On QA-NIZK in the BPK Model

Behzad Abdolmaleki[1], Helger Lipmaa[1], Janno Siim[1,2], and Michał Zając[1]

[1] University of Tartu, Estonia
[2] STACC, Estonia

Abstract. While the CRS model is widely accepted for construction of non-interactive zero knowledge (NIZK) proofs, from the practical viewpoint, a very important question is to minimize the trust needed from the creators of the CRS. Recently, Bellare *et al.* defined subversion-resistance (security in the case the CRS creator may be malicious) for NIZK. First, we observe that subversion zero knowledge (Sub-ZK) in the CRS model corresponds to no-auxiliary-string non-black-box NIZK (also known as nonuniform NIZK) in the Bare Public Key (BPK) model. Due to well-known impossibility results, this observation provides a simple proof that the use of non-black-box techniques is needed to obtain Sub-ZK. Second, we prove that the most efficient known QA-NIZK for linear subspaces by Kiltz and Wee is nonuniform zero knowledge in the BPK model under two alternative novel knowledge assumptions, both secure in the subversion generic bilinear group model. We prove that (for a different set of parameters) a slightly less efficient variant of Kiltz-Wee is nonuniform zero knowledge in the BPK model under a known knowledge assumption that is also secure in the subversion generic bilinear group model.

Keywords: Bare public key model, non-black-box zero knowledge, nonuniform zero knowledge, QA-NIZK, subversion-security

1 Introduction

Zero-knowledge proof systems introduced by Goldwasser *et al.* [GMR85] enable a prover to convince a verifier in veracity of a statement while leaking no additional information. Blum *et al.* [BFM88] introduced non-interactive zero-knowledge (NIZK) proof systems where the prover outputs just one message (the proof) that convinces the verifier in the truth of the statement.

In particular, efficient transferable succinct non-interactive zero knowledge argument systems (SNARGs) are very useful in cryptographic applications, allowing the prover to create a succinct argument π that can be transferred to many different verifiers who can check the correctness of the argument at their leisure time. Among many examples, an efficient SNARG can be used in e-voting to prove the correctness of shuffling and decryption, in cryptocurrencies [BCG$^+$14] to prove the correctness of a transaction, but also to construct UC-secure commitments [FLM11] and verifiable computation.

As it is well-known, NIZKs are impossible in the standard model, and thus in all such applications, one has to rely on some trust assumption like the common reference string (CRS [BFM88, FLS90, BDMP91]) model stating that there exists a trusted third party who has created the CRS from a correct distribution. Other, weaker, trust models include the registered public key (RPK, [BCNP04]) model and the bare public key (BPK, [CGGM00, MR01]) model. However, very few NIZKs are known in the RPK model (see, e.g., [BCNP04, DFN06, VV09]) while black-box zero knowledge [MR01, APV05] and even auxiliary-string non-black-box [Wee07] NIZK is impossible in the BPK model.

Recently, very efficient pairing-based quasi-adaptive NIZKs [JR13, LPJY14, JR14, ABP15, KW15, GHR15] (QA-NIZKs) have been constructed in the CRS model, with the QA-NIZK of Libert *et al.* [LPJY14] being the first QA-NIZK with *constant-length* argument. Although QA-NIZKs for some other languages are known (e.g., the language of bitstrings [GHR15] and the languages of shuffles [GR16]; both requiring a quadratic-length CRS), research on QA-NIZKs has been concentrated on designing more efficient QA-NIZKs for linear subspaces. The latter holds true partially because of the wide applicability of QA-NIZKs for linear subspaces in the design of various cryptographic primitives ranging from UC-secure commitment schemes [FLM11, JR13], dual system fully secure identity-based encryption [JR13], publicly-verifiable fully secure identity-based encryption [JR13], threshold keyed-homomorphic CCA-secure encryption [LPJY14], and KDM-CCA-secure encryption schemes [JR14] to signature schemes that are existentially unforgeable under adaptive chosen message attacks [JR13] and linearly-homomorphic structure-preserving signature schemes [LPJY13, LPJY14, KW15]. As a different example, Fauzi *et al.* [FLSZ17] combined SNARKs and QA-NIZKs for linear subspaces to construct an efficient pairing-based NIZK shuffle argument systems.

Briefly, a (pairing-based) QA-NIZK argument system for linear subspaces allows the prover to convince the verifier that a vector of group elements[3] $[\boldsymbol{y}]_\iota$ belongs to the columnspace of a fixed public matrix $[\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{n \times m}$, i.e., $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x}$ for some vector $\boldsymbol{x} \in \mathbb{Z}_p^m$. A QA-NIZK is *quasi-adaptive* in the sense that the CRS is allowed to depend on the matrix $[\boldsymbol{M}]_\iota$. One consequence of this definition is that up to now, QA-NIZKs have been only considered in the CRS model.

Kiltz and Wee [KW15] proposed two efficient QA-NIZKs, Π_{as} and Π'_{as}, for linear subspaces. Both are perfectly zero-knowledge and (quasi-adaptively) computationally sound in the CRS model under a suitable KerMDH assumption [MRV16]. Π'_{as} is more efficient, with the argument consisting of only k group elements, where k is a small security-assumption-related integer; $k = 1$ in the case of asymmetric pairings. As a drawback, Π'_{as} requires the matrix $[\boldsymbol{M}]_\iota$ to come from a witness-sampleable distribution. (See Section 3 for a definition of witness-sampleability.) Π_{as} works for any matrix distribution but has an argument that consists of $k + 1$ group elements. (Π_{as} was independently proposed by Abdalla *et al.* [ABP15] who proved its soundness under a stronger MDDH [EHK$^+$13] assumption.)

While the CRS model is widely accepted, from the practical viewpoint, a very important question is to minimize the trust needed from the creators of the CRS. There has been a recent surge in the research on this direction due to the use of SNARKs in real-life applications like cryptocurrencies. Ben-Sasson *et al.* [BCG$^+$15] constructed an efficient multi-party protocol for the creation of CRS in the specific case of succinct non-interactive zero knowledge arguments of knowledge (zk-SNARKs, [Gro10, Lip12, GGPR13, PGHR13, Gro16]); however, it assumes that at least one of the CRS creators is honest. Bellare *et al.* [BFS16] defined subversion-resistant soundness (Sub-SND) and subversion-resistant zero knowledge (Sub-ZK) for NIZKs that guarantee either soundness or zero knowledge, respectively, in the case all the creators of the CRS are subverted. In particular, Bellare *et al.* proved that it is impossible to simultaneously obtain Sub-SND and (even non-subversion-resistant) zero knowledge. On the other hand, they constructed a (non-succinct) statistically sound and computationally Sub-ZK NIZK argument system for **NP** where the Sub-ZK property relies on a knowledge assumption [Dam91].

Sub-ZK was further studied by Abdolmaleki *et al.* [ABLZ17] who defined perfect Sub-ZK[4] for zk-SNARK and proposed a Sub-ZK zk-SNARK based on Groth's (non-subversion) zk-SNARK [Gro16] that is essentially as efficient as Groth's original zk-SNARK. They also proposed a general framework to achieve Sub-ZK by constructing a (public) CRS-verification algorithm CV. Essentially, CV accepts the given CRS crs iff crs is correctly computed starting from *some* simulation trapdoor ts. In the Sub-ZK proof of their SNARK, Abdolmaleki *et al.* constructed a simulator that, given crs as the input, first uses a reasonable knowledge assumption BDH-KE to recover ts and after that simulates the behaviour of the prover as in Groth's non-subversion zk-SNARK. Importantly, both the honest prover and the simulator abort given a malformed CRS.

For the knowledge assumption to be usable and for the simulator (and the prover) to be able to decide whether the CRS is malformed, Abdolmaleki *et al.* added extra elements to the CRS which forced them to reprove the soundness of the zk-SNARK in the *Subversion* Generic Bilinear Group Model (Sub-GBGM). Sub-GBGM is a modification of the GBGM [Nec94, Sho97, Mau05], proposed by Bellare *et al.* [BFS16] (who called it *generic group model with hashing into the group*), where the generic adversary is given additional power to create group elements without knowing their discrete logarithms by hashing into an elliptic curve, [Ica09,

[3] We assume pairing-based setting, and use the bracket notation of [EHK$^+$13]. See Section 2 for an explanation of the notation.

[4] We note that since in [ABLZ17], the proof of zero knowledge property involves a knowledge extractor and knowledge extractors are never perfect, their notion corresponds to statistical Sub-ZK. One can define perfect Sub-ZK by requiring that the simulation is perfect whenever the extractor succeeds. In the current paper, to simplify the definitions, we deal with statistical Sub-ZK.

BCI$^+$10, TK17]. Recall that the SNARK of [Gro16] is proven knowledge-sound in the GBGM. See Section 2 for an explanation why Sub-GBGM is a weaker model than the GBGM.

Fuchsbauer [Fuc18] used a similar approach to define another Sub-ZK version of Groth's SNARK that uses a slightly different knowledge assumption, different simulation, and does not require one to add elements to the CRS. Thus, essentially, one obtains Sub-ZK for free. Because of that, it seems that there is no reason to construct and deploy SNARKs that do *not* achieve Sub-ZK. A natural question to ask is if the same holds in the case of (known) QA-NIZKs.

The knowledge assumptions of [ABLZ17, Fuc18] use crucially the fact that for each trapdoor element α, the CRS of Groth's zk-SNARK (but also other well-known zk-SNARKs like [GGPR13, PGHR13]) contains $[\alpha]_\iota$ together with some other α-dependent group elements. This means that these knowledge assumptions (that state that an adversary, who outputs $[\alpha]_\iota$ and some other well-chosen well-formed α-dependent group elements, knows α) are trivially secure in the Sub-GBGM. Due to the known impossibility results [GW11], one needs to use non-falsifiable assumptions (e.g., knowledge assumptions) to prove adaptive soundness of SNARKs and SNARGs. Thus, the additional use of knowledge assumptions to prove the Sub-ZK property does not seem to be "too strong" since non-falsifiable assumptions are needed anyhow to prove knowledge-soundness.

In the case of QA-NIZKs, the situation is different. First, known QA-NIZKs have a very different structure compared to known SNARKs. For example, the Kiltz-Wee QA-NIZKs have a trapdoor matrix $\boldsymbol{K}$ but $[\boldsymbol{K}]_\iota$ is not explicitly given in the CRS. (In fact, the soundness proof of some of their QA-NIZKs relies on the fact that $\boldsymbol{K}$ is ambiguous.) This means that the techniques of [ABLZ17, Fuc18] cannot be directly translated to the case of (Kiltz-Wee) QA-NIZK. In particular, one seems to need quite different knowledge assumptions.

Second, the definition of QA-NIZKs involves the language parameter ϱ that has to be modelled separately from other inputs; no such parameter exists in the case of SNARKs. The most important difference is however that the soundness of existing efficient QA-NIZKs like [JR13, LPJY14, JR14, ABP15, KW15] is based on standard falsifiable assumptions like KerMDH. Thus, intuitively, the use of non-falsifiable assumptions to prove Sub-ZK of a QA-NIZK seems to be less justifiable than in the case of proving Sub-ZK of zk-SNARKs. Moreover, while Bellare *et al.* had a discussion motivating the use of knowledge assumptions to obtain Sub-ZK, they did not have a formal proof of their necessity.

This brings us to the main questions of this work:

– *Are knowledge assumptions or other non-black-box techniques needed to prove Sub-ZK of* NIZK*s for languages outside of* **BPP***?*
– *In particular, can one easily modify existing* QA-NIZK*s for linear subspaces to obtain Sub-ZK?*
– *If so, can it be done by using black-box techniques only?*
– *If not, can one prove that black-box techniques are insufficient?*
– *Can one, similarly to* SNARK*s, get a Sub-ZK* QA-NIZK *for free?*

Our Contributions. We answer to the above main questions (with yes, yes, no, yes, and mostly yes). It turns out that achieving Sub-ZK for state-of-the-art QA-NIZKs is considerably more complicated than for state-of-the-art SNARKs. This follows partially from the nature of QA-NIZKs (e.g., we show that the language parameter ϱ and the CRS behave very differently if one cannot trust the CRS creator; since state-of-the-art SNARKs have no ϱ, this issue does not exist for SNARKs) and from the construction of the concrete QA-NIZK. However, in the most relevant case ($k = 1$), it turns out that the most efficient existing QA-NIZK by Kiltz and Wee [KW15] is Sub-ZK under a novel knowledge assumption given a suitable CV algorithm. Hence, Sub-ZK in this case comes for free.

First, we make a conceptually important observation that Sub-ZK in the CRS model, as defined in [BFS16, ABLZ17, Fuc18], is equal to *no-auxiliary-string non-black-box* zero knowledge (called *nonuniform* zero knowledge in [Wee07]) in the BPK model. This important connection was missed in the previous work on Sub-ZK; we hope it will make it easier to construct and analyse Sub-ZK argument systems including both SNARKs and QA-NIZKs (or their combinations, see [FLSZ17]).

We recall that in the BPK model, only the verifier needs to store her public key and the key authority executes the functionality of an immutable bulletin board by storing the received public keys. In particular, one achieves designated-verifier zero knowledge[5] by using the verifier's own public key and transferable non-interactive zero knowledge by using the public key of a (trusted-by-many-verifiers) third party. (In the latter case, the public key can be generated by using multi-party computation.)

Since in the BPK model, auxiliary-string non-black-box NIZK for languages not in **BPP** is impossible [Wee07], one can only construct no-auxiliary-string non-black-box (i.e., nonuniform) NIZK. In Section 3, we carefully define the security of $\mathsf{QA\text{-}NIZK}$ arguments in the BPK model, following standard $\mathsf{QA\text{-}NIZK}$ definitions. However, we model the definition of nonuniform NIZK after the Sub-ZK definition of Abdolmaleki *et al.* [ABLZ17]. More precisely, we require that for any efficient malicious public-key creator (either the verifier or a third party) Z, there exists an efficient extractor $\mathsf{Ext}_{\mathsf{Z}}$, such that if Z, by using the language parameter ϱ and any random coins r as an input, generates a public key pk (since there is no auxiliary input, pk *has* to be generated by Z) then $\mathsf{Ext}_{\mathsf{Z}}$, given the same input and r, outputs the secret key sk corresponding to pk.

We emphasize that Z obtains ϱ as an input (from a fixed distribution $\mathcal{D}_{\mathsf{p}}$) instead of generating it. This is to be expected since a $\mathsf{QA\text{-}NIZK}$ argument system is defined for a fixed distribution $\mathcal{D}_{\mathsf{p}}$ of ϱ. In their seminal paper, Jutla and Roy [JR13] explicitly say that ϱ should be created by a trusted third party. Moreover, as we will show in Section 5, achieving an intuitively correct level of privacy will be impossible otherwise. In particular, if the malicious public key generator leaks $\boldsymbol{M}$ either to a malicious verifier or even to the extractor (via a knowledge assumption; this seems to be a novel consideration), the intuitive definition of privacy will be breached. More formally, we will assume that $\mathcal{D}_{\mathsf{p}}$ is trusted to not leak information and also works as a black-box (that is, one cannot obtain any extra information about ϱ even when using a knowledge assumption); however, $\mathcal{D}_{\mathsf{p}}$ does not have to generate ϱ from the correct distribution. On the other hand, the rest of the $\mathsf{QA\text{-}NIZK}$ public key pk can be fully subverted. Since in many $\mathsf{QA\text{-}NIZK}$ applications, ϱ is the public key of one of the parties (and the secrecy of the corresponding witness is in the interest of the creator of the public key), this assumption usually makes sense. This result, albeit being somewhat negative, further clarifies the distinction between the language parameter ϱ and the $\mathsf{QA\text{-}NIZK}$ public key pk. Since this distinction is the difference between $\mathsf{QA\text{-}NIZK}$s and (adaptive) NIZKs, it is perhaps not surprising that ϱ and pk need to be handled differently. We also note that because $\mathcal{D}_{\mathsf{p}}$ is a part of $\mathsf{QA\text{-}NIZK}$ definition, ϱ is *not* an auxiliary string. See Sections 3 and 5 for further discussion.

As the second main contribution of the current paper, we study the Kiltz-Wee $\mathsf{QA\text{-}NIZK}$ Π'_{as} [KW15] (that we denote as Π_{kw}) in the BPK model. We consider two different variants of Π_{kw}, and prove their computational soundness and statistical nonuniform zero knowledge property in the BPK model albeit under different knowledge assumptions. We emphasize that we chose to analyse Π'_{as} since it is the most efficient known $\mathsf{QA\text{-}NIZK}$ for linear subspaces. We will leave analysing other $\mathsf{QA\text{-}NIZK}$s (that will hopefully be easier to do following our definitional framework and analysis of Π'_{as}) to the further work.

Kiltz and Wee [KW15] proved that Π_{kw} is perfectly zero knowledge in the CRS model. We show that Π_{kw} is statistically nonuniform zero knowledge in the BPK model under either one of the two new knowledge assumptions $\mathsf{KW\text{-}KE}$ (the *Kiltz-Wee Knowledge of Exponent* assumption) and $\mathsf{sKW\text{-}KE}$ (the *strong Kiltz-Wee Knowledge of Exponent* assumption), assuming that its whole CRS (or rather in this case, a public key pk) is generated by the verifier or a verifier-trusted authority — even if we are set to prove nonuniform zero knowledge that interests the prover.

We achieve this similarly to Abdolmaleki *et al.* [ABLZ17] by allowing V (or her trustee) to construct pk and then designing a public-key verification algorithm PKV. Since we do not modify the public-key generation and the prover, the (non-subversion) soundness of Π_{kw} in the BPK model follows directly from [KW15]. We also prove that Π_{kw} is nonuniform zero knowledge in the BPK model under a new knowledge assumption $\mathsf{KW\text{-}KE}$. While $\mathsf{KW\text{-}KE}$ is a strong assumption, it is weaker than just assuming the Sub-GBGM or even the GBGM. We show that, assuming $k = 1$, the $\mathsf{KW\text{-}KE}$ assumption holds in the Sub-GBGM (see Theorem 1). The proof of Theorem 1 heavily depends on the fact that we work in the Sub-GBGM and is quite intricate.

[5] More precisely, a *publicly verifiable* variation of designated-verifier where one does not need to know the secret key to verify as opposed to say [DFN06]. The verifier only needs to be sure that the prover does not know the secret key.

Table 1. Comparison of Kiltz-Wee nonuniform QA-NIZK variants in the BPK model

Prot.	k	SND assumpt.	ZK assumpt.	$\lvert\mathsf{pk}\rvert$ group elements	PKV pairings	P exp.	V pair.
Π_{kw}	1	KerMDH	sKW-KE or KW-KE	$mn+m+n+1$	$mn+m$	m	$n+1$
$\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$	2	SKerMDH	BDH-KE	$mn+2m+4n+8$	$2mn+4m+4n+10$	$2m$	$2n+4$

Interestingly, under KW-KE we only get the guarantee that the part $\mathsf{pk}^{\mathsf{zk}}$ of the pk, used either by the prover or the simulator[6], has been correctly computed. (Interestingly, this shows that in the case of QA-NIZKs, nonuniform zero knowledge can be achieved even if the correctness of the whole public key cannot be verified.) This however suffices to prove zero knowledge of Π_{kw}. Importantly, this means that in the case $k = 1$ one can get Sub-ZK for free.

Second, we show that under a stronger knowledge assumption sKW-KE, one can guarantee that the whole pk has been correctly computed. However, as an (unexpected) drawback, the sKW-KE assumption holds in the Sub-GBGM only if the language parameter $[\boldsymbol{M}]_\iota$ comes from a suitable hard distribution. (The latter is often the case in QA-NIZK applications, where $[\boldsymbol{M}]_\iota$ is a public key of some cryptographic primitive like an encryption or commitment scheme.) In both cases, the soundness is guaranteed by a KerMDH assumption.

The previous two zero knowledge proofs require that $k = 1$, where k is a security parameter related to the matrix distribution in KerMDH. Since there are also applications where one is interested in setting $k = 2$ (e.g., when one wants to rely on a weaker assumption), we also propose a slightly less efficient variant $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ of the Kiltz-Wee QA-NIZK that is secure when $k = 2$. Essentially, $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ has a few additional elements in the verifier's public key (more precisely, we duplicate some public key elements to both source groups), and the corresponding PKV algorithm checks the correctness of the new public key including the added elements. We then prove that $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ is (under a very natural knowledge assumption BDH-KE known to be Sub-GBGM secure [ABLZ17]) statistically nonuniform zero knowledge. We prove $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ is sound under the SKerMDH assumption of [GHR15]. Thus, in the case $k = 2$, Sub-ZK does not come for free. However, this is expected: the case $k = 2$ is mostly used with symmetric pairings; since we use asymmetric pairings, it is natural that some of the public-key elements have to be duplicated. See Table 1 for the comparison of all mentioned results.

In Section 5, we will provide a thorough discussion about the case ϱ is created by a malicious party. We will show that in this case, one has to be extra careful, in particular, to not employ too strong knowledge assumptions. (The latter insight is, up to our knowledge, novel.)

Finally, nonuniform NIZK argument systems in the BPK model — this includes the QA-NIZK of the current paper and the SNARKs of [ABLZ17,Fuc18] — can be made *black-box zero knowledge* in the stronger Registered Public Key (RPK, [BCNP04]) model by requiring that the key registration authority creates all the secret keys. In the simulation, the simulator Sim emulates the key registration authority and thus will know the secret keys. (Recall in the BPK model we relied on a knowledge assumption to extract these keys.) Alternatively, the verifier can create the public key but then prove its knowledge to the authority in (standalone) interactive zero knowledge in the standard model [BCNP04]. In this case, in the (standalone) simulation, Sim rewinds the verifier to obtain all the secret keys.

The way we use the BPK model is non-standard and one may argue that it is closer to the RPK model due to the use of no auxiliary string (which guarantees the public keys are created "in-system") and knowledge assumptions (which guarantee one can extract the secret keys). In our opinion, there is a big difference between the used BPK model and the RPK model since here, a prover can detect whether using

[6] Recall Π_{kw} is a *split-CRS* QA-NIZK [JR13], meaning that (a) one part of its CRS/public key ($\mathsf{pk}^{\mathsf{zk}}$) is used only by the prover (and the simulator) and another part ($\mathsf{pk}^{\mathsf{snd}}$) only by the verifier, and (b) $\mathsf{pk}^{\mathsf{snd}}$ does not depend on ϱ.

the verifier's public key can breach the zero-knowledge property. Hence, we do not assume malformed public keys will be rejected by honest key registration authorities and thus do not rely on a trust in the latter.[7]

2 Preliminaries

Let PPT denote probabilistic polynomial-time. Let $\lambda \in \mathbb{N}$ be the security parameter. All adversaries will be stateful. For an algorithm $\mathcal{A}$, let $\mathrm{im}(\mathcal{A})$ be the image of $\mathcal{A}$ (the set of of valid outputs of $\mathcal{A}$), let $\mathsf{RND}(\mathcal{A})$ denote the random tape of $\mathcal{A}$, and let $r \leftarrow_{\$} \mathsf{RND}(\mathcal{A})$ denote the random choice of the randomizer r from $\mathsf{RND}(\mathcal{A})$. By $y \leftarrow \mathcal{A}(x; r)$ we denote the fact that $\mathcal{A}$, given an input x and a randomizer r, outputs y. When we use this notation then r represents the full random tape of $\mathcal{A}$. For algorithms $\mathcal{A}$ and $\mathsf{Ext}_{\mathcal{A}}$, we write $(y \,\|\, y') \leftarrow (\mathcal{A} \,\|\, \mathsf{Ext}_{\mathcal{A}})(x; r)$ as a shorthand for $y \leftarrow \mathcal{A}(x; r)$, $y' \leftarrow \mathsf{Ext}_{\mathcal{A}}(x; r)$. Importantly, $\mathsf{Ext}_{\mathcal{A}}$ and $\mathcal{A}$ use the same randomizer r. 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. We denote by $\mathsf{negl}(\lambda)$ an arbitrary negligible function. We write $a \approx_\lambda b$ if $|a - b| \leq \mathsf{negl}(\lambda)$. We follow Bellare *et al.* [BFS16] by using "cryptographic" style in security definitions where all complexity (adversaries, algorithms, assumptions) is uniform but the adversary and the security (say, soundness) is quantified over all inputs chosen by the adversary. See [BFS16] for a discussion.

A bilinear group generator $\mathsf{Pgen}(1^\lambda)$ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic groups of prime order $p = 2^{\Omega(\lambda)}$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate PPT computable bilinear pairing. We assume the bilinear pairing to be Type-3 [GPS08], i.e., that there is no efficient isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$. We use the bracket notation of [EHK+13], i.e., we write $[a]_\iota$ to denote $a g_\iota$ 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 freely use the bracket notation with matrices, e.g., if $\boldsymbol{AB} = \boldsymbol{C}$ then $\boldsymbol{A}[\boldsymbol{B}]_\iota = [\boldsymbol{C}]_\iota$ and $[\boldsymbol{A}]_1[\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$.

Bare Public Key (BPK) Model. In the BPK model [CGGM00,MR01], parties have access to a public file F, a polynomial-size collection of records (id, pk_{id}), where id is a string identifying a party (e.g., a verifier), and pk_{id} is her (alleged) public key. In a typical zero-knowledge protocol in the BPK model, a key-owning party $\mathcal{P}_{id}$ works in two stages. In stage one (the *key-generation stage*), on input a security parameter 1^λ and randomizer r, $\mathcal{P}_{id}$ outputs a public key pk_{id} and stores the corresponding secret key sk_{id}. We assume the *no-auxiliary-string BPK* model where from this it follows that $\mathcal{P}_{id}$ actually created pk_{id}. After that, the public file F will include (id, pk_{id}). In stage two, each party has access to F, while $\mathcal{P}_{id}$ has possibly access to sk_{id} (however, the latter will be not required in the current paper). It is commonly assumed that only the verifier of a NIZK argument system in the BPK model has a public key [MR01]; see also Section 3.

There are several well-known impossibility results about zero knowledge the BPK model. Alwen *et al.* [APV05] proved that any black-box concurrent zero-knowledge argument system satisfying sequential soundness in the BPK model for a language $\mathcal{L}$ outside of **BPP** requires at least 4 rounds. Wee [Wee07] noted that there exists no *auxiliary-string non-black-box* NIZK argument system in the BPK model for a language $\mathcal{L}$ outside of **BPP**. (This explains our reliance on the no-auxiliary-string BPK model.) These results are complemented by a possibility result of Micali and Reyzin [MR01], who proved that if there exist certified trapdoor permutation families secure against subexponentially-strong adversaries then there exists a 4-round black-box resettable zero knowledge protocol, for any $\mathcal{L} \in \mathbf{NP}$, in the BPK model. (See also [SV12].) Here, we recall that resettable zero knowledge is strictly stronger than concurrent zero knowledge, [MR01]. Finally, Wee [Wee07] showed the existence of *weak* (where essentially, the size of the simulator can depend on the size of the distinguisher and of the distinguishing gap) nonuniform NIZK argument systems for **NP** in the BPK model, assuming *subexponential* hardness results; see [Wee07] for a precise statement.

Matrix Diffie-Hellman Assumptions. Kernel Matrix Diffie-Hellman Assumption (KerMDH) is a well-known assumption family formally introduced in [MRV16] and used by Kiltz and Wee in [KW15] to prove soundness of their $\mathsf{QA\text{-}NIZK}$ argument systems for linear subspaces. Informally, a $\mathcal{D}$-KerMDH assumption

[7] Citing [SV12]: "The BPK model is very close to the standard model, indeed the proof phase does not have any requirement beyond the availability of the directory to all provers, and for verifiers, of **a secret key** associated to their identities." and [MR01]: "It suffices for PK to be a string known to the prover, and **chosen** by the verifier prior to any interaction with him."

states that for a matrix $\boldsymbol{A}$ sampled from the distribution $\mathcal{D}$ it is difficult to find a representation of a vector that belongs to the kernel of $\boldsymbol{A}^\top$ provided that the matrix is given in exponents only, i.e., as $[\boldsymbol{A}]_\iota$.

More precisely, let $\mathcal{D}_{\ell k}$ be a probability distribution over matrices in $\mathbb{Z}_p^{\ell \times k}$, where $\ell > k$. We 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*, [JR13].) We denote the lower $(\ell - k) \times k$ submatrix of $\boldsymbol{A}$ as $\underline{\boldsymbol{A}}$. When $\ell = k + 1$, let $\mathcal{D}_k = \mathcal{D}_{\ell k}$.

$\mathcal{D}_{\ell k}$-$\mathsf{KerMDH}_{\mathbb{G}_\iota}$ [MRV16] holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{kermdh}}_{\mathcal{A},\mathcal{D}_{\ell k},\iota,\mathsf{Pgen}}(\lambda) := \Pr\left[\begin{array}{l}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow_\$ \mathcal{D}_{\ell k}; \\ [\boldsymbol{c}]_{3-\iota} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_\iota) : \boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_k \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell\end{array}\right] \approx_\lambda 0 \ .$$

$\mathcal{D}_{\ell k}$-$\mathsf{SKerMDH}$ [GHR15] holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{skermdh}}_{\mathcal{A},\mathcal{D}_{\ell k},\mathsf{Pgen}}(\lambda) :=$

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

According to Lem. 1 of [GHR15], if $\mathcal{D}_{\ell k}$-KerMDH holds in generic symmetric bilinear groups then $\mathcal{D}_{\ell k}$-$\mathsf{SKerMDH}$ holds in generic asymmetric bilinear groups. KerMDH assumption can hold also for Type-1 pairings, where $\mathbb{G}_1 = \mathbb{G}_2$, but then one needs $k \geq 2$, which affects efficiency of the arguments relying on KerMDH.

Bilinear Diffie-Hellman Knowledge of Exponent Assumption. $\mathsf{BDH\text{-}KE}$ [DFGK14, ABLZ17] holds relative to Pgen if $\forall \mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$ and PPT adversary $\mathcal{A}$ there exists a PPT extractor $\mathsf{Ext}_\mathcal{A}$, such that

$$\Pr\left[\begin{array}{l}r \leftarrow_\$ \mathsf{RND}(\mathcal{A}), ([\alpha_1]_1, [\alpha_2]_2 \parallel a) \leftarrow (\mathcal{A} \parallel \mathsf{Ext}_\mathcal{A})(\mathsf{p}; r) : \\ [\alpha_1]_1 [1]_2 = [1]_1 [\alpha_2]_2 \wedge a \neq \alpha_1\end{array}\right] \approx_\lambda 0 \ .$$

Since with the (negligible) probability $\mathsf{poly}(\lambda)/p$, a PPT $\mathcal{A}$ can output correct $([\alpha_1]_1, [\alpha_2]_2)$ such that $[\alpha_1]_1 [1]_2 = [1]_1 [\alpha_2]_2$ by repeated uniform sampling, $\mathsf{BDH\text{-}KE}$ (like most knowledge assumptions) is never perfect.

Generic Bilinear Group Model. In the *Generic Bilinear Group Model* (GBGM) [Nec94, Sho97, Mau05, BBG05], one assumes that the adversary has only access to group elements via generic bilinear-group operations (group operations and the bilinear map) together with an equality test. In the *subversion GBGM* (Sub-GBGM, [BFS16, ABLZ17]; named *generic group model with hashing into the group* in [BFS16]), the adversary has an additional power of creating new indeterminates in bilinear group. The Sub-GBGM is motivated by the existence of elliptic curve hashing algorithms [Ica09, BCI$^+$10, TK17] that allow one to efficiently create elliptic-curve group elements without knowing their discrete logarithms.

Thus, Sub-GBGM is a weaker model than GBGM. As an important example, knowledge assumptions that state that the output group element must belong to the span of input group elements hold in the GBGM but not in the Sub-GBGM. This is since in the Sub-GBGM, the adversary can create new group elements without knowing their discrete logarithms; indeed the output element might be equal to one such created group elements. Hence, a Sub-GBGM adversary is less restricted than a GBGM adversary. Moreover, as we will see later (see Theorem 1), some knowledge assumptions that have a trivial security poof in the GBGM have quite a complicated proof in the Sub-GBGM.

See Appendix A for a long introduction to GBGM and Sub-GBGM.

3 Defining QA-NIZK in the BPK Model

Quasi-adaptive Non-Interactive Zero-Knowledge (QA-NIZK) argument systems [JR13] are quasi-adaptive in the sense that the CRS depends on a language parameter ϱ that has been sampled from a fixed distribution $\mathcal{D}_\mathsf{p}$. QA-NIZKs are of great interest since they are succinct and based on standard assumptions. Since QA-NIZKs have many applications, they have been a subject of intensive study, [JR13, LPJY14, JR14, ABP15, KW15, LPJY15, GHR15]. The main limitation of known QA-NIZKs is that they are only known for a restricted set of languages like the language of linear subspaces (although see [GHR15, GR16] for $\mathsf{QA\text{-}NIZK}$s for other languages).

The original $\mathsf{QA\text{-}NIZK}$ security definitions, [JR13], were given in the CRS model. In what follows, we will lift them to the weaker BPK model. In some of the cases, the only difference compared to the original definitions is in notation (a CRS will be replaced by a public key). The rest of the definitional changes are motivated by the definition of Sub-ZK $\mathsf{zk\text{-}SNARK}$s in [ABLZ17], e.g., a $\mathsf{QA\text{-}NIZK}$ in the BPK model will have a public-key verification algorithm PKV and the zero knowledge definition mentions a subverter and an extractor. Since black-box [MR01, APV05] and even auxiliary-input non-black-box [Wee07] NIZK in the BPK model is impossible we will give an explicit definition of no-auxiliary-string non-black-box NIZK (or, more precisely, *nonuniform* NIZK [Wee07]).

As in [BFS16], we will implicitly assume that the system parameters p are generated deterministically from λ; in particular, the choice of p cannot be subverted. A $\mathsf{QA\text{-}NIZK}$ argument system enables to prove membership in a language defined by a relation $\mathcal{R}_\varrho = \{(\varrho, w_\varrho)\}$, which in turn is completely determined by a parameter ϱ sampled from a distribution $\mathcal{D}_\mathsf{p}$.[8] In the proof of zero knowledge, we will assume that $\mathcal{D}_\mathsf{p}$ works as a black box and one cannot obtain from it any secret keys. As noted by Jutla and Roy [JR13], one needs to assume that $\mathcal{D}_\mathsf{p}$ is reasonable; for example, it should not be the case that all languages $\mathcal{L}_\varrho$ for $\varrho \in \mathcal{D}_\mathsf{p}$ are easy to decide. (See additional discussion at the end of the current section and in Section 5.) We will assume implicitly that ϱ contains p and thus not include p as an argument to algorithms that also input ϱ. A distribution $\mathcal{D}_\mathsf{p}$ on $\mathcal{L}_\varrho$ is *witness-sampleable* [JR13] if there exists a PPT algorithm $\mathcal{D}'_\mathsf{p}$ that samples $(\varrho, w_\varrho) \in \mathcal{R}_\varrho$ such that ϱ is distributed according to $\mathcal{D}_\mathsf{p}$, and membership of ϱ in *the parameter language* $\mathcal{L}_\varrho$ can be verified in PPT given w_ϱ.

While the verifier's public key pk may depend on ϱ (however, we assume that ϱ was not created by the verifier), the zero-knowledge simulator is usually required to be a single (non-black-box) PPT algorithm that works for the whole collection of relations $\mathcal{R}_\mathsf{p} = \{\mathcal{R}_\varrho\}_{\varrho \in \mathrm{Supp}(\mathcal{D}_\mathsf{p})}$; that is, one usually requires *uniform simulation* (see [JR13] for a discussion). We however accompany the universal simulator with an adversary-dependent extractor. The simulator is not allowed to create new ϱ but has to operate with one given to it as an input.

A tuple of PPT algorithms $\Pi = (\mathsf{Pgen}, \mathsf{K}, \mathsf{PKV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ is a *nonuniform zero knowledge* $\mathsf{QA\text{-}NIZK}$ *argument system* in the BPK 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 generation algorithm, K is the public key generation algorithm, PKV is the public key verification algorithm, P is the prover, V is the verifier, and Sim is the simulator.

(i) **Perfect Completeness:** $\forall \lambda$, $\mathsf{p} \in \mathsf{Pgen}(1^\lambda)$, $\varrho \in \mathcal{D}_\mathsf{p}$, and $(x, w) \in \mathcal{R}_\varrho$,

$$\Pr\left[\begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{K}(\varrho); \pi \leftarrow \mathsf{P}(\varrho, \mathsf{pk}, x, w) : \\ \mathsf{PKV}(\varrho, \mathsf{pk}) = 1 \wedge \mathsf{V}(\varrho, \mathsf{pk}, x, \pi) = 1\end{matrix}\right] = 1 \enspace .$$

(ii) **Computational Quasi-Adaptive Soundness:** $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{qasound}}_{\mathcal{A},\Pi}(\lambda) :=$

$$\Pr\left[\begin{matrix}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \varrho \leftarrow_\$ \mathcal{D}_\mathsf{p}; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{K}(\varrho); \\ (x, \pi) \leftarrow \mathcal{A}(\varrho, \mathsf{pk}) : \mathsf{V}(\varrho, \mathsf{pk}, x, \pi) = 1 \wedge \neg(\exists w : \mathcal{R}_\varrho(x, w))\end{matrix}\right] \approx_\lambda 0 \enspace .$$

(iii) **Statistical Nonuniform Zero Knowledge:** for any PPT subverter Z there exists a PPT Ext_Z, such that $\forall \lambda$, $\forall \mathsf{p} \in \mathsf{Pgen}(1^\lambda)$, $\varrho \in \mathcal{D}_\mathsf{p}$, and computationally unbounded $\mathcal{A}$, $\varepsilon_0^{zk} \approx_\lambda \varepsilon_1^{zk}$, where

$$\varepsilon_b^{zk} = \Pr\left[\begin{matrix}r \leftarrow_\$ \mathsf{RND}(\mathsf{Z}); (\mathsf{pk}, \mathsf{aux}_\mathsf{Z} \,\|\, \mathsf{sk}) \leftarrow (\mathsf{Z} \,\|\, \mathsf{Ext}_\mathsf{Z})(\varrho; r) : \\ \mathsf{PKV}(\varrho, \mathsf{pk}) = 1 \wedge \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\varrho, \mathsf{pk}, \mathsf{aux}_\mathsf{Z}) = 1\end{matrix}\right] \enspace .$$

[8] In the $\mathsf{QA\text{-}NIZK}$ literature, it is assumed that samples from $\mathcal{D}_\mathsf{p}$ are generated by a trusted third party (TTP), see [JR13] for a discussion. For example, in the case of the language $\mathcal{L} = ([1]_1, [x]_1, [y]_1, [xy]_1)$ of DDH tuples, $[x]_1$ is created by the TTP. Instead of TTP, one can have a protocol participant who has self-interest in choosing ϱ securely and not leak corresponding secret.

Here, the oracle $\mathsf{O}_0(x, w)$ returns $\perp$ (reject) if $(x, w) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{P}(\varrho, \mathsf{pk}, x, w)$. Similarly, $\mathsf{O}_1(x, w)$ returns $\perp$ (reject) if $(x, w) \notin \mathcal{R}_\varrho$, and otherwise it returns $\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, x)$.

As mentioned before (see the definition of BDH-KE in Section 2), the extractor defined by a knowledge assumption never works with probability 1. However, if it works then in our constructions the simulation will be perfect. For the sake of simplicity, we will not formalize this as perfect zero knowledge. (One reason for this is that is that differently from [ABLZ17], the secret key extracted by Ext_Z is not unique in our case, see discussion in Section 4.)

The existence of PKV is not needed in the CRS model, assuming the CRS creator is trusted by the prover, and thus PKV was not included in the prior art QA-NIZK definitions. Since soundness is proved in the case pk is chosen correctly (by the verifier or a trusted third party, trusted by her), V does not need to execute PKV. However, PKV should be run by P. The simulator is only required to correctly simulate in the case PKV accepts pk.

On Sub-ZK versus Nonuniform Zero Knowledge in the BPK model. Subversion-security was defined by Bellare *et al.* [BFS16] for the CRS model, and further CRS-model subversion-security definitions were given in [ABLZ17, Fuc18]. As proven in [BFS16], one cannot achieve Sub-SND (soundness even if the CRS was generated maliciously) and zero knowledge at the same time. Thus, subsequent efforts have concentrated on achieve either Sub-SND and witness-indistinguishability [BFS16], subversion knowledge-soundness and witness-indistinguishability [FO18], or Sub-ZK (zero knowledge in the case the CRS was generated maliciously) and soundness, [BFS16, ABLZ17, Fuc18]. In the latter case, the CRS is trusted by the verifier V while (following the definitions of [ABLZ17]) the prover checks that the CRS is well-formed by using a publicly available algorithm. Thus, Sub-ZK in the CRS model is the same as zero knowledge in the BPK model: the CRS has to be trusted by (or, even chosen by) V and hence can be equal to the public key of an entity trusted by V (or of V herself). Since black-box NIZK [MR01] and even auxiliary-input non-black-box NIZK [Wee07] in the BPK model is impossible, one has to define nonuniform zero knowledge as above. [9] In particular, the main result of [ABLZ17, Fuc18], reformulated in our language, is that there exist computationally knowledge-sound nonuniform zero knowledge zk-SNARKs for **NP** in the BPK model.

Finally, Wee's definition of nonuniform zero knowledge [Wee07] is slightly different from ours. First, it allows the simulator to depend nonuniformly on the cheating verifier, while we have a universal simulator coupled with an extractor where only the extractor depends nonuniformly on the cheating subverter. This change is motivated by the prior definitions of Sub-ZK by Abdolmaleki *et al.* [ABLZ17] and the standard requirement that QA-NIZKs have a universal simulator [JR13]. Second, we do not require explicitly that there is a polynomial relation between the size of the subverter and that of the extractor although this is implicit due to polynomial dependence on the common security parameter.

Language of linear subspaces. An important application of QA-NIZK is in the case of the following language. Assume we need to show that $[\boldsymbol{y}]_\iota \in \mathrm{im}([\boldsymbol{M}]_\iota)$, where $[\boldsymbol{M}]_\iota$ is sampled from a distribution $\mathcal{D}_\mathsf{p}$ over $\mathbb{G}_\iota^{n \times m}$. We assume, following [JR13], that (n, m) is implicitly fixed by $\mathcal{D}_\mathsf{p}$. That is, a QA-NIZK for linear subspaces handles languages $\mathcal{L}_{[\boldsymbol{M}]_\iota}$ defined as follows:

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

The corresponding relation is defined as $\mathcal{R}_{\boldsymbol{M}} = \{([\boldsymbol{y}]_\iota, \boldsymbol{w}) \in \mathbb{G}_\iota^n \times \mathbb{Z}_p^m : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}\}$. This language is surprisingly useful in many applications, [JR13]. As a typical application, let $[\boldsymbol{M}]_\iota = [1, \mathsf{sk}]_\iota^\top$ be a public key of the Elgamal cryptosystem; then a ciphertext $[\boldsymbol{y}]_\iota \in \mathcal{L}_{[\boldsymbol{M}]_\iota}$ iff it encrypts 0. In this case, $[\boldsymbol{M}]_\iota$ comes from a witness-sampleable distribution $\mathcal{D}_\mathsf{p}$ with $\mathcal{D}_\mathsf{p}$-KerMDH being hard.

Kiltz-Wee QA-NIZK. The most efficient known QA-NIZK for linear subspaces in the CRS model was proposed by Kiltz and Wee [KW15]. In particular, they proposed a QA-NIZK $\mathit{\Pi}_\mathsf{kw}$ (named $\mathit{\Pi}'_\mathsf{as}$ in [KW15]) that

[9] Bellare *et al.* [BFS16] motivated not incorporating auxiliary strings to the definition of Sub-ZK by known impossibility results. Moreover, as noted also in [BFS16], auxiliary-input zero knowledge is usually used to achieve sequential composition in the case of interactive zero knowledge. The given definition of nonuniform zero knowledge guarantees sequential security in the case of NIZK, see [ABLZ17] for a proof.

```
$\mathsf{K}([\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{n\times m})$: $\boldsymbol{A} \leftarrow_\$ \mathcal{D}_k$; $\boldsymbol{K} \leftarrow_\$ \mathbb{Z}_p^{n\times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n\times k}$; $\boldsymbol{P} \leftarrow \boldsymbol{M}^\top \boldsymbol{K} \in \mathbb{Z}_p^{m\times k}$; $\mathsf{pk} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}, [\boldsymbol{P}]_\iota)$; $\mathsf{sk} \leftarrow \boldsymbol{K}$;
    Return $(\mathsf{pk}, \mathsf{sk})$;
$\mathsf{P}([\boldsymbol{M}]_\iota, \mathsf{pk}, [\boldsymbol{y}]_\iota, \boldsymbol{w})$: return $[\boldsymbol{\pi}]_\iota \leftarrow [\boldsymbol{P}]_\iota^\top \boldsymbol{w} \in \mathbb{G}_\iota^k$;
$\mathsf{Sim}([\boldsymbol{M}]_\iota, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_\iota)$: return $[\boldsymbol{\pi}]_\iota \leftarrow \boldsymbol{K}^\top [\boldsymbol{y}]_\iota \in \mathbb{G}_\iota^k$;
$\mathsf{V}([\boldsymbol{M}]_\iota, \mathsf{pk}, [\boldsymbol{y}]_\iota, [\boldsymbol{\pi}]_\iota)$ : check that $[\boldsymbol{y}]_\iota^\top [\boldsymbol{C}]_{3-\iota} = [\boldsymbol{\pi}]_\iota^\top [\bar{\boldsymbol{A}}]_{3-\iota}$;
```

Fig. 1. Kiltz-Wee argument system Π_{kw} for $[\boldsymbol{y}]_\iota = [\boldsymbol{M}]_\iota \boldsymbol{w}$

assumes that the parameter $\varrho = [\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{n\times m}$ is sampled from a witness-sampleable distribution $\mathcal{D}_{\mathsf{p}}$. Π_{kw} results in the argument that consists of k group elements, where k is the parameter ($k = 1$ being usually sufficient in the case of asymmetric pairings) related to the underlying KerMDH distribution. In particular, Π_{kw} is significantly more efficient than the Groth-Sahai NIZK [GS08] for the same language.

For the sake of completeness, Fig. 1 depicts the original Kiltz-Wee QA-NIZK for linear subspaces in the CRS model. Given $n > m$, the Kiltz-Wee QA-NIZK is computationally quasi-adaptively sound under the $\mathcal{D}_k$-$\mathsf{KerMDH}_{\mathbb{G}_\iota}$ assumption relative to Pgen, [KW15].

Discussion: creation of the language parameter. When introducing QA-NIZKs in the CRS model, Jutla and Roy [JR13] claimed that in most of the applications, ϱ is set by a trusted third party. For example, ϱ could be his public key. As also argued by Jutla and Roy, in many applications, that party has no motivation to cheat while generating ϱ since the security is defined with respect to this key. They mention that if ϱ is created say by the prover, then he should as minimum at least prove that $\varrho \in \mathcal{D}_{\mathsf{p}}$.

Now, consider the BPK model definitions of the current paper where pk might be generated by malicious Z. In this case, Z should not generate ϱ, partially since a QA-NIZK argument system is defined for a fixed distribution of ϱ and partially due to simple attacks that become possible if Z just leaks ϱ. We provide thorough discussion on this in Section 5, just noting here that since ϱ is sampled from $\mathcal{D}_{\mathsf{p}}$ it means that $\mathcal{D}_{\mathsf{p}}$ has to be implemented by a trusted third party who does not leak any secret keys to Z.

The notion of QA-NIZK in the BPK model is important in the case where ϱ is not generated by the verifier but either by the prover or some (trusted) third party. In particular, recall that Kiltz-Wee proposed two different QA-NIZKs, Π_{as} (Π_{as} was independently proposed by Abdalla *et al.* [ABP15]) and Π'_{as} where the latter for its *soundness* requires $\varrho = [\boldsymbol{M}]_\iota$ to come from a witness-sampleable distribution. Hence, in the case of Π'_{as}, intuitively, $[\boldsymbol{M}]_\iota$ should be created honestly.

Applications of QA-NIZK in the BPK Model. The simplest example application is that of UC commitments from [JR13] where a trusted third party generates a commitment key ϱ together with a QA-NIZK public key pk and P opens the commitments later by disclosing a QA-NIZK argument of proper commitment under the commitment key ϱ. In this case, ϱ should not be generated by P (who could then equivocate) or by V (who could then extract the message). However, pk can be generated by V. This allows one, securely generated ϱ, to be used in many applications, from UC commitments to efficient identity-based encryption. In each such application, a trusted authority trusted by V (or V herself) can create her pk that takes the particularities of that application into account.

As another example, consider the case of wide-scale e-voting, where the public key pk_e used for encryption belongs to a designated decryption authority (say, trusted hardware; or, in a threshold manner, to authorities). In a typical e-voting protocol, the voters encrypt their ballots (w.r.t. pk_e) and prove in zero knowledge that the plaintext belongs to the set of valid candidates. Later, one can use a mixnet to shuffle (i.e., permute and rerandomize) the encrypted ballots. Each mixserver must prove that he shuffled the encrypted ballots correctly by using a CRS-model shuffle argument [GL07, LZ12, FL16, FLZ16, FLSZ17].

In all such cases, the set of verifiers includes mixservers, the decryption authority, and third-party auditors. Thus, it makes sense to have a separate trusted party (or trusted parties) who generates the public key pk_v used to verify the correctness of all arguments but by using the language parameter ϱ_e that is contained in the public key pk_e of the decryption authority. If e-voting is employed in a big scale, it makes sense that pk_e is fixed ahead of the time and made public, so that pk_v can be later created based on pk_e.

4 Kiltz-Wee QA-NIZK in the BPK Model

In this section, we will show that a minimally changed variant of the Kiltz-Wee QA-NIZK Π_{kw} is secure in the BPK model. More precisely, we assume that the public key (corresponds to the CRS in original Kiltz-Wee) belongs either to the verifier or to a party, trusted by the verifier. Hence, we prove computational soundness in the setting where the verifier trusts the public key (i.e., that the corresponding sk is secret and the CRS is well-formed). Since the public key is not trusted by the prover, we prove nonuniform zero knowledge in the case of a nontrusted (possibly subverted) public key. As motivated in Section 3, we assume that $[\boldsymbol{M}]_\iota$ is sampled honestly, i.e., from a witness-sampleable distribution and moreover, neither the verifier nor the simulator knows the corresponding witness $\boldsymbol{M}$ or any function of $\boldsymbol{M}$ not efficiently computable from $[\boldsymbol{M}]_\iota$.

To modify Π_{kw} in Fig. 1 so that it would be secure in the BPK model instead of the CRS model, the simplest idea is to divide pk into $\mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}$ (the part of pk is used by the verifier and thus intuitively needed to guarantee soundness) and $\mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_\iota$ (the part of pk that is used by the prover and thus intuitively needed to guarantee zero knowledge). Thus, the prover needs to be assured that $\mathsf{pk}^{\mathsf{zk}}$ is generated honestly and the verifier needs to be assured that $\mathsf{pk}^{\mathsf{snd}}$ is generated honestly. Hence, one could use $\mathsf{pk}^{\mathsf{zk}}_{\mathsf{P}}$ from the prover's public key and $\mathsf{pk}^{\mathsf{snd}}_{\mathsf{V}}$ from the verifier's public key to create an argument. However, it is not clear how to do this since both $\mathsf{pk}^{\mathsf{snd}}_{\mathsf{V}}$ and $\mathsf{pk}^{\mathsf{zk}}_{\mathsf{P}}$ depend on the same secret $\boldsymbol{K}$.[10] Moreover, in this case both P and V have public keys while we strive to have a situation, common in the BPK model, where only V has a public key.

In what follows, we assume that the verifier's public key is equal to the whole Kiltz-Wee CRS and then construct a public-key verification algorithm PKV for Π_{kw}. Assume $k = 1$ (this gives the best efficiency and is thus the most interesting in practice). We prove that in the BPK model, Π_{kw} is computationally quasi-adaptively sound under a KerMDH assumption and nonuniform zero knowledge under a novel knowledge assumption. In fact, we define two different knowledge assumptions, KW-KE and sKW-KE.

The assumption KW-KE guarantees that one can extract a secret key $\mathsf{sk} = \boldsymbol{K}$ from which one can compute $\mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_\iota$ (but not necessarily $\mathsf{pk}^{\mathsf{snd}}$) as in Π_{kw}. Since $\mathsf{pk}^{\mathsf{zk}}$ does not fix $\boldsymbol{K}$ uniquely, KW-KE extracts one possible $\boldsymbol{K}$. Since for achieving nonuniform zero knowledge, it is not needed that $\mathsf{pk}^{\mathsf{snd}}$ can be computed from sk, KW-KE will be sufficient. We note that KW-KE is a tautological knowledge assumption for Π_{kw}. To argue that KW-KE is a reasonable knowledge assumption, we prove that it holds in the Sub-GBGM.

We also introduce a stronger knowledge assumption sKW-KE that allows to extract the *unique* secret key $\boldsymbol{K}$ that was used to generate the whole public key pk. We prove sKW-KE holds in the Sub-GBGM given that $\varrho = [\boldsymbol{M}]_\iota$ is chosen from a hard distribution. The latter assumption is undesirable but holds often in practice, e.g., when ϱ corresponds to a randomly chosen public key of a cryptosystem or a commitment scheme (see Section 3 for an example).

After that, we will prove that Π_{kw} is nonuniform zero knowledge under either KW-KE and sKW-KE where in the latter additionally guarantees that the public key is correctly formed. Since we essentially did not modify Π_{kw} (we only defined PKV for Π_{kw}!), its completeness and computational soundness follow from [KW15].

However, the Sub-GBGM proofs for KW-KE and sKW-KE only work if $k = 1$. Since there are applications (e.g., in the setting of symmetric pairing) where one might want to use $k = 2$, we also prove that $\Pi^{\mathsf{bdh}}_{\mathsf{kw}}$, a variant of Π_{kw}, obtained after adding some elements to the public key, is sound under a SKerMDH assumption and nonuniform zero knowledge under a more standard, previously known, knowledge assumption BDH-KE [ABLZ17]. Since in the case of $\Pi^{\mathsf{bdh}}_{\mathsf{kw}}$, one has to rely on the SKerMDH and thus assume that $k = 2$, $\Pi^{\mathsf{bdh}}_{\mathsf{kw}}$ is less efficient than Π_{kw} where $k = 1$ is sufficient.

[10] Jutla and Roy [JR13] say that a QA-NIZK is *split-CRS* if (a) its CRS can be divided into soundness (used only by the verifier) and zero-knowledge part (used only by the prover) that only share common randomness, and (b) the soundness CRS does not depend on ϱ. The QA-NIZK of Jutla and Roy is split-CRS, as are some of the QA-NIZKs of Libert *et al.* [LPJY14] and of Kiltz and Wee. See [JR13] for applications of split-CRS QA-NIZKs.

```
$\mathsf{K}([\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{n\times m})$: $\boldsymbol{A} \leftarrow_\$ \mathcal{D}_k$; $\boldsymbol{K} \leftarrow_\$ \mathbb{Z}_p^{n\times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n\times k}$; $\boldsymbol{P} \leftarrow \boldsymbol{M}^\top \boldsymbol{K} \in \mathbb{Z}_p^{m\times k}$;
    if $\Pi = \Pi_{\mathsf{kw}}$ then $\mathsf{pk}^{\mathsf{pkv}} \leftarrow \epsilon$; elseif $\Pi = \Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ then $\mathsf{pk}^{\mathsf{pkv}} \leftarrow [\bar{\boldsymbol{A}}, \boldsymbol{C}]_\iota$; fi
    $\mathsf{pk}^{\mathsf{zk}} \leftarrow [\boldsymbol{P}]_\iota$; $\mathsf{pk}^{\mathsf{snd}} \leftarrow [\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}$; $\mathsf{pk} \leftarrow (\mathsf{pk}^{\mathsf{snd}}, \mathsf{pk}^{\mathsf{zk}}, \mathsf{pk}^{\mathsf{pkv}})$; $\mathsf{sk} \leftarrow \boldsymbol{K}$; return $(\mathsf{pk}, \mathsf{sk})$;
$\mathsf{P}([\boldsymbol{M}]_\iota, \mathsf{pk}, [\boldsymbol{y}]_\iota, \boldsymbol{w})$: return $[\boldsymbol{\pi}]_\iota \leftarrow [\boldsymbol{P}]_\iota^\top \boldsymbol{w} \in \mathbb{G}_\iota^k$;
$\mathsf{Sim}([\boldsymbol{M}]_\iota, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_\iota)$: return $[\boldsymbol{\pi}]_\iota \leftarrow \boldsymbol{K}^\top [\boldsymbol{y}]_\iota \in \mathbb{G}_\iota^k$;
$\mathsf{V}([\boldsymbol{M}]_\iota, \mathsf{pk}, [\boldsymbol{y}]_\iota, [\boldsymbol{\pi}]_\iota)$ : check that $[\boldsymbol{y}]_\iota^\top [\boldsymbol{C}]_{3-\iota} = [\boldsymbol{\pi}]_\iota^\top [\bar{\boldsymbol{A}}]_{3-\iota}$; // $\in \mathbb{G}_T^{1\times k}$
$\mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk})$: // $k \in \{1, 2\}$
    Check that
    1 :  $[\bar{\boldsymbol{A}}]_{3-\iota} \in \mathbb{G}_{3-\iota}^{k\times k} \wedge [\boldsymbol{C}]_{3-\iota} \in \mathbb{G}_{3-\iota}^{n\times k} \wedge [\boldsymbol{P}]_\iota \in \mathbb{G}_\iota^{m\times k} \wedge [\boldsymbol{M}]_\iota \in \mathbb{G}_\iota^{n\times m}$;
    2 :  if $\Pi = \Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ then check $[\bar{\boldsymbol{A}}]_\iota \in \mathbb{G}_\iota^{k\times k} \wedge [\boldsymbol{C}]_\iota \in \mathbb{G}_\iota^{k\times k}$; fi
    3 :  $[\boldsymbol{M}]_\iota^\top [\boldsymbol{C}]_{3-\iota} = [\boldsymbol{P}]_\iota [\bar{\boldsymbol{A}}]_{3-\iota}$;
    4 :  if $k = 1$ then check $[a_{11}]_{3-\iota} \neq [0]_{3-\iota}$// I.e., $\det \bar{\boldsymbol{A}} \neq 0$
           else check $[a_{11}]_1 [a_{22}]_2 - [a_{12}]_1 [a_{21}]_2 \neq [0]_T$; fi // I.e., $\det \bar{\boldsymbol{A}} \neq 0$
    5 :  if $\Pi = \Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ then check $[\bar{\boldsymbol{A}}]_1 [\boldsymbol{1}]_2 = [\boldsymbol{1}]_1 [\bar{\boldsymbol{A}}]_2$; fi
    6 :  if $\Pi = \Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ then check $[\boldsymbol{C}]_1 [\boldsymbol{1}]_2 = [\boldsymbol{1}]_1 [\boldsymbol{C}]_2$; fi
    return 1 if all checks pass and 0 otherwise.
```

Fig. 2. Variants Π_{kw} and $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ of Kiltz-Wee QA-NIZK for $[\boldsymbol{y}]_\iota = [\boldsymbol{M}]_\iota \boldsymbol{w}$ in the BPK model. Here, $\Pi \in \{\Pi_{\mathsf{kw}}, \Pi_{\mathsf{kw}}^{\mathsf{bdh}}\}$.

4.1 Π_{kw}: QA-NIZK with $k = 1$

Assume the CRS of the original Kiltz-Wee QA-NIZK is now the public key pk of the verifier. Since the amount of public information does not change and the verifier is interested in soundness, there is no need to reprove soundness. On the other hand, a malicious verifier may provide a malformed public key pk such that the nonuniform zero knowledge property does not hold. To protect the prover, we construct a public-key verification algorithm PKV, see Fig. 2, that checks whether $\mathsf{pk}^{\mathsf{zk}}$ is well-formed. Since Π_{kw} is perfectly zero knowledge in the CRS model, it is enough to equip the simulator Sim of Π_{kw} with the correct secret key sk to achieve zero knowledge. In the new construction, in simulation one first uses a (novel) knowledge assumption to retrieve the secret key sk and then runs the original Kiltz-Wee simulator Sim on sk to achieve nonuniform zero knowledge.

To be able to extract $\mathsf{sk} = \boldsymbol{K}$ in the case $k = 1$, we rely on the following novel knowledge *Kiltz-Wee Knowledge of Exponent assumption* that is essentially a tautological knowledge assumption for the Kiltz-Wee QA-NIZK. Nevertheless, it is weaker than the Sub-GBGM itself, and due to the remark in Section 2 it is hence also weaker than the GBGM. Intuitively, we assume that if $\mathcal{A}$ outputs a well-formed pk then there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$ who, knowing the secret coins of $\mathcal{A}$, returns a secret key $\boldsymbol{K}$ that *could* have been used to compute $\mathsf{pk}^{\mathsf{zk}}$. We emphasize that $[\boldsymbol{M}]_\iota$ is given as an input to $\mathcal{A}$.

Definition 1 (KW-KE)**.** *Fix $\iota \in \{1, 2\}$, $k = 1$, and $n > m \geq 1$. Let* PKV *be defined as in Fig. 2. (n, m)-$\mathsf{KW\text{-}KE}_{\mathbb{G}_\iota}$ holds relative to* Pgen *if $\forall \mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$, $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$, and PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, s.t.*

$$\Pr\left[\begin{matrix} r \leftarrow_\$ \mathsf{RND}(\mathcal{A}); (\mathsf{pk} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}, [\boldsymbol{P}]_\iota) \,\|\, \boldsymbol{K}) \leftarrow (\mathcal{A} \,\|\, \mathsf{Ext}_{\mathcal{A}})([\boldsymbol{M}]_\iota; r) : \\ \mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1 \wedge (\boldsymbol{K} \notin \mathbb{Z}_p^{n\times k} \vee \boldsymbol{P} \neq \boldsymbol{M}^\top \boldsymbol{K}) \end{matrix}\right] \approx_\lambda 0 \ .$$

Here, we do not require that $\mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}$ is of correct shape since $\mathsf{pk}^{\mathsf{snd}}$ is not needed for zero knowledge. Recall from Section 2 that the extractor defined by a knowledge assumption never works with probability 1.

We will also use the following knowledge assumption (*strong Kiltz-Wee Knowledge of Exponent assumption*) that is not needed to obtain zero knowledge but is needed for the whole public-key verification.

Definition 2 (sKW-KE)**.** *Fix $\iota \in \{1, 2\}$, $k = 1$, and $n > m \geq 1$. Let* PKV *be as in Fig. 2. Then (n, m)-$\mathsf{sKW\text{-}KE}_{\mathbb{G}_\iota}$ holds relative to* Pgen *if $\forall \mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$, $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$, and PPT adversary $\mathcal{A}$, there exists a*

PPT extractor $\mathsf{Ext}_{\mathcal{A}}$*, s.t.*

$$\Pr\left[\begin{array}{l} r \leftarrow_{\$} \mathsf{RND}(\mathcal{A}); (\mathsf{pk} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}, [\boldsymbol{P}]_\iota) \parallel \boldsymbol{K}) \leftarrow (\mathcal{A} \parallel \mathsf{Ext}_{\mathcal{A}})([\boldsymbol{M}]_\iota; r) : \\ \mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1 \wedge (\boldsymbol{K} \not\in \mathbb{Z}_p^{n \times k} \vee \boldsymbol{P} \neq \boldsymbol{M}^\top \boldsymbol{K} \vee \boldsymbol{C} \neq \boldsymbol{K}\bar{\boldsymbol{A}}) \end{array}\right] \approx_\lambda 0 \enspace .$$

In Theorem 1, we also need the following "weak KerMDH" assumption.

Definition 3. $\mathcal{D}_{\ell k}$-$\mathsf{wKerMDH}_{\mathbb{G}_\iota}$ *holds relative to* Pgen*, if* $\forall$ *PPT* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\mathrm{wkermdh}}_{\mathcal{A},\mathcal{D}_{\ell k},\iota,\mathsf{Pgen}}(\lambda) := \Pr\left[\begin{array}{l} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\ell k}; \boldsymbol{c} \leftarrow \mathcal{A}(\mathsf{p}, [\bar{\boldsymbol{A}}]_\iota) : \\ \boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_k \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell \end{array}\right] \approx_\lambda 0 \enspace .$$

Clearly, $\mathcal{D}_{\ell k}$-$\mathsf{wKerMDH}_{\mathbb{G}_\iota}$ is not stronger and it is ostensibly weaker than $\mathcal{D}_{\ell k}$-$\mathsf{KerMDH}_{\mathbb{G}_\iota}$ since computing $\boldsymbol{c}$ may be more complicated than computing $[\boldsymbol{c}]_{3-\iota}$. Computational Diffie-Hellman (CDH) is a classical example of $\mathsf{wKerMDH}$.

Theorem 1 (Sub-GBGM Security of KW-KE and sKW-KE). *Fix* $\iota \in \{1, 2\}$ *and* $n > m \geq 1$*. Then*

(i) the (n, m)-KW-$\mathsf{KE}_{\mathbb{G}_\iota}$ *assumption holds in the Sub-GBGM.*
(ii) under the $\mathcal{D}_{\mathsf{p}}$-$\mathsf{wKerMDH}_{\mathbb{G}_\iota}$ *assumption, the* (n, m)-sKW-$\mathsf{KE}_{\mathbb{G}_\iota}$ *assumption holds in the Sub-GBGM.*

This statement is straightforward when we replace Sub-GBGM with GBGM. Partially since Sub-GBGM proofs are not common, the following proof contains some novel ideas. In particular, since we work in the Sub-GBGM, the elements output by a KW-KE-adversary $\mathcal{A}$ can be written down as an affine function of all group-element inputs and all indeterminates created the by $\mathcal{A}$. This means that the verification equation gives us a large number of equalities in the coefficients corresponding to the new indeterminates (e.g., $\boldsymbol{P}_j \bar{\boldsymbol{A}}_i = \boldsymbol{0}_{m \times k}$ for each $i \geq 0, j > 0$). The constructed Sub-GBGM extractor $\mathsf{Ext}_{\mathcal{A}}$ returns $\boldsymbol{C}_{i_0} \bar{\boldsymbol{A}}_{i_0}^{-1}$ where i_0 is the smallest index for which $\det \bar{\boldsymbol{A}}_i \neq 0$. In the case of sKW-KE, we (somewhat suprisingly) need to additionally assume that $[\boldsymbol{M}]_\iota$ comes from a hard ($\mathsf{wKerMDH}$) distribution.

Proof. Assume $\mathcal{A}$ is a sKW-KE or KW-KE adversary that succeeds with some probability ε. That is, for any $\boldsymbol{M}$ and $r \leftarrow_{\$} \mathsf{RND}(\mathcal{A})$, with probability ε, $\mathcal{A}([\boldsymbol{M}]_\iota; r)$ outputs $\mathsf{pk} = ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}, [\boldsymbol{P}]_\iota)$, such that $\mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1$. (In particular, $\det \bar{\boldsymbol{A}} \neq 0$ and $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$.)

We now construct the following Sub-GBGM extractor $\mathsf{Ext}_{\mathcal{A}}$, where $Y_{\zeta i}$ are indeterminates created by $\mathcal{A}$ (i.e., group elements created by her for which she does not know the discrete logarithm) in $\mathbb{G}_\zeta$, $\zeta \in \{1, 2\}$, with $Y_{10} = Y_{20} = 1$. (Since $\mathcal{A}$ works in the Sub-GBGM, $\mathsf{Ext}_{\mathcal{A}}$ can extract $\bar{\boldsymbol{A}}$ and $\boldsymbol{C}$.)

```
$\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_\iota; r)$
Extract the coefficients of $\bar{\boldsymbol{A}} = \sum_{i \geq 0} \bar{\boldsymbol{A}}_i Y_{3-\iota,i}$ and $\boldsymbol{C} = \sum_{i \geq 0} \boldsymbol{C}_i Y_{3-\iota,i}$;
Let $i_0$ be the smallest $i$ for which $\det \bar{\boldsymbol{A}}_i \neq 0$;
return $\boldsymbol{K} \leftarrow \boldsymbol{C}_{i_0} \bar{\boldsymbol{A}}_{i_0}^{-1}$;
```

Note that i_0 can be any i for which $\det \bar{\boldsymbol{A}}_i \neq 0$; we made the above choice for the sake of concreteness.

We will now analyse $\mathsf{Ext}_{\mathcal{A}}$, showing that $\mathsf{Ext}_{\mathcal{A}}$ is the extractor in the definition of KW-KE / sKW-KE. Assume that $\mathcal{A}$ was successful with inputs $([\boldsymbol{M}]_\iota; r)$, where $\boldsymbol{M} = \boldsymbol{M} Y_{\iota 0} \in \mathbb{Z}_p$. We now execute $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_\iota)$ and obtain $\boldsymbol{K}$ as above. Since $\mathcal{A}$ works in Sub-GBGM,

$$\boldsymbol{P} = \textstyle\sum_{j \geq 0} \boldsymbol{P}_j Y_{\iota j}$$

(for coefficients $\boldsymbol{P}_j$ that we might not know). From the equation 3 in PKV (i.e., $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$),

$$\boldsymbol{M}^\top Y_{\iota 0} \left(\textstyle\sum_{i \geq 0} \boldsymbol{C}_i Y_{3-\iota,i}\right) - \left(\textstyle\sum_{j \geq 0} \boldsymbol{P}_j Y_{\iota j}\right)^\top \left(\textstyle\sum_{i \geq 0} \bar{\boldsymbol{A}}_i Y_{3-\iota,i}\right) = \boldsymbol{0}_{m \times k} \enspace .$$

Since $Y_{\zeta i}$ are random variables, the coefficients of $Y_{3-\iota,i} Y_{\iota j}$ in the last displayed equation must be equal to $\boldsymbol{0}_{m \times k}$ for each $i, j \geq 0$. Thus,

1. $\boldsymbol{P}_0 \bar{\boldsymbol{A}}_i = \boldsymbol{M}^\top \boldsymbol{C}_i$ for each $i \geq 0$,
2. $\boldsymbol{P}_j \bar{\boldsymbol{A}}_i = \boldsymbol{0}_{m \times k}$ for each $i \geq 0, j > 0$.

Since $k = 1$ (it does not hold in general) and $\det \bar{\boldsymbol{A}} \neq 0$, $\mathcal{N} := \{i \geq 0 : \det \bar{\boldsymbol{A}}_i \neq 0\}$ is non-empty. But then for each $i \in \mathcal{N}$, $\bar{\boldsymbol{A}}_i$ is invertible. Define $\boldsymbol{K}_i := \boldsymbol{C}_i \bar{\boldsymbol{A}}_i^{-1} \in \mathbb{Z}_p^{n \times k}$ for $i \in \mathcal{N}$. Thus,

1. $\boldsymbol{P}_0 = \boldsymbol{M}^\top \boldsymbol{K}_i$ for each $i \in \mathcal{N}$,
2. $\boldsymbol{P}_j = \boldsymbol{0}_{m \times k}$ for each $i \in \mathcal{N}$ and $j > 0$.

Thus, for any $i \in \mathcal{N}$ and thus also for $i = i_0$, $\boldsymbol{P} = \sum_{j \geq 0} \boldsymbol{P}_j Y_{\iota j} = \boldsymbol{M}^\top \boldsymbol{K}_i$ and we have proven the Sub-GBGM security of KW-KE.

To prove that sKW-KE is secure in the Sub-GBGM, we need to also show that $\boldsymbol{C} = \boldsymbol{K} \bar{\boldsymbol{A}}$. We do it by a reduction to $\mathcal{D}_\mathsf{p}$-wKerMDH. We can use the above extractor but in this case we have that $[\boldsymbol{M}]_\iota$ is sampled from $\mathcal{D}_\mathsf{p}$. (This is fine, since the extractor exists for *any* $[\boldsymbol{M}]_\iota$, including also the case $[\boldsymbol{M}]_\iota \leftarrow_\$ \mathcal{D}_\mathsf{p}$.) If $i \not\in \mathcal{N}$, then since $k = 1$, $\boldsymbol{M}^\top \boldsymbol{C}_i = \boldsymbol{0}_{m \times k}$. If $\boldsymbol{C}_i \neq \boldsymbol{0}_{n \times k}$, and since $k = 1$, we have found a non-trivial element in the cokernel of $\boldsymbol{M}$ and thus broken $\mathcal{D}_\mathsf{p}$-wKerMDH. Thus, $\boldsymbol{C}_i = \boldsymbol{0}_{n \times k}$ for $i \not\in \mathcal{N}$. If $|\mathcal{N}| \geq 2$, then for each $i_1, i_2 \in \mathcal{N}$, $\boldsymbol{M}^\top \boldsymbol{K}_{i_1} = \boldsymbol{M}^\top \boldsymbol{K}_{i_2}$ and thus

$$\boldsymbol{M}^\top (\boldsymbol{K}_{i_1} - \boldsymbol{K}_{i_2}) = \boldsymbol{0}_{m \times k} \enspace .$$

For all $i_1, i_2 \in \mathcal{N}$, since $k = 1$, this means that either $\boldsymbol{K}_{i_1} = \boldsymbol{K}_{i_2}$ or $\boldsymbol{K}_{i_1} - \boldsymbol{K}_{i_2}$ is a non-trivial element in the cokernel of $\boldsymbol{M}$. The latter gives us contradiction with the hardness of $\mathcal{D}_\mathsf{p}$-wKerMDH. Thus, there exists $\boldsymbol{K}$, such that for all $i \in \mathcal{N}$, $\boldsymbol{K} = \boldsymbol{K}_i$. (This is trivially true if $|\mathcal{N}| = 1$.) Thus, for each $i \in \mathcal{N}$, $\boldsymbol{K} = \boldsymbol{C}_i \bar{\boldsymbol{A}}_i^{-1}$ and thus $\boldsymbol{C}_i = \boldsymbol{K} \bar{\boldsymbol{A}}_i$. If $i \not\in \mathcal{N}$, $\boldsymbol{C}_i = \boldsymbol{0}_{n \times k}$. Thus,

$$\boldsymbol{C} = \textstyle\sum_{i \in \mathcal{N}} \boldsymbol{K} \bar{\boldsymbol{A}}_i Y_{3-\iota, i} = \boldsymbol{K} \bar{\boldsymbol{A}} \enspace .$$

Since $\boldsymbol{P}_j = \boldsymbol{M}_j^\top \boldsymbol{K}$ for each j, $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$. □

Remark 1. We note that $\varrho = [\boldsymbol{M}]_\iota$, generated by $\mathcal{D}_\mathsf{p}$, is a member of $\mathbb{G}_\iota$ and thus by the definition of Sub-GBGM the only way to make $[\boldsymbol{C}]_{3-\iota}, [\bar{\boldsymbol{A}}]_{3-\iota} \in \mathbb{G}_{3-\iota}$ to depend on it is to consider its hashing to $\mathbb{G}_{3-\iota}$ as a new indeterminate; this is intuitively what we did when we constructed $\mathsf{Ext}_\mathcal{A}$. □

In the case of sKW-KE, we extracted the unique $\boldsymbol{K}$ that was used to compute the CRS. Following the proof idea from Abdolmaleki *et al.* [ABLZ17], it is easy to show that under this assumption, Π_kw is nonuniform zero knowledge.

Theorem 2 (Security of Π_kw). *Let $\Pi = \Pi_\mathsf{kw}$ be the* QA-NIZK *argument system for linear subspaces from Fig. 2. Let $\iota \in \{1, 2\}$ and $k = 1$. Then the following statements hold in the BPK model.*

(i) Π_kw is perfectly complete.
(ii) If the $\mathsf{sKW\text{-}KE}_{\mathbb{G}_\iota}$ *assumption holds relative to* Pgen *then Π_kw is statistically nonuniform zero knowledge. (And thus also in the Sub-GBGM assuming* $\mathcal{D}_\mathsf{p}$-wKerMDH*, i.e., if $[\boldsymbol{M}]_\iota$ comes from a* wKerMDH*-hard distribution.)*
(iii) If the $\mathsf{KW\text{-}KE}_{\mathbb{G}_\iota}$ *assumption holds relative to* Pgen *then Π_kw is statistically nonuniform zero knowledge. (And thus also in the Sub-GBGM.)*
(iv) If the $\mathcal{D}_k$-KerMDH *assumption holds relative to* Pgen *then Π_kw is computationally quasi-adaptively sound.*

Proof. **(i: perfect completeness):** obvious.

(ii: nonuniform zero knowledge under sKW-KE): Let Z be a subverter that computes pk so as to break the nonuniform zero knowledge property. That is, $\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ outputs $(\mathsf{pk}, \mathsf{aux}_\mathsf{Z})$. Let $\mathcal{A}$ be the adversary from Fig. 3. Note that $\mathsf{RND}(\mathcal{A}) = \mathsf{RND}(\mathsf{Z})$. Under the sKW-KE assumption, there exists an extractor $\mathsf{Ext}'_\mathcal{A}$, such that if $\mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1$ then $\mathsf{Ext}'_\mathcal{A}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ outputs $\boldsymbol{K}$, such that $\boldsymbol{C} = \boldsymbol{K} \bar{\boldsymbol{A}}$ and

```
$\mathcal{A}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$
$(\mathsf{pk}, \mathsf{aux}_\mathsf{Z}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$; return $\mathsf{pk}$;

$\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$
return $\mathsf{Ext}'_\mathcal{A}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$;
```

Fig. 3. The extractor and the constructed adversary $\mathcal{A}$ from the nonuniform zero knowledge proof of Theorem 2, for both the sKW-KE and KW-KE case.

$\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$. We construct a trivial extractor $\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ for Z, as depicted in Fig. 3. Clearly, Ext_Z returns $\mathsf{sk} = \boldsymbol{K}$, such that $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$ and $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$.

Fix concrete values of λ, $\mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$, $[\boldsymbol{M}]_\iota \in \mathcal{D}_\mathsf{p}$, $([\boldsymbol{y}]_\iota, \boldsymbol{w}) \in \mathcal{R}_{\boldsymbol{M}}$, $r_\mathsf{Z} \in \mathsf{RND}(\mathsf{Z})$, and run $\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ to obtain $\boldsymbol{K}$. It clearly suffices to show that if $\mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1$ and $([\boldsymbol{y}]_\iota, \boldsymbol{w}) \in \mathcal{R}_{\boldsymbol{M}}$ then

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

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

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

Hence, O_0 and O_1 have the same distribution and thus, Π_kw is nonuniform zero knowledge under sKW-KE.

(iii: nonuniform zero knowledge under KW-KE): The security proof is the same as in the previous case, except that $\mathsf{Ext}'_\mathcal{A}$ is an extractor guaranteed by KW-KE. The only difference in the following is that it is not guaranteed that $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$. The claim follows since $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$ is not used in the proof of (ii).

(iv: soundness under KerMDH): follows from [KW15]. □

4.2 $\Pi_\mathsf{kw}^\mathsf{bdh}$: QA-NIZK with $k = 2$

The KW-KE and sKW-KE assumptions are secure in the Sub-GBGM when $k = 1$. In the case $k = 2$, we will prove nonuniform zero knowledge under the known BDH-KE assumption. To be able to use BDH-KE, we need to include more elements to the public key since the simulator must be able to extract $\bar{\boldsymbol{A}}$ and $\boldsymbol{C}$, and the BDH-KE assumption requires then $[\bar{\boldsymbol{A}}, \boldsymbol{C}]_\iota$ to be available. The corresponding PKV algorithm checks that also these added elements are correct. The argument is depicted in Fig. 2 (the case $\Pi = \Pi_\mathsf{kw}^\mathsf{bdh}$).

Theorem 3 (Security of $\Pi_\mathsf{kw}^\mathsf{bdh}$). *Let $\iota \in \{1, 2\}$ and $k = 2$. Consider $\Pi = \Pi_\mathsf{kw}^\mathsf{bdh}$ from Fig. 2. The following statements hold in the BPK model.*

(i) $\Pi_\mathsf{kw}^\mathsf{bdh}$ is perfectly complete.
(ii) If the $\mathcal{D}_k$-SKerMDH assumption holds relative to Pgen then $\Pi_\mathsf{kw}^\mathsf{bdh}$ is computationally quasi-adaptively sound.
(iii) If the BDH-KE assumption holds relative to Pgen then $\Pi_\mathsf{kw}^\mathsf{bdh}$ is statistically nonuniform zero knowledge.

Proof. **(i: perfect completeness):** obvious.

(ii: soundness): The proof is similar to the soundness proof of Π'_as in [KW15]. There, the authors reduced the soundness of the argument to the KerMDH assumption. Since we added $([\bar{\boldsymbol{A}}, \boldsymbol{C}]_\iota)$ to the public key, we reduce instead to the SKerMDH assumption of [GHR15]; this changes certain aspects of the proof.

Assume that $\mathcal{A}$ breaks the soundness of $\Pi_\mathsf{kw}^\mathsf{bdh}$ with probability ε. We will build an adversary $\mathcal{B}$, see Fig. 4, that breaks SKerMDH with probability $\geq \varepsilon - 1/p$.

Note that in Fig. 4, $[\bar{\boldsymbol{A}}']_\zeta = [\bar{\boldsymbol{A}}]_\zeta \in \mathbb{G}_\zeta^{k \times k}$. Define implicitly (we do not know this value) $\boldsymbol{K} \leftarrow \boldsymbol{K}' + \boldsymbol{M}^\perp \underline{\boldsymbol{A}}' \bar{\boldsymbol{A}}^{-1} \in \mathbb{Z}_p^{n \times k}$. Thus,

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

```
$\mathcal{B}(\mathsf{p}, ([\boldsymbol{A}]_1, [\boldsymbol{A}]_2))$ // $([\boldsymbol{A}]_1, [\boldsymbol{A}]_2) \in \mathbb{G}_1^{(k+1)\times k} \times \mathbb{G}_2^{(k+1)\times k}$
$([\boldsymbol{M}]_\iota, \boldsymbol{M}) \leftarrow_\$ \mathcal{D}'_\mathsf{p}$; // $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$
Let $\boldsymbol{M}^\perp \in \mathbb{Z}_p^{n\times(n-m)}$ be a basis of the kernel of $\boldsymbol{M}^\top$;
$\boldsymbol{K'} \leftarrow_\$ \mathbb{Z}_p^{n\times k}$; $\boldsymbol{R} \leftarrow_\$ \mathbb{Z}_p^{(n-m-1)\times(k+1)}$;
for $\zeta \in \{1, 2\}$ do
    $[\boldsymbol{A'}]_\zeta \leftarrow \begin{pmatrix} [\boldsymbol{A}]_\zeta \\ \boldsymbol{R}\cdot[\boldsymbol{A}]_\zeta \end{pmatrix}$; // $\boldsymbol{A'} \in \mathbb{Z}_p^{(n-m+k)\times k}$
    $[\boldsymbol{C}]_\zeta \leftarrow (\boldsymbol{K'}||\boldsymbol{M}^\perp)[\boldsymbol{A'}]_\zeta$; endfor
$[\boldsymbol{P}]_\iota \leftarrow [\boldsymbol{M}^\top \boldsymbol{K'}]_\iota$;
$\mathsf{pk}' \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_{3-\iota}, [\bar{\boldsymbol{A}}, \boldsymbol{C}, \boldsymbol{P}]_\iota)$;
$([\boldsymbol{y}]_\iota, [\boldsymbol{\pi}]_\iota) \leftarrow \mathcal{A}([\boldsymbol{M}]_\iota, \mathsf{pk}')$; // $[\boldsymbol{y}]_\iota \in \mathbb{G}_\iota^n$, $[\boldsymbol{\pi}]_\iota \in \mathbb{G}_\iota^k$
$[\boldsymbol{c}]_\iota^\top \leftarrow [(\boldsymbol{\pi}^\top - \boldsymbol{y}^\top \boldsymbol{K'})|| - \boldsymbol{y}^\top \boldsymbol{M}^\perp]_\iota$;
Represent $[\boldsymbol{c}]_\iota^\top$ as $[\boldsymbol{c}_1^\top||\boldsymbol{c}_2^\top]_\iota$ with $[\boldsymbol{c}_1]_\iota \in \mathbb{G}_\iota^{k+1}$ and $[\boldsymbol{c}_2]_\iota \in \mathbb{G}_\iota^{n-m-1}$;
$\boldsymbol{s}_{3-\iota} \leftarrow_\$ \mathbb{Z}_p^{k+1}$; $[\boldsymbol{s}_\iota]_\iota \leftarrow [\boldsymbol{c}_1 + \boldsymbol{R}^\top \boldsymbol{c}_2 + \boldsymbol{s}_{3-\iota}]_\iota$;
return $([\boldsymbol{s}_1]_1, [\boldsymbol{s}_2]_2)$;
```

Fig. 4. Adversary $\mathcal{B}$ in the soundness proof of Theorem 3

```
$\mathcal{A}'([\boldsymbol{M}]_\iota; r_\mathsf{Z})$
$(\mathsf{pk}, \mathsf{aux}_\mathsf{Z}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$; return $([\boldsymbol{C}]_1, [\boldsymbol{C}]_2)$;

$\mathcal{A}''([\boldsymbol{M}]_\iota; r_\mathsf{Z})$
$(\mathsf{pk}, \mathsf{aux}_\mathsf{Z}) \leftarrow \mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$; return $([\bar{\boldsymbol{A}}]_1, [\bar{\boldsymbol{A}}]_2)$;

$\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ // Check $r_\mathsf{Z}$ etc
$\boldsymbol{C} \leftarrow \mathsf{Ext}'_{\mathcal{A}'}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$;
$\bar{\boldsymbol{A}} \leftarrow \mathsf{Ext}''_{\mathcal{A}''}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$;
$\boldsymbol{K} \leftarrow \boldsymbol{C}\bar{\boldsymbol{A}}^{-1}$;
return $\boldsymbol{K}$;
```

Fig. 5. The extractor and $\mathcal{A}'$, $\mathcal{A}''$ from the nonuniform zero knowledge proof of Theorem 3.

and

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

Thus, pk' has the same distribution as the real public key.

With probability ε, $\mathcal{A}$ is successful, i.e.,

1. $\boldsymbol{y}^\top \boldsymbol{M}^\perp \neq \boldsymbol{0}_{1\times(n-m)}$ (i.e., $\boldsymbol{y} \not\in \operatorname{im}(\boldsymbol{M})$) and thus also $\boldsymbol{c} \neq \boldsymbol{0}_{n-m+k}$;
2. $\boldsymbol{y}^\top \boldsymbol{C} = \boldsymbol{\pi}^\top \bar{\boldsymbol{A}}$ (V accepts). Thus, $\boldsymbol{0} = \boldsymbol{\pi}^\top \bar{\boldsymbol{A}} - \boldsymbol{y}^\top \boldsymbol{C} = \left(\boldsymbol{\pi}^\top || \boldsymbol{0}_{n-m}^\top\right) \boldsymbol{A'} - \boldsymbol{y}^\top \left(\boldsymbol{K'}||\boldsymbol{M}^\perp\right) \boldsymbol{A'} = \left((\boldsymbol{\pi}^\top - \boldsymbol{y}^\top \boldsymbol{K'})|| - \boldsymbol{y}^\top \boldsymbol{M}^\perp\right) \boldsymbol{A'} = \boldsymbol{c}^\top \boldsymbol{A'}$.

Clearly, $\boldsymbol{s}_\iota - \boldsymbol{s}_{3-\iota} = \boldsymbol{c}_1 + \boldsymbol{R}^\top \boldsymbol{c}_2$ and

$$(\boldsymbol{s}_\iota^\top - \boldsymbol{s}_{3-\iota}^\top)\boldsymbol{A} = (\boldsymbol{c}_1^\top + \boldsymbol{c}_2^\top \boldsymbol{R})\boldsymbol{A} = \boldsymbol{c}^\top \boldsymbol{A'} = \boldsymbol{0}_{1\times k} \enspace .$$

Since $\boldsymbol{c} \neq \boldsymbol{0}_{n-m+k}$ and $\boldsymbol{R}$ leaks only through $\boldsymbol{A'}$ (in definitions of $[\boldsymbol{C}]_1, [\boldsymbol{C}]_2$) as $\boldsymbol{R}\boldsymbol{A}$,

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

where the probability is over $\boldsymbol{R} \leftarrow_\$ \mathbb{Z}_p^{(n-m-1)\times(k+1)}$.

(ii: nonuniform zero knowledge of $\Pi_{\mathsf{kw}}^{\mathsf{bdh}}$ under $\mathsf{BDH\text{-}KE}$**):** Let Z be a subverter aims to break the nonuniform zero knowledge property. That is, $\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ outputs $(\mathsf{pk}, \mathsf{aux}_\mathsf{Z})$. Let $\mathcal{A}'$ and $\mathcal{A}''$ be the adversaries from from Fig. 5. Note that $\mathsf{RND}(\mathcal{A}') = \mathsf{RND}(\mathcal{A}'') = \mathsf{RND}(\mathsf{Z})$. Under the $\mathsf{BDH\text{-}KE}$ assumption, there exist extractors $\mathsf{Ext}'_{\mathcal{A}'}$ and $\mathsf{Ext}''_{\mathcal{A}''}$, such that $\mathsf{Ext}'_{\mathcal{A}'}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ outputs $\boldsymbol{C}$ and $\mathsf{Ext}''_{\mathcal{A}''}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ outputs $\boldsymbol{A}$. We construct an extractor $\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ for Z, as depicted in Fig. 5. Clearly, Ext_Z returns $\mathsf{sk} = \boldsymbol{K}$.

Fix concrete values of λ, $\mathsf{p} \in \operatorname{im}(\mathsf{Pgen}(1^\lambda))$, $[\boldsymbol{M}]_\iota \in \mathcal{D}_\mathsf{p}$, $([\boldsymbol{y}]_\iota, \boldsymbol{w}) \in \mathcal{R}_{\boldsymbol{M}}$, $r_\mathsf{Z} \in \mathsf{RND}(\mathsf{Z})$, and run $\mathsf{Ext}_\mathsf{Z}([\boldsymbol{M}]_\iota; r_\mathsf{Z})$ to obtain sk. It suffices to show that if $\mathsf{PKV}([\boldsymbol{M}]_\iota, \mathsf{pk}) = 1$ and $([\boldsymbol{y}]_\iota, \boldsymbol{w}) \in \mathcal{R}_{\boldsymbol{M}}$ then

$$\mathsf{O}_0([\boldsymbol{y}]_\iota, \boldsymbol{w}) = \mathsf{P}([\boldsymbol{M}]_\iota, \mathsf{pk}, [\boldsymbol{y}]_\iota, \boldsymbol{w}) = [\boldsymbol{P}]_\iota^\top \boldsymbol{w} \enspace ,$$

$$\mathsf{O}_1([\boldsymbol{y}]_\iota, \boldsymbol{w}) = \mathsf{Sim}([\boldsymbol{M}]_\iota, \mathsf{pk}, \mathsf{sk}; [\boldsymbol{y}]_\iota) = \boldsymbol{K}^\top [\boldsymbol{y}]_\iota$$

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

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

Hence, O_0 and O_1 have the same distribution and thus, $\mathit{\Pi}_{\mathsf{kw}}$ is nonuniform zero knowledge under sKW-KE. □

We remark that $k = 2$ is needed to get soundness since $\bar{\boldsymbol{A}}$ is represented in both groups; SKerMDH is not secure with $k = 1$, [GHR15].

5 Discussion: Subverter Choosing ϱ

In Section 3, we defined nonuniform zero knowledge in the BPK model assuming that the language parameter ϱ is generated honestly, that is, from the correct distribution and without any leakage of the secret keys. In this section, we will study whether this assumption is really needed.

For the sake of concreteness, let us first consider $\mathit{\Pi}_{\mathsf{kw}}$ (thus, $\varrho = [\boldsymbol{M}]_\iota$ for some matrix $\boldsymbol{M}$) and the nonuniform zero knowledge definition in Section 3. According to the latter, if Z on input ϱ outputs pk then he can leak information through two different channels: aux_Z (any string of Z's choice that can be sent to a malicious distinguisher) and sk (the secret key extracted from Z by the PPT extractor Ext_Z, where the existence of the latter is stated by the definition).

Leaking Information via aux_Z. If Z leaks (a part of) $\boldsymbol{M}$ to the verifier through aux_Z then V will be able to check whether $[\boldsymbol{y}]_\iota \in \mathrm{im}([\boldsymbol{M}]_\iota)$ or even compute (a part of) $[\boldsymbol{w}]_\iota$ from $[\boldsymbol{y}]_\iota$. This holds since $\mathcal{L}_{[\boldsymbol{M}]_\iota}$ is not necessarily hard if $\boldsymbol{M}$ is public. E.g., consider the case when $[\boldsymbol{M}]_\iota = [M_1, M_2]_\iota^\top$ is an Elgamal public key for $M_i \neq 0$. Then $[y_1, y_2]_\iota^\top =^? [\boldsymbol{M}]_\iota w = [M_1 w, M_2 w]_\iota^\top$ can be decided efficiently, given (M_1, M_2), by checking whether $M_1 [y_2]_\iota = M_2 [y_1]_\iota$. Moreover, one can compute $(1/M_1)[y_1]_\iota = [w]_\iota$.

This attack is possible unless communication between the creator of $[\boldsymbol{M}]_\iota$ and the malicious verifier is limited to not leak any additional information about $\boldsymbol{M}$. Hence, achieving the intuitive notion of zero knowledge is impossible unless $[\boldsymbol{M}]_\iota$ is created by a separate party who does not leak information to V. (Or, the language $\mathcal{L}_{[\boldsymbol{M}]_\iota}$ is easy, which is not interesting.)

Leaking Information via Knowledge Assumptions. There is a more sneaky (and seemingly novel) attack where the subverter, who knows $\boldsymbol{M}$, leaks $\boldsymbol{M}$ to the simulator via sk. Since this attack is less obvious, we will consider it in more detail. In principle, this attack means that one can construct QA-NIZK arguments that are "formally" zero knowledge but intuitively leak information.

For example, consider the case where the pair $([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$ belongs to pk created by Z. Under the BDH-KE assumption, there exists a PPT extractor Ext_Z that extracts $\boldsymbol{M}$ from the pk. Given $[\boldsymbol{y}]_\iota \in \mathcal{L}_{[\boldsymbol{M}]_\iota}$ and $\boldsymbol{M}$, Ext_Z computes $[\boldsymbol{w}]_\iota$ s.t. $[\boldsymbol{y}]_\iota = [\boldsymbol{M}]_\iota \boldsymbol{w}$ (cf. the previous subsubsection). One can now construct a contrived QA-NIZK (see Fig. 6) where the prover and the simulator both output $[\boldsymbol{w}]_\iota$. Since the outputs of P and Sim are the same, this protocol is formally zero knowledge although intuitively it leaks information about $\boldsymbol{w}$.

More generally, a malicious subverter can choose sk to be a function of $\boldsymbol{M}$ and thus leak (partially) $\boldsymbol{M}$ to the simulator who then uses this information to simulate; as above, in this case one can design an argument system that is formally zero knowledge but still leak information.

This is a well-known problem: if the simulator can compute the witness then she can just output the honest proof. Thus, if simulator is allowed to run in time, sufficient to compute witness from the input, there is no reason to construct a zero knowledge argument system. In the case of nonuniform zero knowledge, one also has to make sure that the (PPT) extractor will not be able to extract $\boldsymbol{M}$ (or a part of it). Hence, one should not use a knowledge assumption where the extractor, given pk output by Z, returns some value that depends on $\boldsymbol{M}$. This is impossible to achieve in general: for example in $\mathit{\Pi}_{\mathsf{kw}}$ and $\mathit{\Pi}_{\mathsf{kw}}^{\mathsf{bdh}}$, the subverter who knows $\boldsymbol{M}$ can choose $\boldsymbol{K}$ as a function of $\boldsymbol{M}$.

```
$\mathsf{K}(\mathsf{p})$: // K creates $\mathsf{sk} = \boldsymbol{M} \in \mathbb{Z}_p^{2 \times 1}$ so he does not get $[\boldsymbol{M}]_\iota$ as an input
    $([\boldsymbol{M}]_1, \boldsymbol{M}) \leftarrow_{\$} \mathcal{D}'_{\mathsf{p}}$; $\mathsf{pk} \leftarrow ([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$; $\mathsf{sk} \leftarrow \boldsymbol{M}$; return $(\mathsf{pk}, \mathsf{sk})$;
$\mathsf{P}(\varrho, \mathsf{pk}, [\boldsymbol{y}]_\iota, w)$: return $[\pi]_\iota \leftarrow [w]_\iota \in \mathbb{G}_\iota^1$;
$\mathsf{Ext}_{\mathsf{Z}}(\mathsf{pk}; r)$: Extract $\mathsf{sk} = (M_1, M_2)^\top$ by using BDH-KE; return $\mathsf{sk}$;
$\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_\iota)$: if $M_1^{-1}[y_1]_\iota \neq M_2^{-1}[y_2]_\iota$ then return $\perp$; else return $[\boldsymbol{\pi}]_\iota \leftarrow M_1^{-1}[y_1]_\iota \in \mathbb{G}_\iota^1$; fi
$\mathsf{V}(\varrho, \mathsf{pk}, [\boldsymbol{y}]_\iota, [\pi]_\iota)$ : check that $[\boldsymbol{y}]_\iota^\top [1]_{3-\iota} = [\pi]_\iota^\top [\boldsymbol{M}]_{3-\iota}^\top$;
$\mathsf{PKV}(\varrho, \mathsf{pk})$: check that $[\boldsymbol{M}]_1 [1]_2 = [1]_1 [\boldsymbol{M}]_2$;
```

Fig. 6. A contrived leaky subspace QA-NIZK ($n = 2$, $m = k = 1$)

Thus, we cannot allow the subverter to construct (or even know) $\boldsymbol{M}$ herself since then we can construct an ostensibly nonuniform zero knowledge QA-NIZK argument system where the extractor can use a simple knowledge assumption (like BDH-KE), that is not specific to $\boldsymbol{M}$ at all, to recover $\boldsymbol{M}$ (or a part of it).

6 Conclusion and Open Questions

Bellare, Fuchsbauer, and Scafuro [BFS16] defined the framework of subversion-resistant NIZK. In particular, a NIZK is Sub-ZK if it stays zero knowledge even if the CRS creator is malicious (subverted). After that, Abdolmaleki, Baghery, Lipmaa, and Zając [ABLZ17] and Fuchsbauer [Fuc18] showed how to make the most efficient known SNARKs Sub-ZK. Importantly, as shown in [ABLZ17,Fuc18], Sub-ZK for SNARKs comes for free except that the prover has to check the correctness of the CRS. Since the prover can omit the check assuming that the CRS was correctly generated, one can argue that Sub-ZK is the correct notion of zero knowledge for zk-SNARKs. It is a natural question whether the same is the case for QA-NIZKs. In the current paper, we showed that this is indeed so although the answer is (unexpectedly?) more complicated than in the case of SNARKs.

We first showed that Sub-ZK is equivalent to the previously known notion of no-auxiliary-string non-black-box zero knowledge (also known as non-uniform zero knowledge [Wee07]) in the BPK model; this natural connection was missed in the previous work. In particular, this means that due to the known impossibility results for nonuniform zero knowledge [Wee07], we get that the use of non-falsifiable assumptions in [BFS16, ABLZ17,Fuc18] to prove Sub-ZK is unavoidable. (This is alluded to but not proven by Bellare *et al.* [BFS16].)

After that, we showed that one can achieve nonuniform zero knowledge (and thus Sub-ZK) for free for the most efficient known QA-NIZK for linear subspaces by Kiltz and Wee [KW15]. However, the QA-NIZK case is *more complicated* than the SNARK case: in the case $k = 1$ (where k is a security parameter related to the matrix distributions; $k = 1$ gives the best efficiency) we showed that one can achieve nonuniform zero knowledge for free under a new knowledge assumption. Interestingly, the knowledge assumption that is sufficient to achieve nonuniform zero knowledge does not guarantee that the whole public key is correct. We showed that the latter can be established under a stronger assumption. We also proved that nonuniform zero knowledge in the (less interesting) case $k = 2$ can be obtained from a standard knowledge assumption but it requires one to duplicate some public-key elements to both source groups. Importantly, we noted that all analysed QA-NIZK variants are black-box zero knowledge in the RPK model.

Finally, we showed that the language parameter ϱ of QA-NIZKs *needs* to be generated so that the trapdoor will not be leaked to the verifier or the simulator. This is normal in the case of QA-NIZKs (in particular, a QA-NIZK argument system is only defined for a fixed distribution of $\varrho = [\boldsymbol{M}]_\iota$). Nevertheless, we pointed out some possible attacks (including a non-obvious one where a malicious public-key generator leaks $\boldsymbol{M}$ to the simulator via a knowledge assumption).

Open Problems and Further Work. Since in the important case $k = 1$, Sub-ZK can be achieved for free, we argue that it is the correct notion of zero knowledge for QA-NIZKs even if achieving it is not needed in a concrete application. Still, we mentioned some concrete applications of Sub-ZK QA-NIZK, but we leave their further investigation as an interesting open question. We also leave it to the further work to study whether different versions of QA-NIZKs (like one-time simulation-sound QA-NIZKs [JR13], unbounded simulation-sound QA-NIZK [LPJY14, KW15, LPJY15] or QA-NIZKs for other languages [GHR15, GR16]) can be made Sub-ZK "for free".

Acknowledgement. The authors were supported by the European Union's Horizon 2020 research and innovation programme under grant agreements No 653497 (project PANORAMIX) and No 780477 (project PRIViLEDGE), and by the Estonian Research Council grant (PRG49). Siim has been supported by European Regional Development Fund under the grant no EU48684.

References

ABLZ17. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michał Zając. A Subversion-Resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017 (3)*, volume 10626 of *LNCS*, pages 3–33, Hong Kong, China, December 3–7, 2017. Springer, Cham.

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*, volume 9057 of *LNCS*, pages 69–100, Sofia, Bulgaria, May 26–30, 2015. Springer, Cham.

APV05. Joël Alwen, Giuseppe Persiano, and Ivan Visconti. Impossibility and Feasibility Results for Zero Knowledge with Public Keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 135–151, Santa Barbara, USA, August 14–18, 2005. Springer, Heidelberg.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg.

BCG$^+$14. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *IEEE SP 2014*, pages 459–474, Berkeley, CA, USA, May 18–21, 2014. IEEE Computer Society.

BCG$^+$15. Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs. In *IEEE SP 2015*, pages 287–304, San Jose, CA, USA, May 17–21, 2015. IEEE Computer Society.

BCI$^+$10. Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient Indifferentiable Hashing into Ordinary Elliptic Curves. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 237–254, Santa Barbara, California, USA, August 15–19, 2010. Springer, Heidelberg.

BCNP04. Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally Composable Protocols with Relaxed Set-Up Assumptions. In *FOCS 2004*, pages 186–195, Rome, Italy, October, 17–19 2004. IEEE, IEEE Computer Society Press.

BDMP91. Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive Zero-Knowledge. *SIAM J. Comput.*, 6(20):1084–1118, 1991.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC 1988*, pages 103–112, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.

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 (2)*, volume 10032 of *LNCS*, pages 777–804, Hanoi, Vietnam, December 4–8, 2016. Springer, Cham.

CGGM00. Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable Zero-Knowledge. In *STOC 2000*, pages 235–244, Portland, Oregon, USA, May 21–23 2000. ACM Press.

Dam91. Ivan Damgård. Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In Joan Feigenbaum, editor, *CRYPTO 1991*, volume 576 of *LNCS*, pages 445–456, Santa Barbara, California, USA, August 11–15, 1991. Springer, Heidelberg, 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 (1)*, volume 8873 of *LNCS*, pages 532–550, Kaohsiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg.

DFN06. Ivan Damgård, Nelly Fazio, and Antonio Nicolosi. Non-interactive Zero-Knowledge from Homomorphic Encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 41–59, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg.

EHK$^+$13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge L. Villar. An Algebraic Framework for Diffie-Hellman Assumptions. In Ran Canetti and Juan Garay, editors, *CRYPTO (2) 2013*, volume 8043 of *LNCS*, pages 129–147, Santa Barbara, California, USA, August 18–22, 2013. Springer, Heidelberg.

FL16. Prastudy Fauzi and Helger Lipmaa. Efficient Culpably Sound NIZK Shuffle Argument without Random Oracles. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 200–216, San Franscisco, CA, USA, February 29–March 4, 2016. Springer, Cham.

FLM11. Marc Fischlin, Benoît Libert, and Mark Manulis. Non-interactive and Re-usable Universally Composable String Commitments with Adaptive Security. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 468–485, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg.

FLS90. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String. In *FOCS 1990*, pages 308–317, St. Louis, Missouri, USA, October 22–24, 1990. IEEE Computer Society Press.

FLSZ17. Prastudy Fauzi, Helger Lipmaa, Janno Siim, and Michał Zając. An Efficient Pairing-Based Shuffle Argument. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017 (2)*, volume 10625 of *LNCS*, pages 98–127, Hong Kong, China, December 3–7, 2017. Springer, Cham.

FLZ16. Prastudy Fauzi, Helger Lipmaa, and Michał Zając. A Shuffle Argument Secure in the Generic Model. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016 (2)*, volume 10032 of *LNCS*, pages 841–872, Hanoi, Vietnam, December 4–8, 2016. Springer, Cham.

FO18. Georg Fuchsbauer and Michele Orrù. Non-interactive Zaps of Knowledge. In Bart Preneel and Frederik Vercauteren, editors, *ACNS 2018*, volume 10892 of *LNCS*, pages 44–62, Leuven, Belgium, July 2–4, 2018. Springer, Heidelberg.

Fuc18. Georg Fuchsbauer. Subversion-Zero-Knowledge SNARKs. In Michel Abdalla, editor, *PKC 2018*, volume 10769 of *LNCS*, pages 315–347, Rio de Janeiro, Brazil, March 25–28, 2018. Springer, Cham.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic Span Programs and NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645, Athens, Greece, April 26–30, 2013. Springer, Heidelberg.

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 (1)*, volume 9452 of *LNCS*, pages 605–629, Auckland, New Zealand, November 29–December 3 2015. Springer, Cham.

GL07. Jens Groth and Steve Lu. A Non-interactive Shuffle with Pairing Based Verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, December 2–6, 2007. Springer, Heidelberg.

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In Robert Sedgewick, editor, *STOC 1985*, pages 291–304, Providence, Rhode Island, USA, May 6–8, 1985. ACM Press.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for Cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

GR16. Alonso González and Carla Ràfols. New Techniques for Non-interactive Shuffle and Range Arguments. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 2016*, volume 9696 of *LNCS*, pages 427–444, Guildford, UK, June 19–22, 2016. Springer, Heidelberg.

Gro10. Jens Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340, Singapore, December 5–9, 2010. Springer, Heidelberg.

Gro16. Jens Groth. On the Size of Pairing-based Non-interactive Arguments. In Marc Fischlin and Jean-Sebastien Coron, editors, *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 305–326, Vienna, Austria, May 8–12, 2016. Springer, Cham.

GS08. Jens Groth and Amit Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. In Nigel Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg.

GW11. Craig Gentry and Daniel Wichs. Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions. In Salil Vadhan, editor, *STOC 2011*, pages 99–108, San Jose, California, USA, June 6–8, 2011. ACM Press.

Ica09. Thomas Icart. How to Hash into Elliptic Curves. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316, Santa Barbara, California, USA, August 16–20, 2009. Springer, Heidelberg.

JR13. Charanjit S. Jutla and Arnab Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013 (1)*, volume 8269 of *LNCS*, pages 1–20, Bangalore, India, December 1–5, 2013. Springer, Heidelberg.

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 (2) 2014*, volume 8617 of *LNCS*, pages 295–312, Santa Barbara, California, USA, August 17–21, 2014. Springer, Heidelberg.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-Adaptive NIZK for Linear Subspaces Revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015*, volume 9057 of *LNCS*, pages 101–128, Sofia, Bulgaria, May 26–30, 2015. Springer, Cham.

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, Taormina, Italy, March 18–21, 2012. Springer, Heidelberg.

LPJY13. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly Homomorphic Structure-Preserving Signatures and Their Applications. In Ran Canetti and Juan Garay, editors, *CRYPTO (2) 2013*, volume 8043 of *LNCS*, pages 289–307, Santa Barbara, California, USA, August 18–22, 2013. Springer, Heidelberg.

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, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg.

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 (1)*, volume 9452 of *LNCS*, pages 681–707, Auckland, New Zealand, November 29–December 3 2015. Springer, Cham.

LZ12. Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. In Ivan Visconti and Roberto De Prisco, editors, *SCN 2012*, volume 7485 of *LNCS*, pages 477–502, Amalfi, Italy, September 5–7, 2012. Springer, Heidelberg.

Mau05. Ueli M. Maurer. Abstract Models of Computation in Cryptography. In Nigel P. Smart, editor, *Cryptography and Coding 2005*, pages 1–12, Cirencester, UK, December 19–21, 2005.

MR01. Silvio Micali and Leonid Reyzin. Soundness in the Public-Key Model. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 542–565, Santa Barbara, USA, August 19–23, 2001. Springer, Heidelberg.

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 (1)*, volume 10031 of *LNCS*, pages 729–758, Hanoi, Vietnam, December 4–8, 2016. Springer, Cham.

Nec94. V. I. Nechaev. Complexity of a Determinate Algorithm for the Discrete Logarithm. *Mathematical Notes*, 55(2):165–172, 1994. Translated from *Matematicheskie Zapiski*, 55(2):91–101, 1994.

PGHR13. Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly Practical Verifiable Computation. In *IEEE SP 2013*, pages 238–252, Berkeley, CA, USA, May 19-22, 2013. IEEE Computer Society.

Sho97. Victor Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 256–266, Konstanz, Germany, 11–15 May 1997. Springer, Heidelberg.

SV12. Alessandra Scafuro and Ivan Visconti. On Round-Optimal Zero Knowledge in the Bare Public-Key Model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 153–171, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg.

TK17. Mehdi Tibouchi and Taechan Kim. Improved elliptic curve hashing and point representation. *Des. Codes Cryptography*, 82(1–2):161–177, 2017.

VV09. Carmine Ventre and Ivan Visconti. Co-sound Zero-Knowledge with Public Keys. In Bart Preneel, editor, *AFRICACRYPT 2009*, volume 5580 of *LNCS*, pages 287–304, Gammarth, Tunisia, June 21–25, 2009. Springer, Heidelberg.

Wee07. Hoeteck Wee. Lower Bounds for Non-interactive Zero-Knowledge. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 103–117, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg.

A GBGM and Sub-GBGM

Generic Bilinear Group Model. Next, we will introduce the Generic Bilinear Group Model (GBGM) [Nec94, Sho97, Mau05, BBG05], by following the exposition in [ABLZ17].

We start by picking an asymmetric bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathsf{Pgen}(1^\lambda)$. Consider a black box $\mathbf{B}$ that stores values from additive groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ in internal state variables $\mathsf{cell}_1, \mathsf{cell}_2, \dots$, where for simplicity we allow the storage space to be infinite (this only increases the power of a generic adversary). The initial state consists of some values $(\mathsf{cell}_1, \mathsf{cell}_2, \dots, \mathsf{cell}_{|inp|})$, which are set according to some probability distribution. Each state variable cell_i has an accompanying type $\mathsf{type}_i \in \{1, 2, T, \perp\}$. Initially, $\mathsf{type}_i = \perp$ for $i > |inp|$. The black box allows computation operations on internal state variables and queries about the internal state. No other interaction with $\mathbf{B}$ is possible.

Let $\mathit{\Pi}$ be an allowed set of computation operations. A computation operation consists of selecting a (say, t-ary) operation $f \in \mathit{\Pi}$ together with $t + 1$ indices $i_1, i_2, \dots, i_{t+1}$. Assuming inputs have the correct type, $\mathbf{B}$ computes $f(\mathsf{cell}_{i_1}, \dots, \mathsf{cell}_{i_t})$ and stores the result in $\mathsf{cell}_{i_{t+1}}$. For a set $\mathit{\Sigma}$ of relations, a query consists of selecting a (say, t-ary) relation $\varrho \in \mathit{\Sigma}$ together with t indices $i_1, i_2, \dots, i_t$. Assuming inputs have the correct type, $\mathbf{B}$ replies to the query with $\varrho(\mathsf{cell}_{i_1}, \dots, \mathsf{cell}_{i_t})$. In the GBGM, we define $\mathit{\Pi} = \{+, \hat{e}\}$ and $\mathit{\Sigma} = \{=\}$, where

1. On input $(+, i_1, i_2, i_3)$: if $\mathsf{type}_{i_1} = \mathsf{type}_{i_2} \neq \perp$ then set $\mathsf{cell}_{i_3} \leftarrow \mathsf{cell}_{i_1} + \mathsf{cell}_{i_2}$ and $\mathsf{type}_{i_3} \leftarrow \mathsf{type}_{i_1}$.
2. On input $(\hat{e}, i_1, i_2, i_3)$: if $\mathsf{type}_{i_1} = 1$ and $\mathsf{type}_{i_2} = 2$ then set $\mathsf{cell}_{i_3} \leftarrow \hat{e}(\mathsf{cell}_{i_1}, \mathsf{cell}_{i_2})$ and $\mathsf{type}_{i_3} \leftarrow T$.
3. On input $(=, i_1, i_2)$: if $\mathsf{type}_{i_1} = \mathsf{type}_{i_2} \neq \perp$ and $\mathsf{cell}_{i_1} = \mathsf{cell}_{i_2}$ then return 1. Otherwise return 0.

Since we are proving lower bounds, we will give a generic adversary $\mathcal{A}$ additional power. We assume that all relation queries are for free. We also assume that $\mathcal{A}$ is successful if after τ operation queries, he makes an equality query $(=, i_1, i_2)$, $i_1 \neq i_2$, that returns 1; at this point $\mathcal{A}$ quits. Thus, if $\mathsf{type}_i \neq \perp$, then $\mathsf{cell}_i = F_i(\mathsf{cell}_1, \dots, \mathsf{cell}_{|inp|})$ for a polynomial F_i known to $\mathcal{A}$.

Sub-GBGM. By following [BFS16, ABLZ17], we enhance the power of generic bilinear group model. Since the power of the generic adversary will increase, security proofs in the resulting *Sub-GBGM* are more realistic than in the GBGM, see Section 2.

More precisely, we give the generic model adversary an additional power to effectively create new indeterminates Y_i in groups $\mathbb{G}_1$ and $\mathbb{G}_2$ (e.g., by hashing into elliptic curves), without knowing their values. Since $[Y]_1 [1]_2 = [Y]_T$ and $[1]_1 [Y]_2 = [Y]_T$, the adversary that has generated an indeterminate Y in $\mathbb{G}_\iota$ can also operate with Y in $\mathbb{G}_T$. Formally, $\mathit{\Pi}$ will contain one more operation create, with the following semantics:

4. On input (create, i, t): if $\mathsf{type}_i = \perp$ and $t \in \{1, 2, T\}$ then set $\mathsf{cell}_i \leftarrow_\$ \mathbb{Z}_p$ and $\mathsf{type}_i \leftarrow t$.

The semantics of create dictates that the actual value of the indeterminate Y_i is uniformly random in $\mathbb{Z}_p$, that is, the adversary cannot create indeterminates for which she does not know the discrete logarithm and that yet are not random.