Subversion-Resistant Quasi-Adaptive NIZK and Applications to Modular zk-SNARKs*

Behzad Abdolmaleki[1] and Daniel Slamanig[2]

[1] Max Planck Institute for Security and Privacy, Bochum, Germany
behzad.abdolmaleki@csp.mpg.de
[2] AIT Austrian Institute of Technology, Vienna, Austria
daniel.slamanig@ait.ac.at

Abstract. Quasi-adaptive non-interactive zero-knowledge (QA-NIZK) arguments are NIZK arguments where the common reference string (CRS) is allowed to depend on the language and they can be very efficient for specific languages. Thus, they are for instance used within the modular LegoSNARK toolbox by Campanelli *et al.* (ACM CCS'19) as succinct NIZKs (aka zkSNARKs) for linear subspace languages. Such modular frameworks are interesting, as they provide gadgets for a flexible design of privacy-preserving blockchain applications. Recently, there has been an increasing interest to reduce the trust required in the generator of the CRS. One important line of work in this direction is subversion zero-knowledge by Bellare *et al.* (ASIACRYPT'16), where the zero-knowledge property even holds when the CRS is generated maliciously.
In this paper, we firstly analyze the security of the most efficient QA-NIZK constructions of Kiltz and Wee (EUROCRYPT'15) and the asymmetric QA-NIZKs by González *et al.* (ASIACRYPT'15) when the CRS is subverted and propose subversion versions of them. Secondly, for the first time, we construct unbounded (strong) true-simulation extractable (tSE) variants of them. Thirdly, we show how to integrate our subversion QA-NIZKs into the LegoSNARK toolbox, which so far does not consider subversion resistance. Our results together with existing results on (SE) subversion zk-SNARKS represent an important step towards a subversion variant of the LegoSNARK toolbox.

1 Introduction

Zero-knowledge (ZK) proofs introduced by Goldwasser, Micali and Rackoff [GMR89] are cryptographic protocols between two parties called the prover and the verifier with the purpose that the prover can convince the verifier of the validity of a statement in any language in NP without revealing additional information. Besides this zero-knowledge property, such a system needs to provide soundness, i.e., it must be infeasible for the prover to provide proofs for false statements. While ZK proofs, in general, may require many rounds of interaction, an interesting variant is non-interactive zero-knowledge (NIZK) proofs. They require only a single round, i.e., the prover

* This is the full version of a paper which appears in the proceedings of the 20th International Conference on Cryptology and Network Security - CANS 2021, LNCS, Springer.

outputs a proof, and this proof can then be verified by anybody. A long line of research [Kil92,GOS06,GS08,Gro10,Lip12,GGPR13,Gro16] has led to efficient pairing-based succinct NIZKs called zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs), which are NIZK arguments with $i)$ a stronger notion of soundness called knowledge soundness and, more importantly, $ii)$ in which proofs, as well as the computation of the verifier, are succinct, i.e., ideally a small constant amount of space and computation respectively. Due to these latter properties, zk-SNARKs are a suitable tool to preserve privacy within cryptocurrencies and distributed ledger technologies, most notably used within Zcash [SCG+14] and Ethereum [Buc17], and they increasingly attract interest outside of academia.[3,4] In this paper, we are interested in quasi-adaptive NIZK (QA-NIZK) arguments [JR13a]. These are NIZKs in which the common reference string (CRS) depends on a language parameter and they have many applications and have been intensively studied [JR13a, LPJY14, JR14, ABP15, KW15, LPJY15, GHR15, GHKW16, AJOR18, AJO+19, DGP+19, RS20, BGPR20].

For practical applications of (QA-)NIZKs and zk-SNARKs, an important question is the generation of the CRS. While in theory it is simply assumed that some mutually trusted party will perform the CRS generation, in many real world settings (such as fully decentralized systems) there typically does not exist such a trusted party. Recently, there has been an increasing interest to reduce trust in the generator of the CRS. One of these lines of work is subversion zero-knowledge initiated by Bellare *et al.* in [BFS16], where the zero-knowledge property even holds when the CRS is generated maliciously, i.e., the CRS generator is subverted. Following this initial work, Abdolmaleki *et al.* [ABLZ17, ALSZ21] as well as Fuchsbauer [Fuc18] investigated subversion zk-SNARKs. More recently, Abdolmaleki *et al.* (ALSZ) in [ALSZ20] initiated the study of subversion zero-knowledge QA-NIZK (Sub-ZK QA-NIZK for short). While the latter is an important step, it leaves a number of open problems such as weakening the requires assumptions, stronger soundness guarantees and demonstrating impact for real-world applications.

Our Contribution. Our results can be summarized as follows.

Sub-ZK QA-NIZKs. We investigate the most efficient QA-NIZK constructions of Kiltz and Wee (KW) [KW15] and the asymmetric QA-NIZKs by González *et al.* (GHR) [GHR15] in a subverted setup. We show that for KW we can construct Sub-ZK QA-NIZK arguments for the most efficient their argument Π'_{as} (which requires a witness samplable distribution [JR13a]) by extending the CRS suitably. Thereby, compared to the recent Sub-ZK QA-NIZK based upon KW by ALSZ, we consider a variant where the CRS is subverted, but the language parameter is chosen honestly. We note that latter does not represent a problem for practical applications, as these parameters can typically be obtained in a transparent way such that no trusted setup is needed, e.g., by deriving them using a suitable hash function modelled as a random oracle. In contrast to ALSZ, which relies on a new non-standard knowledge assumption for their subversion zero-knowledge property, our Sub-ZK QA-NIZK can be shown to have this

[3] ZKProof (https://zkproof.org/) being the most notable industry and academic initiative towards a common framework and standards has been founded in 2018.

[4] Zero-knowledge proofs are *on the rise*, cf. https://www.gartner.com/en/documents/3947373/hype-cycle-for-privacy-2019.

property under the Bilinear Diffie-Hellman Knowledge of Exponents (BDH-KE) assumption [ABLZ17, ALSZ21] (being a simple case of the PKE assumption [DFGK14] or viewed differently an asymmetric-pairing version of the KoE assumption [Dam92]). Moreover, we present a Sub-ZK QA-NIZK version of GHR by relying on the same BDH-KE assumption.

Simulation Extractability of Sub-ZK QA-NIZKs. We investigate the construction of Sub-ZK QA-NIZK that satisfies the stronger notions of knowledge soundness and in particular a weakened version of simulation extractability (SE) called true-simulation extractability (tSE) [Har11]. SE for QA-NIZK has to the best of our knowledge only been used in the independent concurrent work by Baghery *et al.* [BGPR20] in a non-subverted setting. We recall that a (QA-)NIZK is called unbounded SE if knowledge soundness holds even if the adversary is allowed to adaptively see an arbitrary number of simulated proofs (restricted to statements inside the language for tSE). The strong tSE notion of QA-NIZKs is important as, similarly to SE, it guarantees non-malleability of proofs thus prevents man-in-the-middle type of attacks, i.e., where an adversary takes a given proof and alters the proof or proven statement without having access to the full witness anyways. Our work is the first treatment of tSE Sub-ZK QA-NIZK and we present unbounded tSE Sub-ZK QA-NIZKs based on KW (also in the non-subversion setting).

Towards Subversion LegoSNARK. LegoSNARK [CFQ19] is a framework for Commit-and-Prove zk-SNARKs (CP-SNARKs) with the aim of constructing a "global" SNARK for some computation C via the linking of "smaller" specialized SNARKs for different subroutines that overall compose to C. The main idea is that by letting each subroutine of C be handled by a different proof system one can choose the one that maximizes a metric (e.g., efficiency) that is important for the concrete application. LegoSNARK uses a knowledge-sound version of the KW QA-NIZK (with succinct proofs) as the zk-SNARKs for linear subspace languages and in particular, they use a knowledge-sound version of the KW QA-NIZK Π'_{as}. We will show how to integrate subversion primitives into LegoSNARK. In particular, we show how to integrate our Sub-ZK QA-NIZKs instead of their non-subversion counterparts. Together with the results on subversion (SE) zk-SNARKs [ABLZ17, Fuc18, GM17a, Lip19, Bag19, ARS20], we thus make an important step towards a complete subversion (SE) variant of the LegoSNARK framework.[5]

2 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter. By $y \leftarrow \mathcal{A}(x;\omega)$ we denote the fact that $\mathcal{A}$, given an input x and random coins ω, outputs y. By $x \leftarrow\!\!\$\ \mathcal{D}$ we denote that x is sampled according to distribution $\mathcal{D}$ or uniformly randomly if $\mathcal{D}$ is a set. Let $\mathsf{RND}(\mathcal{A})$ denote the random tape of $\mathcal{A}$, and let $\omega \leftarrow\!\!\$\ \mathsf{RND}(\mathcal{A})$ denote the random choice of the random coins ω from $\mathsf{RND}(\mathcal{A})$. We denote by $\mathsf{negl}(\lambda)$ an arbitrary negligible function. We write $a \approx_\lambda b$ if $|a-b| \leq \mathsf{negl}(\lambda)$. For algorithms $\mathcal{A}$ and $\mathsf{Ext}_{\mathcal{A}}$, we write $(y||y') \leftarrow (\mathcal{A}||\mathsf{Ext}_{\mathcal{A}})(\cdot)$

[5] We note that there are some tasks, such as fitting existing subversion (SE) zk-SNARKs into the commit-prove framework remaining that need to be worked out in detail. However, we do not expect that one faces significant problems there.

as a shorthand for $y \leftarrow \mathcal{A}(\cdot)$ and $y' \leftarrow \mathsf{Ext}_{\mathcal{A}}(\cdot)$. Algorithm $\mathsf{Pgen}(1^\lambda)$ returns $\mathsf{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic groups of prime order p, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear map (pairing). We use the implicit bracket notation of [EHK+13], that is, we write $[a]_\iota$ to denote ag_ι where g_ι is a fixed generator of $\mathbb{G}_\iota$. We denote $\hat{e}([a]_1, [b]_2)$ as $[a]_1[b]_2$. Thus, $[a]_1[b]_2 = [ab]_T$. We denote $s[a]_\iota = [sa]_\iota$ for $s \in \mathbb{Z}_p$ and $S \cdot [a]_\iota = [Sa]_T$ for $S \in \mathbb{G}_{3-\iota}$ and $\iota \in \{1, 2\}$. We freely use the bracket notation together with matrix notation, e.g., if $\boldsymbol{XY} = \boldsymbol{Z}$ then $[\boldsymbol{X}]_1[\boldsymbol{Y}]_2 = [\boldsymbol{Z}]_T$. Furthermore in our figures, we will not explicitly provide return statements for P and Sim, but output all $\boldsymbol{\pi}$ elements.

Computational Assumptions. We require the following assumptions.

Definition 1 (BDH-KE Assumption [ABLZ17, ALSZ21]). *We say that* BDH-KE *holds relative to* K_0*, if for any PPT adversary* $\mathcal{A}$ *there exists a PPT extractor* $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}}$*, such that*

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow_\$ \mathsf{K}_0(1^\lambda); \omega_{\mathcal{A}} \leftarrow_\$ \mathsf{RND}(\mathcal{A}),\\ ([\alpha_1]_1, [\alpha_2]_2 || a) \leftarrow (\mathcal{A} || \mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}})(\mathsf{p}, \omega_{\mathcal{A}})\end{array} : [\alpha_1]_1[1]_2 = [1]_1[\alpha_2]_2 \wedge a \neq \alpha_1\right] \approx_\lambda 0\,.$$

Where $\mathsf{aux}_{\mathcal{R}}$ is the auxiliary information related to the relation generator of $\mathcal{R}$. Note that the BDH-KE assumption can be considered as a simple case of the PKE assumption of [DFGK14]. Also, BDH-KE can be seen as an asymmetric-pairing version of the original KoE assumption [Dam92].

In the following let $\mathcal{D}_k$ be a matrix distribution in $\mathbb{Z}_p^{(k+1)\times k}$.

Definition 2 ($\mathcal{D}_k$-Matrix Diffie-Hellman ($\mathcal{D}_k$-MDDH) Assumption [MRV16]). *The* $\mathcal{D}_k$-MDDH *assumption for* $\iota \in \{1, 2\}$ *holds relative to* K_0*, if for any PPT adversary* $\mathcal{A}$*,* $|\mathsf{Exp}_{\mathcal{A}}^{\mathsf{MDDH}}(\mathsf{p}) - 1/2| \approx_\lambda 0$*, where* $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{MDDH}}(\mathsf{p}) :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow_\$ \mathsf{K}_0(1^\lambda); \boldsymbol{A} \leftarrow_\$ \mathcal{D}_k; \boldsymbol{v} \leftarrow_\$ \mathbb{Z}_p^k;\\ \boldsymbol{u} \leftarrow_\$ \mathbb{Z}_p^{k+1}; b \leftarrow_\$ \{0, 1\};\\ b^* \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_\iota, [b \cdot \boldsymbol{Av} + (1-b) \cdot \boldsymbol{u}]_\iota)\end{array} : b = b^*\right].$$

Definition 3 ($\mathcal{D}_k$-KerMDH Assumption [MRV16]). *The* $\mathcal{D}_k$-KerMDH *assumption for* $\iota \in \{1, 2\}$ *holds relative to* K_0*, if for any PPT* $\mathcal{A}$*,*

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \boldsymbol{A} \leftarrow_\$ \mathcal{D}_k; [\boldsymbol{s}]_{3-\iota} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_\iota) : \boldsymbol{s} \neq \boldsymbol{0} \wedge \boldsymbol{A}^\top \boldsymbol{s} = \boldsymbol{0}_k\right] \approx_\lambda 0\,.$$

Note that as shown in [MRV16], if $\mathcal{D}_k$-MDDH holds then $\mathcal{D}_k$-KerMDH holds.

Definition 4 ($\mathcal{D}_k$-SKerMDH Assumption [GHR15]). *The* $\mathcal{D}_k$-SKerMDH *assumption holds relative to* K_0*, if for any PPT* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \boldsymbol{A} \leftarrow_\$ \mathcal{D}_k; ([\boldsymbol{s}_1]_1, [\boldsymbol{s}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1, [\boldsymbol{A}]_2) :\\ \boldsymbol{s}_1 - \boldsymbol{s}_2 \neq \boldsymbol{0} \wedge \boldsymbol{A}^\top(\boldsymbol{s}_1 - \boldsymbol{s}_2) = \boldsymbol{0}_k\end{array}\right] \approx_\lambda 0\,.$$

Let $\mathcal{D}_{\ell k}$ be a probability distribution over matrices in $\mathbb{Z}_p^{\ell \times k}$, where $\ell > k$. Next, we define five commonly used distributions (see [EHK+13] for references), where

$a, a_i, a_{ij} \leftarrow_\$ \mathbb{Z}_p^*$: $\mathcal{U}_k$ (uniform), $\mathcal{L}_k$ (linear), $\mathcal{IL}_k$ (incremental linear), $\mathcal{C}_k$ (cascade), $\mathcal{SC}_k$ (symmetric cascade):

$$\mathcal{U}_k\colon \boldsymbol{A} = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \dots & \dots & \dots \\ a_{k1} & \dots & a_{kk} \\ a_{k+1,1} & \dots & a_{k+1,k} \end{pmatrix}, \quad \mathcal{L}_k\colon \boldsymbol{A} = \begin{pmatrix} a_1 & 0 & \dots & 0 & 0 \\ 0 & a_2 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_k \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix},$$

$$\mathcal{IL}_k\colon \boldsymbol{A} = \begin{pmatrix} a & 0 & \dots & 0 & 0 \\ 0 & a+1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a+k-1 \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix}, \quad \mathcal{C}_k\colon \boldsymbol{A} = \begin{pmatrix} a_1 & 0 & \dots & 0 & 0 \\ 1 & a_2 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_k \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix},$$

$$\mathcal{SC}_k\colon \boldsymbol{A} = \begin{pmatrix} a & 0 & \dots & 0 & 0 \\ 1 & a & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

Assume that $\mathcal{D}_{\ell k}$ outputs matrices $\boldsymbol{A}$ where the upper $k \times k$ submatrix $\bar{\boldsymbol{A}}$ is always invertible, i.e., $\mathcal{D}_{\ell k}$ is robust [JR13a]. All the above distributions can be made robust with minimal changes. Denote the lower $(\ell - k) \times k$ submatrix of $\boldsymbol{A}$ as $\underline{\boldsymbol{A}}$ and denote $\mathcal{D}_k = \mathcal{D}_{k+1,k}$.

Quasi-Adaptive NIZK Arguments. We recall the definition of QA-NIZK arguments of Jutla and Roy [JR13a]. A QA-NIZK argument provides a proof for membership of words x with according witnesses $\boldsymbol{w}$ in a language $\mathcal{L}_\varrho$ defined by a relation $\mathcal{R}_\varrho$ which is parametrized by some parameter ϱ chosen from a distribution $\mathcal{D}_{\mathsf{p}}$. The distribution $\mathcal{D}_{\mathsf{p}}$ is witness samplable if there exist an efficient algorithm that samples $(\varrho, \mathtt{tc}_\varrho)$ so that the parameter ϱ is distributed according to $\mathcal{D}_{\mathsf{p}}$ and membership of the language parameter ϱ can be efficiently verified with $\mathtt{tc}_\varrho$. The CRS of QA-NIZKs depends on a language parameter ϱ and as mentioned in [JR13a], it has to be chosen from a correct distribution $\mathcal{D}_{\mathsf{p}}$.

A tuple of PPT algorithms $\Pi = (\mathsf{Pgen}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ is a QA-NIZK argument in the CRS model for a set of witness-relations $\mathcal{R}_{\mathsf{p}} = \{\mathcal{R}_\varrho\}_{\varrho \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})}$ with ϱ sampled from a distribution $\mathcal{D}_{\mathsf{p}}$ over associated parameter language $\mathcal{L}_{\mathsf{p}}$, if the following properties (i-iii) hold. Here, Pgen is the parameter and the CRS generation algorithm, more precisely, Pgen consists of two algorithms K_0 (generates the the parameter p) and K (generates the CRS), P is the prover, V is the verifier, and Sim is the simulator.

(i) **Completeness.** For any λ, and $(x, \boldsymbol{w}) \in \mathcal{R}_\varrho$,

$$\Pr\left[\begin{matrix} \mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow_\$ \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathtt{tc}) \leftarrow \mathsf{K}(\varrho); \pi \leftarrow \mathsf{P}(\varrho, \mathsf{crs}, x, \boldsymbol{w}) : \\ \mathsf{V}(\varrho, \mathsf{crs}, x, \pi) = 1 \end{matrix}\right] = 1\,.$$

(ii) **Statistical Zero-Knowledge.** For any computationally unbounded adversary $\mathcal{A}$, $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| \approx_\lambda 0$, where $\varepsilon_b^{zk} :=$

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow_\$ \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathtt{tc}) \leftarrow \mathsf{K}(\varrho); b \leftarrow_\$ \{0,1\} : \mathcal{A}^{\mathsf{O}_b(\cdot)}(\varrho, \mathsf{crs}) = 1\ \right].$$

The oracle $\mathsf{O}_0(x, \boldsymbol{w})$ returns $\perp$ (reject) if $(x, \boldsymbol{w}) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{P}(\varrho, \mathsf{crs}, x, \boldsymbol{w})$. Similarly, $\mathsf{O}_1(x, \boldsymbol{w})$ returns $\perp$ (reject) if $(x, \boldsymbol{w}) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{Sim}(\varrho, \mathsf{crs}, \mathtt{tc}, x)$.

$\mathsf{K}([\boldsymbol{M}]_1)$

- $\boldsymbol{A} \leftarrow\!\!\$\ \hat{\mathcal{D}}_k$; $\boldsymbol{K} \leftarrow\!\!\$\ \mathbb{Z}_p^{n\times\hat{k}}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\boldsymbol{A} \in \mathbb{Z}_p^{n\times k}$;
- $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in \mathbb{Z}_p^{m\times\hat{k}}$; $\mathsf{crs} \leftarrow ([\boldsymbol{A},\boldsymbol{C}]_2, [\boldsymbol{P}]_1)$; $\mathtt{tc} \leftarrow \boldsymbol{K}$;
- **return** $(\mathtt{tc}, \mathsf{crs})$.

$\mathsf{P}([\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, \boldsymbol{w})$:

- $[\boldsymbol{\pi}]_1 \leftarrow [\boldsymbol{P}]_1^\top \boldsymbol{w} \in \mathbb{G}_1^{\hat{k}}$;

$\mathsf{V}([\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$:

- **if** $[\boldsymbol{y}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{\pi}]_1^\top [\boldsymbol{A}]_2$ **return** 1;

$\mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{crs}, \mathtt{tc}, [\boldsymbol{y}]_1)$:

- $[\boldsymbol{\pi}]_1 \leftarrow \boldsymbol{K}^\top [\boldsymbol{y}]_1 \in \mathbb{G}_1^{\hat{k}}$.

Fig. 1. KW QA-NIZK Π_{as} ($\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$) and Π'_{as} ($\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ and $\hat{k} = k$).

(iii) **Adaptive Soundness.** For any PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow\!\!\$\ \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathtt{tc}) \leftarrow \mathsf{K}(\varrho); (x,\pi) \leftarrow \mathcal{A}(\varrho, \mathsf{crs}) : \\ \mathsf{V}(\varrho, \mathsf{crs}, x, \pi) = 1 \wedge \neg(\exists \boldsymbol{w} : (x, \boldsymbol{w}) \in \mathcal{R}_\varrho)\end{array}\right] \approx_\lambda 0 \ .$$

Additionally, we define a stronger soundness version called knowledge soundness.

Computational Knowledge Soundness. For any PPT $\mathcal{A}$ there exists a non-uniform polynomial time extractor $\mathsf{Ext}_{\mathcal{A}}$ such that,

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow\!\!\$\ \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathtt{tc}) \leftarrow \mathsf{K}(\varrho); \omega_{\mathcal{A}} \leftarrow \mathsf{RND}(\mathcal{A}); \\ ((x,\pi); \boldsymbol{w}) \leftarrow (\mathcal{A}||\mathsf{Ext}_{\mathcal{A}})(\omega_{\mathcal{A}}; \varrho, \mathsf{crs}) : \mathsf{V}(\varrho, \mathsf{crs}, x, \pi) = 1 \wedge (x, \boldsymbol{w}) \notin \mathcal{R}_\varrho)\end{array}\right] \approx_\lambda 0 \ .$$

QA-NIZK Argument for Linear Spaces. Now we recall the two constructions of QA-NIZK arguments of membership in linear spaces given by Kiltz and Wee (KW) [KW15] for the language

$$\mathcal{L}_{[\boldsymbol{M}]_1} = \left\{ [\boldsymbol{y}]_1 \in \mathbb{G}_1^n : \exists \boldsymbol{w} \in \mathbb{Z}_p^m \text{ s.t. } \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w} \right\} \ .$$

The corresponding relation is defined as $\mathcal{R}_{[\boldsymbol{M}]_1} = \{([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathbb{G}_1^n \times \mathbb{Z}_p^m : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}\}$. This language is useful in many applications (cf. [JR13a] and follow up work). We recall the full construction of the Kiltz-Wee QA-NIZK arguments for linear subspaces in the CRS model in Fig. 1. Let $\hat{\mathcal{D}}_k$ and $\bar{\mathcal{D}}_k$ be matrix distributions in $\mathbb{Z}_p^{\hat{k}\times k}$ and $\mathbb{Z}_p^{k\times k}$ respectively. We denote $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ if $\hat{k} = k$, and $\hat{\mathcal{D}}_k = \mathcal{D}_k$ if $\hat{k} = k+1$.

Theorem 1 (Theorem 1 of [KW15]). *If $\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$, Fig. 1 describes a QA-NIZK argument Π_{as} with perfect completeness, computational adaptive soundness based on the $\mathcal{D}_k$-*KerMDH *assumption, perfect zero-knowledge, and proof size $k+1$.*

Theorem 2 (Theorem 2 of [KW15]). *If $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$, $\hat{k} = k$, and $\mathcal{D}_{\mathsf{p}}$ is a witness samplable distribution, Fig. 1 describes a QA-NIZK argument Π'_{as} with perfect completeness, computational adaptive soundness based on the $\mathcal{D}_k$-*KerMDH *assumption, perfect zero-knowledge, and proof size k.*

Asymmetric QA-NIZK for Concatenation Languages. We recall the constructions of asymmetric QA-NIZK arguments of membership in different subspace concatenations

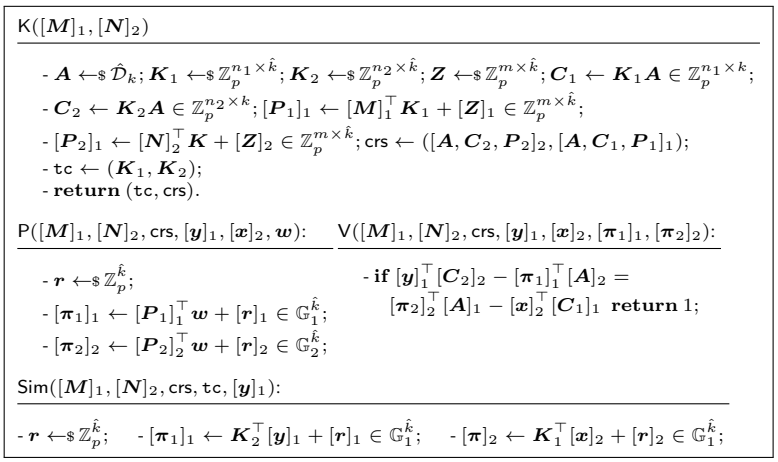

Fig. 2. Asymmetric QA-NIZK $\mathit{\Pi}_{\mathsf{asy}}$ ($\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$) and $\mathit{\Pi}'_{\mathsf{asy}}$ ($\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ and $\hat{k} = k$).

of $\mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2}$ given by Gonzalez *et al.* [GHR15] for the language

$$\mathcal{L}_{[\boldsymbol{M}]_1,[\boldsymbol{N}]_2} = \left\{ ([\boldsymbol{y}]_1, [\boldsymbol{x}]_2) \in \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} : \exists \boldsymbol{w} \in \mathbb{Z}_p^m \text{ s.t. } \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}, \boldsymbol{x} = \boldsymbol{N}\boldsymbol{w} \right\} .$$

This language is also known as the concatenation language, since one can define $\boldsymbol{R}$ as a concatenation of language parameters $[\boldsymbol{M}]_1$ and $[\boldsymbol{N}]_2$ so that $\boldsymbol{R} = \begin{pmatrix} [\boldsymbol{M}]_1 \\ [\boldsymbol{N}]_2 \end{pmatrix}$. In other words $([\boldsymbol{y}]_1, [\boldsymbol{x}]_2) \in \mathcal{L}_{[\boldsymbol{M}]_1,[\boldsymbol{N}]_2}$ iff $\begin{pmatrix} [\boldsymbol{y}]_1 \\ [\boldsymbol{x}]_2 \end{pmatrix}$ is in the span of $\boldsymbol{R}$. We recall the full construction of asymmetric QA-NIZK arguments in the CRS model in Fig. 2.

Notice that the QA-NIZK in Fig. 2 for $\mathcal{L}_{[\boldsymbol{M}]_1,[\boldsymbol{N}]_2}$ is a generalization of $\mathit{\Pi}_{\mathsf{as}}$ of [KW15] in two groups when we set $\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$ (denoted as $\mathit{\Pi}_{\mathsf{asy}}$). Also it is a generalization of $\mathit{\Pi}'_{\mathsf{as}}$ of [KW15] in two groups when we set $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ and $\hat{k} = k$ (denoted as $\mathit{\Pi}'_{\mathsf{asy}}$).

Theorem 3 (Theorem 3 of [GHR15]). *If $\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$, the QA-NIZK proof system in Fig. 2 is perfect complete, computational adaptive soundness based on the $\mathcal{D}_k$-SKerMDH assumption, perfect zero-knowledge.*

Theorem 4 (Theorem 4 of [GHR15]). *If $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$, $\hat{k} = k$ and $\mathcal{D}_{\mathsf{p}}$ is a witness samplable distribution, Fig. 2 describes a QA-NIZK proof system with perfect completeness, computational adaptive soundness based on the $\mathcal{D}_k$-KerMDH assumption, perfect zero-knowledge.*

3 QA-NIZK Arguments in the Subversion Setting

In this section, we investigate QA-NIZK arguments when the CRS is subverted and propose corresponding Sub-ZK QA-NIZK arguments. First we discuss subversion security and then our focus will be on the fundamental and the most efficient QA-NIZK

construction Π'_{as} in [KW15] (cf. Section 2)and the asymmetric QA-NIZK constructions Π_{asy} and Π'_{asy} in [GHR15] (cf. Section 2) for linear subspaces languages

3.1 Security Definitions for Subversion QA-NIZK Arguments

The notion of subversion security for QA-NIZKs in the CRS model was first noted by Jutla and Roy in the full version of [JR13a] (cf. [JR13b]). They have shown that one can obtain both soundness and zero-knowledge (under falsifiable assumptions) when the language parameter ϱ is subverted but the CRS is *generated honestly*. They showed that such a setting can cover a large family of subspace languages. Later Abdolmaleki *et al.* [ALSZ20] (ALSZ) defined the security of QA-NIZKs in the bare public-key (BPK) model, when both ϱ and the CRS are subverted. More precisely, they obtain a version of the Kiltz-Wee QA-NIZK [KW15] when both ϱ and CRS are chosen maliciously, but under a new non-falsifiable KWKE knowledge assumption. ALSZ also obtain (knowledge) soundness when only ϱ is chosen maliciously under a new (non-falsifiable) interactive assumptions $\mathrm{KerMDH}^{\mathrm{dl}}$ and $\mathrm{SKerMDH}^{\mathrm{dl}}$ (cf. [ALSZ20]).

In this paper, we investigate the missing direction, namely the security of QA-NIZKs in the CRS model when the CRS is subverted but with *honestly chosen* ϱ. This can be viewed as a dual version of Jutla and Roy's QA-NIZK in [JR13b, JR13a]. Concretely, we define Sub-ZK QA-NIZKs security with some changes in the CRS model. The most important properties are completeness (an honest prover convinces an honest verifier, and an honestly generated CRS passes the CRS checking), computational (knowledge) soundness, and statistical subversion zero-knowledge (given a possibly subverted CRS, a proof generated by the honest prover reveals no information about the witness). We additionally consider introduce a notion of true-simulation extractability (tSE) [Har11][6]. Therefore, we rely on tag-based QA-NIZKs.

A tuple of PPT algorithms $\Pi = (\mathsf{Pgen}, \mathsf{Vcrs}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ is a Sub-ZK QA-NIZK if properties (i-iii) hold and is a tSE Sub-ZK QA-NIZK if properties (i-ii) and vi hold. Here, Vcrs is a new algorithm that checks the well-formedness of the CRS. We note that since soundness is proved in the case crs is generated correctly (by the verifier or a trusted third party) and V does not need to run Vcrs, so the computational soundness are similar to the original QA-NIZK definitions. We note that similar to ALSZ by a subversion ZK QA-NIZK argument we mean a *no-auxiliary-string non-black-box zero knowledge subversion ZK QA-NIZK argument*. In this paper for the sake of simplicity we just use subversion ZK QA-NIZK or Sub-ZK QA-NIZK for short. Subsequently, we recall only the properties that differ from the definitions of QA-NIZK in $Section\ 2$ (and in particular we omit (iii) adaptive soundness and computational knowledge soundness).

(i) Completeness. For any λ, and $(x, \boldsymbol{w}) \in \mathcal{R}_\varrho$,

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow_{\$} \mathcal{D}_{\mathsf{p}}; (\mathsf{crs}, \mathsf{tc}) \leftarrow \mathsf{K}(\varrho); \pi \leftarrow \mathsf{P}(\varrho, \mathsf{crs}, x, \boldsymbol{w}) : \\ \mathsf{Vcrs}(\varrho, \mathsf{crs}) = 1 \ \wedge \ \mathsf{V}(\varrho, \mathsf{crs}, x, \pi) = 1\end{array}\right] = 1 \ .$$

(ii) Statistical Subversion Zero-Knowledge. For any PPT subverter Z there exists a PPT extractor $\mathsf{Ext}_{\mathsf{Z}}$, such that for any computationally unbounded adversary $\mathcal{A}$, $|\varepsilon_0^{zk} -$

[6] Compared to the one independently introduced by Baghery *et al.* [BGPR20] we use non-black box extraction and guarantee only tSE.

$\mathsf{MATV}([\bar{\boldsymbol{A}}]_2)$ / $\mathcal{D}_k \in \{\mathcal{L}_k, \mathcal{IL}_k, \mathcal{C}_k, \mathcal{SC}_k\}$

check $[a_{11}]_2 \neq [0]_2 \wedge \ldots \wedge [a_{kk}]_2 \neq [0]_2$;
if $\mathcal{D}_k = \mathcal{L}_k$ **then** check $i \neq j \Rightarrow [a_{i,j}]_2 = [0]_2$;
elseif $\mathcal{D}_k = \mathcal{IL}_k$ **then** check $i \neq j \Rightarrow [a_{ij}]_2 = [0]_2$;
$\forall i, [a_{i,i}]_2 = [a_{1,1}]_2 + [i-1]_2$;
elseif $\mathcal{D}_k = \mathcal{C}_k$ **then** check $i \notin \{j, j+1\} \Rightarrow [a_{ij}]_2 = [0]_2$;
$\forall i, [a_{i+1,i}]_2 = [1]_2$;
elseif $\mathcal{D}_k = \mathcal{SC}_k$ **then** check $i \notin \{j, j+1\} \Rightarrow [a_{ij}]_2 = [0]_2$;
$\forall i\,([a_{i+1,i}]_2 = [1]_2 \wedge [a_{ii}]_2 = [a_{11}]_2)$; **fi**
return 1 if all checks pass and 0 otherwise;

Fig. 3. Auxiliary procedure MATV from [ALSZ20] for $\mathcal{D}_k \in \{\mathcal{L}_k, \mathcal{IL}_k, \mathcal{C}_k, \mathcal{SC}_k\}$.

$\varepsilon_1^{zk}| \approx_\lambda 0$, where $\varepsilon_b^{zk} :=$

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow_\$ \mathcal{D}_\mathsf{p}; \omega_\mathsf{Z} \leftarrow_\$ \mathsf{RND}(\mathsf{Z}); (\mathsf{crs}, \mathsf{aux}_\mathsf{Z}) \leftarrow \mathsf{Z}(\varrho; \omega_\mathsf{Z}); \\ \mathtt{tc} \leftarrow \mathsf{Ext}_\mathsf{Z}(\varrho; \omega_\mathsf{Z}); b \leftarrow_\$ \{0,1\} : \mathsf{Vcrs}(\varrho, \mathsf{crs}) = 1 \wedge \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\varrho, \mathsf{crs}, \mathsf{aux}_\mathsf{Z}) = 1\end{array}\right].$$

The oracle $\mathsf{O}_0(x, \boldsymbol{w})$ returns $\perp$ (reject) if $(x, \boldsymbol{w}) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{P}(\varrho, \mathsf{crs}, x, \boldsymbol{w})$. Similarly, $\mathsf{O}_1(x, \boldsymbol{w})$ returns $\perp$ (reject) if $(x, \boldsymbol{w}) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{Sim}(\varrho, \mathsf{crs}, \mathtt{tc}, x)$.

(vi) True-Simulation Extractability. For any PPT $\mathcal{A}$ there exists a non-uniform PPT extractor $\mathsf{Ext}_\mathcal{A}$,

$$\Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{K}_0(1^\lambda); \varrho \leftarrow_\$ \mathcal{D}_\mathsf{p}; (\mathsf{crs}, \mathtt{tc}) \leftarrow \mathsf{K}(\varrho); \omega_\mathcal{A} \leftarrow_\$ \mathsf{RND}(\mathcal{A}); \\ (\tau', x', \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\cdot,\cdot)}(\omega_\mathcal{A}; \varrho, \mathsf{crs}); \boldsymbol{w} \leftarrow \mathsf{Ext}_\mathcal{A}(\omega_\mathcal{A}; \varrho, \mathsf{crs}) : (x', \boldsymbol{w}) \notin \mathcal{R}_\varrho \\ \wedge\, (\tau', x') \notin Q \wedge \mathsf{V}(\varrho, \mathsf{crs}, \tau', x', \pi') = 1\end{array}\right] \approx_\lambda 0 \ .$$

where $\mathsf{O}(\tau, (x, \boldsymbol{w}))$ outputs $\mathsf{Sim}(\varrho, \mathsf{crs}, \tau, x, \mathtt{tc})$ if $(x, \boldsymbol{w}) \in \mathcal{R}_\varrho$ and adds (τ, x) to the set Q keeping track of the queries. If $(x, \boldsymbol{w}) \notin \mathcal{R}_\varrho$ it outputs $\perp$. One can also define a stronger variant called strong tSE which changes the winning condition to $(\tau', x', \pi') \notin Q$ and O records (τ, x, π) into Q.

3.2 QA-NIZKs with Subverted Setup

In this part, we construct a Sub-ZK QA-NIZK based on the QA-NIZK from KW [KW15], where we focus on the most efficient version $\mathit{\Pi}'_{\mathsf{as}}$. Intuitively, for constructing such a system, one needs two properties. Firstly one needs to make the CRS publicly verifiable, and secondly the trapdoor of the CRS should be extractable under some knowledge assumption (the latter is required to simulate proofs in the subversion zero-knowledge game).

We achieve the first property by defining a Vcrs algorithm which takes the CRS crs and the language parameter ϱ of the QA-NIZK's language and checks the well-formedness of the crs. If the possibly maliciously generated crs (from the prover's

$\mathsf{K}([\boldsymbol{M}]_1)$

- $\boldsymbol{A} \leftarrow\!\!\$\ \mathcal{D}_k$; $\boldsymbol{K} \leftarrow\!\!\$\ \mathbb{Z}_p^{n\times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n\times k}$;
- $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in \mathbb{Z}_p^{m\times k}$; $\mathsf{crs} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [\boldsymbol{P}, \boldsymbol{C}, \bar{\boldsymbol{A}}]_1)$; $\mathtt{tc} \leftarrow \boldsymbol{K}$;
- **return** $(\mathtt{tc}, \mathsf{crs})$.

$\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs})$:

- **if** $[\bar{\boldsymbol{A}}]_1 \in \mathbb{G}_1^{k\times k} \wedge [\boldsymbol{P}]_1 \in \mathbb{G}_1^{m\times k} \wedge [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k\times k}$
  $\wedge\, [\boldsymbol{C}]_1 \in \mathbb{G}_1^{n\times k} \wedge [\boldsymbol{C}]_2 \in \mathbb{G}_2^{n\times k}$
  $\wedge\, [\bar{\boldsymbol{A}}]_1[1]_2 = [1]_1[\bar{\boldsymbol{A}}]_2 \wedge [\boldsymbol{C}]_1[1]_2 = [1]_1[\boldsymbol{C}]_2$
  $\wedge\, [\boldsymbol{M}]_1^\top[\boldsymbol{C}]_2 = [\boldsymbol{P}]_1[\bar{\boldsymbol{A}}]_2 \wedge \mathsf{MATV}([\bar{\boldsymbol{A}}]_2) = 1$ **return** 1;

$\mathsf{P}([\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, \boldsymbol{w})$:

- $[\boldsymbol{\pi}]_1 \leftarrow [\boldsymbol{P}]_1^\top \boldsymbol{w} \in \mathbb{G}_1^k$;

$\mathsf{V}([\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$:

- **if** $[\boldsymbol{y}]_1^\top[\boldsymbol{C}]_2 = [\boldsymbol{\pi}]_1^\top[\bar{\boldsymbol{A}}]_2$ **return** 1;

$\mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{crs}, \mathtt{tc}, [\boldsymbol{y}]_1)$:

- $[\boldsymbol{\pi}]_1 \leftarrow \boldsymbol{K}^\top[\boldsymbol{y}]_1 \in \mathbb{G}_1^k$;

Fig. 4. Sub-ZK QA-NIZK Π_{sub}: Sub-ZK Π'_{as}.

point of view) passes the Vcrs algorithm, it is guaranteed that there exists a trapdoor tc for crs. Then, by using the BDH-KE assumption, we can extract the trapdoor tc from crs which realizes the second property. As in [ABLZ17] in context of subversion zk-SNARKs, we also need to add some extra elements $[\bar{\boldsymbol{A}}]_1 \in \mathbb{G}_1^{k\times k}$ and $[\boldsymbol{C}]_1 \in \mathbb{G}_1^{n\times k}$ to the CRS (assume that $\mathcal{D}_k$ outputs matrices $\boldsymbol{A}$ where the upper $k \times k$ submatrix $\bar{\boldsymbol{A}}$ is always invertible). Then, we prove that the new construction is complete, subversion zero-knowledge and adaptive sound in Theorem 5. We note, however, that there are also subversion zk-SNARKs [Fuc18] where one can achieve the public verifiability property of the CRS for free, i.e., without adding some extra elements to the CRS. We show that this can also be the case for subversion ZK QA-NIZKs and in particular the asymmetric QA-NIZKsdiscussed in Section 3.3.

Before describing the full construction of our Sub-ZK QA-NIZK argument Π_{sub}, we recall the definition of an efficiently verifiable distribution $\mathcal{D}_k$ from [ALSZ20]. This guarantees that for $\boldsymbol{A} \leftarrow\!\!\$\ \mathcal{D}_k$ there exists an algorithm $\mathsf{MATV}([\bar{\boldsymbol{A}}]_2)$ that outputs 1 if $\bar{\boldsymbol{A}}$ is invertible (we assume that the matrix distribution is robust) and well-formed with respect to $\mathcal{D}_k$ and otherwise outputs 0. Clearly, the distributions $\mathcal{D}_1$, $\mathcal{L}_k$, $\mathcal{IL}_k$, $\mathcal{C}_k$, and $\mathcal{SC}_k$ (for any k) are verifiable, as can be seen in Fig. 3 that allow one to verify whether $[\bar{\boldsymbol{A}}]_2$ is invertible.

Fig. 4 describes Sub-ZK QA-NIZK argument Π_{sub}, which is the subversion ZK version of the KW QA-NIZK argument Π'_{as} [KW15].

In Lemma 1, we show that from any adversary producing a valid CRS crs it is possible to extract the trapdoor $\boldsymbol{K}$ (simulation trapdoors). We will use it in the proof of subversion zero-knowledge in Theorem 5.

Lemma 1. *Let* BDH-KE *assumption hold and let* $[\boldsymbol{M}]_1 \leftarrow\!\!\$\ \mathcal{D}_{\mathsf{p}}$. *Then for any PPT adversary* $\mathcal{A}$ *there exists extractor* $\mathsf{Ext}_{\mathcal{A}}$ *such that the probability that* $\mathcal{A}$ *on input* $[\boldsymbol{M}]_1$ *and randomness* ω *outputs* crs *such that* $\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs}) = 1$ *and that* $\mathsf{Ext}_{\mathcal{A}}$ *on the same input, outputs* $\mathtt{tc} = \boldsymbol{K}$, *is overwhelming.*

Proof. Let adversary $\mathcal{A}$ output crs such that $\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs}) = 1$, which guarantees that elements from $\boldsymbol{P}$, $\bar{\boldsymbol{A}}$ and $\boldsymbol{C}$ are consistent and in particular that $[\boldsymbol{M}]_1^\top[\boldsymbol{C}]_2 =$

$\mathcal{A}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$

$(\mathsf{crs},\mathsf{aux}_{\mathcal{A}}) \leftarrow \mathcal{A}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$; **return** crs;

$\mathcal{A}_{\mathsf{BDH\text{-}KE}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$

$(\mathsf{crs},\mathsf{aux}_{\mathcal{A}}) \leftarrow \mathcal{A}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$;
return $([\bar{\boldsymbol{A}}]_1,[\bar{\boldsymbol{A}}]_2,[\boldsymbol{C}]_1,[\boldsymbol{C}]_2)$;

$\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$

- $(\bar{\boldsymbol{A}},\boldsymbol{C}) \leftarrow \mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$;
Compute $\boldsymbol{K} = \boldsymbol{C}\bar{\boldsymbol{A}}^{-1}$;
- **return** $\mathtt{tc} = \boldsymbol{K}$;

Fig. 5. The extractors and the constructed adversary $\mathcal{A}$ for Lemma 1.

$[\boldsymbol{P}]_1[\bar{\boldsymbol{A}}]_2$ and $\bar{\boldsymbol{A}}$ is invertible. Beside the main $\mathcal{A}$, we use an internal subverter $\mathcal{A}_{\mathsf{BDH\text{-}KE}}$. We note that both the subverter and the adversary are in connection and separating them is just for readability of the proof. Let $\omega_{\mathcal{A}} = \omega_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$. Let $\mathcal{A}_{\mathsf{BDH\text{-}KE}}$ run $\mathcal{A}$ and output $([\bar{\boldsymbol{A}}]_1,[\bar{\boldsymbol{A}}]_2,[\boldsymbol{C}]_1,[\boldsymbol{C}]_2)$. Then under the BDH-KE assumption, there exists an extractor $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$, such that if $\mathsf{Vcrs}([\boldsymbol{M}]_1,\mathsf{crs}) = 1$ then $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ outputs $(\bar{\boldsymbol{A}},\boldsymbol{C})$.

Let $\mathsf{Ext}_{\mathcal{A}}$ be an extractor that with input $([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ and running $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$ as subroutine, extracts $\mathtt{tc} = \boldsymbol{K}$. For the sake of simplicity, the full description of the algorithms is depicted in Fig. 5. More precisely, the extractor $\mathsf{Ext}_{\mathcal{A}}$ first runs $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ which outputs $(\bar{\boldsymbol{A}},\boldsymbol{C})$. Then, $\mathsf{Ext}_{\mathcal{A}}$ computes $\boldsymbol{K}$. Indeed, by having $\bar{\boldsymbol{A}}$, $\boldsymbol{CM}$, and the fact that $\boldsymbol{A}$ is invertible, the extractor $\mathsf{Ext}_{\mathcal{A}}$ can compute $\boldsymbol{K} = \boldsymbol{C}\bar{\boldsymbol{A}}^{-1}$.

Theorem 5. *Let Π_{sub} be a Sub-ZK QA-NIZK argument for linear subspaces from Fig. 4. Let $\mathcal{D}_{\mathsf{p}}$ be a witness samplable distribution. (i) Π_{sub} is subversion complete, (ii) if* BDH-KE *holds, then Π_{sub} is statistically subversion zero-knowledge, and (iii) if $\mathcal{D}_k$-SKerMDH holds then Π_{sub} is computationally sound.*

Proof. **(i: Completeness):** This is straightforward.

(ii: Subversion Zero-Knowledge:) Let the BDH-KE assumption hold. Let $\mathcal{A}$ be an adversary that computes crs so as to break the subversion zero-knowledge property of the Sub-ZK QA-NIZK in Fig. 4. That is, $\mathcal{A}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ outputs $(\mathsf{crs}^*,\mathsf{aux}_{\mathcal{A}})$. Let $\mathcal{A}$ be the adversary from Fig. 5 of Lemma 1. Let $\mathsf{RND}(\mathcal{A}) = \mathsf{RND}(\mathcal{A}_{\mathsf{BDH\text{-}KE}})$ in Lemma 1. Note that the subverter and the adversary are in connection. Underlying Lemma 1, if $\mathsf{Vcrs}([\boldsymbol{M}]_1,\mathsf{crs}^*) = 1$ then $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ from Fig. 5 outputs $\boldsymbol{K}$.

Fix concrete values of λ, $\mathsf{p} \in \mathrm{im}(\mathsf{K}_0(1^\lambda))$, $[\boldsymbol{M}]_1 \leftarrow_{\$} \mathcal{D}_{\mathsf{p}}$, $([\boldsymbol{y}]_1,\boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1}$, $\omega_{\mathcal{A}} \in \mathsf{RND}(\mathcal{A})$, and run $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1;\omega_{\mathcal{A}})$ to obtain $\boldsymbol{K}$. Thus, it suffices to show that if $\mathsf{Vcrs}([\boldsymbol{M}]_1,\mathsf{crs}^*) = 1$ and $([\boldsymbol{y}]_1,\boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1}$ then

$$\begin{aligned}\mathsf{O}_0([\boldsymbol{y}]_1,\boldsymbol{w}) =& \mathsf{P}([\boldsymbol{M}]_1,\mathsf{crs}^*,[\boldsymbol{y}]_1,\boldsymbol{w}) = [\boldsymbol{P}]_1^\top \boldsymbol{w}\ ,\\ \mathsf{O}_1([\boldsymbol{y}]_1,\boldsymbol{w}) =& \mathsf{Sim}([\boldsymbol{M}]_1,\mathsf{crs}^*,[\boldsymbol{y}]_1,\boldsymbol{K}) = \boldsymbol{K}^\top[\boldsymbol{y}]_1\end{aligned}$$

have the same distribution. This holds since from $\mathsf{Vcrs}([\boldsymbol{M}]_1,\mathsf{crs}^*) = 1$ it follows that $\boldsymbol{P} = \boldsymbol{M}^\top\boldsymbol{K}$ and from $([\boldsymbol{y}]_1;\boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1}$ it follows that $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}$. Thus,

$$\mathsf{O}_0([\boldsymbol{y}]_1,\boldsymbol{w}) = [\boldsymbol{P}]_1^\top\boldsymbol{w} = [\boldsymbol{K}^\top\boldsymbol{M}\boldsymbol{w}]_1 = \boldsymbol{K}^\top[\boldsymbol{y}]_1 = \mathsf{O}_1([\boldsymbol{y}]_1,\boldsymbol{w})\ .$$

Hence, O_0 and O_1 have the same distribution and thus, Π_{sub} is Sub-ZK under the BDH-KE assumption.

(iii: Adaptive Soundness:) The proof is similar to the adaptive soundness proof of Π'_{as} in [KW15, ALSZ20] but with some modifications in a way that instead of KerMDH, similar to [ALSZ20], the adaptive soundness proof of Π'_{as} is based on the $\mathcal{D}_k$-SKerMDH assumption (due to adding $[\bar{\boldsymbol{A}}]_1$ to the CRS). Assume that $\mathcal{A}$ breaks the adaptive soundness of subversion Π'_{as} with probability ε. We will build an adversary $\mathcal{B}$, that breaks $\mathcal{D}_k$-SKerMDH with probability $\geq \varepsilon - 1/p$.

Let $\mathcal{B}([\boldsymbol{A}]_1 \in \mathbb{G}_1^{(k+1)\times k}, [\boldsymbol{A}]_2 \in \mathbb{G}_2^{(k+1)\times k})$ generate $\boldsymbol{M} \leftarrow_{\$} \mathcal{D}'_{\mathsf{p}}$. Note that the $\mathcal{D}'_{\mathsf{p}}$ exists since $\mathcal{D}_{\mathsf{p}}$ is witness sampleable. Let $\boldsymbol{M}^{\perp}$ be the basis for the kernel of $\boldsymbol{M}^{\top}$ where $\boldsymbol{M}^{\top}\boldsymbol{M}^{\perp} = \mathbf{0}$. Then it computes $[\boldsymbol{A'}]_\iota = \begin{pmatrix} [\boldsymbol{A}]_\iota \\ \boldsymbol{R}\cdot[\boldsymbol{A}]_\iota \end{pmatrix} \in \mathbb{Z}_p^{(n-m+k)\times k}$ for $\iota = \{1,2\}$ where $\boldsymbol{R} \leftarrow_{\$} \mathbb{G}_\iota^{(n-m-1)\times(k+1)}$.

Let $[\bar{\boldsymbol{A}}']_\iota = [\bar{\boldsymbol{A}}]_\iota \in \mathbb{G}_\iota^{k\times k}$. Define implicitly (we do not know this value) $\boldsymbol{K} \leftarrow \boldsymbol{K'} + \boldsymbol{M}^{\perp}\underline{\boldsymbol{A}'}\bar{\boldsymbol{A}}^{-1} \in \mathbb{Z}_p^{n\times k}$ where $\boldsymbol{K'} \leftarrow_{\$} \mathbb{Z}_p^{n\times k}$. Thus,

$$[\boldsymbol{C}]_\iota = (\boldsymbol{K'}||\boldsymbol{M}^{\perp})[\boldsymbol{A'}]_\iota = [\boldsymbol{K'}\bar{\boldsymbol{A}'} + \boldsymbol{M}^{\perp}\underline{\boldsymbol{A}'}]_\iota = [(\boldsymbol{K'} + \boldsymbol{M}^{\perp}\underline{\boldsymbol{A}'}\bar{\boldsymbol{A}}^{-1})\bar{\boldsymbol{A}}]_\iota = [\boldsymbol{K}\bar{\boldsymbol{A}}]_\iota$$

and

$$[\boldsymbol{P}]_1 = [\boldsymbol{M}^{\top}\boldsymbol{K'}]_1 = [\boldsymbol{M}^{\top}(\boldsymbol{K} - \boldsymbol{M}^{\perp}\underline{\boldsymbol{A}'}\bar{\boldsymbol{A}}^{-1})]_1 = [\boldsymbol{M}^{\top}\boldsymbol{K}]_1 \ .$$

Thus, $\mathsf{crs}' = ([\boldsymbol{A},\boldsymbol{C}]_2, [\boldsymbol{A},\boldsymbol{C},\boldsymbol{P}]_1)$ has the same distribution as the real crs.

With probability ε, $([\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1) \leftarrow \mathcal{A}([\boldsymbol{M}]_1, \mathsf{crs}')$ is successful, so, for $\boldsymbol{y} \notin \mathrm{span}(\boldsymbol{M})$ we have that $\boldsymbol{y}^{\top}\boldsymbol{M}^{\perp} \neq \mathbf{0}_{1\times(n-m)}$. Since $\mathcal{A}$ wins, $\boldsymbol{y}^{\top}\boldsymbol{C} = \boldsymbol{\pi}^{\top}\bar{\boldsymbol{A}}$. Thus,

$$\begin{aligned}\boldsymbol{\pi}^{\top}\bar{\boldsymbol{A}} - \boldsymbol{y}^{\top}\boldsymbol{C} &= \left(\boldsymbol{\pi}^{\top}||\mathbf{0}_{n-m}^{\top}\right)\boldsymbol{A'} - \boldsymbol{y}^{\top}\left(\boldsymbol{K'}||\boldsymbol{M}^{\perp}\right)\boldsymbol{A'} \\ &= \left((\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top}\boldsymbol{K'})|| - \boldsymbol{y}^{\top}\boldsymbol{M}^{\perp}\right)\boldsymbol{A'} = \boldsymbol{c}^{\top}\boldsymbol{A'} = \mathbf{0}\end{aligned}$$

where $[\boldsymbol{c}]_1^{\top} \leftarrow [(\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top}\boldsymbol{K'})|| - \boldsymbol{y}^{\top}\boldsymbol{M}^{\perp}]_1$. Define $[\boldsymbol{c}]_1^{\top}$ as $[\boldsymbol{c}_1^{\top}||\boldsymbol{c}_2^{\top}]_1$ with $[\boldsymbol{c}_1]_1 \in \mathbb{G}_1^{k+1}$ and $[\boldsymbol{c}_2]_1 \in \mathbb{G}_1^{n-m-1}$. Set $\boldsymbol{s}_2 \leftarrow_{\$} \mathbb{Z}_p^{k+1}$; $[\boldsymbol{s}_1]_1 \leftarrow [\boldsymbol{c}_1 + \boldsymbol{R}^{\top}\boldsymbol{c}_2 + \boldsymbol{s}_2]_1$.

Clearly, $\boldsymbol{s}_1 - \boldsymbol{s}_2 = \boldsymbol{c}_1 + \boldsymbol{R}^{\top}\boldsymbol{c}_2$ and

$$(\boldsymbol{s}_1^{\top} - \boldsymbol{s}_2^{\top})\boldsymbol{A} = (\boldsymbol{c}_1^{\top} + \boldsymbol{c}_2^{\top}\boldsymbol{R})\boldsymbol{A} = \boldsymbol{c}^{\top}\boldsymbol{A'} = \mathbf{0}_{1\times k} \ .$$

Since $\boldsymbol{c} \neq \mathbf{0}_{n-m+k}$ and $\boldsymbol{R}$ leaks only through $\boldsymbol{A'}$ as $\boldsymbol{RA}$,

$$\Pr[\boldsymbol{c}_1 + \boldsymbol{R}^{\top}\boldsymbol{c}_2 = \mathbf{0} \mid \boldsymbol{RA}] \leq 1/p \ ,$$

where the probability is over $\boldsymbol{R} \leftarrow_{\$} \mathbb{Z}_p^{(n-m-1)\times(k+1)}$. Finally $\mathcal{B}$ outputs the pair $([\boldsymbol{s}_1]_1, [\boldsymbol{s}_2]_2)$ as the answer to the $\mathcal{D}_k$-SKerMDH problem.

3.3 Asymmetric QA-NIZK in the Subversion Setting

Now, we consider the asymmetric QA-NIZK argument in [GHR15] and show how one can achieve asymmetric Sub-ZK QA-NIZK, i.e., subversion versions of Π_{asy} and Π'_{asy}.

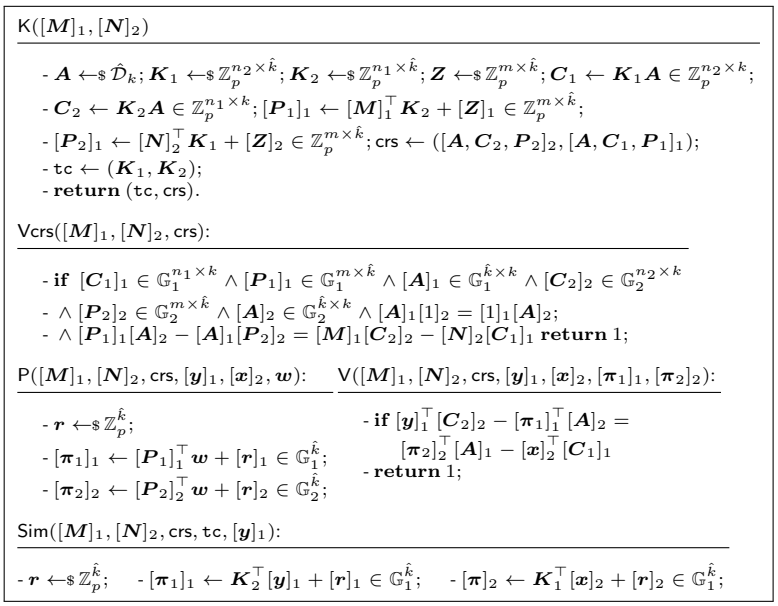

Fig. 6. Asymmetric Subversion QA-NIZK $\Pi_{\mathsf{asy\text{-}sub}}$: Sub Π_{asy} ($\hat{\mathcal{D}}_k = \mathcal{D}_k$ and $\hat{k} = k+1$) and Sub Π'_{asy} ($\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ and $\hat{k} = k$).

To this aim, similar to Sub-zk-SNARKs [Fuc18], we present a new Vcrs algorithm that does not require adding extra elements into the CRS. For extractability, we then again use the well-known $\mathsf{BDH\text{-}KE}$ and KoE knowledge assumptions and show that if the possibly maliciously generated crs passes the Vcrs algorithm, then under the knowledge assumptions there exists an extractor that extracts the trapdoor $\mathtt{tc}$ of crs. In Theorem 6 we prove completeness and subversion zero-knowledge of the asymmetric Sub-ZK QA-NIZKs. Since we do not add any new elements to the CRS, the soundness proof of the asymmetric Sub-ZK QA-NIZKs will be the same as the one in [GHR15]. We depict the full construction of the asymmetric Sub-ZK QA-NIZK arguments in Fig. 6.

We also want to stress that one can adapt the asymmetric Sub-ZK QA-NIZKs construction in Fig. 6 to the *sum in subspace language* and obtain the subversion version of the *argument of sum in subspace* of [GHR15]. In Lemma 2, we show that from any adversary producing a valid CRS crs from scratch it is possible to extract the trapdoors $(\boldsymbol{K}_1, \boldsymbol{K}_2)$. We will use it in the proof of subversion zero-knowledge in Theorem 6.

Lemma 2. *For any PPT adversary $\mathcal{A}$ that outputs a CRS crs^*, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$, such that if $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$ then $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathcal{A}})$ outputs $\mathtt{tc} = (\boldsymbol{K}_1, \boldsymbol{K}_2)$.*

Proof. Let $\mathcal{A}$ be the adversary from Fig. 7. The subverter Z outputs crs^* such that $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$. For sake of simplicity, the same as Lemma 1 we assume there are some internal $\mathsf{Z}_{\mathsf{BDH\text{-}KE}}$ and $\mathsf{Z}_{\mathsf{KoE}}$ which can compute some part of the CRS. The adversary $\mathcal{A}$ and all the subverters Z

$\mathcal{A}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$

$(\mathsf{crs}^*, \mathsf{aux}_{\mathsf{Z}}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$;
return crs^*;

$\mathsf{Ext}_{\mathsf{Z}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$

$(\boldsymbol{K}'_1, \boldsymbol{K}'_2) \leftarrow \mathsf{Ext}_{\mathsf{Z}_{\mathsf{KoE}}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}_{\mathsf{KoE}}})$;
return $\mathtt{tc} = (\boldsymbol{K}'_1, \boldsymbol{K}'_2)$;

$\mathsf{Z}_{\mathsf{BDH\text{-}KE}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}_{\mathsf{BDH\text{-}KE}}})$

$(\mathsf{crs}^*, \mathsf{aux}_{\mathsf{Z}}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$;
return $([\boldsymbol{A}]_1, [\boldsymbol{A}]_2)$;

$\mathsf{Z}_{\mathsf{KoE}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}_{\mathsf{KoE}}})$

$(\mathsf{crs}^*, \mathsf{aux}_{\mathsf{Z}}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$;
$([\boldsymbol{A}]_1, [\boldsymbol{A}]_2) \leftarrow \mathsf{Z}_{\mathsf{BDH\text{-}KE}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}_{\mathsf{BDH\text{-}KE}}})$;
$\boldsymbol{A} \leftarrow \mathsf{Ext}_{\mathsf{Z}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}_{\mathsf{BDH\text{-}KE}}})$;
return
$([\boldsymbol{K}'_{1,ij}\boldsymbol{A}_{jt}]_2, [\boldsymbol{K}'_{2,ij}]_2), ([\boldsymbol{K}'_{1,ij}\boldsymbol{A}_{jt}]_2, [\boldsymbol{K}'_{2,ij}]_2)$;

Fig. 7. The extractors and the constructed adversary $\mathcal{A}$ for Lemma 2.

are in connection. Assume $\mathsf{Z}_{\mathsf{BDH\text{-}KE}}$ runs Z and outputs $([\boldsymbol{A}]_1, [\boldsymbol{A}]_2)$. Then from the BDH-KE assumption, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}}$, such that if $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$ then $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}}([\boldsymbol{M}, [\boldsymbol{N}]_2]_1; \omega_{\mathsf{Z}_{\mathsf{BDH\text{-}KE}}})$ outputs $\boldsymbol{A}$. Let $\mathsf{Z}_{\mathsf{KoE}}$ runs the subverter Z and $\mathsf{Z}_{\mathsf{BDH\text{-}KE}}$ and the extractor $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}}$, and outputs $([\boldsymbol{K}'_{1,ij}\boldsymbol{A}_{jt}]_1, (\boldsymbol{K}'_1)_{1,ij})_{i\in[1,n],j\in[1,\hat{k}],t\in[1,k]}$ and $([\boldsymbol{K}'_{2,ij}\boldsymbol{A}_{jt}]_2, (\boldsymbol{K}'_2)_{2,ij})_{i\in[1,n],j\in[1,\hat{k}],t\in[1,k]}$. Roughly speaking, the subverter $\mathsf{Z}_{\mathsf{KoE}}$ runs $\mathsf{Z}_{\mathsf{BDH\text{-}KE}}$ and $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{BDH\text{-}KE}}$, obtains $\boldsymbol{A}$. By having $\boldsymbol{A}$, and solving the system of linear equations of $([\boldsymbol{C}_1]_1, [\boldsymbol{C}_2]_2)$ (i.e $\left(\begin{smallmatrix}\boldsymbol{X}_1\\ \boldsymbol{X}_2\end{smallmatrix}\right)\left(\begin{smallmatrix}\boldsymbol{A}\end{smallmatrix}\right) = \left(\begin{smallmatrix}[\boldsymbol{C}_1]_1\\ [\boldsymbol{C}_2]_2\end{smallmatrix}\right)$), $\mathsf{Z}_{\mathsf{BDH\text{-}KE}}$ computes $([\boldsymbol{K}'_1]_1 = \boldsymbol{X}_1, [\boldsymbol{K}_2]_1 = \boldsymbol{X}'_2)$ such that $[\boldsymbol{K}'_1]_1\boldsymbol{A} = [\boldsymbol{K}_1]_1\boldsymbol{A} = [\boldsymbol{C}_1]_1$ and $[\boldsymbol{K}'_2]_2\boldsymbol{A} = [\boldsymbol{K}_2]_2\boldsymbol{A} = [\boldsymbol{C}_2]_2$. The $\mathsf{Z}_{\mathsf{KoE}}$ finally outputs $([\boldsymbol{K}'_{1,ij}\boldsymbol{A}_{jt}]_1, (\boldsymbol{K}'_1)_{1,ij})_{i\in[1,n],j\in[1,\hat{k}],t\in[1,k]}$ and $([\boldsymbol{K}'_{2,ij}\boldsymbol{A}_{jt}]_2, (\boldsymbol{K}'_2)_{2,ij})_{i\in[1,n],j\in[1,\hat{k}],t\in[1,k]}$. Based on KoE assumption, that if $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{KoE}}$ knowing the random coins of $\mathsf{Z}_{\mathsf{KoE}}$, outputs $(\boldsymbol{K}'_1, \boldsymbol{K}'_2)$.

Theorem 6. *Let $\Pi_{\mathsf{asy\text{-}sub}}$ be a asymmetric Sub-ZK QA-NIZK argument for linear subspaces from Fig. 6. (i) $\Pi_{\mathsf{asy\text{-}sub}}$ is subversion complete, (ii) if the* BDH-KE *and* KoE *assumptions hold, then $\Pi_{\mathsf{asy\text{-}sub}}$ is statistically subversion zero-knowledge, and (iii) if the $\mathcal{D}_k$-SKerMDH, (for the case $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$, the distribution $\mathcal{D}_{\mathsf{p}}$ should be WS) then $\Pi_{\mathsf{asy\text{-}sub}}$ is computationally sound.*

Proof. **(i: Completeness):** This is straightforward from the construction.

(ii: Subversion Zero-Knowledge:) Let BDH-KE and KoE assumptions hold. Let Z be a subverter that computes crs^* so as to break the subversion zero-knowledge of Fig. 6. That is, $\mathsf{Z}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$ outputs $(\mathsf{crs}^*, \mathsf{aux}_{\mathsf{Z}})$. Let $\mathcal{A}$ be the same adversary as in Lemma 2. Note that $\mathsf{RND}(\mathcal{A}) = \mathsf{RND}(\mathsf{Z})$. Underlying Lemma 2, if $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$ then $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$ from Fig. 7 outputs $((\boldsymbol{K}'_1, \boldsymbol{K}'_2))$ such that $[\boldsymbol{K}'_1]_1\boldsymbol{A} = [\boldsymbol{K}_1]_1\boldsymbol{A} = [\boldsymbol{C}_1]_1$ and $[\boldsymbol{K}'_2]_2\boldsymbol{A} = [\boldsymbol{K}_2]_2\boldsymbol{A} = [\boldsymbol{C}_2]_2$. Since $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$, one concludes that $[\boldsymbol{M}]_1^\top\boldsymbol{K}'_2 = [\boldsymbol{M}]_1^\top\boldsymbol{K}_2$ and $[\boldsymbol{N}]_2^\top\boldsymbol{K}'_1 = [\boldsymbol{N}]_2^\top\boldsymbol{K}_1$ which these properties are enough for simulating the proof.

Fix concrete values of λ, $\mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$, $([\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1,[\boldsymbol{N}]_2}$, $\omega_{\mathsf{Z}} \in \mathsf{RND}(\mathsf{Z})$, and run $\mathsf{Ext}_{\mathsf{Z}}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2; \omega_{\mathsf{Z}})$ to obtain $(\boldsymbol{K}'_1, \boldsymbol{K}'_1)$. Thus, it suffices to show

$\mathsf{K}([\boldsymbol{M}]_1)$

- $\boldsymbol{A} \leftarrow_\$ \mathcal{D}_k$; $(\boldsymbol{K}_i)_{i=0}^{i=\ell} \leftarrow_\$ \mathbb{Z}_p^{n\times(k+1)}$; $(\boldsymbol{C}_i)_{i=0}^{i=\ell} \leftarrow \boldsymbol{K}_i\boldsymbol{A} \in \mathbb{Z}_p^{n\times k}$;
- $[(\boldsymbol{P}_i)_{i=0}^{i=\ell}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K}_i \in \mathbb{Z}_p^{m\times(k+1)}$; $\mathsf{crs} \leftarrow ([\boldsymbol{A}, (\boldsymbol{C}_i)_{i=0}^{i=\ell}]_2, [\boldsymbol{A}, (\boldsymbol{P}_i)_{i=0}^{i=\ell}]_1)$;
- $\mathtt{tc} \leftarrow (\boldsymbol{K}_i)_{i=0}^{i=\ell}$;
- **return** $(\mathtt{tc}, \mathsf{crs})$

$\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs})$:

- **if** $[\boldsymbol{A}]_1 \in \mathbb{G}_1^{(k+1)\times k} \wedge [(\boldsymbol{P}_i)_{i=0}^{i=\ell}]_1 \in \mathbb{G}_1^{m\times k} \wedge [\boldsymbol{A}]_2 \in \mathbb{G}_2^{(k+1)\times k} \wedge [(\boldsymbol{C}_i)_{i=0}^{i=\ell}]_2 \in \mathbb{G}_2^{n\times k}$
- $\wedge\, [\boldsymbol{A}]_1[1]_2 = [1]_1[\boldsymbol{A}]_2 \wedge [\boldsymbol{M}]_1^\top [(\boldsymbol{C}_i)_{i=0}^{i=\ell}]_2 = [(\boldsymbol{P}_i)_{i=0}^{i=\ell}]_1[\boldsymbol{A}]_2$ **return** 1;

$\mathsf{P}(\tau, [\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, \boldsymbol{w})$:

- $[\boldsymbol{\pi}]_1 \leftarrow (\sum_{i=0}^{i=\ell} \tau^i [\boldsymbol{P}_i]_1^\top)\boldsymbol{w} \in \mathbb{G}_1^{k+1}$;

$\mathsf{V}(\tau, [\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$:

- **if** $[\boldsymbol{y}]_1^\top \sum_{i=0}^{i=\ell} \tau^i [\boldsymbol{C}_i]_2 = [\boldsymbol{\pi}]_1^\top [\boldsymbol{A}]_2$ **return** 1 :

$\mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{crs}, \mathtt{tc}, [\boldsymbol{y}]_1)$:

- $[\boldsymbol{\pi}]_1 \leftarrow \sum_{i=0}^{i=\ell} \tau^i \boldsymbol{K_i}^\top [\boldsymbol{y}]_1 \in \mathbb{G}_1^{k+1}$;

Fig. 8. ℓ-time simulation sound Sub-ZK QA-NIZK argument $\Pi_{\mathsf{ls\text{-}sub}}$.

that if $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$ and $([\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1,[\boldsymbol{N}]_2}$ then

$$\begin{aligned}\mathsf{O}_0([\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{w}) =& \mathsf{P}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{w}) \ ,\\ \mathsf{O}_1([\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{w}) =& \mathsf{Sim}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{x}]_2, \boldsymbol{K}'_1, \boldsymbol{K}'_2)\end{aligned}$$

have the same distribution. This holds since from $\mathsf{Vcrs}([\boldsymbol{M}]_1, [\boldsymbol{N}]_2, \mathsf{crs}^*) = 1$. Hence, O_0 and O_1 have the same distribution and thus, $\Pi_{\mathsf{asy\text{-}sub}}$ is Sub-ZK under BDH-KE and KoE assumptions.

(iii: Adaptive Soundness:) If $\hat{\mathcal{D}}_k = \mathcal{D}_k$, it follows directly from the adaptive soundness proof in [GHR15]. If $\hat{\mathcal{D}}_k = \bar{\mathcal{D}}_k$, and $\mathcal{D}_{\mathsf{p}}$ is WS, it follows directly from the adaptive soundness proof in [GHR15].

4 Subversion True-Simulation Extractable QA-NIZK

In this section, we present an unbounded true-simulation extractable Sub-ZK QA-NIZK (tSE Sub-ZK QA-NIZK) version of the Sub-ZK QA-NIZK. To this aim, we rely on the discrete logarithm assumption, in the algebraic group model (AGM) [FKL18]. Roughly speaking, inspired by [KW15], we first modify the Sub-ZK QA-NIZK Π'_{as} in Section 3.2 to make it unbounded tSE, then we add some new elements in the CRS to make

$\mathsf{K}([\boldsymbol{M}]_1)$

- $\bar{\boldsymbol{A}}, \boldsymbol{B} \leftarrow\!\!\$\ \bar{\mathcal{D}}_k$; $\boldsymbol{K} \leftarrow\!\!\$\ \mathbb{Z}_p^{n\times k}$; $\boldsymbol{K}_0, \boldsymbol{K}_1 \leftarrow\!\!\$\ \mathbb{Z}_p^{k\times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n\times k}$;
- $(\boldsymbol{C}_i)_{i=0}^{i=1} \leftarrow \boldsymbol{K}_i\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{k\times k}$; $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in \mathbb{G}_1^{m\times k}$;
- $[(\boldsymbol{P}_i)_{i=0}^{i=1}]_1 \leftarrow [\boldsymbol{B}]_1\boldsymbol{K}_i \in \mathbb{G}_1^{k\times k}$; $\mathsf{crs} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}, (\boldsymbol{C}_i)_{i=0}^{i=1}]_2, [\bar{\boldsymbol{A}}, \boldsymbol{C}, \boldsymbol{B}, \boldsymbol{P}, (\boldsymbol{P}_i)_{i=0}^{i=1}]_1)$; $\mathtt{tc} \leftarrow \boldsymbol{K}$;
- **return** $(\mathtt{tc}, \mathsf{crs})$.

$\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs})$:

- **if** $[\bar{\boldsymbol{A}}]_1 \in \mathbb{G}_1^{k\times k} \wedge [\boldsymbol{B}]_2 \in \mathbb{G}_2^{k\times k} \wedge [\boldsymbol{P}]_1 \in \mathbb{G}_1^{m\times k} \wedge [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k\times k} \wedge\ [\boldsymbol{C}]_1 \in \mathbb{G}_1^{n\times k} \wedge [\boldsymbol{C}]_2 \in \mathbb{G}_2^{n\times k}$
- $\wedge\ [(\boldsymbol{P}_i)_{i=0}^{i=1}]_1 \in \mathbb{G}_1^{k\times k} \wedge [(\boldsymbol{C}_i)_{i=0}^{i=1}]_2 \in \mathbb{G}_2^{k\times k} \wedge [\bar{\boldsymbol{A}}]_1[1]_2 = [1]_1[\bar{\boldsymbol{A}}]_2 \wedge\ [\boldsymbol{C}]_1[1]_2 = [1]_1[\boldsymbol{C}]_2$
- $\wedge\ [\boldsymbol{M}]_1^\top[\boldsymbol{C}]_2 = [\boldsymbol{P}]_1[\bar{\boldsymbol{A}}]_2 \wedge\ [\boldsymbol{B}]_1^\top[(\boldsymbol{C}_i)_{i=0}^{i=1}]_2 = [(\boldsymbol{P}_i)_{i=0}^{i=1}]_1[\bar{\boldsymbol{A}}]_2 \wedge \mathsf{MATV}(\bar{\boldsymbol{A}}) = 1$ **return** 1;

$\mathsf{P}(\tau, [\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, \boldsymbol{w})$:

- $\boldsymbol{r} \leftarrow\!\!\$\ \mathbb{Z}_p^k$; $[\boldsymbol{\pi}]_1 \leftarrow ([\boldsymbol{\pi}_1]_1, [\boldsymbol{\pi}_2]_1) \leftarrow ([\boldsymbol{P}]_1^\top\boldsymbol{w} + (\sum_{i=0}^{i=1}\tau^i[\boldsymbol{P}_i]_1^\top)\boldsymbol{r}, [\boldsymbol{B}]_1\boldsymbol{r}) \in (\mathbb{G}_1^k)^2$;

$\mathsf{V}(\tau, [\boldsymbol{M}]_1, \mathsf{crs}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$:

- Parse $[\boldsymbol{\pi}]_1 = ([\boldsymbol{\pi}_1]_1, [\boldsymbol{\pi}_2]_1)$; **if** $[\boldsymbol{y}]_1^\top[\boldsymbol{C}]_2 + [\boldsymbol{\pi}_2]_1^\top \sum_{i=0}^{i=1}\tau^i[\boldsymbol{C}_i]_2 = [\boldsymbol{\pi}_1]_1^\top[\bar{\boldsymbol{A}}]_2$ **return** 1.

$\mathsf{Sim}(\tau, [\boldsymbol{M}]_1, \mathsf{crs}, \mathtt{tc}, [\boldsymbol{y}]_1)$:

- $\boldsymbol{r} \leftarrow\!\!\$\ \mathbb{Z}_p^k$; $[\boldsymbol{\pi}]_1 = ([\boldsymbol{\pi}_1]_1, [\boldsymbol{\pi}_2]_1) \leftarrow (\boldsymbol{K}^\top[\boldsymbol{y}]_1 + \sum_{i=0}^{i=1}\tau^i[\boldsymbol{P}_i]_1^\top\boldsymbol{r}, [\boldsymbol{B}]_1\boldsymbol{r}) \in (\mathbb{G}_1^k)^2$.

Fig. 9. Unbounded true-simulation extractable Sub-ZK QA-NIZK argument $\Pi_{\mathsf{utse\text{-}sub}}$.

it publicly verifiable. We define a new Vcrs algorithm to check whether the CRS is well-formed. Then by applying the technique from Lemma 1, we show the extractability of the CRS. We present the full construction of unbounded SE Sub-ZK QA-NIZK in Fig. 9. We note that we overcome the problem in [KW15] of requiring that $n > m$ and the lack of knowledge soundness, which does not make then usable within LegoSNARK. So we avoid the $n > m$ restriction, but for knowledge soundness, the matrix $[\boldsymbol{M}]_1$ must be generated using a witness sampleable distribution $\mathcal{D}_{\mathsf{p}}$, i.e., there must exist a polynomial time algorithm that samples $\boldsymbol{M}$ in $\mathbb{Z}_p$ such that $[\boldsymbol{M}]_1$ has the same distribution as the one sampled with $\mathcal{D}_p$. However, we note that this is satisfied for the use-case within LegoSNARK where $\boldsymbol{M}$ includes bases of a Pedersen-like commitment schemes (cf. Section 5). Finally, we discuss how to obtain strong true-simulation extractability (tSE) for our construction.

In Lemma 3, we show that from any adversary producing a valid CRS crs it is possible to extract the trapdoor $\boldsymbol{K}$ (simulation trapdoors). We will use it in the proof of subversion zero-knowledge in Theorem 7.

Lemma 3. *Let* $\mathsf{BDH\text{-}KE}$ *assumption hold and let* $[\boldsymbol{M}]_1 \leftarrow\!\!\$\ \mathcal{D}_{\mathsf{p}}$. *Then for any PPT adversary* $\mathcal{A}$ *there exists extractor* $\mathsf{Ext}_{\mathcal{A}}$ *such that the probability that* $\mathcal{A}$ *on input* $[\boldsymbol{M}]_1$ *and randomness* ω *outputs* crs *such that* $\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs}) = 1$ *and that* $\mathsf{Ext}_{\mathcal{A}}$ *on the same input, outputs* $\mathtt{tc} = \boldsymbol{K}$ *is overwhelming.*

Proof. Let adversary $\mathcal{A}$ output crs such that $\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs}) = 1$, which guarantees that elements from $\boldsymbol{P}$, $\boldsymbol{C}$, and $\bar{\boldsymbol{A}}$ are consistent and in particular that $[\boldsymbol{M}]_1^\top[\boldsymbol{C}]_2 =$

$\mathcal{A}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$

$(\mathsf{crs}, \mathsf{aux}_{\mathcal{A}}) \leftarrow \mathcal{A}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$; **return** crs;

$\mathcal{A}_{\mathsf{BDH\text{-}KE}}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$

$(\mathsf{crs}, \mathsf{aux}_{\mathcal{A}}) \leftarrow \mathcal{A}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$;
return $([\bar{\boldsymbol{A}}]_1, [\bar{\boldsymbol{A}}]_2, [\boldsymbol{C}]_1, [\boldsymbol{C}]_2)$;

$\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$

- $(\bar{\boldsymbol{A}}, \boldsymbol{C}) \leftarrow \mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$;
Compute $\boldsymbol{K} = \boldsymbol{C}\bar{\boldsymbol{A}}^{-1}$;
- **return** $\mathtt{tc} = \boldsymbol{K}$;

Fig. 10. The extractors and the constructed adversary $\mathcal{A}$ for Lemma 3.

$[\boldsymbol{P}]_1[\bar{\boldsymbol{A}}]_2$. Beside the main $\mathcal{A}$, we use an internal subverter $\mathcal{A}_{\mathsf{BDH\text{-}KE}}$. We note that both the subverter and the adversary are in connection and separating them is just for readability of the proof. Let $\omega_{\mathcal{A}} = \omega_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$. Let $\mathcal{A}_{\mathsf{BDH\text{-}KE}}$ runs $\mathcal{A}$ and outputs $([\bar{\boldsymbol{A}}]_1, [\bar{\boldsymbol{A}}]_2, [\boldsymbol{C}]_1, [\boldsymbol{C}]_2)$. Then under the BDH-KE assumption, there exists an extractor $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$, such that if $\mathsf{Vcrs}([\boldsymbol{M}]_1, \mathsf{crs}) = 1$ then $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$ outputs $(\bar{\boldsymbol{A}}, \boldsymbol{C})$.

Let $\mathsf{Ext}_{\mathcal{A}}$ be an extractor that with input $([\boldsymbol{M}]_1; \omega_{\mathcal{A}})$ and running $\mathsf{Ext}^{\mathsf{BDH\text{-}KE}}_{\mathcal{A}_{\mathsf{BDH\text{-}KE}}}$ as subroutine, extracts $\mathtt{tc} = \boldsymbol{K} = \boldsymbol{C}\bar{\boldsymbol{A}}^{-1}$. Thus, from the Kronecker-Capelli theorem we know that this system has a unique solution. For the sake of simplicity, the full description of the algorithms is depicted in Fig. 10.

Theorem 7. *Let $\Pi_{\mathsf{utse\text{-}sub}}$ be the unbounded tSE Sub-ZK QA-NIZK argument for linear subspaces from Fig. 9. (i) $\Pi_{\mathsf{utse\text{-}sub}}$ is subversion complete, (ii) if* BDH-KE *holds, then $\Pi_{\mathsf{utse\text{-}sub}}$ is Sub-ZK, and (iii) if the discrete logarithm assumption, in the AGM holds then $\Pi_{\mathsf{utse\text{-}sub}}$ is unbounded true-simulation extractable.*

Proof. Completeness and Sub-ZK proofs are straightforward from Theorem 5 (for the Sub-ZK proof, one first extract $\mathtt{tc}$ underlying Lemma 3 and then follows the Sub-ZK proof of Theorem 5).

(iii: Unbounded True Simulation Extractability:) We show this under the discrete logarithm assumption in asymmetric bilinear groups in the AGM [FKL18].

Without loss of generality, we consider $\Pi_{\mathsf{utse\text{-}sub}}$ for $k = 1$. Assume an algebraic adversary $\mathcal{A}([\boldsymbol{M}]_1, \mathsf{crs}, \mathsf{aux})$ against the simulation extractability of $\Pi_{\mathsf{utse\text{-}sub}}$ where aux is an associated auxiliary input and $\mathsf{crs} = ([a, \boldsymbol{C}, (C_i)_{i=0}^{i=1}]_2, [a, b, \boldsymbol{C}, \boldsymbol{P}, (P_i)_{i=0}^{i=1}]_1$ and she accesses her simulation oracle on the instances $([\boldsymbol{y}_1]_1, \ldots, [\boldsymbol{y}_q]_1)$ to obtain the responses $(([\boldsymbol{\pi}_1]_1, \tau_1), \ldots, ([\boldsymbol{\pi}_q]_1, \tau_q))$. Let $[\boldsymbol{\zeta}]_1$ be a vector that contains $\boldsymbol{M}$ and the portion of aux that has elements from the group $\mathbb{G}_1$ and assume $[\boldsymbol{\zeta}]_1$ includes $[1]_1$. $\mathcal{A}$ returns a tuple $(\tau, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1 = ([\boldsymbol{\pi}_1]_1, [\boldsymbol{\pi}_2]_1))$ along with coefficients that explain these elements as linear combinations of its input in the group $\mathbb{G}_1$. Let $r \leftarrow_{\$} \mathbb{Z}_p$ and these coefficients be:

$$[\boldsymbol{y}]_1 = \boldsymbol{Y}_0[\boldsymbol{P}]_1 + \boldsymbol{Y}_1[\boldsymbol{\zeta}]_1 + \boldsymbol{Y}_2[a]_1 + \boldsymbol{Y}_3[b]_1 + \boldsymbol{Y}_{4i}[(P_i)_{i=0}^{i=1}]_1 + \boldsymbol{Y}_5[\boldsymbol{C}]_1 + \sum_{j=0}^{j=q} \boldsymbol{Y}'_j[\boldsymbol{y}_j]_1$$

$$[\boldsymbol{\pi}_1]_1 = \boldsymbol{Z}_0[\boldsymbol{P}]_1 + \boldsymbol{Z}_1[\boldsymbol{\zeta}]_1 + \boldsymbol{Z}_2[a]_1 + \boldsymbol{Z}_3[b]_1 + \sum_{i=0}^{i=1} \boldsymbol{Z}_{4,i}\tau^i[P_i]_1 r + \boldsymbol{Z}_{10}[\boldsymbol{C}]_1 + \sum_{j=0}^{j=q} \boldsymbol{Z}'_j[\boldsymbol{\pi}_{\mathbf{1}j}]_1$$

$$[\boldsymbol{\pi}_2]_1 = \boldsymbol{Z}_5[\boldsymbol{P}]_1 + \boldsymbol{Z}_6[\boldsymbol{\zeta}]_1 + \boldsymbol{Z}_7[a]_1 + \boldsymbol{Z}_{8i}[(P_i)_{i=0}^{i=1}]_1 + \boldsymbol{Z}_9[b]_1 r + \boldsymbol{Z}_{11}[\boldsymbol{C}]_1 + \sum_{j=0}^{j=q} \boldsymbol{Z}''_j[\boldsymbol{\pi}_{\mathbf{2}j}]_1 \quad (1)$$

Let the extractor $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1, \mathsf{crs}, \mathsf{aux})$ be the algorithm that runs $\mathcal{A}$ and returns $\boldsymbol{w} = \boldsymbol{Z}_0$. Then, we have to show that the probability that the output of $(\mathcal{A}, \mathsf{Ext}_{\mathcal{A}})$ satisfies verification while $\boldsymbol{y} \neq \boldsymbol{M}\boldsymbol{w}$ is negligible. In other words, assume that the output of $\mathcal{A}$ is such that $[\boldsymbol{y}]_1$ is not queried before, and $[\boldsymbol{y}]_1 \neq [\boldsymbol{M}]_1\boldsymbol{Z}_0$, and plugged into the verification equation we have:

$$[\boldsymbol{y}^\top \boldsymbol{K} + \boldsymbol{\pi}_2^\top \sum_{i=0}^{i=1} \tau^i K_i - \boldsymbol{\pi}_1^\top]_1 [a]_2 = [0]_T.$$

This means that $\boldsymbol{y}^\top \boldsymbol{K} + \boldsymbol{\pi}_2^\top \sum_{i=0}^{i=1} \tau^i K_i - \boldsymbol{\pi}_1^\top = 0$. If it happens with non-negligible probability, we can construct an algorithm $\mathcal{B}$ that on input $([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ outputs nonzero elements $\boldsymbol{\alpha} \in \mathbb{Z}_p^{n \times n}$, $\boldsymbol{\beta} \in \mathbb{Z}_p^n$, and $\gamma \in \mathbb{Z}_p$ s.t.

$$\boldsymbol{K}^\top \boldsymbol{\alpha} \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{\beta} + \gamma = 0$$

and then we can construct an algorithm $\mathcal{C}$ against the discrete logarithm assumption in asymmetric bilinear groups, which given elements $([t]_1, [t]_2)$ returns the exponent $t \in \mathbb{Z}_p$. More precisely $\mathcal{B}([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ proceeds as follows:

- Choose $([\boldsymbol{M}]_1, \mathsf{aux})$ from $\mathcal{D}_{\mathsf{p}}$ along with its $\mathbb{G}_1$ elements (i.e., a vector $\boldsymbol{\zeta}$ of entries in $\mathbb{Z}_p$).
- Sample $a, b \leftarrow_{\$} \bar{\mathcal{D}}_k$, $K_0, K_1 \leftarrow_{\$} \mathbb{Z}_p$, set $(C_i)_{i=0}^{i=1} \leftarrow aK_i$, and $(P_i)_{i=0}^{i=1} \leftarrow bK_i$. Run $\mathcal{A}([\boldsymbol{\zeta}, \boldsymbol{P}, \boldsymbol{C}, (P_i)_{i=0}^{i=1}, a, b]_1, [a, a\boldsymbol{K}, (C_i)_{i=0}^{i=1}]_2)$. We note that $\mathcal{A}$'s input can be efficiently simulated.
- Once received the output of $\mathcal{A}$, it sets $\boldsymbol{\alpha} := \boldsymbol{Y}_0\boldsymbol{M}^\top$, $\boldsymbol{\beta} := \boldsymbol{Y}_1\boldsymbol{\zeta} + \boldsymbol{Y}_2 a + \boldsymbol{Y}_3 b + \boldsymbol{Y}_{4i}(P_i)_{i=0}^{i=1} + \boldsymbol{Y}_5\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Y}'_j \boldsymbol{y}_j - \boldsymbol{M}\boldsymbol{Z}_0$ and $\gamma := -(\boldsymbol{Z}_1\boldsymbol{\zeta} + \boldsymbol{Z}_2 a + \boldsymbol{Z}_3 b + \sum_{i=0}^{i=1} \boldsymbol{Z}_{4,i}\tau^i P_i r + \boldsymbol{Z}_{10}\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Z}'_j \boldsymbol{\pi}_{\mathbf{1}j}) + \sum_{i=0}^{i=1} \tau^i K_i (\boldsymbol{Z}_5\boldsymbol{P} + \boldsymbol{Z}_6\boldsymbol{\zeta} + \boldsymbol{Z}_7 a + \boldsymbol{Z}_{8i}(P_i)_{i=0}^{i=1} + \boldsymbol{Z}_9 br + \boldsymbol{Z}_{11}\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Y}''_j \boldsymbol{\pi}_{\mathbf{2}j})$

Notice that $\boldsymbol{K}^\top \boldsymbol{\alpha} \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{\beta} + \gamma =$

$$\begin{aligned}
&\boldsymbol{K}^\top \boldsymbol{Y}_0 \boldsymbol{M}^\top \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{Y}_1 \boldsymbol{\zeta} + \boldsymbol{K}^\top \boldsymbol{Y}_2 a + \boldsymbol{K}^\top \boldsymbol{Y}_3 b + \boldsymbol{K}^\top \boldsymbol{Y}_{4i}(P_i)_{i=0}^{i=1} + \boldsymbol{K}^\top \boldsymbol{Y}_5 \boldsymbol{C} \\
&+ \boldsymbol{K}^\top \sum_{j=0}^{j=q} \boldsymbol{Y}'_j \boldsymbol{y}_j - \boldsymbol{K}^\top \boldsymbol{M}\boldsymbol{Z}_0 - (\boldsymbol{Z}_1\boldsymbol{\zeta} + \boldsymbol{Z}_2 a + \boldsymbol{Z}_3 b + \sum_{i=0}^{i=1} \boldsymbol{Z}_{4,i}\tau^i P_i r + \boldsymbol{Z}_{10}\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Z}'_j \boldsymbol{\pi}_{\mathbf{1}j}) \\
&+ \sum_{i=0}^{i=1} \tau^i K_i (\boldsymbol{Z}_5\boldsymbol{P} + \boldsymbol{Z}_6\boldsymbol{\zeta} + \boldsymbol{Z}_7 a + \boldsymbol{Z}_{8i}(P_i)_{i=0}^{i=1} + \boldsymbol{Z}_9 br + \boldsymbol{Z}_{11}\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Y}''_j \boldsymbol{\pi}_{\mathbf{2}j}) \\
&= \boldsymbol{K}^\top (\boldsymbol{Y}_0 \boldsymbol{M}^\top \boldsymbol{K} + \boldsymbol{Y}_1\boldsymbol{\zeta} + \boldsymbol{Y}_2 a + \boldsymbol{Y}_3 b + \boldsymbol{Y}_{4i}(P_i^\top)_{i=0}^{i=1} + \boldsymbol{Y}_5\boldsymbol{C} + \sum_{j=0}^{j=q} \boldsymbol{Y}'_j \boldsymbol{y}_j) - \boldsymbol{\pi}_1
\end{aligned}$$

$$+\sum_{i=0}^{i=1}\tau^i K_i\boldsymbol{\pi}_2 = \boldsymbol{K}^\top\boldsymbol{y} - \boldsymbol{\pi}_1 + \sum_{i=0}^{i=1}\tau^i K_i\boldsymbol{\pi}_2 = 0.$$

Note that, one among $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ must be nonzero. Indeed, if they are all zero then $\boldsymbol{Y}_0 = \boldsymbol{0}$ and also $\boldsymbol{Y}_1\boldsymbol{\zeta}+\boldsymbol{Y}_2 a+\boldsymbol{Y}_3 b+\boldsymbol{Y}_{4i}(P_i)_{i=0}^{i=1}+\boldsymbol{Y}_5\boldsymbol{C}+\sum_{j=0}^{j=q}\boldsymbol{Y}'_j\boldsymbol{y}_j-\boldsymbol{M}\boldsymbol{Z}_0 = \boldsymbol{0}$, thus from Eq. (1), we have $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{Z}_0$, which contradicts our assumption on $\mathcal{A}$'s output. If $\boldsymbol{\gamma} = 0$ then from Eq. (1), we have $\boldsymbol{\pi}_1 = \boldsymbol{Z}_0\boldsymbol{P} + \sum_{i=0}^{i=1}\tau^i K_i\boldsymbol{\pi}_2$ which means the adversary has output one of the simulated proofs and so the queried τ, and contradicts our assumption on $\mathcal{A}$'s output.

Finally we show how the above problem can be reduced to discrete logarithm problem, i.e., the adversary $\mathcal{C}$ on input $([t]_1, [t]_2)$ returns t. Indeed $\mathcal{C}$ samples $\boldsymbol{r}, \boldsymbol{s} \in \mathbb{Z}_p^n$ and implicitly sets $\boldsymbol{K} = t\boldsymbol{r} + \boldsymbol{s}$. We see that $([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ can be efficiently simulated with a distribution identical to the one expected by $\mathcal{B}$. Next, given a solution $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ such that $\boldsymbol{K}^\top\boldsymbol{\alpha} + \boldsymbol{K}^\top\boldsymbol{\beta} + \gamma = 0$, one can find $e_1, e_2, e_3 \in \mathbb{Z}_p$ such that:

$$0 = (t\boldsymbol{r}+\boldsymbol{s})^\top\boldsymbol{\alpha}(t\boldsymbol{r}+\boldsymbol{s})+(t\boldsymbol{r}+\boldsymbol{s})^\top\boldsymbol{\beta}+\gamma = t^2(\boldsymbol{r}^\top\boldsymbol{\alpha}\boldsymbol{r})+t(\boldsymbol{r}^\top\boldsymbol{\alpha}\boldsymbol{s}+\boldsymbol{s}^\top\boldsymbol{\alpha}\boldsymbol{r}+\boldsymbol{r}^\top\boldsymbol{\beta})$$
$$+ (\boldsymbol{s}^\top\boldsymbol{\alpha}\boldsymbol{s} + \boldsymbol{s}^\top\boldsymbol{\beta} + \gamma) = e_1t^2 + e_2t + e_3$$

In particular, with overwhelming probability (over the choice of $\boldsymbol{s}$ that is information theoretically hidden from $\mathcal{B}$'s view) $e_3 \neq 0$. From this solution, $\mathcal{C}$ can solve the system and extract t.

On Achieving Strong True-Simulation Extractability. We recall that for strong tSE we additionally require non-malleability on the proofs π in that our winning condition is changed to $(\tau', x', \pi') \notin Q$. Now to achieve this we can use the generic compiler from [Har11] and in particular we additionally use a strongly unforgeable (sEUF-CMA-secure) one-time signature (sOTS) scheme (e.g., Groth's sOTS [Gro06] or Boneh-Boyen signatures [BB04]). The prover P is now changed so that in addition to computing the proof of the tSE Sub-ZK QA-NIZK it samples a key pair of the sOTS and signs the proof, where the signature and the verification key are attached to the proof. Moreover, instead of randomly choosing the tag τ, P uses a collision-resistant hash function and computes the tag as the hash of the verification key of the sOTS and the word x. Verification is then straightforward.

5 Integrating Sub-ZK QA-NIZK into LegoSNARK

5.1 The LegoSNARK Framework

We recall that LegoSNARK [CFQ19] is a framework for Commit-and-Prove zk-SNARKs (CP-SNARKs) with the aim of constructing a "global" SNARK for some computation C via the linking of "smaller" specialized SNARKs for different subroutines that overall compose to C. LegoSNARK denotes these specialized SNARKs by proof gadgets which form the basic building blocks that can be reused and composed as required. The main idea is that by letting each subroutine of C be handled by a different proof system chosen such that one that maximizes a metric (e.g., efficiency) important for the concrete application.

Therefore, LegoSNARK relies on the commit-and-prove (CP) methodology [CLOS02], i.e., one proves statements of the form *commitment* $c_{\mathsf{ck}}(x)$ *contains* x *such that* $\mathcal{R}(x, \boldsymbol{w}) = 1$. LegoSNARK considers new CP-SNARKs for several basic relations, where the main one is $\mathsf{CP}_{\mathsf{link}}$ for proving that two different commitments (i.e., Pedersen-like commitments) open to the same vector. More precisely, $\mathsf{CP}_{\mathsf{link}}$ proves that a linear relation $\boldsymbol{F}\boldsymbol{u} = x$ holds for a committed vector $\boldsymbol{u}$, a public matrix $\boldsymbol{F}$ and public vector x.

Using $\mathsf{CP}_{\mathsf{link}}$ LegoSNARK obtains CP versions of popular efficient zkSNARKs, such as Groth's [Gro16], and zkSNARKs for linear subspaces (QA-NZIKs) [KW15], latter which can prove statements about data committed using the Pedersen scheme for vectors [Ped92]. Such commit-and-prove schemes are useful in applications where one needs to commit before the SNARK keys for a relation are created, e.g., to post commitments on a blockchain so that one can later prove statements about the committed data.

5.2 Integration of Sub-ZK QA-NIZK into LegoSNARK

We now show how to integrate our Sub-ZK QA-NIZK (as well as unbounded tSE Sub-ZK QA-NIZK discussed in Section 4) into the LegoSNARK framework of CP-SNARKs [CFQ19]. LegoSNARK uses a knowledge-sound version of the Kiltz-Wee QA-NIZK Π'_{as}. They show how to use this QA-NIZK to construct CP-SNARKs that work for any commitment scheme whose verification algorithm is the same as the generalized Pedersen commitment and present two schemes. The first scheme $\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{link}}$ allows proving that commitments under different keys open to the same vector and the second more general scheme $\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ allows proving the correctness of a linear function of a committed vector.

Subsequently, we will show how to transform our Sub-ZK QA-NIZK and (strong) tSE Sub-ZK QA-NIZK into Sub-CP-SNARKs and (strong) tSE Sub-CP-SNARKs. Then, we will construct subversion variants of the more general $\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ which we denote Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ and tSE Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ respectively. For the Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ version, we note that our result can be applied equivalently to the more specific first scheme. Technically, we, therefore, need to show that our Π_{sub} based on Π'_{as} is knowledge-sound. With regard to the potentially malicious generation of the respective commitment keys, as mentioned in [CFQ19], for Pedersen commitments they can easily be sampled in a transparent way such that no trusted setup is needed, e.g., by deriving them using a suitable hash function modelled as a random oracle. Consequently, we obtain a subversion variant of LegoSNARK for the QA-NIZK part and stress that using other recent results on subversion zk-SNARKs in [GM17b,Lip19,Bag19,ARS20], one can further extend the toolbox of a subversion variant of the LegoSNARK framework.

We now demonstrate how to construct a Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ and (strong) tSE Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ for the linear relation $\mathcal{R}^{\mathsf{Lin}}$, which checks linear properties of some committed vectors: for a fixed public matrix $\boldsymbol{M} \in \mathbb{Z}_p^{n \times m}$, relation $\mathcal{R}^{\mathsf{Lin}}_{\boldsymbol{M}}$ over public input $[\boldsymbol{y}]_1 \in \mathbb{G}_1^n$ and witness $\boldsymbol{w} \in \mathbb{Z}_p^m$, with $\boldsymbol{w} := (\boldsymbol{w}_j)_{j \in [\ell]}$ and $\boldsymbol{w}_j \in \mathbb{Z}_p^{n_j}$, holds iff $[\boldsymbol{y}]_1 = [\boldsymbol{M}]_1 \boldsymbol{w}$.

For simplicity, we mostly use the notation in [CFQ19]. Let Com be a commitment scheme such that $\mathsf{Com.VerCommit} = \mathsf{Ped.VerCommit}$. Let $\mathsf{pk} = [\boldsymbol{h}]_1 \in \mathbb{G}_1^{n+1}$ be the key of the global commitment Com. In our subversion $\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$, the public inputs of the

prover are ℓ commitments $(c_j)_{j\in[\ell]}$ and another commitment $\boldsymbol{c}'$; the witness is a set of openings $((\boldsymbol{w}_j)_{j\in[\ell]};(o_j)_{j\in[\ell]})$ for commitments $(c_j)_{j\in[\ell]}$ and $[\boldsymbol{y}]_1$. In particular, the prover must prove: $\mathcal{R}^{\mathsf{lin}}_{\mathsf{ped}}(\boldsymbol{y},(c_j)_{j=1}^{\ell},(\boldsymbol{w}_j)_{j=1}^{\ell},(o_j)_{j=1}^{\ell})=1 \iff$

$$\bigwedge_{j=1}^{\ell} c_j = (o_j,\boldsymbol{w}_j)\cdot[\boldsymbol{h}_{[0\,..\,n_j]}]_\iota \wedge \boldsymbol{y}=[\boldsymbol{M}]_\iota\cdot(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_\ell)\ .$$

Our scheme, called subversion Commit-and-Prove (Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$), is quite similar to $\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ of [CFQ19] but it uses a Sub-ZK QA-NIZK in the prove phase. The Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ essentially consists of the following algorithms:

$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}.\mathsf{K}(\mathcal{R}^{\mathsf{Lin}}_{\boldsymbol{M}},\mathsf{pk})$**:** parse $\mathsf{pk}=[\boldsymbol{h}]_1\in\mathbb{G}_1^{m+1}$. Use $[\boldsymbol{h}]_1$ and $\mathcal{R}^{Lin}_{\boldsymbol{M}}$ to construct $[\boldsymbol{M}^*]_1$ as in Eq. (2). Run $(\mathsf{crs},\mathtt{tc})\leftarrow\Pi_{\mathsf{sub}}.\mathsf{K}([\boldsymbol{M}^*]_1)$. Return $(\mathsf{crs},\mathtt{tc})$.
$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}.\mathsf{Vcrs}([\boldsymbol{M}^*]_1,\mathsf{crs})$**:** return $\Pi_{\mathsf{sub}}.\mathsf{Vcrs}([\boldsymbol{M}^*]_1,\mathsf{crs})$.
$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}.\mathsf{P}([\boldsymbol{M}^*]_1,\mathsf{crs},[\boldsymbol{y}^*]_1,\boldsymbol{w}^*)$**:** return $\pi\leftarrow\Pi_{\mathsf{sub}}.\mathsf{P}(\boldsymbol{M}^*,\mathsf{crs},[\boldsymbol{y}^*]_1,\boldsymbol{w}^*)$.
$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}.\mathsf{V}([\boldsymbol{M}^*]_1,\mathsf{crs},[\boldsymbol{y}^*]_1,\pi)$**:** return $\Pi_{\mathsf{sub}}.\mathsf{V}([\boldsymbol{M}^*]_1,\mathsf{crs},\boldsymbol{y}^*,\pi)$.

Notice that the scheme Sub-$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ considers each $\boldsymbol{w}_j$ to be committed using a Pedersen commitment scheme whose key is $\mathsf{pk}=[\boldsymbol{h}]_1\in\mathbb{G}_1^{m+1}$. The general idea is to express such a commit-prove relation with the linear subspace relation $\mathcal{R}_{[\boldsymbol{M}^*]_1}(x^*,\boldsymbol{w}^*)$ that holds iff $[\boldsymbol{y}^*]_1=[\boldsymbol{M}^*]_1\boldsymbol{w}^*$, where $[\boldsymbol{y}^*]_1\in\mathbb{G}_1^l$, $[\boldsymbol{M}^*]_1\in\mathbb{G}_1^{l\times t}$, and $\boldsymbol{w}^*\in\mathbb{Z}_p^t$ can be built from the inputs of $\mathcal{R}^{Lin}_{\boldsymbol{F}}$ for $l=\ell+n$ and $t=m+\ell$, as follows:

$$\overbrace{\begin{pmatrix}c_1\\ \vdots\\ c_\ell\\ \boldsymbol{y}\end{pmatrix}}^{\boldsymbol{y}^*} = \overbrace{\left(\begin{array}{cccc|cccc} h_0 & 0 & \cdots & 0 & h_{[1,n_1]} & 0 & \cdots & 0\\ 0 & h_0 & \cdots & 0 & 0 & h_{[1,n_2]} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & \cdots & h_0 & 0 & 0 & \cdots & h_{[1,n_\ell]}\\ \hline 0 & 0 & \cdots & 0 & \boldsymbol{M} & \boldsymbol{M} & \cdots & \boldsymbol{M}\end{array}\right)}^{\boldsymbol{M}^*} \overbrace{\begin{pmatrix}o_1\\ \vdots\\ o_\ell\\ \boldsymbol{w}\end{pmatrix}}^{\boldsymbol{w}^*} \tag{2}$$

Subsequently, we show that we can obtain a Sub-CP-SNARK suitable for LegoSNARK when using a suitable knowledge-sound Sub-ZK QA-NIZK Π_{sub}.

Theorem 8. *Let $\boldsymbol{M}\in\mathbb{Z}_p^{n\times m}$ be a matrix from a distribution $\mathcal{D}_{\mathsf{p}}$, and* aux *be an auxiliary input distribution. If Π_{sub} is subversion zero-knowledge and knowledge sound, then the Sub-*CP-SNARK *construction Sub-*$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ *given above is (i) subversion zero-knowledge and (ii) knowledge sound.*

We present the proof in Appendix A. Additionally, we show that we can obtain a (strong) tSE Sub-CP-SNARK suitable for LegoSNARK when using a (strong) tSE Sub-ZK QA-NIZK $\Pi_{\mathsf{utse\text{-}sub}}$.

Theorem 9. *Let $\boldsymbol{M}\in\mathbb{Z}_p^{n\times m}$ be a matrix from a distribution $\mathcal{D}_{\mathsf{p}}$, and* aux *be an auxiliary input distribution. If Π_{sub} is subversion zero-knowledge and knowledge sound, then the tSE Sub-*CP-SNARK *construction is tSE Sub-*$\mathsf{CP}^{\mathsf{Ped}}_{\mathsf{lin}}$ *given above is (i) subversion zero-knowledge and (ii) unbounded true-simulation extractable.*

Proof. The proof is straightforward from subversion zero-knowledge and unbounded true-simulation extractability of $\Pi_{\mathsf{utse\text{-}sub}}$ in Theorem 7.

Remark. LegoSNARK does not consider the integration of the asymmetric QA-NIZK (Π'_{asy}) by González *et al.* [GHR15]. We note, however, that this can be done analogously to Π'_{as}, which further helps to increase the expressiveness for languages supported by QA-NIZKs in LegoSNARK. Furthermore, we want to remark that our subversion version of Π'_{asy} can be integrated into LegoSNARK analogously to the integration of the subversion version of Π'_{as}.

Acknowledgements. We would like to thank Antonio Faonio for helpful discussion. This work received funding from the European Union's Horizon 2020 ECSEL Joint Undertaking under grant agreement n° 783119 (SECREDAS), from the European Union's Horizon 2020 research and innovation programme under grant agreement n°871473 (KRAKEN), and by the Austrian Science Fund (FWF) and netidee SCIENCE under grant agreement P31621-N38 (PROFET).

References

ABLZ17. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017.

ABP15. Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 69–100. Springer, Heidelberg, April 2015.

AJO+19. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, Jiaxin Pan, Arnab Roy, and Yuyu Wang. Shorter QA-NIZK and SPS with tighter security. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 669–699. Springer, Heidelberg, December 2019.

AJOR18. Masayuki Abe, Charanjit S. Jutla, Miyako Ohkubo, and Arnab Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 627–656. Springer, Heidelberg, December 2018.

ALSZ20. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 590–620. Springer, Heidelberg, May 2020.

ALSZ21. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On subversion-resistant snarks. *J. Cryptol.*, 34(3):17, 2021.

ARS20. Behzad Abdolmaleki, Sebastian Ramacher, and Daniel Slamanig. Lift-and-shift: Obtaining simulation extractable subversion and updatable SNARKs generically. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1987–2005. ACM Press, November 2020.

Bag19. Karim Baghery. Subversion-resistant simulation (knowledge) sound nizks. In *IMA International Conference on Cryptography and Coding*, pages 42–63. Springer, 2019.

BB04. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, Heidelberg, May 2004.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016.

BGPR20. Karim Baghery, Alonso González, Zaira Pindado, and Carla Ràfols. Signatures of knowledge for boolean circuits under standard assumptions. In Abderrahmane Nitaj and Amr M. Youssef, editors, *AFRICACRYPT 20*, volume 12174 of *LNCS*, pages 24–44. Springer, Heidelberg, July 2020.

Buc17. Jon Buck. Ethereum upgrade byzantium is live, verifes'first zk-snark proof, 2017. https://cointelegraph.com/news/ethereum-upgrade-byzantium-is-live-verifies-first-zk-snark-proof.

CFQ19. Matteo Campanelli, Dario Fiore, and Anaïs Querol. LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2075–2092. ACM Press, November 2019.

CLOS02. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002.

Dam92. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992.

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, December 2014.

DGP+19. Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. Shorter quadratic QA-NIZK proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, April 2019.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

Fuc18. Georg Fuchsbauer. Subversion-zero-knowledge SNARKs. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 315–347. Springer, Heidelberg, March 2018.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013.

GHKW16. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 1–27. Springer, Heidelberg, May 2016.

GHR15. Alonso González, Alejandro Hevia, and Carla Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629. Springer, Heidelberg, November / December 2015.

GM17a. Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. Cryptology ePrint Archive, Report 2017/540, 2017. https://eprint.iacr.org/2017/540.

GM17b. Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 581–612. Springer, Heidelberg, August 2017.

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

GOS06. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Heidelberg, May / June 2006.

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, December 2006.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

Har11. Kristiyan Haralambiev. *Efficient Cryptographic Primitives for Non-Interactive Zero-Knowledge Proofs and Applications*. PhD thesis, New York University, 2011.

JR13a. Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013.

JR13b. Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. Cryptology ePrint Archive, Report 2013/109, 2013. https://eprint.iacr.org/2013/109.

JR14. Charanjit S. Jutla and Arnab Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, August 2014.

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015.

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012.

Lip19. Helger Lipmaa. Simulation-extractable snarks revisited. Cryptology ePrint Archive, Report 2019/612, 2019. https://eprint.iacr.org/2019/612.

LPJY14. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532. Springer, Heidelberg, May 2014.

LPJY15. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, November / December 2015.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016.

Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, August 1992.

RS20. Carla Ràfols and Javier Silva. QA-NIZK arguments of same opening for bilateral commitments. In Abderrahmane Nitaj and Amr M. Youssef, editors, *AFRICACRYPT 20*, volume 12174 of *LNCS*, pages 3–23. Springer, Heidelberg, July 2020.

SCG+14. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

A Omitted Proof of Theorem 5

Proof. **(i: Subversion Zero-knowledge):** This is straightforward from subversion zero-knowledge proof in Theorem 5.

(ii: Knowledge Soundness): We show the theorem under the discrete logarithm assumption in asymmetric bilinear groups in the AGM [FKL18]. Without lose of generality, we consider the Sub-ZK QA-NIZK scheme for linear subspaces Π_{sub} for $\bar{\mathcal{D}}_k = \bar{\mathcal{D}}_k$ (Sub-ZK QA-NIZK Π'_{as} in Fig. 4), in the MDDH setting where $k = 1$. The proof follows the argumentation in [CFQ19]. Assume an algebraic adversary $\mathcal{A}([\boldsymbol{M}]_1, \mathsf{crs}, \mathsf{aux})$ against the knowledge soundness of Π_{sub} where aux is an associated auxiliary input and $\mathsf{crs} = \{[\bar{\boldsymbol{A}}, \boldsymbol{P}]_1, [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2\}$. Let $[\boldsymbol{\zeta}]_1$ be a vector that contains $\boldsymbol{M}$ and the portion of aux that has elements from the group $\mathbb{G}_1$. Assume $[\boldsymbol{\zeta}]_1$ includes $[1]_1$. $\mathcal{A}$ returns a pair $([\boldsymbol{y}]_1, [\pi]_1)$ along with coefficients that explain these elements as linear combinations of its input in the group $\mathbb{G}_1$. Let these coefficients be:

$$[\boldsymbol{y}]_1 = \boldsymbol{Y}_0[\boldsymbol{P}]_1 + \boldsymbol{Y}_1[\boldsymbol{\zeta}]_1 + \boldsymbol{Y}_2[\bar{\boldsymbol{A}}]_1 + \boldsymbol{Y}_3[\boldsymbol{C}]_1$$
$$[\boldsymbol{\pi}]_1 = \boldsymbol{Z}_0[\boldsymbol{P}]_1 + \boldsymbol{Z}_1[\boldsymbol{\zeta}]_1 + \boldsymbol{Z}_2[\bar{\boldsymbol{A}}]_1 + \boldsymbol{Z}_3[\boldsymbol{C}]_1$$

Let the extractor $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1, \mathsf{crs}, \mathsf{aux})$ be the algorithm that runs $\mathcal{A}$ and returns $\boldsymbol{w} = \boldsymbol{Z}_0$. Then, we have to show that the probability that the output of $(\mathcal{A}, \mathsf{Ext}_{\mathcal{A}})$ satisfies verification while $\boldsymbol{y} \neq \boldsymbol{M}\boldsymbol{w}$ is negligible. In other words, assume that the output of $\mathcal{A}$ is such that $[\boldsymbol{y}]_1 \neq [\boldsymbol{M}]_1\boldsymbol{Z}_0$ and, $[\boldsymbol{y}]_1^\top[a\boldsymbol{K}]_2 = [\boldsymbol{\pi}]_1[a]_2$. If it happens with non-negligible

probability, we can construct an algorithm $\mathcal{B}$ that on input $([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ outputs nonzero elements $\boldsymbol{\alpha} \in \mathbb{Z}_p^{\ell \times \ell}$, $\boldsymbol{\beta} \in \mathbb{Z}_p^{\ell}$, and $\gamma \in \mathbb{Z}_p$ such that

$$\boldsymbol{K}^\top \boldsymbol{\alpha} \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{\beta} + \gamma = 0.$$

Then we can construct an algorithm $\mathcal{C}$ against the discrete logarithm assumption in asymmetric bilinear groups such that given elements $([t]_1, [t]_2)$ it returns the exponent $t \in \mathbb{Z}_p$. More precisely the algorithm $\mathcal{B}([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ proceeds as follows:

- Choose $([\boldsymbol{M}]_1, \mathsf{aux})$ from $\mathcal{D}_{\mathsf{p}}$ along with its $\mathbb{G}_1$ (i.e., a $\mathbb{Z}_p$ vector $\boldsymbol{\zeta}$).
- Sample $a \leftarrow_{\$} \mathbb{Z}_p$ and run $\mathcal{A}([\boldsymbol{\zeta}, \boldsymbol{C}, \boldsymbol{P}, a]_1, [a, a\boldsymbol{K}]_2)$. We note that $\mathcal{A}$'s input can be efficiently simulated.
- Once received the output of $\mathcal{A}$, it sets $\boldsymbol{\alpha} := \boldsymbol{Y}_0 \boldsymbol{M}^\top$, $\boldsymbol{\beta} := \boldsymbol{Y}_1 \boldsymbol{\zeta} + \boldsymbol{Y}_2 a + \boldsymbol{Y}_3 \boldsymbol{C} - \boldsymbol{M}\boldsymbol{Z}_0$ and $\gamma := -\boldsymbol{Z}_1 \boldsymbol{\zeta} - \boldsymbol{Z}_2 a - \boldsymbol{Z}_3 \boldsymbol{C}$

Notice that

$$\begin{aligned}\boldsymbol{K}^\top \boldsymbol{\alpha} \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{\beta} + \gamma &= \boldsymbol{K}^\top \boldsymbol{Y}_0 \boldsymbol{M}^\top \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{Y}_1 \boldsymbol{\zeta} + \boldsymbol{K}^\top \boldsymbol{Y}_2 a + \boldsymbol{K}^\top \boldsymbol{Y}_3 \boldsymbol{C} \\ &\quad - \boldsymbol{K}^\top \boldsymbol{M}\boldsymbol{Z}_0 - \boldsymbol{Z}_1 \boldsymbol{\zeta} - \boldsymbol{Z}_2 a - \boldsymbol{Z}_3 \boldsymbol{C} = \boldsymbol{K}^\top \boldsymbol{Y}_0 \boldsymbol{M}^\top \boldsymbol{K} + \boldsymbol{K}^\top \boldsymbol{Y}_1 \boldsymbol{\zeta} \\ &\quad + \boldsymbol{K}^\top \boldsymbol{Y}_2 a + \boldsymbol{K}^\top \boldsymbol{Y}_3 \boldsymbol{C} - \boldsymbol{\pi} = \boldsymbol{K}^\top \boldsymbol{y} - \boldsymbol{\pi} = 0.\end{aligned}$$

Note that, one among $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and γ must be nonzero. Indeed, if they are all zero then $\boldsymbol{Y}_1 \boldsymbol{\zeta} + \boldsymbol{Y}_2 a + \boldsymbol{Y}_3 \boldsymbol{C} - \boldsymbol{M}\boldsymbol{Z}_0 = 0$, that is $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{Z}_0$, which contradicts our assumption on $\mathcal{A}$'s output.

Finally we show how the above problem can be reduced to discrete logarithm problem in asymmetric groups, i.e., the adversary $\mathcal{C}$ on input $([t]_1, [t]_2)$ returns t. Indeed $\mathcal{C}$ samples $\boldsymbol{r}, \boldsymbol{s} \in \mathbb{Z}_p^{\ell}$ and implicitly sets $\boldsymbol{K} = t\boldsymbol{r} + \boldsymbol{s}$. We see that $([\boldsymbol{K}]_1, [\boldsymbol{K}]_2)$ can be efficiently simulated with a distribution identical to the one expected by $\mathcal{B}$. Next, given a solution $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$ such that $\boldsymbol{K}^\top \boldsymbol{\alpha} + \boldsymbol{K}^\top \boldsymbol{\beta} + \gamma = 0$, one can find $e_1, e_2, e_3 \in \mathbb{Z}_p$ such that:

$$\begin{aligned}0 &= (t\boldsymbol{r} + \boldsymbol{s})^\top \boldsymbol{\alpha} (t\boldsymbol{r} + \boldsymbol{s}) + (t\boldsymbol{r} + \boldsymbol{s})^\top \boldsymbol{\beta} + \gamma = t^2 (\boldsymbol{r}^\top \boldsymbol{\alpha} \boldsymbol{r}) + t(\boldsymbol{r}^\top \boldsymbol{\alpha} \boldsymbol{s} + \boldsymbol{s}^\top \boldsymbol{\alpha} \boldsymbol{r} + \boldsymbol{r}^\top \boldsymbol{\beta}) \\ &\quad + (\boldsymbol{s}^\top \boldsymbol{\alpha} \boldsymbol{s} + \boldsymbol{s}^\top \boldsymbol{\beta} + \gamma) = e_1 t^2 + e_2 t + e_3.\end{aligned}$$

In particular, with overwhelming probability (over the choice of $\boldsymbol{s}$ that is information theoretically hidden from $\mathcal{B}$'s view) $e_3 \neq 0$. From this solution, $\mathcal{C}$ can solve the system and extract t.