# On QA-NIZK in the BPK Model

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

[1] University of Tartu, Tartu, Estonia
[2] Clearmatics, London, UK

**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. In particular, an S-ZK NIZK is zero knowledge even in the case of subverted CRS. We propose new definitions for S-ZK Quasi-Adaptive NIZKs (QA-NIZKs) where the CRS can depend on the language parameter. First, we observe that subversion zero knowledge (S-ZK) in the CRS model corresponds to no-auxiliary-string non-black-box 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 S-ZK. Second, we show that the language parameter $\varrho$ must be generated honestly. Importantly, this emphasizes the difference of $\varrho$ and the CRS. Third, we prove that the most efficient known QA-NIZK for linear subspaces by Kiltz and Wee (after possibly adding some new elements to its public key) is no-auxiliary-string non-black-box zero knowledge in the BPK model under a novel knowledge assumption that is secure in the subversion generic bilinear group model of Bellare *et al.* Hence, S-ZK can be achieved (almost) for free and is thus arguably the correct security definition for QA-NIZKs.

**Keywords:** Bare public key model, no-auxiliary-string zero knowledge, non-black-box 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 of knowledge (zk-SNARKs, [Gro10, Lip12, GGPR13, PHGR13, Lip13, DFGK14, Gro16]) are very useful in cryptographic applications, allowing the prover to create a succinct argument $\pi$ that can be transferred to many different verifiers who can check the correctness of the argument at their leisure time.

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 NIZK [MR01,APV05] and even auxiliary-string non-black-box [GO94, Wee07] (see Lemma 1) 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.

A pairing-based QA-NIZK argument system for linear subspaces allows the prover to convince the verifier that a vector of group elements[3] $[\boldsymbol{y}]_1$ belongs to the column space of a fixed public matrix $\varrho = [\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m}$, 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 may depend on $[\boldsymbol{M}]_1$. 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, $\mathit{\Pi}_{\mathsf{as}}$ and $\mathit{\Pi}'_{\mathsf{as}}$, for linear subspaces. Both are perfectly zero-knowledge and (quasi-adaptively) computationally sound in the CRS model under a suitable KerMDH assumption [MRV16]. $\mathit{\Pi}'_{\mathsf{as}}$ 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. $\mathit{\Pi}_{\mathsf{as}}$ works for any matrix distribution but has an argument that consists of $k + 1$ group elements. ($\mathit{\Pi}_{\mathsf{as}}$ 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, 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 succinct non-

[3] We assume pairing-based setting, and use the bracket notation of [EHK$^+$13] (see Section 2).

interactive zero knowledge arguments of knowledge (zk-SNARKs) in real-life applications like cryptocurrencies [BCG$^+$14]. Ben-Sasson *et al.* [BCG$^+$15] constructed an efficient multi-party protocol for the creation of CRS for (a subclass of) zk-SNARKs; however, it assumes that at least one of the CRS creators is honest. Bellare *et al.* [BFS16] defined subversion-resistant soundness (S-SND) and subversion-resistant zero knowledge (S-ZK) for NIZKs that guarantee either soundness or zero knowledge, resp., in the case all the creators of the CRS are subverted. In particular, Bellare *et al.* proved that it is impossible to simultaneously obtain S-SND and (even non-subversion-resistant) zero knowledge. On the other hand, they constructed a (non-succinct) computationally sound and computationally S-ZK NIZK argument system for $\mathsf{NP}$ where the S-ZK property relies on a knowledge assumption [Dam92].

S-ZK was further studied by Abdolmaleki *et al.* [ABLZ17] who defined S-ZK for zk-SNARK and proposed an S-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 S-ZK by constructing a (public) CRS-verification algorithm $\mathsf{CV}$. Essentially, $\mathsf{CV}$ accepts the given CRS $\mathsf{crs}$ iff $\mathsf{crs}$ is correctly computed starting from *some* simulation trapdoor $\mathsf{td}$. In the S-ZK proof of their SNARK, Abdolmaleki *et al.* constructed a simulator that, given $\mathsf{crs}$ as the input, first uses a knowledge assumption to recover $\mathsf{td}$ 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 (S-GBGM). S-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 has additional power to create group elements without knowing their discrete logarithms by hashing into an elliptic curve, [Ica09, BCI$^+$10, TK17]. See Section 2 for an explanation why S-GBGM is a weaker model than the GBGM.

Fuchsbauer [Fuc18] used a similar approach to define another S-ZK version of Groth's SNARK using a slightly different knowledge assumption, different simulation, and not requiring one to add elements to the CRS. Thus, essentially, one obtains S-ZK for free. Thereby, it seems that there is no reason to construct and deploy SNARKs that do *not* achieve S-ZK. It is only natural to ask if the same holds in the case of QA-NIZKs.

The knowledge assumptions of [ABLZ17,Fuc18] use crucially the fact that for each trapdoor element $\alpha$, the CRS of Groth's zk-SNARK and other well-known zk-SNARKs like [GGPR13, PHGR13] contains $[\alpha]_1$ together with some other $\alpha$-dependent group elements. Thus, these knowledge assumptions (that state that an adversary, who outputs $[\alpha]_1$ and some other well-formed $\alpha$-dependent group elements, knows $\alpha$) are trivially secure in the 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, relying on knowledge assumptions to prove the S-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-NIZK $\Pi_{\mathsf{as}}$ has a trapdoor matrix $\boldsymbol{K}$ but $[\boldsymbol{K}]_1$ is not explicitly given in the CRS. (In fact, the soundness proof of $\Pi_{\mathsf{as}}$ relies on the fact that $\boldsymbol{K}$ is ambiguous.) In the case of $\Pi'_{\mathsf{as}}$, $\boldsymbol{K}$ is uniquely fixed by the CRS via $[\bar{\boldsymbol{A}}, \bar{\boldsymbol{A}}\boldsymbol{K}]_2$, however, $[\boldsymbol{K}]_1$ is still not published. 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 a language parameter $\varrho$ that has to be modeled separately from other inputs; no such parameter exists in the case of SNARKs. Another important difference is 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 S-ZK of a QA-NIZK seems to be less justifiable than in the case of proving S-ZK of zk-SNARKs. Moreover, while Bellare *et al.* had a discussion motivating the use of knowledge assumptions to obtain S-ZK, they did not have a formal proof of their necessity. This brings us to the main questions of the current work:

*(i) Are knowledge assumptions or other non-black-box techniques needed to prove S-ZK of NIZKs for languages outside of* $\mathsf{BPP}$*?*
*(ii) Does the definition of S-ZK make sense if one is also allowed to subvert* $\varrho$*?*
*(iii) Can one easily modify existing QA-NIZKs for linear subspaces to obtain S-ZK?*
*(iv) Can one, similarly to SNARKs, get S-ZK for free?*

**Our Contributions.** We answer to the above main questions (with yes, no, yes, and mostly yes). It turns out that achieving S-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 $\varrho$ and the CRS behave very differently if one cannot trust the CRS creator; since state-of-the-art SNARKs have no $\varrho$, 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 S-ZK under a novel knowledge assumption given a suitable $\mathsf{CV}$ algorithm. Hence, S-ZK in this case comes for free.

First, we make a conceptually important observation that S-ZK in the CRS model, as defined in [BFS16,ABLZ17,Fuc18], is equal to *no-auxiliary-string non-black-box* zero knowledge in the BPK model [CGGM00, MR01]. Recall that in the BPK model, only the verifier needs to have a public key and the key authority executes the functionality of an immutable bulletin board by storing the received public keys. One achieves designated-verifier zero knowledge 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. BPK is significantly weaker than the CRS model and it is arguably the weakest (public key or parameter based) trust model possible.

This important positive connection between no-auxiliary-string non-black-box zero knowledge and S-ZK was missed in the prior work on S-ZK; we hope it will simplify the construction and analysis of future S-ZK argument systems. Because of that connection, we will usually use the abbreviation S-ZK to denote no-auxiliary-string non-black-box zero knowledge, but we will mostly emphasize that we are working in the BPK model.

Since three messages are needed to achieve auxiliary-string zero knowledge in the plain model for languages outside of $\mathsf{BPP}$ [GO94], it follows that in the BPK model, auxiliary-string non-black-box NIZK is possible only for languages in $\mathsf{BPP}$. This provides a simple proof that one can only construct non-auxiliary-string non-black-box NIZK for languages outside of $\mathsf{BPP}$ and thus provides an answer to the open question (i).

In Section 3, we carefully define the security of QA-NIZK arguments in the BPK model, modifying standard QA-NIZK definitions. However, we model the definition of no-auxiliary-string non-black-box zero knowledge (i.e., S-ZK) for QA-NIZK after the S-ZK definition for SNARK of Abdolmaleki *et al.* [ABLZ17]. More precisely, we require that for any efficient malicious public-key creator (either the verifier or a third party) $\mathcal{Z}$, there exists an efficient extractor $\mathsf{Ext}_{\mathcal{Z}}$, s.t. if $\mathcal{Z}$, by using a correctly sampled language parameter $\varrho$ and any random coins $r$ as an input, generates a public key $\mathsf{pk}$ (since there is no auxiliary input, $\mathsf{pk}$ *has* to be generated by $\mathcal{Z}$) then $\mathsf{Ext}_{\mathcal{Z}}$, given the same input and $r$, outputs the secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$.

We emphasize that $\mathcal{Z}$ obtains $\varrho$ as an input (from a fixed distribution $\mathscr{D}_{\mathsf{p}}$) instead of generating it. This is to be expected since a QA-NIZK argument system is defined for a fixed distribution $\mathscr{D}_{\mathsf{p}}$ of $\varrho$. Jutla and Roy [JR13] explicitly say that $\varrho$ 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 $\boldsymbol{M}$ (the discrete logarithm of the language parameter) 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 $\mathscr{D}_{\mathsf{p}}$ is trusted to not leak information and also works as a black-box (that is, one cannot obtain any extra information about $\varrho$ even when using a knowledge assumption). However, $\mathsf{pk}$ can be fully subverted. Since this distinction is at the core of the difference between QA-NIZKs and (adaptive) NIZKs, it is perhaps not surprising that $\varrho$ and $\mathsf{pk}$ need to be handled differently. Our results, albeit being somewhat negative, further clarify the distinction between $\varrho$ and $\mathsf{pk}$. This answers to the open question (ii). Moreover, as all impossibility results, it hopefully helps to focus subsequent work by narrowing down possible approaches in constructing S-ZK NIZK argument systems. See Sections 3 and 5 for further discussion.

As the next main contribution, we study a variant $\Pi_{\mathsf{bpk}}$ of the Kiltz-Wee QA-NIZK $\Pi'_{\mathsf{as}}$ [KW15] in the BPK model. More precisely, $\mathsf{pk}$ of $\Pi_{\mathsf{bpk}}$ includes a new component $\mathsf{pk}^{\mathsf{pkv}}$ that helps to publicly check that an adversarially generated matrix $[\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k \times k}$ in $\mathsf{pk}$ has full rank $k$. Similarly to [ABLZ17], we also define an efficient public-key verification algorithm $\mathsf{PKV}$. We emphasize that we chose to analyse $\Pi'_{\mathsf{as}}$ since it is the most efficient known QA-NIZK for linear subspaces. We will leave analysing other QA-NIZKs (that will hopefully be easier to do following our definitional framework and analysis of $\Pi'_{\mathsf{as}}$) to the further work.

Since in the case $k = 1$, we do not modify the public-key generation and the prover (then, essentially $\Pi'_{\mathsf{as}} = \Pi_{\mathsf{bpk}}$), the (non-subversion) soundness of $\Pi_{\mathsf{bpk}}$ in the BPK model follows directly from [KW15]. In the case $k = 2$ (then $\mathsf{pk}$ contain some extra elements), we prove the (non-subversion) soundness of $\Pi_{\mathsf{bpk}}$ under the SKerMDH assumption of [GHR15].

We prove that $\Pi_{\mathsf{bpk}}$ is statistically S-ZK in the BPK model under either one of the two new knowledge assumptions KWKE (the *Kiltz-Wee Knowledge of Exponent* assumption) and SKWKE (the *strong* KWKE assumption), assuming that its whole $\mathsf{pk}$ is generated by the verifier or a verifier-trusted authority — even if we are set to prove S-ZK that interests the prover. Intuitively, (S)KWKE guarantees that if an adversary $\mathcal{A}$ has succeeded in creating a $\mathsf{pk}$ accepted by $\mathsf{PKV}$ then one can extract corresponding $\mathsf{sk} = \boldsymbol{K}$. We prove that both assumptions hold in the S-GBGM (see Theorem 1). The quite intricate proof of Theorem 1 heavily depends on the fact that we work in the S-GBGM. More precisely, in the S-GBGM we can extract some outputs of $\mathcal{A}$ as polynomials in indeterminates created by $\mathcal{A}$. To extract an integer $\mathsf{sk}$, we use the Schwartz-Zippel lemma and let the extractor output an evaluation of the polynomials at a random point. We then use the specific form of $\mathsf{PKV}$ to argue that such $\mathsf{sk}$ is correct. In the case of SKWKE, we evaluate the polynomials at two random points and use a more complicated argument, see Theorem 1.

Interestingly, under KWKE we only get the guarantee that the part $\mathsf{pk}^{\mathsf{zk}}$ of the $\mathsf{pk}$, used either by the prover or the simulator, has been correctly computed. This however suffices to prove that $\Pi_{\mathsf{bpk}}$ is S-ZK. (Thus, S-ZK can be achieved even if the correctness of the whole public key cannot be verified.) Hence, in the case $k = 1$ (but not when $k = 2$) one can get S-ZK for free. This is important since it means that in the case $k = 1$ we get a stronger security property (S-ZK) without having to design a new, more complicated and less efficient QA-NIZK. Arguably, the most efficient case $k = 1$ is the only really interesting case, and the case $k = 2$ is only needed in some applications (e.g., when one wants to rely on a weaker assumption). This answers to the open questions (iii) and (iv).

Then, we show that under a stronger knowledge assumption SKWKE, one can guarantee that the whole $\mathsf{pk}$ has been correctly computed. However, as a drawback, the SKWKE assumption holds in the S-GBGM only if the language parameter $[\boldsymbol{M}]_1$ comes from a suitable hard distribution. (The latter is often the case in QA-NIZK applications, where $[\boldsymbol{M}]_1$ 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.

In Section 6, we mention that the QA-NIZKs of the current paper can be made black-box zero-knowledge in the stronger Registered Public Key (RPK, [BCNP04]). We also discuss the relation between the BPK model (as used in the current paper) and the RPK model.

**Recent Work.** Our results have been used in several recent (though, up to our knowledge, yet unpublished) eprints. Lipmaa [Lip19] has used the new S-ZK QA-NIZK to construct an S-ZK QA-NIZK where, additionally, the public key and the argument can be updated. Independently, González and Ràfols [GR19] constructed an updatable S-ZK SNARK. Interestingly, both papers used the KWKE assumption of (an early version of) the current paper in situations where it is not tautological, showing that the KWKE assumption may have wider applications. It also confirms that S-ZK QA-NIZK is an interesting building block that can be used to construct other, more complicated, protocols like SNARKs.

## 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 $\mathcal{A}$. By $x \leftarrow_{\$} \mathscr{D}$ we denote that $x$ is sampled according to distribution $\mathscr{D}$ or uniformly at random if $\mathscr{D}$ 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 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 $g_\iota$ 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 Bare Public Key (BPK) model [CGGM00,MR01], parties have access to a public file $F$, a polynomial-size collection of records $(id, \mathsf{pk}_{id})$, where $id$ is a string identifying a party (e.g., a verifier), and $\mathsf{pk}_{id}$ 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 $1^\lambda$ and randomizer $r$, $\mathcal{P}_{id}$ outputs a public key $\mathsf{pk}_{id}$ and stores the corresponding secret key $\mathsf{sk}_{id}$. We assume the *no-auxiliary-string BPK* model where from this it follows that $\mathcal{P}_{id}$ actually created $\mathsf{pk}_{id}$. After that, $F$ will include $(id, \mathsf{pk}_{id})$. In stage two, each party has access to $F$, while $\mathcal{P}_{id}$ has possibly access to $\mathsf{sk}_{id}$ (however, the latter is not required by us). 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.

**Zero Knowledge.** In a zero-knowledge proof or argument system [GMR85, BCC88], a prover convinces the verifier in the veracity of a statement without leaking any side information except that the statement is true. Here, a proof (resp., an argument) system guarantees soundness against an unbounded (resp., a PPT) cheating prover. The zero-knowledge property is proven by constructing a simulator that can simulate the view of a cheating verifier without knowing the secret information (witness) of the prover. A non-interactive zero knowledge proof or argument system [BFM88] consists of just one message by the prover.

We will only deal with no-auxiliary-string non-black-box NIZK argument systems in the plain model, but to explain this choice, it is important to know that there are many possibility and impossibility results about zero knowledge in 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 $\mathsf{BPP}$ requires at least 4 rounds. Goldreich and Oren [GO94] proved that three rounds are needed for auxiliary-string zero knowledge in the plain model. From this it follows that there exists no *auxiliary-string non-black-box* NIZK argument system in the BPK model for a language $\mathcal{L}$ outside of $\mathsf{BPP}$, see Lemma 1. (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 $\mathcal{L} \in \mathsf{NP}$, 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 that weak nonuniform NIZK proof systems are only possible for languages in $\mathsf{P/poly}$. On the other hand, Wee constructed a weak nonuniform NIZK argument system for $\mathsf{NP}$, assuming subexponential hardness results. Here, nonuniformity means that the simulator can depend on the adversary and weakness means that the simulator size can both depend on the distinguisher and the distinguishing gap; in particular, in Wee's weak nonuniform NIZK argument system for $\mathsf{NP}$, the simulator works in the quasipolynomial time. While quasipolynomial-time simulation [Pas03] *per se* is a valid property, it means that Wee's argument system combines the use of quasipolynomial-time simulation with the use of subexponential hardness assumptions, an adversary-dependent simulator, and of the BPK. Moreover, it is unknown how to instantiate Wee's argument system efficiently.

**Matrix Diffie-Hellman Assumptions.** Kernel Matrix Diffie-Hellman Assumption (KerMDH) is a well-known assumption family formally introduced in [MRV16]. Let $\mathscr{D}_{\ell k}$ be a probability distribution over matrices in $\mathbb{Z}_p^{\ell \times k}$, where $\ell > k$. Assume that $\mathscr{D}_{\ell k}$ outputs matrices $\boldsymbol{A}$ where the upper $k \times k$ submatrix $\bar{\boldsymbol{A}}$ is always invertible. (I.e., $\mathscr{D}_{\ell k}$ is *robust*, [JR13].) Denote the lower $(\ell - k) \times k$ submatrix of $\boldsymbol{A}$ as $\underline{\boldsymbol{A}}$. Denote $\mathscr{D}_k = \mathscr{D}_{k+1,k}$.

$\mathscr{D}_{\ell k}$-KerMDH$_{\mathbb{G}_1}$ [MRV16] holds relative to $\mathsf{Pgen}$, if for any PPT $\mathcal{A}$,

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

$\mathscr{D}_{\ell k}$-SKerMDH [GHR15] holds relative to $\mathsf{Pgen}$, if for any PPT $\mathcal{A}$,

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

According to Lem. 1 of [GHR15], if $\mathscr{D}_{\ell k}$-KerMDH holds in generic symmetric bilinear groups then $\mathscr{D}_{\ell k}$-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.

**Generic 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* (S-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 S-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, S-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 S-GBGM. This is since in the S-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, an S-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 proof in the GBGM have quite a complicated proof in the S-GBGM.

See Appendix A for a long introduction to GBGM and S-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 $\varrho$ that has been sampled from a fixed distribution $\mathscr{D}_{\mathsf{p}}$. 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 QA-NIZKs for other languages).

The original QA-NIZK security definitions [JR13] were given in the CRS model. In what follows, we will lift them to the weaker BPK model. Sometimes, 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 S-ZK zk-SNARKs in [ABLZ17], e.g., a QA-NIZK in the BPK model will have a public-key verification algorithm $\mathsf{PKV}$ and the zero knowledge definition mentions a subverter and an extractor. Since black-box [MR01, APV05] and even auxiliary-input non-black-box [GO94] (see Lemma 1) NIZK in the BPK model is impossible we will give an explicit definition of no-auxiliary-string non-black-box NIZK.

As in [BFS16], we will implicitly assume that the system parameters $\mathsf{p}$ are generated deterministically from $\lambda$; in particular, the choice of $\mathsf{p}$ cannot be subverted. A QA-NIZK argument system enables to prove membership in a language defined by a relation $\mathcal{R}_\varrho = \{(x, w)\}$, which in turn is completely determined by a parameter $\varrho$ sampled from a distribution $\mathscr{D}_{\mathsf{p}}$.[4] In the proof of zero knowledge, we will assume that $\mathscr{D}_{\mathsf{p}}$ 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 $\mathscr{D}_{\mathsf{p}}$ is reasonable; for example, it should not be the case that all languages $\mathcal{L}_\varrho$ for $\varrho \in \mathscr{D}_{\mathsf{p}}$ are easy to decide. (See additional discussion at the end of the current section and in Section 5.) We will assume implicitly that $\varrho$ contains $\mathsf{p}$ and thus not include $\mathsf{p}$ as an argument to algorithms that also input $\varrho$. A distribution $\mathscr{D}_{\mathsf{p}}$ on $\mathcal{L}_\varrho$ is *witness-sampleable* [JR13] if there exists a PPT algorithm $\mathscr{D}'_{\mathsf{p}}$ that samples $(x, w) \in \mathcal{R}_\varrho$ such that $\varrho$ is distributed according to $\mathscr{D}_{\mathsf{p}}$, and membership of $x$ in *the parameter language* $\mathcal{L}_\varrho$ can be verified in PPT given $w$.

While the verifier's public key $\mathsf{pk}$ may depend on $\varrho$ (however, we assume that $\varrho$ 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 $\mathcal{R}_{\mathsf{p}} = \{\mathcal{R}_\varrho\}_{\varrho \in \mathrm{Supp}(\mathscr{D}_{\mathsf{p}})}$; that is, one usually requires *uniform simulation* (see [JR13] for a discussion).Following [ABLZ17], we accompany the universal simulator with an adversary-dependent extractor. The simulator is not allowed to create new $\varrho$ 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 *no-auxiliary-string non-black-box zero knowledge (S-ZK) QA-NIZK argument system* in the

[4] In the QA-NIZK literature, it is assumed that samples from $\mathscr{D}_{\mathsf{p}}$ 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 $\varrho$ securely and not leak corresponding secret.

BPK model for a set of witness-relations $\mathcal{R}_{\mathsf{p}} = \{\mathcal{R}_\varrho\}_{\varrho \in \mathrm{Supp}(\mathscr{D}_{\mathsf{p}})}$ with $\varrho$ sampled from a distribution $\mathscr{D}_{\mathsf{p}}$ over associated parameter language $\mathcal{L}_{\mathsf{p}}$, if the following properties (i-iii) hold. Here, $\mathsf{Pgen}$ is the parameter generation algorithm, $\mathsf{K}$ is the public key generation algorithm, $\mathsf{PKV}$ is the public key verification algorithm, $\mathsf{P}$ is the prover, $\mathsf{V}$ is the verifier, and $\mathsf{Sim}$ is the simulator.

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

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

(ii) **Computational Quasi-Adaptive Soundness:** for any PPT $\mathcal{A}$,

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

(iii) **Statistical No-Auxiliary-String Non-Black-Box Zero Knowledge (S-ZK):** for any PPT subverter $\mathcal{Z}$ there exists a PPT extractor $\mathsf{Ext}_{\mathcal{Z}}$, such that for any $\lambda$, $\mathsf{p} \in \mathsf{Pgen}(1^\lambda)$, and computationally unbounded adversary $\mathcal{A}$, $\varepsilon_0^{zk} \approx_\lambda \varepsilon_1^{zk}$, where $\varepsilon_b^{zk} =$

$$\Pr \begin{bmatrix} \varrho \leftarrow_{\$} \mathscr{D}_{\mathsf{p}}; r \leftarrow_{\$} \mathsf{RND}(\mathcal{Z}); (\mathsf{pk}, \mathsf{aux}_{\mathcal{Z}}) \leftarrow \mathcal{Z}(\varrho; r); \mathsf{sk} \leftarrow \mathsf{Ext}_{\mathcal{Z}}(\varrho; r) : \\ \mathsf{PKV}(\varrho, \mathsf{pk}) = 1 \wedge \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\varrho, \mathsf{pk}, \mathsf{aux}_{\mathcal{Z}}) = 1 \end{bmatrix} \enspace .$$

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 $\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, x)$.

The extractor never works with probability 1 since $\mathcal{Z}$ can randomly sample (with a non-zero but negligible probability) a well-formed $\mathsf{pk}$. 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 $\mathsf{Ext}_{\mathcal{Z}}$ is not unique in our case, see discussion in Section 4.)

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

**Comparison to Earlier S-ZK Definitions.** 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 S-SND (soundness even if the CRS was generated maliciously) and zero knowledge at the same time. Thus, subsequent efforts have concentrated on achieving either S-SND and witness-indistinguishability [BFS16], subversion knowledge-soundness and witness-indistinguishability [FO18], or S-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 $\mathsf{V}$ while (following the definitions of [ABLZ17]) the prover checks that the CRS is well-formed by using a publicly available algorithm. Thus, S-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) $\mathsf{V}$ and hence can be equal to the public key of an entity trusted by $\mathsf{V}$ (or of $\mathsf{V}$ herself). Since black-box NIZK [MR01] and even auxiliary-string non-black-box NIZK [GO94] in the BPK model is impossible, one has to define no-auxiliary-string non-black-box zero knowledge (S-ZK) as above. Bellare *et al.* [BFS16] motivated not incorporating auxiliary strings to the definition of S-ZK by known impossibility results. We will formalize this (folklore, see [Wee07] for discussion) impossibility result as the following straightforward lemma.

**Lemma 1.** *Auxiliary-string non-black-box NIZK in the BPK model is only possible for languages in* $\mathsf{BPP}$*.*

*Proof.* The notions of (no-)auxiliary-string and (non-)-black-box zero knowledge were defined by Goldreich and Oren [GO94] who proved that auxiliary-string (even non-black-box) zero knowledge argument systems for languages outside of $\mathsf{BPP}$ require at least three messages in the plain model. An auxiliary-string (non-black-box) NIZK argument system in the BPK model can be interpreted as a two-message auxiliary-string (non-black-box) zero knowledge argument system in the plain model, where the verifier creates BPK and sends it as her first message. Thus, an auxiliary-string NIZK argument system for languages outside of $\mathsf{BPP}$ would contradict the impossibility result of [GO94]. □

Note that [Wee07, Section 1.2] (falsely) claimed that the impossibility result of Goldreich and Oren rules out NIZK argument systems in the BPK model for languages outside $\mathsf{BPP}$, without qualifying that [GO94] only rules out auxiliary-string NIZK.

As noted in [BFS16], auxiliary-input zero knowledge is usually used to achieve sequential composition in the case of interactive zero knowledge. The given definition of S-ZK guarantees sequential security in the case of NIZK, see [ABLZ17] for a proof. In particular, the main result of [ABLZ17,Fuc18], reformulated in our language, is that there exist computationally knowledge-sound S-ZK zk-SNARKs for $\mathsf{NP}$ in the BPK model.

**Language of linear subspaces and Kiltz-Wee QA-NIZK.** An important application of QA-NIZK is in the case of the following language. Assume we need to show that $[\boldsymbol{y}]_1 \in \mathrm{colspace}([\boldsymbol{M}]_1)$, where $[\boldsymbol{M}]_1$ is sampled from a distribution $\mathscr{D}_{\mathsf{p}}$ over $\mathbb{G}_1^{n \times m}$. We assume, following [JR13], that $(n, m)$ is implicitly fixed by $\mathscr{D}_{\mathsf{p}}$. That is, a QA-NIZK for linear subspaces handles languages

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

```
$\mathsf{K}([\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m})$: $\boldsymbol{A} \leftarrow_{\$} \mathcal{D}_k$; $\boldsymbol{K} \leftarrow_{\$} \mathbb{Z}_p^{n \times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n \times k}$; $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in$
    $\mathbb{Z}_p^{m \times k}$; $\mathsf{pk} \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [\boldsymbol{P}]_1)$; $\mathsf{sk} \leftarrow \boldsymbol{K}$; return $(\mathsf{pk}, \mathsf{sk})$;
$\mathsf{P}([\boldsymbol{M}]_1, \mathsf{pk}, [\boldsymbol{y}]_1, \boldsymbol{w})$: return $[\boldsymbol{\pi}]_1 \leftarrow [\boldsymbol{P}]_1^\top \boldsymbol{w} \in \mathbb{G}_1^k$;
$\mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_1)$: return $[\boldsymbol{\pi}]_1 \leftarrow \boldsymbol{K}^\top [\boldsymbol{y}]_1 \in \mathbb{G}_1^k$;
$\mathsf{V}([\boldsymbol{M}]_1, \mathsf{pk}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$ : check that $[\boldsymbol{y}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{\pi}]_1^\top [\bar{\boldsymbol{A}}]_2$;
```

**Fig. 1.** Kiltz-Wee QA-NIZK argument system $\Pi'_{\mathsf{as}}$ for $[\boldsymbol{y}]_1 = [\boldsymbol{M}]_1 \boldsymbol{w}$

The corresponding relation is defined as $\mathcal{R}_{[\boldsymbol{M}]_1} = \{([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathbb{G}_1^n \times \mathbb{Z}_p^m : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}\}$. This language is useful in many applications, [JR13]. As a typical application, let $[\boldsymbol{M}]_1 = [1, \mathsf{sk}]_1^\top$ be public key of the Elgamal cryptosystem; then ciphertext $[\boldsymbol{y}]_1 \in \mathcal{L}_{[\boldsymbol{M}]_1}$ iff it encrypts 0. Here, $[\boldsymbol{M}]_1$ comes from a KerMDH-hard witness-sampleable distribution $\mathcal{D}_{\mathsf{p}}$.

The most efficient known QA-NIZK for linear subspaces in the CRS model was proposed by Kiltz and Wee [KW15] (see also [ABP15,Ben16]). In particular, they proposed a QA-NIZK $\Pi'_{\mathsf{as}}$ that assumes that the parameter $\varrho = [\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m}$ is sampled from a witness-sampleable distribution $\mathcal{D}_{\mathsf{p}}$. $\Pi'_{\mathsf{as}}$ 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. More precisely, given $n > m$, the Kiltz-Wee QA-NIZK is computationally quasi-adaptively sound under the $\mathcal{D}_k$-KerMDH$_{\mathbb{G}_1}$ assumption relative to $\mathsf{Pgen}$, [KW15]. Importantly, $\Pi'_{\mathsf{as}}$ is significantly more efficient than the Groth-Sahai NIZK [GS08] for the same language. For the sake of completeness, Fig. 1 describes the Kiltz-Wee QA-NIZK argument system $\Pi'_{\mathsf{as}}$ for linear subspaces in the CRS model.

**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, $\varrho$ is set by a trusted third party. For example, $\varrho$ could be his public key. As also argued by Jutla and Roy, in many applications, that party has no motivation to cheat while generating $\varrho$ since the security is defined with respect to this key. They mention that if $\varrho$ 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 $\mathsf{pk}$ might be generated by malicious $\mathcal{Z}$. In this case, $\mathcal{Z}$ should not generate $\varrho$, partially since a QA-NIZK argument system is defined for a fixed distribution of $\varrho$ and partially due to simple attacks that become possible if $\mathcal{Z}$ just leaks $\varrho$. We provide thorough discussion on this in Section 5, just noting here that since $\varrho$ 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 $\mathcal{Z}$.

The notion of QA-NIZK in the BPK model is important when $\varrho$ is not generated by the verifier but either by the prover or some (trusted) third party. In particular, recall that [KW15] proposed two different QA-NIZKs, $\Pi_{\mathsf{as}}$ and $\Pi'_{\mathsf{as}}$ where the latter for its *soundness* requires $\varrho = [\boldsymbol{M}]_1$ to come from a witness-sampleable distribution. Thus, in the case of $\Pi'_{\mathsf{as}}$, $[\boldsymbol{M}]_1$ should be created honestly.

**An Application 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 $\varrho$ together with a QA-NIZK public key $\mathsf{pk}$ and $\mathsf{P}$ opens the commitments later by disclosing a QA-NIZK argument of proper commitment under the commitment key $\varrho$. Here, $\varrho$ should not be generated by $\mathsf{P}$ (who could then equivocate) or by $\mathsf{V}$ (who could then extract the message). However, $\mathsf{pk}$ can be generated by $\mathsf{V}$. This allows one, securely generated $\varrho$, to be used in many applications, from UC commitments to identity-based encryption. In each such application, a trusted authority trusted by $\mathsf{V}$ (e.g., $\mathsf{V}$ herself) can create her $\mathsf{pk}$ that takes the particularities of that application into account.

## 4 A QA-NIZK in the BPK Model

In this section, we will show that if $k \in \{1, 2\}$ then a slight variant $\Pi_{\mathsf{bpk}}$ of $\Pi'_{\mathsf{as}}$ is secure in the BPK model. We assume that the public key $\mathsf{pk}$ (corresponds to the CRS in $\Pi'_{\mathsf{as}}$ together with $\mathsf{pk}^{\mathsf{pkv}}$ that makes it possible to evaluate $\mathsf{PKV}$ efficiently) belongs either to the verifier $\mathsf{V}$ or to a party, trusted by $\mathsf{V}$. That is, we prove computational soundness in the setting where $\mathsf{V}$ trusts $\mathsf{pk}$ is honestly generated, i.e., that the corresponding $\mathsf{sk}$ is secret and $\mathsf{pk}$ is well-formed. Since $\mathsf{pk}$ is not trusted by the prover $\mathsf{P}$, we prove S-ZK in the case of a maliciously generated $\mathsf{pk}$. As motivated in Section 3 (see also Section 5), we assume that $[\boldsymbol{M}]_1$ is sampled honestly from a witness-sampleable distribution and moreover, neither $\mathsf{V}$ nor the simulator knows the corresponding witness $\boldsymbol{M}$ or any function of $\boldsymbol{M}$ not efficiently computable from $[\boldsymbol{M}]_1$.

To modify $\Pi'_{\mathsf{as}}$ so that it would be secure in the BPK model instead of the CRS model, the simplest idea is to divide $\mathsf{pk}$ into $\mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_1$ (the part of $\mathsf{pk}$ that is used by $\mathsf{P}$ and thus intuitively needed to guarantee zero knowledge) and $\mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2$ (the part of $\mathsf{pk}$ is used by $\mathsf{V}$ and thus intuitively needed to guarantee soundness). Thus, $\mathsf{P}$ (resp., $\mathsf{V}$) has to be assured that $\mathsf{pk}^{\mathsf{zk}}$ (resp., $\mathsf{pk}^{\mathsf{snd}}$) is generated honestly. Hence, one could use $\mathsf{pk}^{\mathsf{zk}}_{\mathsf{P}}$ from $\mathsf{P}$'s public key and $\mathsf{pk}^{\mathsf{snd}}_{\mathsf{V}}$ from the $\mathsf{V}$'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 $\boldsymbol{K}$. Moreover, in this case both $\mathsf{P}$ and $\mathsf{V}$ have public keys while we want to have a situation, common in the BPK model, where only $\mathsf{V}$ has a public key.

Instead, we assume that $\mathsf{V}$'s public key $\mathsf{pk}$ is equal to the whole CRS and then construct a public-key verification algorithm $\mathsf{PKV}$. For this, we also need to add some new elements (collectively denoted as $\mathsf{pk}^{\mathsf{pkv}}$) to $\mathsf{pk}$. We prove that in the BPK model, the resulting QA-NIZK $\Pi_{\mathsf{bpk}}$ is computationally quasi-adaptively sound under either a KerMDH assumption ($k = 1$) or a SKerMDH assumption ($k = 2$) and S-ZK under a novel knowledge assumption. In fact, we define two different (tautological) knowledge assumptions, KWKE (Kiltz-Wee Knowledge of Exponent assumption) and SKWKE (Strong Kiltz-Wee Knowledge of Exponent assumption). The knowledge assumption is used to equip the simulator $\mathsf{Sim}$ of $\Pi'_{\mathsf{as}}$ with the correct secret key $\mathsf{sk} = \boldsymbol{K}$.

The assumption KWKE guarantees that one can extract a secret key $\mathsf{sk} = \boldsymbol{K}$ from which one can compute $\mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_1$ but not necessarily $\mathsf{pk}^{\mathsf{snd}}$. Since $\mathsf{pk}^{\mathsf{zk}}$ does not fix $\boldsymbol{K}$ uniquely, KWKE extracts one possible $\boldsymbol{K}$. Since for achieving S-ZK, it is not needed that $\mathsf{pk}^{\mathsf{snd}}$ can be computed from $\mathsf{sk}$, KWKE is sufficient. To argue that KWKE is a reasonable knowledge assumption, we prove that it holds in the S-GBGM.

We also introduce a stronger knowledge assumption SKWKE that allows to extract the *unique* secret key $\boldsymbol{K}$ that was used to generate the *whole* public key $\mathsf{pk}$. We prove SKWKE holds in the S-GBGM given that $\varrho = [\boldsymbol{M}]_1$ is chosen from a hard distribution. The latter assumption often holds in practice, e.g., when $\varrho$ 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 $\Pi_{\mathsf{bpk}}$ is S-ZK under either KWKE or SKWKE where in the latter case we additionally get a guarantee that the public key is correctly formed.

Since in the case $k = 1$, we did not modify $\Pi_{\mathsf{bpk}}$ (we only defined $\mathsf{PKV}$ for $\Pi_{\mathsf{bpk}}$), its completeness and computational soundness follow from [KW15]. Since there are applications (e.g., in the setting of symmetric pairing) where one might want to use $k = 2$, we prove that in this case, $\Pi_{\mathsf{bpk}}$ is sound under a SKerMDH assumption. Intuitively, $\mathsf{pk}^{\mathsf{pkv}}$ contains additional elements, needed to efficiently check that $[\bar{\boldsymbol{A}}]_2$ has full rank. If $k = 1$ then $\mathsf{pk}^{\mathsf{pkv}} = \varepsilon$ (empty string). The larger is $k$, the more elements $\mathsf{pk}^{\mathsf{pkv}}$ will contain. Since for efficiency reasons, one is interested in only small values of $k$, we will not consider the case $k > 2$ at all.

We will now define new knowledge assumptions. In KWKE, we assume that if $\mathcal{A}$ outputs a $\mathsf{pk}$ accepted by $\mathsf{PKV}$ 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 $\mathsf{pk}^{\mathsf{zk}}$. SKWKE will additionally guarantee that the same $\boldsymbol{K}$ was used to compute $\mathsf{pk}^{\mathsf{snd}}$. We emphasize that $[\boldsymbol{M}]_1 \leftarrow_{\$} \mathscr{D}_{\mathsf{p}}$ is given as an input to $\mathcal{A}$.

**Definition 1.** *Fix $k \in \{1, 2\}$ and $n > m \geq 1$. Let* $\mathsf{PKV}$ *be as in Fig. 3. Then* $(\mathscr{D}_{\mathsf{p}}, k)$-KWKE$_{\mathbb{G}_1}$ *(resp.,* $\boxed{(\mathscr{D}_{\mathsf{p}}, k)\text{-SKWKE}_{\mathbb{G}_1}}$*) holds relative to* $\mathsf{Pgen}$ *if for any* $\mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$ *and PPT adversary $\mathcal{A}$, there exists a PPT extractor* $\mathsf{Ext}_{\mathcal{A}}$*, s.t.*

$$\Pr\left[\begin{aligned} &[\boldsymbol{M}]_1 \leftarrow_{\$} \mathscr{D}_{\mathsf{p}}; r \leftarrow_{\$} \mathsf{RND}(\mathcal{A}); \mathsf{pk} \leftarrow \mathcal{A}([\boldsymbol{M}]_1; r); \boldsymbol{K} \leftarrow \mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1; r) : \\ &\mathsf{pk} = ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [\boldsymbol{P}]_1, \mathsf{pk}^{\mathsf{pkv}}) \wedge \mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1 \wedge \\ &(\boldsymbol{P} \neq \boldsymbol{M}^\top \boldsymbol{K} \boxed{\vee \boldsymbol{C} \neq \boldsymbol{K}\bar{\boldsymbol{A}}}) \end{aligned}\right] \approx_\lambda 0 \enspace .$$

*Here, the* $\boxed{\textit{boxed}}$ *part is only present in the definition of* SKWKE.

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

**Definition 2.** $\mathscr{D}_{\ell k}$-WKerMDH$_{\mathbb{G}_1}$ *holds relative to* $\mathsf{Pgen}$*, if for any PPT $\mathcal{A}$,* $\Pr[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathscr{D}_{\ell k}; \boldsymbol{c} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1) : \boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_k \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell] \approx_\lambda 0$.

```
$\mathsf{isinvertible}([\bar{\boldsymbol{A}}]_2, \mathsf{pk}^{\mathsf{pkv}})$
$\textbf{if } k = 1 \textbf{ then}$ check $\mathsf{pk}^{\mathsf{pkv}} = \epsilon \wedge [a_{11}]_2 \neq [0]_2$
$\textbf{else}$ check $\mathsf{pk}^{\mathsf{pkv}} = [a_{11}, a_{12}]_1 \in \mathbb{G}_1^{1 \times 2} \wedge [a_{11}]_1[1]_2 = [1]_1[a_{11}]_2 \wedge$
        $[a_{12}]_1[1]_2 = [1]_1[a_{12}]_2 \wedge [a_{11}]_1[a_{22}]_2 - [a_{12}]_1[a_{21}]_2 \neq [0]_T; \textbf{fi}$
```

**Fig. 2.** Auxiliary procedure isinvertible for $k \in \{1, 2\}$.

```
$\mathsf{K}([\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m})$: $\boldsymbol{A} \leftarrow_\$ \mathcal{D}_k$; $\boldsymbol{K} \leftarrow_\$ \mathbb{Z}_p^{n \times k}$; $\boldsymbol{C} \leftarrow \boldsymbol{K}\bar{\boldsymbol{A}} \in \mathbb{Z}_p^{n \times k}$; $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in \mathbb{Z}_p^{m \times k}$;
    $\textbf{if } k = 1 \textbf{ then } \mathsf{pk}^{\mathsf{pkv}} \leftarrow \epsilon$; $\textbf{elseif } k = 2 \textbf{ then } \mathsf{pk}^{\mathsf{pkv}} \leftarrow [a_{11}, a_{12}]_1$; $\textbf{fi}$ ;
    $\mathsf{pk}^{\mathsf{snd}} \leftarrow [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2$; $\mathsf{pk}^{\mathsf{zk}} \leftarrow [\boldsymbol{P}]_1$; $\mathsf{pk} \leftarrow (\mathsf{pk}^{\mathsf{snd}}, \mathsf{pk}^{\mathsf{zk}}, \mathsf{pk}^{\mathsf{pkv}})$; $\mathsf{sk} \leftarrow \boldsymbol{K}$; $\textbf{return } (\mathsf{pk}, \mathsf{sk})$;
$\mathsf{P}([\boldsymbol{M}]_1, \mathsf{pk}, [\boldsymbol{y}]_1, \boldsymbol{w})$: return $[\boldsymbol{\pi}]_1 \leftarrow [\boldsymbol{P}]_1^\top \boldsymbol{w} \in \mathbb{G}_1^k$;
$\mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_1)$: // sk is extracted from $\mathcal{Z}$ by using a knowledge assumption;
    return $[\boldsymbol{\pi}]_1 \leftarrow \boldsymbol{K}^\top [\boldsymbol{y}]_1 \in \mathbb{G}_1^k$;
$\mathsf{V}([\boldsymbol{M}]_1, \mathsf{pk}, [\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1)$ : check that $[\boldsymbol{y}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{\pi}]_1^\top [\bar{\boldsymbol{A}}]_2$; // $\in \mathbb{G}_T^{1 \times k}$
$\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk})$: Return 1 only if the following checks all succeed:
            $\mathsf{pk} = (\mathsf{pk}^{\mathsf{snd}}, \mathsf{pk}^{\mathsf{zk}}, \mathsf{pk}^{\mathsf{pkv}}) \wedge \mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2 \wedge \mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_1$;
            $[\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m} \wedge [\boldsymbol{P}]_1 \in \mathbb{G}_1^{m \times k} \wedge [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k \times k} \wedge [\boldsymbol{C}]_2 \in \mathbb{G}_2^{n \times k}$;
    $(*)$   $[\boldsymbol{M}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{P}]_1 [\bar{\boldsymbol{A}}]_2$;
            $\mathsf{isinvertible}([\bar{\boldsymbol{A}}]_2, \mathsf{pk}^{\mathsf{pkv}})$;
```

**Fig. 3.** S-ZK QA-NIZK $\Pi_{\mathsf{bpk}}$ for $[\boldsymbol{y}]_1 = [\boldsymbol{M}]_1 \boldsymbol{w}$ in the BPK model, where $k \in \{1, 2\}$.

Clearly, $\mathcal{D}_{\ell k}$-WKerMDH$_{\mathbb{G}_1}$ is not stronger and it is ostensibly weaker than $\mathcal{D}_{\ell k}$-KerMDH$_{\mathbb{G}_1}$ since computing $\boldsymbol{c}$ may be more complicated than computing $[\boldsymbol{c}]_2$. The Discrete Logarithm assumption is a classical example of WKerMDH (consider matrices $\boldsymbol{A} = \left( \begin{smallmatrix} a \\ -1 \end{smallmatrix} \right)$ for $a \leftarrow_\$ \mathbb{Z}_p$). In the case of common matrix distributions $\mathcal{D}_{\ell k}$, $\mathcal{D}_{\ell k}$-WKerMDH can be shown to be secure in the (S-)GBGM against $o(\sqrt{p/\mathsf{poly}(\lambda)})$-time generic adversaries.

Finally, Fig. 3 describes the new QA-NIZK argument system $\Pi_{\mathsf{bpk}}$. It also makes use of the auxiliary procedure isinvertible depicted in Fig. 2.

**Theorem 1 (S-GBGM Security of** KWKE **and** SKWKE**).** *Let $k \in \{1, 2\}$ and $k/p = \mathsf{negl}(\lambda)$. Then*

*(i) the $(\mathcal{D}_{\mathsf{p}}, k)$-KWKE$_{\mathbb{G}_1}$ assumption holds in the S-GBGM.*

*(ii) assuming that the $\mathcal{D}_{\mathsf{p}}$-WKerMDH$_{\mathbb{G}_1}$ assumption holds in the S-GBGM (thus, $\varrho = [\boldsymbol{M}]_1$ has been chosen from a WKerMDH$_{\mathbb{G}_1}$-hard distribution) against $\tau(\lambda)$-time generic adversaries, the $(\mathcal{D}_{\mathsf{p}}, k)$-SKWKE$_{\mathbb{G}_1}$ assumption holds in the S-GBGM against $O(\tau(\lambda))$-time generic adversaries.*

This statement is straightforward when we replace S-GBGM with GBGM. Partially since S-GBGM proofs are not common (yet), the following proof contains some novel and intricate ideas. In particular, since we work in the S-GBGM,

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

    if $\mathsf{PKV}(\varrho; \mathsf{pk}) = 0$ then return $\perp$; fi ;
    Extract the coefficients of $\bar{\boldsymbol{A}} = \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Y_i$ and of $\boldsymbol{C} = \sum_{i \geq 0} \boldsymbol{C}[i] Y_i$;
    For each $i$, sample random $y_i \leftarrow_\$ \mathbb{Z}_p$;
(♯)if $\det(\bar{\boldsymbol{A}}(\boldsymbol{y})) = 0$ then return $\perp$; fi ; // Probability $k/p$
    return $\boldsymbol{K} \leftarrow \boldsymbol{C}(\boldsymbol{y}) / \bar{\boldsymbol{A}}(\boldsymbol{y})$;
```

```
$\mathsf{Ext}^2_{\mathcal{A}}([\boldsymbol{M}]_1; r)$

if $\mathsf{PKV}(\varrho; \mathsf{pk}) = 0$ then return $\perp$; fi ;
Extract the coefficients of $\bar{\boldsymbol{A}} = \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Y_i$ and of $\boldsymbol{C} = \sum_{i \geq 0} \boldsymbol{C}[i] Y_i$;
For each $i$, sample $y_i \leftarrow_\$ \mathbb{Z}_p$ and $y'_i \leftarrow_\$ \mathbb{Z}_p$;
if $\det(\bar{\boldsymbol{A}}(\boldsymbol{y})) = 0 \vee \det(\bar{\boldsymbol{A}}(\boldsymbol{y}')) = 0$ then return $\perp$; fi ; // Probability $\leq 2k/p$
$\boldsymbol{K} \leftarrow \boldsymbol{C}(\boldsymbol{y}) / \bar{\boldsymbol{A}}(\boldsymbol{y}); \boldsymbol{K}' \leftarrow \boldsymbol{C}(\boldsymbol{y}') / \bar{\boldsymbol{A}}(\boldsymbol{y}')$;
if $\boldsymbol{K} \neq \boldsymbol{K}'$ then return $\boldsymbol{K} - \boldsymbol{K}'$; else return $\boldsymbol{K}$; fi ;
```

**Fig. 4.** Extractors $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1; r)$ and $\mathsf{Ext}^2_{\mathcal{A}}([\boldsymbol{M}]_1; r)$ in the S-GBGM proof

the discrete logarithms of the elements $[\bar{\boldsymbol{A}}]_2$ and $[\boldsymbol{C}]_2$ output by a (S)KWKE-adversary $\mathcal{A}$ can be written down as affine functions $\bar{\boldsymbol{A}}(\boldsymbol{Y})$ and $\boldsymbol{C}(\boldsymbol{Y})$ of all $\mathbb{G}_2$-indeterminates $Y_i$ created by $\mathcal{A}$. The extractor returns the evaluation of the rational function $\boldsymbol{K}(\boldsymbol{Y}) := \bar{\boldsymbol{C}}(\boldsymbol{Y}) / \bar{\boldsymbol{A}}(\boldsymbol{Y})$ at a uniformly random vector $\boldsymbol{y}$. The main intricacy in the proof consists of constructing the required extractor and then showing that if $\mathsf{PKV}$ accepts the public key then, w.h.p., $\bar{\boldsymbol{A}}(\boldsymbol{y})$ is invertible (in the case of KWKE) and moreover, $\boldsymbol{K}(\boldsymbol{Y})$ is a constant function (in the case of SKWKE). In the case of SKWKE, we somewhat surprisingly need to additionally assume that $[\boldsymbol{M}]_1$ comes from a hard (WKerMDH) distribution.

*Proof (of Theorem 1).* Assume $\mathcal{A}$ is a KWKE or SKWKE adversary, s.t.: given $\varrho = [\boldsymbol{M}]_1 \leftarrow_\$ \mathscr{D}_{\mathsf{p}}$ and $r \leftarrow_\$ \mathsf{RND}(\mathcal{A})$, $\mathcal{A}(\varrho; r)$ outputs with probability $\varepsilon_{\mathcal{A}}$ a public key $\mathsf{pk}$, such that $\mathsf{PKV}(\varrho; \mathsf{pk}) = 1$ (in particular, $\det \bar{\boldsymbol{A}} \neq 0$ and $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$).

**(i: GBGM-security of** KWKE**):** Assume $\mathcal{A}$ is a KWKE adversary. Fig. 4 depicts an S-GBGM extractor $\mathsf{Ext}_{\mathcal{A}}$, where $X_i$ (resp., $Y_i$) are indeterminates created by $\mathcal{A}$ (i.e., group elements created by her for which she does not know the discrete logarithm) in $\mathbb{G}_1$ (resp., $\mathbb{G}_2$), with $X_0 = Y_0 = 1$. Since $\mathcal{A}$ works in the S-GBGM, $\mathsf{Ext}_{\mathcal{A}}$ can extract all coefficients $\bar{\boldsymbol{A}}[i]$ and $\boldsymbol{C}[i]$ of $\bar{\boldsymbol{A}}$ and $\boldsymbol{C}$.

We will now analyse $\mathsf{Ext}_{\mathcal{A}}$, showing that $\mathsf{Ext}_{\mathcal{A}}$ satisfies the requirements to the extractor in the definition of KWKE. Assume that $\mathcal{A}$ was successful with inputs $(\varrho = [\boldsymbol{M}]_1; r)$, where $[\boldsymbol{M}]_1 = [\boldsymbol{M} X_0]_1 \leftarrow_\$ \mathscr{D}_{\mathsf{p}}$, that is, $\mathsf{PKV}(\varrho; \mathsf{pk}) = 1$. We execute $\mathsf{Ext}_{\mathcal{A}}([\boldsymbol{M}]_1; r)$ and obtain either $\boldsymbol{K}$ or $\perp$. Note that $\boldsymbol{P} = \sum_{j \geq 0} \boldsymbol{P}[j] X_j$ for unique coefficients $\boldsymbol{P}[j]$ that might not be known since $[\boldsymbol{M}]_1$ is an auxiliary string to $\mathcal{A}$. From (*) in $\mathsf{PKV}$ (i.e., $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$),

$$\boldsymbol{M}^\top X_0 \cdot \left( \sum_{i \geq 0} \boldsymbol{C}[i] Y_i \right) - \left( \sum_{j \geq 0} \boldsymbol{P}[j] X_j \right) \cdot \left( \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Y_i \right) = \boldsymbol{0}_{m \times k} \ . \qquad (1)$$

Since $X_j$ and $Y_i$ are indeterminates for all $i, j > 0$, the coefficients of $X_j Y_i$ in Eq. (1) must be equal to $\mathbf{0}_{m\times k}$ for all $i, j \geq 0$. In particular,

(i) $\boldsymbol{P}[0] \cdot \bar{\boldsymbol{A}}[i] = \boldsymbol{M}^\top \boldsymbol{C}[i]$ for all $i \geq 0$,
(ii) $\boldsymbol{P}[j] \cdot \bar{\boldsymbol{A}}[i] = \mathbf{0}_{m\times k}$ for all $i \geq 0, j > 0$.

Let $\bar{\boldsymbol{A}}(\boldsymbol{Y}) = \sum \bar{\boldsymbol{A}}[i] Y_i \in \mathbb{Z}_p^{k\times k}[\boldsymbol{Y}]$ be a multivariate matrix polynomial and let the polynomial $d(\boldsymbol{Y}) := \det(\bar{\boldsymbol{A}}(\boldsymbol{Y})) \in \mathbb{Z}_p[\boldsymbol{Y}]$ be its determinant. Clearly, $d(\boldsymbol{Y})$ has degree at most $k$ and that the matrix $\bar{\boldsymbol{A}}(\boldsymbol{Y})$ is invertible iff $d(\boldsymbol{Y}) \neq 0$ as a polynomial. Since $\mathsf{PKV}(\varrho; \mathsf{pk}) = 1$, $d(\boldsymbol{Y}) \neq 0$ and thus $\bar{\boldsymbol{A}}(\boldsymbol{Y})$ is invertible. This is obvious in the case $k = 1$. If $k = 2$, then $[a_{1s}]_1[1]_2 = [1]_1[a_{1s}]_2$, for $s \in \{1, 2\}$, and $[a_{11}]_1[a_{22}]_2 = [a_{12}]_1[a_{21}]_2$ guarantee that $d(\boldsymbol{Y}) \neq 0$.

By the Schwartz-Zippel lemma [Zip79,Sch80], $d(\boldsymbol{y}) = 0$ for uniformly sampled $y_i \leftarrow_{\$} \mathbb{Z}_p$ (and thus $\mathsf{Ext}_{\mathcal{A}}$ aborts in step ($\sharp$)) with probability at most $k/p$. Thus, $\bar{\boldsymbol{A}}(\boldsymbol{y})$ is invertible with probability at least $\varepsilon_{\mathcal{A}} - k/p$.

Assume now that $\bar{\boldsymbol{A}}(\boldsymbol{y})$ is invertible. Define

$$\boldsymbol{K}(\boldsymbol{Y}) := \boldsymbol{C}(\boldsymbol{Y})\bar{\boldsymbol{A}}^{-1}(\boldsymbol{Y}) = \left(\sum_{i\geq 0} \boldsymbol{C}[i] Y_i\right) \left(\sum_{i\geq 0} \bar{\boldsymbol{A}}[i] Y_i\right)^{-1} \in \mathbb{Z}_p^{n\times k}(\boldsymbol{Y})$$

and let $\boldsymbol{K} := \boldsymbol{K}(\boldsymbol{y})$. Since $\bar{\boldsymbol{A}}(\boldsymbol{y})$ is invertible then from Items i and ii,

(i') $\boldsymbol{P}[0] \cdot \bar{\boldsymbol{A}}(\boldsymbol{y}) = \boldsymbol{P}[0] \cdot (\sum_i \bar{\boldsymbol{A}}[i] y_i = \boldsymbol{M}^\top (\sum_i \boldsymbol{C}[i] y_i) = \boldsymbol{M}^\top \boldsymbol{C}(\boldsymbol{y})$ and thus $\boldsymbol{P}[0] = \boldsymbol{M}^\top \boldsymbol{K}$,
(ii') $\boldsymbol{P}[j] \cdot \bar{\boldsymbol{A}}(\boldsymbol{y}) = \boldsymbol{P}[j] \cdot (\sum_i \bar{\boldsymbol{A}}[i] y_i = \mathbf{0}_{m\times k}$ and thus $\boldsymbol{P}[j] = \mathbf{0}_{m\times k}$ for all $j > 0$.

Hence, with probability $\varepsilon_{\mathsf{Ext}_{\mathcal{A}}} \geq \varepsilon_{\mathcal{A}} - k/p$,

$$\boldsymbol{P} = \sum_{j\geq 0} \boldsymbol{P}[j] X_j = \boldsymbol{P}[0] = \boldsymbol{M}^\top \boldsymbol{K} \enspace .$$

Thus, $|\varepsilon_{\mathsf{Ext}_{\mathcal{A}}} \geq \varepsilon_{\mathcal{A}}| \leq k/p$ and the S-GBGM security of KWKE follows.

**(ii: GBGM-security of** SKWKE**):** Let $\mathcal{A}$ be a generic SKWKE adversary that works in time $\tau(\lambda)$ and outputs a $\mathsf{pk}$ accepted by $\mathsf{PKV}$ with probability $\varepsilon_{\mathcal{A}}$. To prove that SKWKE is secure in the S-GBGM, we need to additionally show that $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$. In the process, we need to assume that $\mathscr{D}_{\mathsf{p}}$-WKerMDH is hard.

More precisely, the main idea is that in the proof step () we already established that $\boldsymbol{C}(\boldsymbol{Y}) = \boldsymbol{K}(\boldsymbol{Y})\bar{\boldsymbol{A}}(\boldsymbol{Y})$ as polynomials. In the current step, we need to show that $\boldsymbol{C}(\boldsymbol{Y}) = \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Y})$ holds, that is, $\boldsymbol{K}(\boldsymbol{Y})$ is a constant function. To guarantee the latter, we check the value of the rational function $\boldsymbol{K}(\boldsymbol{Y})$ at two positions. If the two values are different, we can break $\mathscr{D}_{\mathsf{p}}$-WKerMDH. Otherwise, w.h.p., $\boldsymbol{K}(\boldsymbol{X})$ is a constant function.

More precisely, consider the extractor $\mathsf{Ext}^2_{\mathcal{A}}$ in Fig. 4. Here, $\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{y})$ and $\boldsymbol{K}' = \boldsymbol{K}(\boldsymbol{y}')$. Let $\varepsilon_{\mathcal{A}}$ be the success probability of $\mathcal{A}$. Analogously to the security proof of KWKE, with probability $\varepsilon_{\mathcal{A}} - 2k/p$, both $\bar{\boldsymbol{A}}(\boldsymbol{y})$ and $\bar{\boldsymbol{A}}(\boldsymbol{y}')$ are invertible and thus $\mathsf{Ext}^2_{\mathcal{A}}$ does not return $\perp$.

Assume now that $\mathsf{Ext}^2_{\mathcal{A}}$ does not return $\perp$. Then, by following similar analysis as in the case (i), we have that $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$ and $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}'$ which means that

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

If $\boldsymbol{K} \neq \boldsymbol{K}'$ then $\mathsf{Ext}_{\mathcal{A}}$ has computed a non-zero element $\boldsymbol{K} - \boldsymbol{K}'$ in the cokernel of $[\boldsymbol{M}]_1$ and thus broken $\mathscr{D}_{\mathsf{p}}$-WKerMDH$_{\mathbb{G}_1}$ in the S-GBGM. Since breaking $\mathscr{D}_{\mathsf{p}}$-WKerMDH in the S-GBGM is hard within $\tau(\lambda)$ steps, the probability $\varepsilon_{\text{WKerMDH}}$ that $\mathsf{Ext}_{\mathcal{A}}$ returns $\boldsymbol{K} - \boldsymbol{K}'$ is negligible unless $\mathcal{A}$ has computational complexity $\omega(\tau(\lambda))$. Otherwise, $\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{y}) = \boldsymbol{K}(\boldsymbol{y}')$, which means $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{f}(\boldsymbol{y}') = \boldsymbol{0}$, where

$$\boldsymbol{f}(\boldsymbol{Y}) := \boldsymbol{C}(\boldsymbol{Y}) \bar{\boldsymbol{A}}^{-1}(\boldsymbol{Y}) - \boldsymbol{K} \enspace .$$

Denote the $(i, j)$th coefficient of the matrix $\boldsymbol{f}(\boldsymbol{Y})$ by $f_{ij}(\boldsymbol{Y}) = \sum_s C_{is}(\boldsymbol{Y}) \bar{A}^{-1}_{sj}(\boldsymbol{Y}) - K_{ij}$. Note that $f_{ij}(\boldsymbol{Y}) = f'_{ij}(\boldsymbol{Y}) / \det(\bar{\boldsymbol{A}}(\boldsymbol{Y}))$, where $f'_{ij}(\boldsymbol{Y})$ is some polynomial of degree $\leq k$.

Now, $\boldsymbol{f}(\boldsymbol{Y}) = \boldsymbol{0}$ iff $\boldsymbol{C}(\boldsymbol{Y}) - \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Y}) = \boldsymbol{0}$ and $\det(\bar{\boldsymbol{A}}(\boldsymbol{Y})) \neq 0$. At this point we know that $\det(\bar{\boldsymbol{A}}(\boldsymbol{Y})) \neq 0$. Thus, $\boldsymbol{f}(\boldsymbol{Y}) \neq \boldsymbol{0}$ iff $\boldsymbol{C}(\boldsymbol{Y}) - \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Y}) \neq \boldsymbol{0}$. From this and the Schwartz-Zippel lemma it follows that if $f_{ij}(\boldsymbol{Y}) \neq 0$ then $\Pr_{\boldsymbol{y}}[f_{ij}(\boldsymbol{y}) = 0] \leq k/p$. If $\boldsymbol{f}(\boldsymbol{Y}) \neq \boldsymbol{0}$ then there exists at least one $(i_0, j_0)$ such that $f_{i_0, j_0}(\boldsymbol{Y}) \neq 0$ and thus $\Pr_{\boldsymbol{y}}[f_{i_0, j_0}(\boldsymbol{y}) = 0] \leq k/p$. Thus, if $\boldsymbol{f}(\boldsymbol{Y}) \neq \boldsymbol{0}$ then $\Pr_{\boldsymbol{y}}[\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{0}] \leq k/p$.

Hence, with probability $\varepsilon_{\mathsf{Ext}^2_{\mathcal{A}}} \geq \varepsilon_{\mathcal{A}} - 3k/p - \varepsilon_{\text{WKerMDH}}$, $\boldsymbol{C}(\boldsymbol{Y}) = \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Y})$ and thus $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$ and $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$. Thus, $|\varepsilon_{\mathsf{Ext}^2_{\mathcal{A}}} - \varepsilon_{\mathcal{A}}| \leq 3k/p + \varepsilon_{\text{WKerMDH}}$ and the S-GBGM security of SKWKE follows. □

In the case of SKWKE, we extracted the *unique* $\boldsymbol{K}$ used to compute the CRS. Following a proof idea from [ABLZ17], it is easy to show that under either the KWKE or the SKWKE assumption, $\Pi_{\mathsf{bpk}}$ is S-ZK.

**Theorem 2 (Security of $\Pi_{\mathsf{bpk}}$).** *Let $\Pi_{\mathsf{bpk}}$ be the QA-NIZK argument system for linear subspaces from Fig. 3. Let $k \in \{1, 2\}$. The following statements hold in the BPK model.*

*(i) $\Pi_{\mathsf{bpk}}$ is perfectly complete.*

*(ii) If the $(\mathscr{D}_{\mathsf{p}}, k)$-SKWKE$_{\mathbb{G}_1}$ assumption holds relative to* Pgen *(and thus also assuming $\mathscr{D}_{\mathsf{p}}$-WKerMDH, i.e., that $[\boldsymbol{M}]_1$ comes from a* WKerMDH*-hard distribution) then $\Pi_{\mathsf{bpk}}$ is statistically S-ZK.*

*(iii) If the $(\mathscr{D}_{\mathsf{p}}, k)$-KWKE$_{\mathbb{G}_1}$ assumption holds relative to* Pgen *then $\Pi_{\mathsf{bpk}}$ is statistically S-ZK.*

*(iv) Let $k = 1$ (resp., $k = 2$). If the $\mathscr{D}_k$-*KerMDH *(resp., $\mathscr{D}_k$-*SKerMDH*) assumption holds relative to* Pgen *then $\Pi_{\mathsf{bpk}}$ is computationally quasi-adaptively sound.*

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

**(ii: S-ZK under** SKWKE**):** Let $\mathcal{Z}$ be a subverter that computes $\mathsf{pk}$ so as to break the S-ZK property. That is, $\mathcal{Z}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$ outputs $(\mathsf{pk}, \mathsf{aux}_{\mathcal{Z}})$.

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

$\mathsf{Ext}_{\mathcal{Z}}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$
    return $\mathsf{Ext}^2_{\mathcal{A}}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$;
```

**Fig. 5.** The extractor and the constructed adversary $\mathcal{A}$ from the S-ZK proof of Theorem 2, for both the SKWKE and KWKE case.

Let $\mathcal{A}$ be the adversary from Fig. 5. Note that $\mathsf{RND}(\mathcal{A}) = \mathsf{RND}(\mathcal{Z})$. Under the $(\mathscr{D}_{\mathsf{p}}, k)$-SKWKE assumption, there exists an extractor $\mathsf{Ext}^2_{\mathcal{A}}$, such that if $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$ then $\mathsf{Ext}^2_{\mathcal{A}}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$ outputs $\boldsymbol{K}$, such that $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$ and $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$. We construct a trivial extractor $\mathsf{Ext}_{\mathcal{Z}}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$ for $\mathcal{Z}$, as depicted in Fig. 5. Clearly, $\mathsf{Ext}_{\mathcal{Z}}$ 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 $\lambda$, $\mathsf{p} \in \mathrm{im}(\mathsf{Pgen}(1^\lambda))$, $[\boldsymbol{M}]_1 \leftarrow_{\$} \mathscr{D}_{\mathsf{p}}$, $([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1}$, $r_{\mathcal{Z}} \in \mathsf{RND}(\mathcal{Z})$, and run $\mathsf{Ext}_{\mathcal{Z}}([\boldsymbol{M}]_1; r_{\mathcal{Z}})$ to obtain $\boldsymbol{K}$. It clearly suffices to show that if $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$ and $([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathcal{R}_{[\boldsymbol{M}]_1}$ then

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

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

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

Hence, $\mathsf{O}_0$ and $\mathsf{O}_1$ have the same distribution and thus, $\Pi_{\mathsf{bpk}}$ is S-ZK under SKWKE.

**(iii: S-ZK under** KWKE**):** The security proof is the same as in the previous case, except that $\mathsf{Ext}_{\mathcal{A}}$ is an extractor guaranteed by KWKE. 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: $k = 1$):** follows directly from the soundness proof of $\Pi'_{\mathsf{as}}$ in [KW15].

**(iv: $k = 2$, soundness under** SKerMDH**):** In the case $k = 2$, the proof is *similar* to the soundness proof of $\Pi'_{\mathsf{as}}$ in [KW15]. However, since we added $[a_{11}, a_{12}]_1$ to the public key, we reduce instead to the SKerMDH assumption of [GHR15]; this complicates the proof.

Assume that $\mathcal{A}$ breaks the soundness of $\Pi_{\mathsf{bpk}}$ with probability $\varepsilon$. We will build an adversary $\mathcal{B}$, see Fig. 6, that breaks SKerMDH with probability $\geq \varepsilon - 1/p$.

Note that in Fig. 6, $[\bar{\boldsymbol{A}}']_2 = [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k \times k}$. Define *implicitly* (since 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,

$$\begin{aligned} [\boldsymbol{C}]_2 =& (\boldsymbol{K}' || \boldsymbol{M}^\perp)[\boldsymbol{A}']_2 = [\boldsymbol{K}'\bar{\boldsymbol{A}}' + \boldsymbol{M}^\perp \underline{\boldsymbol{A}}']_2 = [(\boldsymbol{K}' + \boldsymbol{M}^\perp \underline{\boldsymbol{A}}' \bar{\boldsymbol{A}}^{-1})\bar{\boldsymbol{A}}]_2 = [\boldsymbol{K}\bar{\boldsymbol{A}}]_2 \enspace , \\ [\boldsymbol{P}]_1 =& [\boldsymbol{M}^\top \boldsymbol{K}']_1 = [\boldsymbol{M}^\top (\boldsymbol{K} - \boldsymbol{M}^\perp \underline{\boldsymbol{A}}' \bar{\boldsymbol{A}}^{-1})]_1 = [\boldsymbol{M}^\top \boldsymbol{K}]_1 \enspace . \end{aligned}$$

Thus, $\mathsf{pk}'$ has the same distribution as the real public key.

With probability $\varepsilon$, $\mathcal{A}$ is successful, that is,

```
$\mathcal{B}(\mathsf{p}, ([\boldsymbol{A}]_1, [\boldsymbol{A}]_2))$ // $([\boldsymbol{A}]_1, [\boldsymbol{A}]_2) \in \mathbb{G}_1^{(k+1)\times k} \times \mathbb{G}_2^{(k+1)\times k}$ with $\boldsymbol{A} = (a_{ij})$
$([\boldsymbol{M}]_1, \boldsymbol{M}) \leftarrow_{\$} \mathscr{D}'_{\mathsf{p}}$; // $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$
Let $\boldsymbol{M}^{\perp} \in \mathbb{Z}_p^{n\times(n-m)}$ be a basis of the kernel of $\boldsymbol{M}^{\top}$;
$\boldsymbol{K}' \leftarrow_{\$} \mathbb{Z}_p^{n\times k}$; $\boldsymbol{R} \leftarrow_{\$} \mathbb{Z}_p^{(n-m-1)\times(k+1)}$;
$[\boldsymbol{A}']_2 \leftarrow \begin{pmatrix} [\boldsymbol{A}]_2 \\ \boldsymbol{R}\cdot[\boldsymbol{A}]_2 \end{pmatrix}$; // $\boldsymbol{A}' \in \mathbb{Z}_p^{(n-m+k)\times k}$
$[\boldsymbol{C}]_2 \leftarrow (\boldsymbol{K}' || \boldsymbol{M}^{\perp})[\boldsymbol{A}']_2$;
$[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}^{\top}\boldsymbol{K}']_1$;
$\mathsf{pk}' \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [a_{11}, a_{12}, \boldsymbol{P}]_1)$;
$([\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1) \leftarrow \mathcal{A}([\boldsymbol{M}]_1, \mathsf{pk}')$; // $[\boldsymbol{y}]_1 \in \mathbb{G}_1^n$, $[\boldsymbol{\pi}]_1 \in \mathbb{G}_1^k$
$[\boldsymbol{c}]_1^{\top} \leftarrow [(\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top}\boldsymbol{K}') || - \boldsymbol{y}^{\top}\boldsymbol{M}^{\perp}]_1$;
Represent $[\boldsymbol{c}]_1^{\top}$ as $[\boldsymbol{c}_1^{\top} || \boldsymbol{c}_2^{\top}]_1$ with $[\boldsymbol{c}_1]_1 \in \mathbb{G}_1^{k+1}$ and $[\boldsymbol{c}_2]_1 \in \mathbb{G}_1^{n-m-1}$;
$\boldsymbol{s}_2 \leftarrow_{\$} \mathbb{Z}_p^{k+1}$; $[\boldsymbol{s}_1]_1 \leftarrow [\boldsymbol{c}_1 + \boldsymbol{R}^{\top}\boldsymbol{c}_2 + \boldsymbol{s}_2]_1$;
return $([\boldsymbol{s}_1]_1, [\boldsymbol{s}_2]_2)$;
```

**Fig. 6.** Adversary $\mathcal{B}$ in the soundness proof of Theorem 2 (reduction to SKerMDH)

1. $\boldsymbol{y}^{\top}\boldsymbol{M}^{\perp} \neq \boldsymbol{0}_{1\times(n-m)}$ (that is, $\boldsymbol{y} \notin \operatorname{colspace}(\boldsymbol{M})$) and thus also $\boldsymbol{c} = ((\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top}\boldsymbol{K}') || - \boldsymbol{y}^{\top}\boldsymbol{M}^{\perp}) \neq \boldsymbol{0}_{n-m+k}$;
2. $\boldsymbol{y}^{\top}\boldsymbol{C} = \boldsymbol{\pi}^{\top}\bar{\boldsymbol{A}}$ ($\mathsf{V}$ accepts). Thus, $\boldsymbol{0}_{1\times k} = \boldsymbol{\pi}^{\top}\bar{\boldsymbol{A}} - \boldsymbol{y}^{\top}\boldsymbol{C} = \left(\boldsymbol{\pi}^{\top} || \boldsymbol{0}_{n-m}^{\top}\right)\boldsymbol{A}' - \boldsymbol{y}^{\top}\left(\boldsymbol{K}' || \boldsymbol{M}^{\perp}\right)\boldsymbol{A}' = \left((\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top}\boldsymbol{K}') || - \boldsymbol{y}^{\top}\boldsymbol{M}^{\perp}\right)\boldsymbol{A}' = \boldsymbol{c}^{\top}\boldsymbol{A}'$.

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

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

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

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

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

We note SKerMDH is not secure when $k = 1$, [GHR15].

## 5 On Subverting $\varrho$

In Section 3, we defined S-ZK in the BPK model assuming that the language parameter $\varrho$ is generated honestly, i.e., from the correct distribution and without any leakage of the secret keys. Next, we will study whether this assumption is really needed.

For the sake of concreteness, let us first consider $\Pi_{\mathsf{bpk}}$ (thus, $\varrho = [\boldsymbol{M}]_1$ for some $\boldsymbol{M}$) and the S-ZK definition in Section 3. According to the latter, if $\mathcal{Z}$ on input $\varrho$ outputs $\mathsf{pk}$ then he can leak information through two different channels: $\mathsf{aux}_{\mathcal{Z}}$ (any string of $\mathcal{Z}$'s choice that can be sent to a malicious distinguisher) and $\mathsf{sk}$ (the secret key extracted from $\mathcal{Z}$ by the PPT extractor $\mathsf{Ext}_{\mathcal{Z}}$, where the existence of $\mathsf{Ext}_{\mathcal{Z}}$ is stated by the definition).

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

**Fig. 7.** A contrived leaky subspace QA-NIZK ($n = 2$, $m = k = 1$)

**Leaking Information via $\mathsf{aux}_{\mathcal{Z}}$.** If $\mathcal{Z}$ leaks (a part of) $\boldsymbol{M}$ to the verifier through $\mathsf{aux}_{\mathcal{Z}}$ then $\mathsf{V}$ will be able to check whether $[\boldsymbol{y}]_1 \in \mathrm{colspace}([\boldsymbol{M}]_1)$ or even compute (a part of) $[\boldsymbol{w}]_1$ from $[\boldsymbol{y}]_1$. This holds since $\mathcal{L}_{[\boldsymbol{M}]_1}$ is not necessarily hard if $\boldsymbol{M}$ is public. E.g., consider the case when $[\boldsymbol{M}]_1 = [M_1, M_2]_1^\top$ is an Elgamal public key for $M_i \neq 0$. Then $[y_1, y_2]_1^\top =^? [\boldsymbol{M}]_1 w = [M_1 w, M_2 w]_1^\top$ can be decided efficiently, given $(M_1, M_2)$, by checking whether $M_1 [y_2]_1 = M_2 [y_1]_1$. Moreover, one can compute $(1/M_1)[y_1]_1 = [w]_1$.

This attack is possible unless communication between the creator of $[\boldsymbol{M}]_1$ and the malicious verifier is limited to leak no side information about $\boldsymbol{M}$. Thus, achieving the intuitive notion of zero knowledge is impossible unless $[\boldsymbol{M}]_1$ is created by a separate party who does not leak information to $\mathsf{V}$. (Or, the language $\mathcal{L}_{[\boldsymbol{M}]_1}$ is easy, which is not interesting.)

**Leaking Information via Knowledge Assumptions.** There is a more sneaky (and novel?) attack where the subverter, who knows $\boldsymbol{M}$, leaks $\boldsymbol{M}$ to the simulator via $\mathsf{sk}$. Since this attack is less obvious, we will consider it in more detail. This attack shows 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 $\mathsf{pk}$ created by $\mathcal{Z}$. Under the BDHKE assumption of [ABLZ17], there exists a PPT extractor $\mathsf{Ext}_{\mathcal{Z}}$ that, given access to the inputs and the random coins of $\mathcal{Z}$, extracts $\boldsymbol{M}$ from $\mathsf{pk} = ([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$. Given $[\boldsymbol{y}]_1 \in \mathcal{L}_{[\boldsymbol{M}]_1}$ and $\boldsymbol{M}$, one can compute $[\boldsymbol{w}]_1$, such that $[\boldsymbol{y}]_1 = [\boldsymbol{M}]_1 \boldsymbol{w}$ (cf. the previous subsubsection). One can now construct a contrived QA-NIZK (see Fig. 7) where the prover and the simulator both output $[\boldsymbol{w}]_1$. Since $\mathsf{P}$ and $\mathsf{Sim}$ have the same output, this protocol is formally zero knowledge although intuitively it leaks information about $\boldsymbol{w}$.

More generally, a malicious subverter can choose $\mathsf{sk}$ 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, one can then 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 S-ZK, 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 $\mathsf{pk}$ output by $\mathcal{Z}$, returns some value that depends on $\boldsymbol{M}$. This is impossible to achieve in general: for example in $\Pi_{\mathsf{bpk}}$, the subverter who knows $\boldsymbol{M}$ can choose $\boldsymbol{K}$ as a function of $\boldsymbol{M}$.

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

## 6 On QA-NIZK in the RPK Model

S-ZK 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 $\mathsf{Sim}$ 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 (stand-alone) interactive zero knowledge in the standard model [BCNP04]. In this case, in the (stand-alone) simulation, $\mathsf{Sim}$ rewinds the verifier to obtain all the secret keys. This is important since the RPK model is still substantially weaker than the CRS model, [BCNP04]. Interestingly, it seems that this simple observation is novel.

Finally, Wee [Wee07] proved that NIZK proof systems in the RPK model exist only for languages for $\mathsf{BPP}$. Additionally, he constructed a NIZK argument system for $\mathsf{NP}$ making subexponential hardness assumptions. Due to the observation in the previous paragraph, the NIZK argument systems of [BFS16,ABLZ17,Fuc18] and of the current paper are secure in the RPK model without relying on subexponential hardness assumptions to get zero knowledge, however, they use nonfalsifiable assumptions in the soundness proof.

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 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.

To confirm our position, we cite Scafuro and Visconti [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 Micali and Reyzin [MR01]: “It suffices for PK to be a string known to the prover, and **chosen** by the verifier prior to any interaction with him.”

**Open Problems.** Since in the important case $k = 1$, S-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. We mentioned some concrete applications of S-ZK QA-NIZK, but we leave their further investigation as an interesting open question. However, recall from the introduction that the results of the current work have been used recently [Lip19, GR19] to construct updatable S-ZK QA-NIZKs and updatable S-ZK SNARKs. 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 S-ZK “for free”.

**Acknowledgments.** The authors were supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE), and by the Estonian Research Council grant (PRG49). Most of this work was performed while Zając was working at the University of Tartu.

## References

ABLZ17. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70700-6_1.

ABP15. Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 69–100. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_3.

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. Springer, Heidelberg, August 2005. doi:10.1007/11535218_9.

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. Springer, Heidelberg, May 2005. doi:10.1007/11426639_26.

BCC88. Guilles Brassard, David Chaum, and Claude Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

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 *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. doi:10.1109/SP.2014.36.

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 *2015 IEEE Symposium on Security and Privacy*, pages 287–304. IEEE Computer Society Press, May 2015. doi:10.1109/SP.2015.25.

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. Springer, Heidelberg, August 2010. doi:10.1007/978-3-642-14623-7_13.

BCNP04. Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *45th FOCS*, pages 186–195. IEEE Computer Society Press, October 2004. doi:10.1109/FOCS.2004.71.

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

Ben16. Fabrice Ben Hamouda-Guichoux. *Diverse Modules and Zero-Knowledge.* PhD thesis, PSL Research University, 2016.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988. doi:10.1145/62212.62222.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53890-6_26.

CGGM00. Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *32nd ACM STOC*, pages 235–244. ACM Press, May 2000. doi:10.1145/335305.335334.

Dam92. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992. doi:10.1007/3-540-46766-1_36.

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, December 2014. doi:10.1007/978-3-662-45611-8_28.

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. Springer, Heidelberg, March 2006. doi:10.1007/11681878_3.

EHK$^+$13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume

8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_8.

FLM11. Marc Fischlin, Benoît Libert, and Mark Manulis. Non-interactive and reusable universally composable string commitments with adaptive security. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 468–485. Springer, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_25.

FLS90. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. doi:10.1109/FSCS.1990.89549.

FLSZ17. Prastudy Fauzi, Helger Lipmaa, Janno Siim, and Michal Zajac. An efficient pairing-based shuffle argument. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 97–127. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70697-9_4.

FO18. Georg Fuchsbauer and Michele Orrù. Non-interactive zaps of knowledge. In Bart Preneel and Frederik Vercauteren, editors, *ACNS 18*, volume 10892 of *LNCS*, pages 44–62. Springer, Heidelberg, July 2018. doi:10.1007/978-3-319-93387-0_3.

Fuc18. Georg Fuchsbauer. Subversion-zero-knowledge SNARKs. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 315–347. Springer, Heidelberg, March 2018. doi:10.1007/978-3-319-76578-5_11.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_37.

GHR15. Alonso González, Alejandro Hevia, and Carla Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629. Springer, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_25.

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. doi:10.1145/22145.22178.

GO94. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994. doi:10.1007/BF00195207.

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 16*, volume 9696 of *LNCS*, pages 427–444. Springer, Heidelberg, June 2016. doi:10.1007/978-3-319-39555-5_23.

GR19. Antonio González and Carla Ràfols. Sublinear Pairing-based Arguments with Updatable CRS and Weaker Assumptions. Technical Report 2019/326, IACR, March 25, 2019. Available from https://eprint.iacr.org/2019/326, last checked version from Mar 29, 2019.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_19.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008. doi:10.1007/978-3-540-78967-3_24.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. doi:10.1145/1993636.1993651.

Ica09. Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316. Springer, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_18.

JR13. Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_1.

JR14. Charanjit S. Jutla and Arnab Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_17.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_4.

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_10.

Lip13. Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 41–60. Springer, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_3.

Lip19. Helger Lipmaa. Key-and-Argument-Updatable QA-NIZKs. Technical Report 2019/333, IACR, March 27, 2019. Available from https://eprint.iacr.org/2019/333, last checked version from Mar 31, 2019.

LPJY13. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Linearly homomorphic structure-preserving signatures and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_17.

LPJY14. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532. Springer, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_29.

LPJY15. Benoît Libert, Thomas Peters, Marc Joye, and Moti Yung. Compactly hiding linear spans - tightly secure constant-size simulation-sound QA-NIZK proofs and applications. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 681–707. Springer, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_28.

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 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. Springer, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_32.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_27.

Nec94. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

Pas03. Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2003. doi:10.1007/3-540-39200-9_10.

PHGR13. Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. doi:10.1109/SP.2013.47.

Sch80. Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_18.

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. Springer, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_11.

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 09*, volume 5580 of *LNCS*, pages 287–304. Springer, Heidelberg, June 2009.

Wee07. Hoeteck Wee. Lower bounds for non-interactive zero-knowledge. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 103–117. Springer, Heidelberg, February 2007. doi:10.1007/978-3-540-70936-7_6.

Zip79. Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In Edward W. Ng, editor, *EUROSM 1979*, volume 72 of *LNCS*, pages 216–226, Marseille, France, June 1979. Springer, Heidelberg.

## A GBGM and S-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 $\mathsf{cell}_i$ 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 $\Pi$ be an allowed set of computation operations. A computation operation consists of selecting a (say, $t$-ary) operation $f \in \Pi$ 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 $\Sigma$ of relations, a query consists of selecting a (say, $t$-ary) relation $\varrho \in \Sigma$ 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 $\Pi = \{+, \hat{e}\}$ and $\Sigma = \{=\}$, 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 $\mathcal{A}$ additional power. We assume that all relation queries are for free. We also assume that $\mathcal{A}$ is successful if after $\tau$ 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 $\mathcal{A}$.

**S-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 *S-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 $\mathbb{G}_1$ can also operate with $Y$ in $\mathbb{G}_T$. Formally, $\mathit{\Pi}$ will contain one more operation $\mathsf{create}$, with the following semantics:

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

The semantics of $\mathsf{create}$ 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.