{x}| + |\mathbb{w}|)$ for some constant $c < 1$) for $\mathbf{L}$, then there is a black-box reduction from the adaptive soundness of Π to a falsifiable assumption X only when X is false.

3 Sub-ZK Bilateral Subspace QA-SNARK

A bilateral subspace argument system, with $\mathsf{lpar} = [\boldsymbol{M}]_*$, allows to prove that $[\mathbf{c}_1]_1 \in \mathbb{G}_1^{n_1}$ and $[\mathbf{c}_2]_2 \in \mathbb{G}_2^{n_2}$ satisfy $\left(\begin{smallmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{smallmatrix}\right) \in \mathrm{colspace}\left(\begin{smallmatrix}\boldsymbol{M}_1\\ \boldsymbol{M}_2\end{smallmatrix}\right)$. Following [10,12], we will use it to construct QA-SNARGs. Next, we prove that BLS, a variant of the González-Hevia-Ráfols bilateral subspace QA-SNARG, satisfies stronger properties, needed for FANA to be non-adaptively knowledge-sound and Sub-ZK.

First, let σ be any efficiently computable function. A distribution $\mathcal{D}_{\mathsf{p}}$ is *σ-witness-sampleable* if (1) there exists a PPT algorithm $\mathsf{setup.ltrap}_\sigma$ that samples $(\mathsf{lpar}, \sigma(\mathsf{ltrap}))$ such that lpar is distributed according to $\mathcal{D}_{\mathsf{p}}$, and (2) for any language trapdoor ltrap', such that the membership of lpar in the parameter language $\mathbf{L}_{\mathsf{p}}$ can be efficiently verified given ltrap', it holds that $\sigma(\mathsf{ltrap}) = \sigma(\mathsf{ltrap}')$. (In the context of the current paper, think of ltrap as the discrete logarithm of lpar, and $\sigma(\mathsf{ltrap})$ as an efficient—fixed—leakage function of ltrap.) We will prove that BLS satisfies the following new security property that follows from the quasi-adaptive strong soundness (see page 14):

Computational Quasi-Adaptive σ-Strong Soundness: defined if lpar is σ-witness-sampleable. For any stateful PPT $\mathcal{A}$, $\mathsf{Adv}^{\sigma-\mathrm{strsound}}_{\mathsf{Pgen},\Pi,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathsf{lpar}, \sigma(\mathsf{ltrap})) \leftarrow \mathsf{setup.ltrap}_\sigma(\mathsf{p}); (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{lpar});\\ (\mathbb{x}, \pi) \leftarrow \mathcal{A}(\mathsf{lpar}, \sigma(\mathsf{ltrap}), \mathsf{crs}) : \mathsf{V}(\mathsf{lpar}, \mathsf{crs}, \mathbb{x}, \pi) = 1 \land\\ \neg(\exists \mathbb{w}. \mathbf{R}_{\mathsf{lpar}}(\mathbb{x}, \mathbb{w}) = 1)\end{array}\right] \approx_\lambda 0 \enspace .$$

This notion agrees with the quasi-adaptive strong soundness when $\sigma = id$ is the identity function and with the quasi-adaptive soundness if σ is a constant function. While BLS is quasi-adaptively strongly sound and thus also quasi-adaptively σ-strongly sound for any efficient σ, we find it instructive to define σ-strong soundness. In particular, for the non-adaptive soundness of FANA, we will need BLS to be σ_x-strongly sound for a well-defined function σ_x. It is possible that one can find a more efficient version of BLS that is quasi-adaptively σ_x-strongly sound but not quasi-adaptively strong sound.

Assume that the matrix security parameter is $\kappa = 2$ (if $\kappa = 1$ then SKerMDH does not hold, [19]). Assume $\tau := \operatorname{corank}(\boldsymbol{M}) = n_1 + n_2 - \operatorname{rank}(\boldsymbol{M}) \geq 1$; here, n_1, n_2 can be smaller or larger (only the latter case was studied in [19]) than m. For $\mathsf{lpar} \in \mathbb{G}_1^{n_1 \times m} \times \mathbb{G}_2^{n_2 \times m}$, where $\mathsf{lpar} = [\boldsymbol{M}]_*$, define the bilateral subspace language (also known as the subspace concatenation language, [19])

$$\mathbf{L}_{\mathsf{lpar}} := \left\{ ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2) \in \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} : \exists \mathbb{w} \in \mathbb{Z}_p^m . \left(\begin{smallmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{smallmatrix} \right) = \left(\begin{smallmatrix} \boldsymbol{M}_1 \\ \boldsymbol{M}_2 \end{smallmatrix} \right) \mathbb{w} \right\} \enspace .$$

That is, $\mathbf{c}_1 = \boldsymbol{M}_1 \mathbb{w}$ and $\mathbf{c}_2 = \boldsymbol{M}_2 \mathbb{w}$.

A distribution $\mathcal{D}_\kappa$ is *efficiently verifiable* [2], if there exists a PPT algorithm $\mathsf{MATV}([\bar{\boldsymbol{A}}]_2)$ that outputs 1 if $\bar{\boldsymbol{A}}$ is invertible (recall that we assume that the matrix distribution is robust) and well-formed with respect to $\mathcal{D}_\kappa$, and otherwise outputs 0. Clearly, the standard distributions (see the full version [35]) $\mathcal{U}_1$, $\mathcal{L}_\kappa$, $\mathcal{IL}_\kappa$, $\mathcal{C}_\kappa$, and $\mathcal{SC}_\kappa$ (for any κ) are verifiable [2], while the verification whether $[\bar{\boldsymbol{A}}]_2$ is invertible is intractable for $\mathcal{U}_\kappa$ if $\kappa > 1$. To be able to handle $\mathcal{U}_\kappa$, [2] added parts of $[\bar{\boldsymbol{A}}]_1$ to crs. However, in the $\mathcal{U}_\kappa$ case, they proved adaptive soundness under the SKerMDH$^{\mathrm{dl}}$ (that we will define in Section 3.1) assumption instead of the KerMDH$^{\mathrm{dl}}$ assumption (see [2] for more discussion), which resulted in the choice $\kappa = 2$. $[\bar{\boldsymbol{A}}]_1$ is always in crs of a bilateral subspace argument system and thus the adaptive soundness relies on (a variant of) the SKerMDH$^{\mathrm{dl}}$ assumption.

As before, assume that the distribution $\mathcal{D}_\kappa$ is robust. Extending the definition of [2], we say that $\mathcal{D}_\kappa$ is *efficiently verifiable*, if there exists an algorithm $\mathsf{MATV}([\bar{\boldsymbol{A}}]_1, [\bar{\boldsymbol{A}}]_2)$ that outputs 1 if $\bar{\boldsymbol{A}}$ is invertible and well-formed with respect to $\mathcal{D}_\kappa$ and otherwise outputs 0. Here, MATV gets two inputs, $[\bar{\boldsymbol{A}}]_1$ and $[\bar{\boldsymbol{A}}]_2$; there are cases when an efficient MATV does not exist when only $[\bar{\boldsymbol{A}}]_1$ is given as the input. In particular, under this definition, also $\mathcal{U}_2$ is efficiently verifiable.

We depict a slight variant of the González-Hevia-Ràfols bilateral subspace QA-SNARG argument system BLS for $\mathbf{L}_{[\boldsymbol{M}_1]_1, [\boldsymbol{M}_2]_2}$ in Fig. 2. Compared to [19], we add the CRS verification algorithm CV and assume the existence of $\mathsf{setup.ltrap}_\sigma$ for some efficiently computable function σ. As in [19], the prover's work is dominated by $2m\kappa$ exponentiations, the verifier's work is dominated by $(n_1 + n_2 + 2\kappa)\kappa$ pairings, and the argument consists of 2κ group elements. Theorem 1 generalizes a theorem from [19] to any $n_\iota \times m$ matrices $\boldsymbol{M}_\iota$ (even if $m > n_\iota$), given that $\tau := n_1 + n_2 - \operatorname{rank}(\boldsymbol{M}) \geq 1$. This generalization is important since in FANA (see Eq. (4)), $m > n_2$. On top of the known results that BLS is quasi-adaptively (strongly) sound and zero-knowledge, we prove that BLS is quasi-adaptively σ-strongly sound (for any efficient σ), adaptively sound, adaptively knowledge-sound, persistent zero-knowledge, and thus Sub-ZK. To state Theorem 1, we will first need to define several security assumptions.

```
$\mathbf{L}_{\mathsf{p}} = \{[\boldsymbol{M}]_* \in \mathbb{G}_1^{n_1 \times m} \times \mathbb{G}_2^{n_2 \times m} : \tau := n_1 - \mathrm{rank}(\boldsymbol{M}_1) = n_2 - \mathrm{rank}(\boldsymbol{M}_2) \geq 1\}$

$\mathsf{setup.lpar}(\mathsf{p})$
    $([\boldsymbol{M}]_*, \sigma(\boldsymbol{M}_1, \boldsymbol{M}_2)) \leftarrow_\$ \mathsf{setup.ltrap}_\sigma(\mathsf{p})$;
    $\mathbf{return}\ \mathsf{lpar} \leftarrow [\boldsymbol{M}]_*$;

$\mathsf{BLS.K_{crs}}(\mathsf{p}, \mathsf{lpar} = ([\boldsymbol{M}_1]_1, [\boldsymbol{M}_2]_2))$
    $\boldsymbol{A} \leftarrow_\$ \mathcal{D}_\kappa$;   // $\boldsymbol{A} \in \mathbb{Z}_p^{(\kappa+1)\times\kappa}$, $\bar{\boldsymbol{A}}$ is invertible
    $\boldsymbol{K}_1 \leftarrow_\$ \mathbb{Z}_p^{n_1 \times \kappa}; \boldsymbol{K}_2 \leftarrow_\$ \mathbb{Z}_p^{n_2 \times \kappa}; \boldsymbol{\Delta} \leftarrow_\$ \mathbb{Z}_p^{\kappa \times m}$;
    $\boldsymbol{C}_1 \leftarrow \boldsymbol{K}_1 \bar{\boldsymbol{A}}; \boldsymbol{C}_2 \leftarrow \boldsymbol{K}_2 \bar{\boldsymbol{A}}$;   // $\boldsymbol{C}_\iota \in \mathbb{Z}_p^{n_\iota \times \kappa}$
    $[\boldsymbol{P}_1]_1 \leftarrow \boldsymbol{K}_1^\top [\boldsymbol{M}_1]_1 + [\boldsymbol{\Delta}]_1$;
    $[\boldsymbol{P}_2]_2 \leftarrow \boldsymbol{K}_2^\top [\boldsymbol{M}_2]_2 - [\boldsymbol{\Delta}]_2$; // $[\boldsymbol{P}_\iota]_\iota \in \mathbb{G}_\iota^{\kappa \times m}$
    $\mathsf{crs} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}_2, \boldsymbol{P}_1]_1, [\bar{\boldsymbol{A}}, \boldsymbol{C}_1, \boldsymbol{P}_2]_2)$;
    $\mathsf{td} \leftarrow (\boldsymbol{K}_1, \boldsymbol{K}_2)$;
    $\mathbf{return}\ (\mathsf{crs}, \mathsf{td})$;

$\mathsf{BLS.P}(\mathsf{p}, \mathsf{crs}; ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2), \mathbb{w})$
    $\boldsymbol{\zeta} \leftarrow_\$ \mathbb{Z}_p^\kappa$;
    $[\boldsymbol{\psi}_1]_1 \leftarrow [\boldsymbol{P}_1]_1 \mathbb{w} + [\boldsymbol{\zeta}]_1$;
    $[\boldsymbol{\psi}_2]_2 \leftarrow [\boldsymbol{P}_2]_2 \mathbb{w} - [\boldsymbol{\zeta}]_2$;   // $[\boldsymbol{\psi}_\iota]_\iota \in \mathbb{G}_\iota^\kappa$
    $\mathbf{return}\ \psi \leftarrow ([\boldsymbol{\psi}_1]_1, [\boldsymbol{\psi}_2]_2)$;

$\mathsf{BLS.Sim}(\mathsf{p}, \mathsf{crs}; ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2), \mathsf{td})$
    $\boldsymbol{\zeta}' \leftarrow_\$ \mathbb{Z}_p^\kappa$;
    $[\boldsymbol{\psi}'_1]_1 \leftarrow \boldsymbol{K}_1^\top [\mathbf{c}_1]_1 + [\boldsymbol{\zeta}']_1$;   // $[\mathbf{c}_\iota]_\iota \in \mathbb{G}_\iota^{n_\iota}$
    $[\boldsymbol{\psi}'_2]_2 \leftarrow \boldsymbol{K}_2^\top [\mathbf{c}_2]_2 - [\boldsymbol{\zeta}']_2$;   // $[\boldsymbol{\psi}'_\iota]_\iota \in \mathbb{G}_\iota^\kappa$
    $\mathbf{return}\ \psi' \leftarrow ([\boldsymbol{\psi}'_1]_1, [\boldsymbol{\psi}'_2]_2)$;

$\mathsf{BLS.V}(\mathsf{p}, \mathsf{crs}; ([\mathbf{c}_1]_1, [\mathbf{c}_2]_2), \psi)$
    $\mathbf{return}\ [\mathbf{c}_1]_1^\top \bullet [\boldsymbol{C}_1]_2 + ([\boldsymbol{C}_2]_1^\top \bullet [\mathbf{c}_2]_2)^\top \overset{?}{=} [\boldsymbol{\psi}_1]_1^\top \bullet [\bar{\boldsymbol{A}}]_2 + ([\bar{\boldsymbol{A}}]_1^\top \bullet [\boldsymbol{\psi}_2]_2)^\top$; // in $\mathbb{G}_T^{1 \times \kappa}$

$\mathsf{BLS.CV}([\boldsymbol{M}]_*, \mathsf{crs})$:
    $\mathbf{return}\ 1\ \mathbf{if}$ the following checks all succeed
            $\mathsf{crs} = ([\bar{\boldsymbol{A}}, \boldsymbol{C}_2, \boldsymbol{P}_1]_1, [\bar{\boldsymbol{A}}, \boldsymbol{C}_1, \boldsymbol{P}_2]_2)$;
            $[\boldsymbol{P}_1]_1 \in \mathbb{G}_1^{\kappa \times m} \wedge [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{\kappa \times \kappa} \wedge [\boldsymbol{C}_1]_2 \in \mathbb{G}_2^{n_1 \times \kappa}$;
            $[\boldsymbol{P}_2]_2 \in \mathbb{G}_2^{\kappa \times m} \wedge [\bar{\boldsymbol{A}}]_1 \in \mathbb{G}_2^{\kappa \times \kappa} \wedge [\boldsymbol{C}_2]_1 \in \mathbb{G}_1^{n_2 \times \kappa}$;
    $(\sharp)$  $[\bar{\boldsymbol{A}}]_1 \bullet [1]_2 = [1]_1 \bullet [\bar{\boldsymbol{A}}]_2$;
    $(*)$  $[\boldsymbol{M}_1]_1^\top \bullet [\boldsymbol{C}_1]_2 + [\boldsymbol{M}_2]_2^\top \bullet [\boldsymbol{C}_2]_1 = [\boldsymbol{P}_1]_1^\top \bullet [\bar{\boldsymbol{A}}]_2 + [\boldsymbol{P}]_2^\top \bullet [\bar{\boldsymbol{A}}]_1$;
            $\mathsf{MATV}([\boldsymbol{A}]_2) = 1$;
```

Fig. 2. The Sub-ZK bilateral subspace QA-SNARG BLS, for efficiently verifiable $\mathcal{D}_\kappa$.

3.1 New Security Assumptions

To state Theorem 1, we will first need to define two (non-falsifiable) non-adaptive security assumptions, SKerMDH$^{\mathrm{dl}}$ and SDL$^{\mathrm{dl}}$, that state that the SKerMDH and SDL [6] assumptions stay secure even if one is given a non-adaptive access to a discrete logarithm oracle in both $\mathbb{G}_1$ and $\mathbb{G}_2$. [30] used KerMDH to prove the quasi-adaptive soundness of their QA-SNARG Π_{kw} (assuming that lpar is honestly generated and witness-sampleable), and [2] used (non-falsifiable) non-adaptive interactive assumptions KerMDH$^{\mathrm{dl}}$ and SDL$^{\mathrm{dl}}$ to prove the adaptive soundness and knowledge-soundness of Π_{kw}. Witness-sampleability makes it possible for the reduction to generate lpar together with ltrap, and then use the knowledge of ltrap. The use of a non-falsifiable but reasonable looking non-adaptive interactive assumption allows the reduction to obtain ltrap by using the (non-polynomial-time) discrete logarithm oracles. Thus, one does not have to assume anymore that lpar is honestly generated. See [2] for discussion.

The intuition behind using different assumptions, compared to [2], is similar to the reason why BLS is sound under the SKerMDH and not under the KerMDH assumption. See [19] for discussion.

For $\iota \in \{1, 2\}$, the oracle $\mathrm{dl}_\iota([y]_\iota)$ returns the discrete logarithm y of $[y]_\iota$. *The* $\mathcal{D}_{\kappa^*,\kappa}$-SKerMDH$^{\mathrm{dl}}$ *assumption [2]* holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{skermdhdl}}_{\mathcal{D}_{\kappa^*,\kappa},\mathsf{Pgen},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathsf{st} \leftarrow \mathcal{A}^{\mathrm{dl}_1(\cdot),\mathrm{dl}_2(\cdot)}(\mathsf{p});\\ \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\kappa^*,\kappa}; ([\boldsymbol{c}_1]_1, [\boldsymbol{c}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, \mathsf{st}, [\boldsymbol{A}]_1, [\boldsymbol{A}]_2) :\\ \boldsymbol{A}^\top(\boldsymbol{c}_1 - \boldsymbol{c}_2) = \boldsymbol{0}_\kappa \wedge \boldsymbol{c}_1 - \boldsymbol{c}_2 \neq \boldsymbol{0}_{\kappa^*}\end{array}\right] \approx_\lambda 0 \enspace .$$

The SDL$^{\mathrm{dl}}$ *assumption* [2] holds relative to Pgen, if for any PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{sdldl}}_{\mathsf{Pgen},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathsf{st} \leftarrow \mathcal{A}^{\mathrm{dl}_1(\cdot),\mathrm{dl}_2(\cdot)}(\mathsf{p}); x \leftarrow_{\$} \mathbb{Z}_p :\\ \mathcal{A}(\mathsf{p}, \mathsf{st}, [x]_1, [x]_2) = x\end{array}\right] \approx_\lambda 0 \enspace .$$

In the version of SKerMDH$^{\mathrm{dl}}$ and SDL$^{\mathrm{dl}}$ from [2], $\mathcal{A}$ was only given access to the oracle dl_1. We decided to not change the name of the assumption.

[2] proved the persistent zero-knowledge of the Kiltz-Wee QA-SNARG argument system [30] under a new knowledge assumption KW-KE and then proved KW-KE's security in the algebraic group model, [15]. Since BLS is sufficiently different from [30], we need to define another knowledge assumption, GHR-KE (the *González-Hevia-Ràfols knowledge-of-exponent*). Intuitively, GHR-KE states that if one outputs an lpar and a crs, such that PARV and CV accept $(\mathsf{lpar}, \mathsf{crs})$ correspondingly, then one must know $\mathsf{td} = (\boldsymbol{K}_1, \boldsymbol{K}_2)$. This also gives an intuition of the role that is filled by PARV and CV. In the full version [35], we prove the security of GHR-KE in the AGM.

Definition 1. *Fix $\kappa \geq 1$, $n > m \geq 1$, and a distribution $\mathcal{D}_\kappa$. Let* $\mathsf{BLS.PARV}$ *and* $\mathsf{BLS.CV}$ *be as in Fig. 2.* $(\mathcal{D}_{\mathsf{p}}, \kappa, \mathcal{D}_\kappa)$-GHR-KE *holds relative to* Pgen *if for any PPT $\mathcal{A}$, there exists a PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, s.t.* $\mathsf{Adv}^{\mathrm{ghrke}}_{\mathcal{D}_{\mathsf{p}},\kappa,\mathcal{D}_\kappa,\mathsf{Pgen},\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A}); (\mathsf{lpar} := [\boldsymbol{M}]_*, \mathsf{crs}) \leftarrow \mathcal{A}(\mathsf{p}, r);\\ (\boldsymbol{K}_1, \boldsymbol{K}_2) \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{p}, r) : \mathsf{crs} = ([\bar{\boldsymbol{A}}, \boldsymbol{C}_1, \boldsymbol{P}_2]_1, [\bar{\boldsymbol{A}}, \boldsymbol{C}_2, \boldsymbol{P}_1]_2) \wedge\\ \mathsf{BLS.PARV}(\mathsf{lpar}) = 1 \wedge \mathsf{BLS.CV}(\mathsf{lpar}, \mathsf{crs}) = 1 \wedge\\ (\boldsymbol{P}_1 + \boldsymbol{P}_2 \neq \boldsymbol{K}_1^\top \boldsymbol{M}_1 + \boldsymbol{K}_2^\top \boldsymbol{M}_2)\end{array}\right] \approx_\lambda 0 \enspace .$$

3.2 Security Proof of BLS

Theorem 1. *Fix λ, n_1, n_2, m, let $\tilde{n} = n_1 + n_2$. Let $\kappa = 2$. Let σ be any efficient function. Let $\mathcal{D}_{\mathsf{p}}$ be a matrix distribution on $[\boldsymbol{M}]_* \in \mathbb{G}_1^{n_1 \times m} \times \mathbb{G}_2^{n_2 \times m}$, such that $\tilde{n} - \mathrm{rank}(\boldsymbol{M}) \geq 1$, where $\boldsymbol{M} := \left(\begin{smallmatrix}\boldsymbol{M}_1\\ \boldsymbol{M}_2\end{smallmatrix}\right)$. Then (1)* BLS *is perfectly complete and perfectly zero-knowledge. (2) If* $(\mathcal{D}_{\mathsf{p}}, \kappa, \mathcal{D}_\kappa)$-GHR-KE *holds relative to* Pgen*, then* BLS *is perfectly persistent zero-knowledge. (3) Assume $\mathcal{D}_\kappa$ is efficiently verifiable. If* $\mathcal{D}_\kappa$-SKerMDH$^{\mathrm{dl}}$ *holds relative to* Pgen*, then* BLS *is computationally adaptively sound. (4) Assume $\mathcal{D}_{\mathsf{p}}$ is σ-witness-sampleable and $\mathcal{D}_\kappa$ is efficiently verifiable. If* $\mathcal{D}_\kappa$-SKerMDH *holds relative to* Pgen *then* BLS *is computationally quasi-adaptively σ-strongly sound. (5) Assume that $\mathcal{D}_\kappa$ is robust. If* BLS *is computationally adaptively sound and* SDL$^{\mathrm{dl}}$ *holds relative to* Pgen*, then* BLS *is computationally adaptively knowledge-sound in the AGM.*

4 A Non-Adaptive SNARK FANA for QAP

Next, we propose a non-adaptively sound Sub-ZK SNARK FANA for QAP by following the ideas from [10,12] who proposed (quasi-adaptively sound) QA-SNARGs for SSP and SAP. A significant difference between QAP and SSP/SAP is that in QAP, one has to deal with different polynomials $u_j(X)$ and $v_j(X)$ in groups $\mathbb{G}_1$ and $\mathbb{G}_2$; this complicates the argument system since one has to include a functional SSB commitment in both groups. (In both [10,12], a functional SSB commitment is only given in $\mathbb{G}_1$.) On top of doing a version of the usual zk-SNARK with perfectly-hiding commitments to the evaluations $\mathsf{a} = A(x)$, $\mathsf{b} = B(x)$, and $\mathsf{c} = C(x)$ of three polynomials $A(X), B(X), C(X)$ (see Eq. (3); here, x is a trapdoor), we add (in both groups) a functional SSB commitment to specific values, explained later. We then use a bilateral subspace argument system [19] to show that all commitments are consistent.

More precisely, let $u(X)$, $v(X)$, and $w(X)$ be defined as in Section 2. In the new zk-SNARG, we define the following polynomials with randomizers r_a, r_b, r_c:

$$\begin{aligned} A(X) =& u(X) + r_a Z(X) \ , \\ B(X) =& v(X) + r_b Z(X) \ , \\ C(X) =& w(X) + r_c Z(X) \ , \\ h(X) =& (A(X)B(X) - C(X))/Z(X) \\ =& (u(X)v(X) - w(X))/Z(X) + (r_a v(X) + r_b u(X) - r_c) + r_a r_b Z(X) \ . \end{aligned} \tag{3}$$

V checks $[\mathsf{a}]_1 \bullet [\mathsf{b}]_2 - [\mathsf{c}]_1 \bullet [1]_2 = [h(x)]_1 [Z(x)]_2$, where $[\mathsf{a} = A(x), \mathsf{c} = C(x)]_1$, $[\mathsf{b} = B(x)]_2$ are circuit-dependent perfectly-hiding commitments. Intuitively, V checks $[V(x)]_2 = [0]_2$, where $V(X) := A(X)B(X) - C(X) - h(X)Z(X)$.

Let $[\boldsymbol{g}_u]_1 \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\boldsymbol{\mathcal{N}}_u]_1)$ and $[\boldsymbol{g}_v]_2 \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\boldsymbol{\mathcal{N}}_v]_2)$ be commitment keys of the functional SSB commitment scheme, with $\boldsymbol{g}_u \in \mathbb{Z}_p^{(N_1+1)\times(m+3)}$ and $\boldsymbol{g}_v \in \mathbb{Z}_p^{(N_2+1)\times(m+2)}$. Here, N_1 and N_2 are locality parameters set to $N_1 := 4$ and $N_2 := 2$. (We define $[\boldsymbol{\mathcal{N}}_u]_1$ and $[\boldsymbol{\mathcal{N}}_v]_2$ in Lemma 3; the choice of N_1 and N_2 will become later.)

In addition, we commit to the bases $\boldsymbol{u}(X) = (u_i(X))_{i=1}^m$, $\boldsymbol{v}(X) = (v_i(X))_{i=1}^m$, and $\boldsymbol{w}(X) = (w_i(X))_{i=1}^m$. For example, $[\mathsf{c}]_1 = \sum_{j=1}^m \mathbb{z}_j [w_j(x)]_1 + r_c [Z(x)]_1$. Since $w_j(X) = \sum_{i=1}^n W_{ij} \ell_i(X)$, then $[\mathsf{c}]_1 = \sum_{i=1}^n \sum_{j=1}^m W_{ij} \mathbb{z}_j [\ell_i(X)]_1 + r_c [Z(x)]_1 = \sum_{i=1}^n \sum_{j=1}^m (\boldsymbol{W} \mathbb{z})_i [\ell_i(X)]_1 + r_c [Z(x)]_1 = [\boldsymbol{g}_\ell]_1 \binom{\boldsymbol{W}\mathbb{z}}{r_c}$, where $[\boldsymbol{g}_\ell]_\iota := [\ell_1(x), \dots, \ell_n(x), Z(x)]_\iota$. Thus, $[\mathsf{c}]_1$ is an interpolation commitment [33] to the vector $\boldsymbol{W}\mathbb{z}$ (i.e., the vector of all output wires of all multiplication gates) with the randomness r_c. Similar formulas hold for $[\mathsf{a}]_1$ and $[\mathsf{b}]_2$.

Let $\hat{\boldsymbol{G}} = (\boldsymbol{I}_{m_0} \| \boldsymbol{0}_{m_0 \times (m-m_0)}) \in \mathbb{Z}_p^{m_0 \times m}$. Let $\hat{u}_i(X) = 0$ for $i \le m_0$ and $\hat{u}_i(X) = u_i(X)$ for $i > m_0$. Using $\hat{u}_i(X)$ instead of $u_i(X)$ helps us to prove efficiently that the prover used the correct public input $(\mathbb{z}_1, \dots, \mathbb{z}_{m_0})^\top = \mathbb{x}$. We use BLS to prove that several commitments commit to the same message while using different commitment keys, with

$$\begin{aligned}
\boldsymbol{H}_1 &= \left(\begin{array}{ccc|ccc|cc}
\hat{\boldsymbol{G}}^{(1)} & \dots & \hat{\boldsymbol{G}}^{(m)} & \mathbf{0}_{m_0} & \mathbf{0}_{m_0} & \mathbf{0}_{m_0} & \mathbf{0}_{m_0} & \mathbf{0}_{m_0} \\
\hat{u}_1(x) & \dots & \hat{u}_m(x) & Z(x) & 0 & 0 & 0 & 0 \\
w_1(x) & \dots & w_m(x) & 0 & 0 & Z(x) & 0 & 0 \\
\boldsymbol{g}_u^{(1)} & \dots & \boldsymbol{g}_u^{(m)} & \boldsymbol{g}_u^{(m+1)} & \mathbf{0}_{N_1+1} & \boldsymbol{g}_u^{(m+2)} & \boldsymbol{g}_u^{(m+3)} & \mathbf{0}_{N_1+1}
\end{array}\right) \enspace . \\
\boldsymbol{H}_2 &= \left(\begin{array}{ccc|ccc|cc}
v_1(x) & \dots & v_m(x) & 0 & Z(x) & 0 & 0 & 0 \\
\boldsymbol{g}_v^{(1)} & \dots & \boldsymbol{g}_v^{(m)} & \mathbf{0}_{N_2+1} & \boldsymbol{g}_v^{(m+1)} & \mathbf{0}_{N_2+1} & \mathbf{0}_{N_2+1} & \boldsymbol{g}_v^{(m+2)}
\end{array}\right) \enspace .
\end{aligned} \tag{4}$$

Here, $\boldsymbol{H}_1 \in \mathbb{Z}_p^{(m_0+N_1+3)\times(m+5)}$ and $\boldsymbol{H}_2 \in \mathbb{Z}_p^{(N_2+2)\times(m+5)}$. The witness is $(\mathbb{z}, r_a, r_b, r_c, r_u, r_v)$, where r_u and r_v are randomizers needed to randomize additional commitments. [4]

We use the bilateral subspace argument system to guarantee that

$$(\mathbb{x} /\!/ \mathsf{a} /\!/ \mathsf{c} /\!/ \tilde{\boldsymbol{c}}_u /\!/ \mathsf{b} /\!/ \tilde{\boldsymbol{c}}_v) \in \operatorname{colspace}\left(\begin{smallmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{smallmatrix}\right) \enspace . \tag{5}$$

That is, there exists $\mathsf{BLS}.\mathbb{w} = (\mathbb{z} = \left(\begin{smallmatrix} \mathbb{x} \\ \mathbb{w} \end{smallmatrix}\right), r_a, r_b, r_c, r_u, r_v)$, such that

- $[\mathsf{a}]_1 = \sum_{j=1}^m \mathbb{z}_j [\hat{u}_j(x)]_1 + r_a [Z(x)]_1 = \sum_{j=m_0+1}^m \mathbb{z}_j [u_j(x)]_1 + r_a [Z(x)]_1$,
- $[\mathsf{c}]_1 = \sum_{j=1}^m \mathbb{z}_j [w_j(x)]_1 + r_c [Z(x)]_1$,
- $[\tilde{\boldsymbol{c}}_u]_1 = \mathsf{FSSB}_1.\mathsf{Com}([\boldsymbol{g}_u]_1; \mathbb{z} /\!/ r_a /\!/ r_c; r_u)$,
- $[\mathsf{b}]_2 = \sum_{j=1}^m \mathbb{z}_j [v_j(x)]_2 + r_b [Z(x)]_2$, and
- $[\tilde{\boldsymbol{c}}_v]_2 = \mathsf{FSSB}_1.\mathsf{Com}([\boldsymbol{g}_v]_2; \left(\begin{smallmatrix} \mathbb{z} \\ r_b \end{smallmatrix}\right); r_v)$.

4.1 Description of FANA

We depict FANA in Fig. 3. The CRS of FANA consists of the public elements needed to compute all the commitments, $[h(x)]_1$, and the bilateral subspace argument system. The input of P and V is $\mathbb{x}$. The argument π includes $[\mathsf{a}, \mathsf{c}]_1$ and $[\mathsf{b}]_2$ (perfectly-hiding commitments to $\boldsymbol{U}\mathbb{z}$, $\boldsymbol{W}\mathbb{z}$, and $\boldsymbol{V}\mathbb{z}$, with randomizers r_a, r_c, and r_b) and $[\tilde{\boldsymbol{c}}_u]_1$ and $[\tilde{\boldsymbol{c}}_v]_2$ (functional SSB commitments to $\mathbb{z} /\!/ r_a /\!/ r_c$ and $\left(\begin{smallmatrix} \mathbb{z} \\ r_b \end{smallmatrix}\right)$). On top of that, the argument also contains $[h(x)]_1$ and a bilateral subspace argument $\mathsf{BLS}.\pi$. Here, $h(X)$ is as in the description of the QAP.

5 FANA: Assumptions and Soundness Proofs

5.1 The QA-LINRES Assumption

In the full version [35], we reproduce the known assumptions n-TSDH [37] (a well-known, relatively standard pairing-based assumption), n-S-TSDH (Assumption 7 and Assumption 8 in [10]; used to prove the soundness of the SNARG

[4] In [12], the structure of corresponding matrices was different, and thus one ended up with dimensions $[\boldsymbol{H}_1]_1 \in \mathbb{G}_1^{(2m+2)\times(2m+3)}, [\boldsymbol{H}_2]_2 \in \mathbb{G}_2^{5\times(2m+3)}$. In particular, they used Elgamal encryption as a perfectly-binding commitment in $\mathbb{G}_1$ (resulting in the addend $2m$ in the number of rows of $\mathbb{G}_1$).

```
$\mathsf{K_{crs}}(\mathsf{p}, \mathbf{R}_{\mathcal{I}_{\mathsf{qap}}})$:    // $n$ is implicit in $\mathsf{p}, \mathbf{R}$, matrices are as in Eq. (4)
    $\mathbf{N}_u \leftarrow_{\$} \mathbb{Z}_p^{N_1 \times (m+1)}$; $([\boldsymbol{g}_u]_1, \mathsf{td}_u) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\mathbf{N}_u]_1)$;
    $\mathbf{N}_v \leftarrow_{\$} \mathbb{Z}_p^{N_2 \times (m+1)}$; $([\boldsymbol{g}_v]_2, \mathsf{td}_v) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\mathbf{N}_v]_2)$;
    $x \leftarrow_{\$} \mathbb{Z}_p^*$; Create $\mathsf{BLS.lpar} \leftarrow [\boldsymbol{H}]_*$ as in Eq. (4);
    $(\mathsf{BLS.crs}, \mathsf{BLS.td}) \leftarrow \mathsf{BLS.K_{crs}}(\mathsf{p}, \mathsf{BLS.lpar})$;
    $\mathsf{crs} \leftarrow ([\boldsymbol{g}_u, (x^i)_{i=0}^n]_1, [\boldsymbol{g}_v, (x^i)_{i=0}^n]_2, \mathsf{BLS.lpar}, \mathsf{BLS.crs})$; $\mathsf{td} \leftarrow \mathsf{BLS.td}$;
    return $(\mathsf{crs}, \mathsf{td})$;
--------------------------------------------------------------------------------
$\mathsf{CV}(\mathsf{crs})$: Create $\mathsf{BLS.lpar} \leftarrow [\boldsymbol{H}]_*$ as in Eq. (4); Check $\mathsf{BLS.CV}(\mathsf{BLS.lpar}, \mathsf{BLS.crs}) = 1$;
--------------------------------------------------------------------------------
$\mathsf{P}(\mathsf{crs}, \mathbb{x} = (\mathbb{z}_1, \dots, \mathbb{z}_{m_0}); \mathbb{w} = (\mathbb{z}_j)_{j=m_0+1}^m)$:
     1. $r_a, r_b, r_c, r_u, r_v \leftarrow_{\$} \mathbb{Z}_p$;
     2. $A(X) \leftarrow \sum_{j=1}^m \mathbb{z}_j u_j(X) + r_a Z(X)$; $B(X) \leftarrow \sum_{j=1}^m \mathbb{z}_j v_j(X) + r_b Z(X)$;
     3. $C(X) \leftarrow \sum_{j=1}^m \mathbb{z}_j w_j(X) + r_c Z(X)$;
     4. $h(X) \leftarrow (A(X)B(X) - C(X))/Z(X)$;
     5. $[\mathsf{a}]_1 \leftarrow \sum_{j=m_0+1}^m \mathbb{z}_j [u_j(x)]_1 + r_a [Z(x)]_1$; $[\mathsf{c}]_1 \leftarrow \sum_{j=1}^m \mathbb{z}_j [w_j(x)]_1 +$
        $r_c [Z(x)]_1$; $[\mathsf{b}]_2 \leftarrow \sum_{j=1}^m \mathbb{z}_j [v_j(x)]_2 + r_b [Z(x)]_2$;
     6. $[\tilde{\boldsymbol{c}}_u]_1 \leftarrow \mathsf{FSSB}_1.\mathsf{Com}([\boldsymbol{g}_u]_1; \mathbb{z} // r_a // r_c; r_u)$;
     7. $[\tilde{\boldsymbol{c}}_v]_2 \leftarrow \mathsf{FSSB}_2.\mathsf{Com}([\boldsymbol{g}_v]_2; \left(\begin{smallmatrix} \mathbb{z} \\ r_b \end{smallmatrix}\right); r_v)$;
     8. $[h(x)]_1 \leftarrow \sum_{i=0}^{n-2} h_i [x^i]_1$;
     9. $\mathsf{BLS}.\mathbb{x} \leftarrow [\mathbb{x} // \mathsf{a} // \mathsf{c} // \tilde{\boldsymbol{c}}_u]_1 // [\mathsf{b} // \tilde{\boldsymbol{c}}_v]_2$;
    10. $\mathsf{BLS}.\pi \leftarrow \mathsf{BLS.P}(\mathsf{BLS.lpar}, \mathsf{BLS.crs}; \mathsf{BLS}.\mathbb{x}; (\mathbb{z}, r_a, r_b, r_c, r_u, r_v))$;
    11. $\pi \leftarrow ([\mathsf{a}, \mathsf{c}, \tilde{\boldsymbol{c}}_u, h(x)]_1, [\mathsf{b}, \tilde{\boldsymbol{c}}_v]_2, \mathsf{BLS}.\pi)$.
--------------------------------------------------------------------------------
$\mathsf{V}(\mathsf{crs}, \mathbb{x} = (\mathbb{z}_1, \dots, \mathbb{z}_{m_0}); \pi)$:
    $\mathsf{BLS}.\mathbb{x} \leftarrow [\mathbb{x} // \mathsf{a} // \mathsf{c} // \tilde{\boldsymbol{c}}_u]_1 // [\mathsf{b} // \tilde{\boldsymbol{c}}_v]_2$;
    check $\mathsf{BLS.V}(\mathsf{BLS.lpar}, \mathsf{BLS.crs}, \mathsf{BLS}.\mathbb{x}, \mathsf{BLS}.\pi)$ accepts and $([\mathsf{a}]_1 +$
    $\sum_{j=1}^{m_0} \mathbb{z}_j [u_j(x)]_1) \bullet [\mathsf{b}]_2 - [\mathsf{c}]_1 \bullet [1]_2 = [h(x)]_1 \bullet [Z(x)]_2$.
--------------------------------------------------------------------------------
$\mathsf{Sim}(\mathsf{crs}, \mathsf{td} = \mathsf{BLS.td}, \mathbb{x} = (\mathbb{z}_1, \dots, \mathbb{z}_{m_0}))$:
     1. $r_u, r_v, \mu_1, \mu_2, \mu_3 \leftarrow_{\$} \mathbb{Z}_p$;
     2. $[\mathsf{a}]_1 \leftarrow \mu_1 [Z(x)]_1 - \sum_{j=1}^{m_0} \mathbb{z}_j [u_j(x)]_1$; $[\mathsf{b}]_2 \leftarrow \mu_2 [Z(x)]_2$;
     3. $[\mathsf{c}]_1 \leftarrow \mu_3 [Z(x)]_1$;
     4. $[\tilde{\boldsymbol{c}}_u]_1 \leftarrow \mathsf{FSSB}_1.\mathsf{Com}([\boldsymbol{g}_u]_1; \mathbf{0}_{m+2}; r_u)$;
     5. $[\tilde{\boldsymbol{c}}_v]_2 \leftarrow \mathsf{FSSB}_2.\mathsf{Com}([\boldsymbol{g}_v]_2; \mathbf{0}_{m+1}; r_v)$;
     6. $[h(x)]_1 \leftarrow \mu_1 \mu_2 [Z(x)]_1 - \mu_3 [1]_1$;
     7. $\mathsf{BLS}.\mathbb{x} \leftarrow [\mathbb{x} // \mathsf{a} // \mathsf{c} // \tilde{\boldsymbol{c}}_u]_1 // [\mathsf{b} // \tilde{\boldsymbol{c}}_v]_2$;
     8. $\mathsf{BLS}.\pi \leftarrow \mathsf{BLS.Sim}(\mathsf{BLS.lpar} = [\boldsymbol{H}]_*, \mathsf{BLS.crs}, \mathsf{BLS.td}; \mathsf{BLS}.\mathbb{x})$.
     9. $\pi \leftarrow ([\mathsf{a}, \mathsf{c}, \tilde{\boldsymbol{c}}_u, h(x)]_1, [\mathsf{b}, \tilde{\boldsymbol{c}}_v]_2, \mathsf{BLS}.\pi)$.
```

Fig. 3. New zk-SNARK FANA for QAP.

DGPRS for SSP), n-Q-TSDH (Assumption 8 in [10]; used to prove the soundness of range proofs and some other argument systems), and n-SA-TSDH [12] (used to prove the soundness of a SNARG for SAP). The last three assumptions are known to hold under if n-TSDH and a suitable knowledge assumption hold. One can similarly define a new TSDH-related assumption QA-TSDH (see the full version [35]) to prove the non-adaptive soundness of FANA.

While $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH naturally extend the well-known assumption TSDH, they look complicated. Each of them is intrinsically related to the underlying language: S-TSDH is related to the SSP language, SA-TSDH is related to the SAP language, and QA-TSDH is related to the QAP language. Since SAP is a more involved language than SSP, SA-TSDH is more involved than S-TSDH.

Most importantly, in $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH, $\mathcal{A}$ returns an element $[\nu]_T$ of the target group $\mathbb{G}_T$. The widely-accepted way to motivate the security of an assumption like $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH is to analyze its security in the generic group model GGM, or in some of its weakenings like the algebraic group mode, AGM [15]. As explained in [26], in pairing-based settings, $\mathbb{G}_T$, being a subgroup of the multiplicative group of a finite field, should not be thought of as a generic group. Instead, [26] proposed the *semi-GGM*, where only $\mathbb{G}_1$ and $\mathbb{G}_2$ are considered to be generic groups. Since an $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH adversary returns $[\nu]_T$ in the target group, $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH is not secure in the semi-GGM.

Fortunately, this is a problem of the concrete assumptions, not intrinsic to the QA-SNARGs. We prove that FANA is sound under a different assumption, QA-LINRES, where the adversary *only returns elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

Definition 2 (QA-LINRES). n-Quadratic Arithmetic Linear Residuosity (n-QA-LINRES) *holds relative to* Pgen*, if* $\forall$ *PPT* $\mathcal{A}$*,* $\mathsf{Adv}^{\text{qa-linres}}_{\mathsf{Pgen},n,\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$*, where* $\mathsf{Adv}^{\text{qa-linres}}_{\mathsf{Pgen},n,\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); x, y \leftarrow_{\$} \mathbb{Z}_p^*; \mathsf{ck} \leftarrow (([x^i]_1, [x^i]_2)_{i=0}^n, [y]_1, [y]_2); \pi \leftarrow \mathcal{A}(\mathsf{p}, \mathsf{ck}) : \\ \pi = \Big(\mathsf{J}, [A(x), \alpha_u(x), \hat{\beta}_u, C(x), \alpha_w(x), \hat{\beta}_w, h(x)]_1, [B(x), \alpha_v(x), \hat{\beta}_v]_2\Big) \wedge \\ A(x) = \alpha_u(x)(x - \omega^{\mathsf{J}-1}) + \hat{\beta}_u/y \wedge B(x) = \alpha_v(x)(x - \omega^{\mathsf{J}-1}) + \hat{\beta}_v/y \wedge \\ C(x) = \alpha_w(x)(x - \omega^{\mathsf{J}-1}) + \hat{\beta}_w/y \wedge A(x)B(x) - C(x) = h(x)Z(x) \wedge \\ \hat{\beta}_u\hat{\beta}_v \neq \hat{\beta}_w y \end{array}\right] .$$

QA-LINRES is falsifiable since the challenger who created x and y can efficiently verify that the conditions hold. Like $\{*\}$TSDH, QA-LINRES is parameterized by n (the size of the instance) but does not depend on the instance otherwise.

Next, we will motivate the choice of the assumption. The penultimate equality above, $A(x)B(x) - C(x) = h(x)Z(x)$, is the key equation of the QAP. The first three equalities are explicitly motivated by the soundness proof of FANA; they intuitively guarantee that (say) when one divides the polynomial $A(X)$ with $X - \omega^{\mathsf{J}-1}$, then the remainder is (integer) β_u and the quotient is the polynomial $\alpha_u(X)$. To guarantee that (say) β_u is an integer (and thus does not depend on x), the $\{\mathsf{S}, \mathsf{Q}, \mathsf{SA}, \mathsf{QA}\}$TSDH assumptions introduce a new indeterminate y and require that the adversary also outputs $[\beta_u y]_1$. Since ck only contains $[y]_1$ (and no $[x^i y]_1$), it means that an algebraic adversary must know the integer β_u.

This trick means that in the case of the say SA-TSDH assumption (see the full version, [35]), the adversary has to return $[\beta_u, \beta_w]_1$ together with knowledge-components $[\hat{\beta}_w, \hat{\beta}_w]_1$; this makes the assumption more complicated. Moreover, in the soundness proof, the reduction has to extract all four elements. While

$[\beta_u, \beta_w]_1$ can be extracted from the perfectly-binding commitment scheme, the other two are extracted from the functional SSB commitment scheme, making the output of the functional SSB commitment longer. Following this blueprint, in the case of the QA-SNARG for QAP, there are three elements $[\beta_u, \beta_w]_1, [\beta_v]_2$ and thus there would be also three extra knowledge components $[\hat{\beta}_u, \hat{\beta}_w]_1, [\hat{\beta}_v]_2$.

In QA-LINRES, the adversary only has to return the knowledge-components $[\hat{\beta}_u, \hat{\beta}_w]_1, [\hat{\beta}_v]_2$ but not $[\beta_u, \beta_w]_1, [\beta_v]_2$. This results in a cleaner assumption (the adversary has to return fewer elements) and a more efficient QA-SNARG (the length of the functional SSB commitment is reduced by one group element).

Since the adversary of QA-LINRES does not output elements like $[\beta_u]_1$ together with their knowledge components anymore, the security of QA-LINRES cannot be directly ascertained under Damgård's knowledge-of-exponent assumptions. Hence, in the full version [35], we will prove that QA-LINRES holds in the AGM under the standard PDL assumption.

Theorem 2. *(1)* FANA *is perfectly complete. (2) If* BLS *is perfectly zero-knowledge and* FSSB_1 *and* FSSB_2 *are almost everywhere perfectly-hiding then* FANA *is perfectly zero-knowledge. (3) If* BLS *is perfectly persistent zero-knowledge,* FANA *is perfectly zero-knowledge, and* FSSB_1 *and* FSSB_2 *are computationally-hiding then* FANA *is Sub-ZK.*

Recall that if FSSB_ι is function-set hiding and almost everywhere perfectly-hiding, then it is also computationally-hiding; thus, for (3) it suffices if we assume function-set hiding and almost everywhere perfectly-hiding properties.

5.2 Non-Adaptive Soundness of FANA

Our non-adaptive soundness proof proceeds in four games. In the last game, Game_4, we construct two reductions. The first reduction is to the quasi-adaptive σ-strong soundness (for a fixed σ_x) of BLS that guarantees that no PPT non-adaptive soundness adversary $\mathcal{A}$ is successful if there exists no witness $\mathsf{BLS}.\mathbb{w}$, s.t. Eq. (5) holds; this includes the case $\mathcal{A}$ used a wrong public input.

Assume now that there exists at least one witness $\mathsf{BLS}.\mathbb{w}$, such that Eq. (5) holds. Then, a successful $\mathcal{A}$ left at least one constraint unsatisfied. The adversary $\mathcal{B}_3$ (constructed in the second, QA-LINRES, reduction) samples $\mathsf{J} \leftarrow_\$ [1, n]$ and guesses that the Jth QAP constraint $(\boldsymbol{U}_\mathsf{J}\mathbb{z})(\boldsymbol{V}_\mathsf{J}\mathbb{z}) = \boldsymbol{W}_\mathsf{J}\mathbb{z}$ is unsatisfied. $\mathcal{B}_3$ aborts in Game_4 if the guess was correct. In Game_3 and Game_4, we modify the functional SSB scheme's commitment key to be able to extract six elements (namely, $([\hat{\alpha}_u(x), \hat{\beta}_u, \hat{\alpha}_w(x), \hat{\beta}_w]_1, [\hat{\alpha}_v(x), \hat{\beta}_v]_2$; other elements can be computed in a straightforward way) needed to break QA-LINRES. $\mathcal{B}_3$ works with the modified commitment keys; inside the QA-LINRES experiment, $\mathcal{B}_3$ aborts if $\mathcal{A}$ satisfied the Jth constraint. This incurs an n-times security loss.

Crucially, $\mathcal{B}_3$ makes the decision to abort based on the information in modified functional SSB commitment keys. Thus, we can only abort in the last game Game_4. This is the main reason why we have both a succinct argument (abortion is not based on information, extracted from the perfectly-binding commitment as in [10,12]) and non-adaptive soundness (in the adaptive case, $\mathcal{A}$ sees the modified commitment key before creating the input, and then it can covertly choose the unsatisfied constraint based on it).

With the modified commitment keys, $\hat{\beta}_u = \boldsymbol{U}_{\mathsf{J}}\mathbb{z}$, $\hat{\beta}_v = \boldsymbol{V}_{\mathsf{J}}\mathbb{z}$, and $\hat{\beta}_w = \boldsymbol{W}_{\mathsf{J}}\mathbb{z}$ for some $\mathbb{z}$. If $\mathcal{B}_3$ aborts, then $\hat{\beta}_u\hat{\beta}_v \neq \hat{\beta}_w$. (The quasi-adaptive σ_x-strong soundness of BLS guarantees that such a $\mathbb{z}$ exists.) Since $A(X) = \sum_{i=1}^{n} \boldsymbol{U}_i\mathbb{z}\ell_i(X) + r_a Z(X)$, we get $A(X) \equiv \boldsymbol{U}_{\mathsf{J}}\mathbb{z} \pmod{Z(X)}$ and thus $A(X) \equiv \boldsymbol{U}_{\mathsf{J}}\mathbb{z} \pmod{X - \omega^{\mathsf{J}-1}}$. Similarly, $B(X) \equiv \boldsymbol{V}_{\mathsf{J}}\mathbb{z} \pmod{X - \omega^{\mathsf{J}-1}}$ and $C(X) \equiv \boldsymbol{W}_{\mathsf{J}}\mathbb{z} \pmod{X - \omega^{\mathsf{J}-1}}$. Thus, for some polynomials $\alpha_u(X)$, $\alpha_v(X)$, and $\alpha_w(X)$,

$$\begin{aligned} A(X) =&\alpha_u(X)(X-\omega^{\mathsf{J}-1}) + \hat{\beta}_u \enspace, \quad B(X) = \alpha_v(X)(X-\omega^{\mathsf{J}-1}) + \hat{\beta}_v \enspace, \\ C(X) =&\alpha_w(X)(X-\omega^{\mathsf{J}-1}) + \hat{\beta}_w \enspace. \end{aligned}$$

In the malicious case, $[\hat{\beta}_u]_1$, $[\hat{\beta}_v]_2$, and $[\hat{\beta}_w]_1$ can depend on x; e.g., $[\hat{\beta}_u]_1 = [\hat{\beta}_u(x)]_1$. Consider first the case $y = 1$. Then, the verification equation $A(X)B(X) - C(X) = h(X)Z(X)$ guarantees that $\hat{\beta}_u(X)\hat{\beta}_v(X) - \hat{\beta}_w(X) \equiv 0 \pmod{X - \omega^{\mathsf{J}-1}}$ as a polynomial while the QA-LINRES assumption states $\hat{\beta}_u\hat{\beta}_v \neq \hat{\beta}_w$. To obtain a contradiction, we need to guarantee that $\mathcal{B}_3$ returned $([\hat{\beta}_u, \hat{\beta}_v]_1, [\hat{\beta}_w]_2)$, such that $\hat{\beta}_u$, $\hat{\beta}_v$, and $\hat{\beta}_w$ do not depend on x. We achieve this by sampling a random y and adding $([y]_1, [y]_2)$ to crs; then an algebraic adversary can create (say) $[\hat{\beta}_u]_1 = [\boldsymbol{U}_{\mathsf{J}}\mathbb{z}y]_1$, such that $\hat{\beta}_u/y$ is in a non-trivial relation only if $\hat{\beta}_u$ does not depend on the trapdoor x.

Importantly, the non-adaptive soundness of FANA follows from falsifiable assumptions. Knowing which constraint J was unsatisfied, we use the local extractability of the functional SSB scheme to recover a succinct local witness that allows one to reduce the non-adaptive soundness to QA-LINRES. Thus, we do not need to have a perfectly-binding commitment. In comparison, [10,12] used witness-sampleability to extract some elements of that local witness from the perfectly-binding commitment scheme.

Let $\sigma_x : \mathsf{ltrap} \mapsto x$. Clearly, σ_x can be computed efficiently: given $(\boldsymbol{H}_1, \boldsymbol{H}_2)$, σ_x uses one of the entries of $\boldsymbol{H}_1$ that contains $Z(x)$ to compute the value of x. For $\iota \in \{1,2\}$, let FSSB_ι be the Fauzi-Lipmaa-Pindado-Siim functional SSB commitment scheme in $\mathbb{G}_\iota$. Let BLS be the González-Hevia-Ràfols bilateral subspace argument system. Let $N_1 = 4$ and $N_2 = 2$.

Theorem 3. *Assume that FSSB_ι is locally $[\cdot]_\iota$-extractable and function-set hiding for $\iota \in \{1,2\}$, BLS is quasi-adaptively σ_x-strongly sound, and n-QA-LINRES holds relative to Pgen. Then the QA-SNARK FANA from Fig. 3 is non-adaptively sound. More precisely, there exist PPT adversaries $\mathcal{B}_1, \mathcal{B}_1', \mathcal{B}_2, \mathcal{B}_3$ against the function-set hiding property of FSSB_1, the function-set hiding property of FSSB_2, the quasi-adaptive σ_x-strong soundness of BLS, and the n-QA-LINRES assumption, respectively, such that*

$$\begin{aligned} \mathsf{Adv}^{\mathrm{nas}}_{\mathsf{Pgen},\mathsf{FANA},\mathcal{A}}(\lambda) \leq& \mathsf{Adv}^{\mathrm{fsh}}_{\mathsf{FSSB}_1,m+2,N_1,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathrm{fsh}}_{\mathsf{FSSB}_2,m+1,N_2,\mathcal{B}_1'}(\lambda) + \\ & \mathsf{Adv}^{\sigma_{\mathrm{x}}-\mathrm{strsound}}_{\mathsf{Pgen},\mathsf{BLS},\mathcal{B}_2}(\lambda) + n \cdot \mathsf{Adv}^{\text{qa-linres}}_{\mathsf{Pgen},n,\mathcal{B}_3}(\lambda) \enspace. \end{aligned}$$

For the quasi-adaptive σ_x-strong soundness of BLS, the language parameter distribution of BLS must be σ_x-witness-sampleable. On the other hand, [10,12] assumed that $\mathsf{BLS.lpar}$ is witness-sampleable and thus also lpar for their QA-SNARG (for SSP/SAP) is witness-sampleable; by this reason alone, their SNARGs are only quasi-adaptively sound (i.e., sound, assuming lpar is honestly generated). FANA is not a QA-SNARG and thus has no language parameter.

Proof (of Theorem 3). The non-adaptive soundness proof consists of the following games. Let $\mathcal{A}$ be an adversary against the non-adaptive soundness. We recall that in the terminology of arithmetic circuits, $\mathcal{A}$ has two avenues of cheating: either by using a wrong public input or by leaving some constraints unsatisfied.

Game_1**:** this is the non-adaptive soundness game for non-QA NIZKs (see page 14 but remember that in the case of NIZKs, there is no lpar). The output is 1 if $\mathcal{A}$ produces a false accepting proof, i.e., either (1) there exists at least one constraint i, such that $(\boldsymbol{U}\mathbb{z})_i(\boldsymbol{V}\mathbb{z})_i \neq (\boldsymbol{W}\mathbb{z})_i$, or (2) the various committed values are either different or do not start with $\mathbb{x}$.

Game_2**:** This game also samples $\mathsf{J} \leftarrow_{\$} [1, n]$ as a guess for the unsatisfied equation i in the case (1).

Game_3**:** Let $\delta_{uj}(X)$ (resp., $\delta_{wj}(X)$ / $\delta_Z(X)$) be the quotient of the division of $u_j(X)$ (resp., $w_j(X)$ / $Z(X)$) with $X - \omega^{\mathsf{J}-1}$. We will show later that the remainder is $U_{\mathsf{J}j}$ (resp., $W_{\mathsf{J}j}$ / 0). We redefine the commitment key of the FSSB_1 scheme as $([\boldsymbol{g}_u]_1, \mathsf{td}_u) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\boldsymbol{\mathfrak{N}}_u]_1)$ for

$$[\boldsymbol{\mathfrak{N}}_u]_1 \leftarrow \begin{bmatrix} \delta_{u1}(x) & \dots & \delta_{um}(x) & \delta_Z(x) & 0 \\ U_{\mathsf{J}1}y & \dots & U_{\mathsf{J}m}y & 0 & 0 \\ \delta_{w1}(x) & \dots & \delta_{wm}(x) & 0 & \delta_Z(x) \\ W_{\mathsf{J}1}y & \dots & W_{\mathsf{J}m}y & 0 & 0 \end{bmatrix}_1 \in \mathbb{G}_1^{N_1 \times (m+2)} \quad . \tag{6}$$

In Lemma 3, this change allows us to use the local extractability of FSSB_1 to extract $[\hat{\alpha}_u, \hat{\beta}_u, \hat{\alpha}_w, \hat{\beta}_w]_1 (= [\alpha_u(x), \hat{\beta}_u, \alpha_w(x), \hat{\beta}_w]_1)$ related to the QA-LINRES assumption (see Definition 2).

Game_4**:** Let $\delta_{vj}(X)$ be the quotient of the division of $v_j(X)$ with $X - \omega^{\mathsf{J}-1}$. We will show later that the remainder is $V_{\mathsf{J}j}$. We redefine the commitment key of the FSSB_2 scheme as $([\boldsymbol{g}_v]_2, \mathsf{td}_v) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\boldsymbol{\mathfrak{N}}_v]_2)$ for

$$[\boldsymbol{\mathfrak{N}}_v]_2 \leftarrow \begin{bmatrix} \delta_{v1}(x) & \dots & \delta_{vm}(x) & \delta_Z(x) \\ V_{\mathsf{J}1}y & \dots & V_{\mathsf{J}m}y & 0 \end{bmatrix}_2 \in \mathbb{G}_2^{N_2 \times (m+1)} \quad . \tag{7}$$

In Lemma 3, this change allows us to use the local extractability of FSSB_2 to extract $[\hat{\alpha}_v, \hat{\beta}_v]_2 (= [\alpha_v(x), \hat{\beta}_v]_2)$ related to the QA-LINRES assumption.

We show that in Game_4, either one can (1) break the quasi-adaptive σ_x-strong soundness of BLS or (2) with probability $1/n$, compute $[\hat{\beta}_u, \hat{\beta}_w]_1$ and $[\hat{\beta}_v]_2$, where $\hat{\beta}_u / y = \boldsymbol{U}_{\mathsf{J}}\mathbb{z}$, $\hat{\beta}_v / y = \boldsymbol{V}_{\mathsf{J}}\mathbb{z}$, and $\hat{\beta}_w / y = \boldsymbol{W}_{\mathsf{J}}\mathbb{z}$, and thus break QA-LINRES. (Here, we need FSSB_ι to be locally $[\cdot]_\iota$-extractable.)

See Fig. 4 for the formal description of all games.

Game_1 and Game_2 are clearly indistinguishable.

Lemma 1. *There exist a PPT adversary* $\mathcal{B}_1$*, such that* $|\Pr[\mathsf{Game}_3(\mathcal{A}) = 1] - \Pr[\mathsf{Game}_2(\mathcal{A}) = 1]| \leq \mathsf{Adv}^{\mathrm{fsh}}_{\mathsf{FSSB}_1, m+2, N_1, \mathcal{B}_1}(\lambda)$.

```
$\mathsf{Game}_1$ / $\boxed{\mathsf{Game}_2}$ / $\boxed{\mathsf{Game}_3}$ / $\boxed{\mathsf{Game}_4}$
$\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \mathbb{x} \leftarrow \mathcal{A}(\mathsf{p});$    // Non-adaptive NIZK soundness adversary $\mathcal{A}$ (no $\mathsf{lpar}$)
$\boxed{\boxed{\boxed{\mathsf{J} \leftarrow_\$ [1, n];}}}$ $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{p}); \pi \leftarrow \mathcal{A}(\mathsf{crs});$
if $\mathsf{V}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \wedge \neg(\exists \mathbb{w}.\mathbf{R}(\mathbb{x}, \mathbb{w}) = 1)$ then return $1$; else return $0$; fi ;
$\mathsf{K}_{\mathsf{crs}}(\mathsf{p})$
$\mathbf{N}_u \leftarrow_\$ \mathbb{Z}_p^{N_1 \times (m+1)}; ([\boldsymbol{g}_u]_1, \mathsf{td}_u) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\mathbf{N}_u]_1);$
$\boxed{\boxed{\text{Choose } [\mathbf{N}_u]_1 \text{ as in Eq. (6)}; ([\boldsymbol{g}_u]_1, \mathsf{td}_u) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\mathbf{N}_u]_1);}}$
$\mathbf{N}_v \leftarrow_\$ \mathbb{Z}_p^{N_2 \times (m+1)}; ([\boldsymbol{g}_v]_2, \mathsf{td}_v) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\mathbf{N}_v]_2);$
$\boxed{\text{Choose } [\mathbf{N}_v]_2 \text{ as in Eq. (7)}; ([\boldsymbol{g}_v]_2, \mathsf{td}_v) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\mathbf{N}_v]_2);}$
$x \leftarrow_\$ \mathbb{Z}_p^*;$ Create $\mathsf{BLS.lpar} \leftarrow [\boldsymbol{H}]_*$ as in Eq. (4);
$(\mathsf{BLS.crs}, \mathsf{BLS.td}) \leftarrow \mathsf{BLS.K}_{\mathsf{crs}}(\mathsf{p}, \mathsf{BLS.lpar});$
$\mathsf{crs} \leftarrow ([\boldsymbol{g}_u, (x^i)_{i=0}^n]_1, [\boldsymbol{g}_v, (x^i)_{i=0}^n]_2, \mathsf{BLS.lpar}, \mathsf{BLS.crs}); \mathsf{td} \leftarrow \mathsf{BLS.td};$
return $(\mathsf{crs}, \mathsf{td})$;
```

Fig. 4. Games in the proof of Theorem 3. $\boxed{\text{Dotted boxed}}$ part is only in Game_2, $\boxed{\text{dashed boxed}}$ part is only in Game_3, and $\boxed{\text{boxed}}$ part is only in Game_4. The parts with several boxes are present in all corresponding games.

Proof. If $\mathcal{A}$'s success in the two games differs then one can distinguish between two different $[\boldsymbol{g}_u]_1$'s: the distinguisher $\mathcal{B}_1$ obtains $\mathbb{x}$ from $\mathcal{A}(\mathsf{p})$, creates crs from the correct Game_2 or Game_3 distribution but embedding $[\boldsymbol{g}_u]_1$ to it, and then obtains π from $\mathcal{A}$. If $\mathcal{A}$ succeeds, then $\mathcal{B}_1$ guesses that $[\boldsymbol{g}_u]_1$ was modified. Clearly, $\mathcal{B}_1$ has at least the same advantage as $\mathcal{A}$. □

The analysis of Lemma 2 is similar.

Lemma 2. *There exist a PPT adversary $\mathcal{B}'_1$, such that* $|\Pr[\mathsf{Game}_4(\mathcal{A}) = 1] - \Pr[\mathsf{Game}_3(\mathcal{A}) = 1]| \leq \mathsf{Adv}^{\mathrm{fsh}}_{\mathsf{FSSB}_2, m+1, N_2, \mathcal{B}'_1}(\lambda)$.

Finally, we bound the advantage of $\mathcal{A}$ in Game_4.

Lemma 3. *Assume* FSSB_1 *is locally* $[\cdot]_1$*-extractable and* FSSB_2 *is locally* $[\cdot]_2$*-extractable. There exist PPT adversaries $\mathcal{B}_2$ and $\mathcal{B}_3$, such that*

$$|\Pr[\mathsf{Game}_4(\mathcal{A}) = 1] \leq \mathsf{Adv}^{\sigma_{\mathbb{x}}-\mathrm{strsound}}_{\mathsf{Pgen}, \mathsf{BLS}, \mathcal{B}_2}(\lambda) + n \cdot \mathsf{Adv}^{\text{qa-linres}}_{\mathsf{Pgen}, n, \mathcal{B}_3}(\lambda) \enspace .$$

Proof. Let $\mathcal{A}$ be a non-adaptive soundness adversary in Game_4. Let ev be the event that Eq. (5) does not hold, that is, there does not exist $\mathsf{BLS}.\mathbb{w} = (\mathbb{z}, r_a, r_b, r_c, r_{\mathsf{sph}}, r_u, r_v)$, such that Eq. (5) (and the paragraph after it) holds. Clearly,

$$\Pr[\mathsf{Game}_4(\mathcal{A}) = 1] \leq \Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \mathsf{ev}] + \Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \overline{\mathsf{ev}}] \enspace .$$

First Reduction. We bound the first addend $\Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \mathsf{ev}]$ by the advantage of an adversary $\mathcal{B}_2$ against the quasi-adaptive σ_x-strong soundness

```
$\mathcal{B}_2(\mathsf{p}, \mathsf{BLS.lpar} = [\boldsymbol{H}]_*, \sigma_x(\mathsf{BLS.ltrap}) = x, \mathsf{BLS.crs})$    // QA $\sigma_x$-strong soundness
$\mathbb{x} \leftarrow \mathcal{A}(\mathsf{p})$;    // $\mathbb{x} = (\mathbb{z}_1, \dots, \mathbb{z}_{m_0})$
$\boldsymbol{\mathcal{N}}_u \leftarrow_\$ \mathbb{Z}_p^{N_1 \times (m+1)}; \boldsymbol{\mathcal{N}}_v \leftarrow_\$ \mathbb{Z}_p^{N_2 \times (m+1)}$;    // Generate $\mathsf{crs}$
$([\boldsymbol{g}_u]_1, \mathsf{td}_u = (\mathsf{ek}_u, \mathsf{tk}_u)) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\boldsymbol{\mathcal{N}}_u]_1)$;
$([\boldsymbol{g}_v]_2, \mathsf{td}_v = (\mathsf{ek}_v, \mathsf{tk}_v)) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\boldsymbol{\mathcal{N}}_v]_2)$;
Create $\mathsf{BLS.lpar} \leftarrow [\boldsymbol{H}]_*$ as in Eq. (4);
$\mathsf{FANA.crs} \leftarrow ([\boldsymbol{g}_u, (x^i)_{i=0}^n]_1, [\boldsymbol{g}_v, (x^i)_{i=0}^n]_2, \mathsf{BLS.lpar}, \mathsf{BLS.crs})$;
$\pi \leftarrow \mathcal{A}(\mathsf{FANA.crs})$;    // $\pi = ([\mathsf{a}, \mathsf{c}, \tilde{\boldsymbol{c}}_u, h(x)]_1, [\mathsf{b}, \tilde{\boldsymbol{c}}_v]_2, \mathsf{BLS}.\pi)$
$\mathsf{BLS}.\mathbb{x} \leftarrow ([\mathbb{x} /\!/ \mathsf{a} /\!/ \mathsf{c} /\!/ \tilde{\boldsymbol{c}}_u]_1 /\!/ [\mathsf{b} /\!/ \tilde{\boldsymbol{c}}_v]_2)$;
return $(\mathsf{BLS}.\mathbb{x}, \mathsf{BLS}.\pi)$;

$\mathcal{B}_3(\mathsf{p}, ([x^i]_1, [x^i]_2)_{i=0}^n, [y]_1, [y]_2)$    // QA-LINRES
$\mathbb{x} \leftarrow \mathcal{A}(\mathsf{p})$;    // $\mathbb{x} = (\mathbb{z}_1, \dots, \mathbb{z}_{m_0})$
$\boldsymbol{\mathcal{N}}_u \leftarrow_\$ \mathbb{Z}_p^{N_1 \times (m+1)}; \boldsymbol{\mathcal{N}}_v \leftarrow_\$ \mathbb{Z}_p^{N_2 \times (m+1)}$;    // Generate $\mathsf{crs}$
$([\boldsymbol{g}_u]_1, \mathsf{td}_u = (\mathsf{ek}_u, \mathsf{tk}_u)) \leftarrow \mathsf{FSSB}_1.\mathsf{KC}(\mathsf{p}, m+2, N_1, [\boldsymbol{\mathcal{N}}_u]_1)$;
$([\boldsymbol{g}_v]_2, \mathsf{td}_v = (\mathsf{ek}_v, \mathsf{tk}_v)) \leftarrow \mathsf{FSSB}_2.\mathsf{KC}(\mathsf{p}, m+1, N_2, [\boldsymbol{\mathcal{N}}_v]_2)$;
Create $\mathsf{BLS.lpar} = [\boldsymbol{H}]_*$ as in Eq. (4);
$\mathsf{BLS.crs} \leftarrow \mathsf{BLS.K}_{\mathsf{crs}}(\mathsf{p}, \mathsf{BLS.lpar})$;
$\mathsf{FANA.crs} \leftarrow ([\boldsymbol{g}_u, (x^i)_{i=0}^n]_1, [\boldsymbol{g}_v, (x^i)_{i=0}^n]_2, \mathsf{BLS.lpar}, \mathsf{BLS.crs})$;
$\pi \leftarrow \mathcal{A}(\mathsf{FANA.crs})$;    // $\pi = ([\mathsf{a}, \mathsf{c}, \tilde{\boldsymbol{c}}_u, h(x)]_1, [\mathsf{b}, \tilde{\boldsymbol{c}}_v]_2, \mathsf{BLS}.\pi)$
$[\hat{\alpha}_u, \hat{\beta}_u, \hat{\alpha}_w, \hat{\beta}_w]_1^\top \leftarrow \mathsf{FSSB}_1.\mathsf{LExt}(\mathsf{ek}_u; [\tilde{\boldsymbol{c}}_u]_1)$;
$[\hat{\alpha}_v, \hat{\beta}_v]_2^\top \leftarrow \mathsf{FSSB}_2.\mathsf{LExt}(\mathsf{ek}_v; [\tilde{\boldsymbol{c}}_v]_2)$;
$[\mathsf{a}']_1 \leftarrow [\mathsf{a}]_1 + \sum_{j=1}^{m_0} \mathbb{z}_j [u_j(x)]_1$;
if $[\hat{\beta}_u]_1 \bullet [\hat{\beta}_v]_2 = [\hat{\beta}_w]_1 \bullet [y]_2$ then return $\perp$;
else return $(\mathsf{J}, [\mathsf{a}', \hat{\alpha}_u, \hat{\beta}_u, \mathsf{c}, \hat{\alpha}_w, \hat{\beta}_w, h(x)]_1, [\mathsf{b}, \hat{\alpha}_v, \hat{\beta}_v]_2)$; fi ;
```

Fig. 5. The quasi-adaptive σ_x-strong soundness adversary $\mathcal{B}_2$ and the n-QA-LINRES adversary $\mathcal{B}_3$ in Lemma 3. $\mathcal{A}$ is a non-adaptive soundness adversary in Game_4.

(see page 15 for the definition) of BLS. In Fig. 5, we depict $\mathcal{B}_2$. $\mathcal{B}_2$ receives its input, sampled according to the distribution specified by Game_4. (The necessity to have $\sigma_x(\mathsf{ltrap}) = x$ as part of the input is precisely why BLS needs to be quasi-adaptively σ_x-*strongly* sound.) Given $\sigma_x(\mathsf{ltrap}) = x$, $\mathcal{B}_2$ constructs the rest of $\mathsf{FANA.crs}$. Finally, $\mathcal{B}_2$ uses the output of $\mathcal{A}$ to break the quasi-adaptively σ_x-strong soundness of BLS. Thus, $\Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \mathsf{ev}] \leq \mathsf{Adv}^{\sigma_x-\mathrm{strsound}}_{\mathsf{Pgen},\mathsf{BLS},\mathcal{B}_2}(\lambda)$.

Notably, quasi-adaptive σ_x-strong soundness of BLS suffices since $\mathsf{BLS.lpar}$ is a part of $\mathsf{FANA.crs}$ and thus honestly generated; moreover, σ_x is efficient.

Second Reduction. Assume $\mathsf{ev} = \mathsf{false}$. To bound the second addend $\Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \overline{\mathsf{ev}}]$, we construct an adversary $\mathcal{B}_3$ (see Fig. 5) against the n-QA-LINRES assumption. $\mathcal{B}_3$ queries $\mathcal{A}$ to obtain $\mathbb{x}$. After that, $\mathcal{B}_3$ uses its input to create $\mathsf{FANA.crs}$ according to the CRS distribution specified by the game (Game_3 or Game_4). $\mathcal{B}_3$ sends $\mathsf{FANA.crs}$ to $\mathcal{A}$, who outputs π. $\mathcal{B}_3$ uses the local extractability of FSSB_1 and FSSB_2 to extract certain values and then finishes as in Fig. 5, aborting when $\hat{\beta}_u \hat{\beta}_v \neq \hat{\beta}_w y$.

Let us explain why $\mathcal{B}_3$ succeeds with probability at least $1/n$. First, since FSSB_1 is locally $[\cdot]_1$-extractable and FSSB_2 is locally $[\cdot]_2$-extractable, $\mathcal{B}_3$ can extract $[\hat{\alpha}_u, \hat{\beta}_u, \hat{\alpha}_w, \hat{\beta}_w]_1 := [\boldsymbol{\mathfrak{N}}_{\boldsymbol{u}}(\mathbb{z} // r_a // r_c)]_1 \leftarrow \mathsf{FSSB}_1.\mathsf{LExt}(\mathsf{ek}_u; [\tilde{\boldsymbol{c}}_u]_1)$ and $[\hat{\alpha}_v, \hat{\beta}_v]_2 := [\boldsymbol{\mathfrak{N}}_{\boldsymbol{v}}(\begin{smallmatrix} \mathbb{z} \\ r_b \end{smallmatrix})]_2 \leftarrow \mathsf{FSSB}_2.\mathsf{LExt}(\mathsf{ek}_v; [\tilde{\boldsymbol{c}}_v]_2)$. ($\mathsf{LExt}$ is defined as in Fig. 1.)

We will next show that if that $\mathcal{B}_3$ does not abort, then it succeeds in breaking QA-LINRES. That is, the following conditions lifted from Definition 2 hold in relation to the values output by $\mathcal{B}_3$:

(a) $\hat{\beta}_u \hat{\beta}_v \neq \hat{\beta}_w y$,

(b) $\mathsf{a}' = \hat{\alpha}_u \cdot (x - \omega^{\mathsf{J}-1}) + \hat{\beta}_u / y$, where a' is as in Fig. 5 description of $\mathcal{B}_3$),

(c) $\mathsf{b} = \hat{\alpha}_v \cdot (x - \omega^{\mathsf{J}-1}) + \hat{\beta}_v / y$,

(d) $\mathsf{c} = \hat{\alpha}_w \cdot (x - \omega^{\mathsf{J}-1}) + \hat{\beta}_w / y$,

(e) $\mathsf{a}'\mathsf{b} - \mathsf{c} = h(x)Z(x)$.

Trivially, if $\mathcal{B}_3$ does not abort, then Item a holds. Since the FANA verifier accepts, $[\mathsf{a}']_1 \bullet [\mathsf{b}]_2 - [\mathsf{c}]_1 \bullet [1]_2 = [h(x)]_1 \bullet [Z(x)]_2$. Thus, Item e holds.

Next, since $\mathsf{ev} = \mathsf{false}$, there exists at least one $\mathsf{BLS}.\mathbb{w} = (\mathbb{z} = (\begin{smallmatrix} \mathbb{x} \\ \mathbb{w} \end{smallmatrix}), r_a, r_b, r_c, r_{\mathsf{sph}}, r_u, r_v)$, such that Eq. (5) holds. Fix *any* such $\mathsf{BLS}.\mathbb{w}$ (it does not have to be known to the reduction or even the one used by the adversary). Due to Eqs. (6) and (7), $\hat{\alpha}_u = \sum_{j=1}^{m} \mathbb{z}_j \delta_{uj}(x) + r_a \delta_Z(x)$ and $\hat{\beta}_u = (\sum_{j=1}^{m} \mathbb{z}_j U_{\mathsf{J}j}) y = \boldsymbol{U}_{\mathsf{J}} \mathbb{z} y$. The quotient of $u_j(X)/(X - \omega^{\mathsf{J}-1})$ is $\delta_{uj}(X)$. Since $u_j(X) = \sum_{i=1}^{n} U_{ij} \ell_i(X)$, the remainder of $u_j(X)/(X - \omega^{\mathsf{J}-1})$ is $U_{\mathsf{J}j}$. Clearly, $Z(X) = \delta_Z(X)(X - \omega^{\mathsf{J}-1})$. Since Eq. (5) holds, $\mathsf{a}' = \sum_{j=1}^{m} \mathbb{z}_j u_j(x) + r_a Z(x) = \sum_{j=1}^{m} \mathbb{z}_j (\delta_{uj}(x)(x - \omega^{\mathsf{J}-1}) + U_{\mathsf{J}j}) + r_a \delta_Z(x)(x - \omega^{\mathsf{J}-1}) = \hat{\alpha}_u \cdot (x - \omega^{\mathsf{J}-1}) + \hat{\beta}_u / y$. Thus, Item b holds. Similarly, Items c and d hold. Hence, if $\mathcal{B}_3$ does not abort, then all five conditions hold.

Finally, we need to argue that $\mathcal{B}_3$ does not abort with a probability of at least $1/n$. Since $\mathsf{ev} = \mathsf{false}$, we have that $\mathsf{BLS}.\mathbb{w}$ starts with $\mathbb{w}$. Thus, according to Eq. (1), for $\mathcal{A}$ to be successful, there must exist an i such that $(\boldsymbol{U}\mathbb{z})_i (\boldsymbol{V}\mathbb{z})_i \neq (\boldsymbol{W}\mathbb{z})_i$. Since J is chosen uniformly at random and the non-adaptive soundness adversary $\mathcal{A}$ chooses the input before seeing crs, with probability $\geq 1/n$, the Jth constraint is not satisfied. Thus, with probability $\geq 1/n$, $A(X)B(X) - C(X)$ does not divide by $X - \omega^{\mathsf{J}-1}$, where A, B, C are defined as always. Then, $\beta := A(X)B(X) - C(X) \mod (X - \omega^{\mathsf{J}-1})$ is non-zero. However, $\beta = (\boldsymbol{U}_{\mathsf{J}} \mathbb{z})(\boldsymbol{V}_{\mathsf{J}} \mathbb{z}) - \boldsymbol{W}_{\mathsf{J}} \mathbb{z} = \hat{\beta}_u \hat{\beta}_v / y^2 - \hat{\beta}_w / y$ and thus $\mathcal{B}_3$ does not abort with probability $\geq 1/n$. Thus, Item a holds with probability $\geq 1/n$.

Since (1) if $\mathcal{B}_3$ does not abort, then all five conditions hold, and (2) $\mathcal{B}_3$ does not abort with probability $\geq 1/n$, $\Pr[\mathsf{Game}_4(\mathcal{A}) = 1 | \mathsf{ev}] \leq n \cdot \mathsf{Adv}^{\text{qa-linres}}_{\mathsf{Pgen}, n, \mathcal{B}_3}(\lambda)$. □

Combining the lemmas proves the theorem. □

5.3 Adaptive Knowledge-Soundness of FANA

Theorem 4. *Assume the setting of Theorem 3. If* FANA *is non-adaptively sound and* BLS *is adaptively knowledge-sound, then* FANA *is non-adaptively knowledge-sound.*

Acknowledgment. We thank Prastudy Fauzi, Zaira Pindado, Carla Ràfols, Janno Siim, and anonymous reviewers for helpful comments.

References

1. Abdolmaleki, B., Baghery, K., Lipmaa, H., Zajac, M.: A subversion-resistant SNARK. In: ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 3–33
2. Abdolmaleki, B., Lipmaa, H., Siim, J., Zajac, M.: On QA-NIZK in the BPK model. In: PKC 2020, Part I. LNCS, vol. 12110, pp. 590–620
3. Abdolmaleki, B., Lipmaa, H., Siim, J., Zajac, M.: On Subversion-Resistant SNARKs. J. Cryptology **34**(3) (2021) pp. 1–42
4. Abe, M., Fehr, S.: Perfect NIZK with adaptive soundness. In: TCC 2007. LNCS, vol. 4392, pp. 118–136
5. Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: Security in the face of parameter subversion. In: ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 777–804
6. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get shorty via group signatures without encryption. In: SCN 10. LNCS, vol. 6280, pp. 381–398
7. Chung, K.M., Lin, H., Mahmoody, M., Pass, R.: On the power of nonuniformity in proofs of security. In: ITCS 2013, pp. 389–400
8. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: CRYPTO'91. LNCS, vol. 576, pp. 445–456
9. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square span programs with applications to succinct NIZK arguments. In: ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 532–550
10. Daza, V., González, A., Pindado, Z., Ràfols, C., Silva, J.: Shorter quadratic QA-NIZK proofs. In: PKC 2019, Part I. LNCS, vol. 11442, pp. 314–343
11. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147
12. Fauzi, P., Lipmaa, H., Pindado, Z., Siim, J.: Somewhere Statistically Binding Commitment Schemes with Applications. In: FC 2021. LNCS, vol. ?, pp. ?–?
13. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: 31st FOCS, pp. 308–317
14. Fuchsbauer, G.: Subversion-zero-knowledge SNARKs. In: PKC 2018, Part I. LNCS, vol. 10769, pp. 315–347
15. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62
16. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645
17. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: 43rd ACM STOC, pp. 99–108
18. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. Journal of Cryptology **7**(1) (1994) pp. 1–32
19. González, A., Hevia, A., Ràfols, C.: QA-NIZK arguments in asymmetric groups: New tools and new constructions. In: ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 605–629
20. González, A., Ràfols, C.: New techniques for non-interactive shuffle and range arguments. In: ACNS 16. LNCS, vol. 9696, pp. 427–444
21. González, A., Ràfols, C.: Shorter pairing-based arguments under standard assumptions. In: ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 728–757

22. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340
23. Groth, J.: On the size of pairing-based non-interactive arguments. In: EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326
24. Groth, J., Maller, M.: Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In: CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 581–612
25. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: CRYPTO 2006. LNCS, vol. 4117, pp. 97–111
26. Jager, T., Rupp, A.: The semi-generic group model and applications to pairing-based cryptography. In: ASIACRYPT 2010. LNCS, vol. 6477, pp. 539–556
27. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20
28. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. Cryptology ePrint Archive, Report 2013/109 (2013) https://eprint.iacr.org/2013/109.
29. Kalai, Y.T., Paneth, O., Yang, L.: How to delegate computations publicly. In: 51st ACM STOC, pp. 1115–1124
30. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128
31. Lipmaa, H.: Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In: TCC 2012. LNCS, vol. 7194, pp. 169–189
32. Lipmaa, H.: Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In: ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 41–60
33. Lipmaa, H.: Prover-efficient commit-and-prove zero-knowledge SNARKs. In: AFRICACRYPT 16. LNCS, vol. 9646, pp. 185–206
34. Lipmaa, H.: Simulation-Extractable ZK-SNARKs Revisited. Technical Report 2019/612, IACR (2019) https://ia.cr/2019/612, updated on 8 Feb 2020.
35. Lipmaa, H., Pavlyk, K.: Gentry-Wichs Is Tight: A Falsifiable Non-Adaptively Sound SNARG. Technical report, IACR (2021)
36. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: CRYPTO 2003. LNCS, vol. 2729, pp. 96–109
37. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy, pp. 238–252
38. Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: TCC 2013. LNCS, vol. 7785, pp. 334–354
39. Stachowiak, G.: Proofs of knowledge with several challenge values. Cryptology ePrint Archive, Report 2008/181 (2008) https://eprint.iacr.org/2008/181.