On QA-NIZK in the BPK Model

Behzad Abdolmaleki[1], Helger Lipmaa[1,2(✉)], Janno Siim[1], and Michał Zając[3]

[1] University of Tartu, Tartu, Estonia
abdolmaleki.behzad.ir@gmail.com, helger.lipmaa@gmail.com,
jannosiim@gmail.com
[2] Simula UiB, Bergen, Norway
[3] Clearmatics, London, UK
m.p.zajac@gmail.com

Abstract. Recently, Bellare *et al.* defined subversion-resistance (security in the case the CRS creator may be malicious) for NIZK. In particular, a Sub-ZK NIZK is zero-knowledge, even in the case of subverted CRS. We study Sub-ZK QA-NIZKs, where the CRS can depend on the language parameter. First, we observe that subversion zero-knowledge (Sub-ZK) in the CRS model corresponds to no-auxiliary-string non-black-box NIZK in the Bare Public Key model, and hence, the use of non-black-box techniques is needed to obtain Sub-ZK. Second, we give a precise definition of Sub-ZK QA-NIZKs that are (knowledge-)sound if the language parameter but not the CRS is subverted and zero-knowledge even if both are subverted. Third, we prove that the most efficient known QA-NIZK for linear subspaces by Kiltz and Wee is Sub-ZK under a new knowledge assumption that by itself is secure in (a weaker version of) the algebraic group model. Depending on the parameter setting, it is (knowledge-)sound under different non-falsifiable assumptions, some of which do not belong to the family of knowledge assumptions.

Keywords: Bare Public Key model · No-auxiliary-string zero knowledge · Non-black-box zero knowledge · QA-NIZK · Subversion-security

1 Introduction

Zero-knowledge argument systems introduced by Goldwasser *et al.* [22] enable a prover to convince a verifier of the veracity of a statement while leaking no additional information. Blum *et al.* [6] introduced non-interactive zero-knowledge (NIZK) argument systems where the prover outputs just one message (the argument) that convinces the verifier of the truth of the statement. Unfortunately, NIZKs are impossible in the standard model [21], and thus in all such applications, one has to rely on some trust assumption like the common reference string (CRS) 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, [3], where the authority is trusted to check that a party knows the secret key corresponding to the public key and then store her key)

A. Kiayias et al. (Eds.): PKC 2020, LNCS 12110, pp. 590–620, 2020.
https://doi.org/10.1007/978-3-030-45374-9_20

model and the bare public key (BPK, [9], where the authority is only trusted to store the public key of each party) model. However, very few NIZKs are known in the RPK model while black-box NIZK [38] (the simulator uses adversarial algorithm only by giving inputs and receiving outputs) and even auxiliary-string non-black-box [21,42] (the simulator may use the code of the adversary, who has access to an arbitrary auxiliary string) NIZK is impossible in the BPK model.

There has been a recent surge of the research to decrease the trust in the CRS model due to the use of succinct non-interactive zero knowledge argument systems of knowledge (zk-SNARKs, [11,18,26,27,35,36,40]) in real-life applications like verifiable computation and cryptocurrencies. Recently, [2,15] constructed subversion-zero knowledge (Sub-ZK) zk-SNARKs, where the prover does not have to trust the CRS creator. According to an impossibility result of [4], this means that such SNARKs cannot have soundness when the CRS has been maliciously generated. Abdolmaleki *et al.* [2] proposed the following concrete recipe for constructing Sub-ZK zk-SNARKs: first, construct an efficient public CRS verification algorithm CV that rejects malformed CRSs. Second, when proving Sub-ZK, use a non-falsifiable knowledge assumption [10] to obtain an extractor that recovers the CRS trapdoor td from a CV-accepted CRS; td is then used by the simulator (that works when the CRS has been honestly generated) to simulate the argument. Based on this recipe, [2,15] showed that the most efficient known zk-SNARK by Groth [27] is Sub-ZK. One principal weakness of zk-SNARKs is that zk-SNARKs for languages outside of BPP have to rely on non-falsifiable assumptions, based on the impossibility result of [19]. *However, we are not aware of any prior result indicating whether non-falsifiable assumptions are needed to obtain Sub-ZK.*

Another important recent direction in the NIZK arena is that of quasi-adaptive NIZKs (QA-NIZKs, [28]). In a QA-NIZK, the CRS can depend on a language parameter ϱ, where ϱ can be thought of as a properly distributed public key of some cryptosystem. One consequence of this definition is that up to now, QA-NIZKs have been only considered in the CRS model. The dependence of CRS on correctly generated ϱ means that one can construct very efficient QA-NIZKs for non-trivial languages based on standard assumptions like KerMDH [39]. Importantly, very efficient pairing-based QA-NIZKs [1,23,28,30–32] for the linear subspace language have been constructed in the CRS model. A QA-NIZK argument system for linear subspaces allows the prover to convince the verifier that a vector of group elements[1] $[\boldsymbol{y}]_1$ 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 $\boldsymbol{x} \in \mathbb{Z}_p^m$.

Although QA-NIZKs for other languages are known (e.g., the language of bitstrings [23] and the languages of shuffles [24], both requiring a quadratic-length CRS, and a recent QA-NIZK [12] for SSP [11], that relies on non-succinct commitment), research on QA-NIZKs has been mostly concentrated on designing efficient QA-NIZKs for linear subspaces. Such focus is motivated because of the broad applicability of QA-NIZKs for linear subspaces in the design of various cryptographic primitives (see [28,30–32] for examples and references). In addition, [14]

[1] We use pairing-based setting and the additive bracket notation of [13] (see Sect. 2).

combined SNARKs and QA-NIZKs for linear subspaces to construct an efficient pairing-based NIZK shuffle argument systems. This and other recent work [8,25,37] that use QA-NIZKs to construct SNARKs shows that the study of different properties of QA-NIZKs can be also beneficial in the world of SNARKs.

In particular, Campanelli *et al.* [8] proposed a toolbox called LegoSNARK that allows building complex zk-SNARK arguments from other zk-SNARKs given that the building blocks of the final zk-SNARK are so-called commit-and-prove SNARKs (CP-SNARKs). A linear subspace QA-NIZK plays a crucial role in the Campanelli *et al.* framework. First, it is used in a transformation that makes commit-carrying SNARKs (CC-SNARKs), like [27], CP-SNARKs. Second, it is used as a building block in several CP-SNARKs proposed in [8]. Thus, one interested in having Sub-ZK LegoSNARK or Sub-ZK CP-SNARKs inevitably needs a Sub-ZK QA-NIZK for linear subspaces. Importantly, in [8,14], one uses a QA-NIZK to prove that an element belongs to the trivial full space; in this case, a QA-NIZK is sound by default. Instead, one has to prove that the stronger property of knowledge-soundness holds.

The main goal of the current paper is the study and construction of subversion-secure QA-NIZKs. According to the original security definitions of QA-NIZKs [28], one aims for soundness (alternatively, knowledge-soundness in applications like [8,14]) and zero-knowledge in the case when both ϱ and the CRS are honestly generated. In reality, it means that in the case of QA-NIZKs, one will have one more subversion-attack vector than in the case of SNARKs: namely, one has to consider both the case of a subverted language parameter (the Sub-PAR case) and the case of a subverted CRS. The Sub-PAR case with *honestly generated CRS* was tackled in [29] (updated full version of [28] from September 2018) where both Sub-PAR soundness and Sub-PAR zero-knowledge were shown to be achievable for a large family of subspace languages.[2] Since the simulator does not need access to a language parameter trapdoor td_ϱ, one does not have to extract td_ϱ for the simulation to be possible. Moreover, in the Sub-PAR case, the CRS is still honestly generated, which means that the simulator has access to the CRS trapdoor td.

Translated to the language of QA-NIZKs, by the impossibility result of [4], one cannot achieve both soundness and zero-knowledge in the case both ϱ and the CRS have been subverted. Therefore, in the rest of the paper, we study the slightly more relaxed case when (knowledge-)soundness holds if only ϱ has been subverted and zero-knowledge holds when both ϱ and the CRS have been subverted. It is unclear whether one can use existing techniques to construct a Sub-ZK version of the most efficient QA-NIZKs like Π_{kw} by Kiltz and Wee [31] in this case. First, ϱ has to be modeled separately from other inputs; no such parameter exists in the case of SNARKs. The existence of ϱ (and the dependence of the CRS on it) is the main reason why falsifiable QA-NIZKs are so efficient.

[2] This does not contradict the impossibility result of [4] (that achieving Sub-CRS soundness and Sub-CRS zero-knowledge at the same time is impossible for non-trivial languages) since ϱ plays a different role compared to CRS.

Second, known QA-NIZKs have a very different structure compared to SNARKs. For example, the most efficient known QA-NIZK for linear subspaces Π_{kw} by Kiltz and Wee [31] has a trapdoor matrix $\boldsymbol{K}$, but $[\boldsymbol{K}]_1$ is not explicitly given in the CRS. This means that the knowledge assumptions of [2,15] or knowledge-of-exponent assumptions [10] (that all rely on $[\alpha]_\iota$ being in the CRS for each trapdoor α) cannot be directly translated to the case of (Kiltz-Wee) QA-NIZK, and thus one seems to need quite different knowledge assumptions.

Third, another significant difference is that the soundness of efficient QA-NIZKs like [1,28,30–32] is based on standard falsifiable assumptions like KerMDH. Thus, intuitively, the use of non-falsifiable assumptions to prove Sub-ZK of a (sound) QA-NIZK seems to be less justifiable than in the case of proving Sub-ZK of zk-SNARKs since in the case of zk-SNARKs, non-falsifiable assumptions are needed to get soundness anyhow [19]. Moreover, while Bellare *et al.* had a discussion motivating the use of knowledge assumptions to obtain Sub-ZK, they did not have a formal proof of their necessity. Can one base Sub-ZK QA-NIZKs on falsifiable assumptions or prove it is impossible? (Non-subversion zero-knowledge) QA-NIZKs do not always rely on falsifiable assumptions: in the applications of QA-NIZKs in [8,14,25,37], one proves the "membership" in the full space that only makes sense under knowledge assumptions.

This brings us to the main questions of the current work:

(i) Are non-black-box techniques needed to prove Sub-ZK of NIZKs for languages outside of BPP*?*
(ii) Are (knowledge-)soundness and zero-knowledge achievable in the previously described model, i.e., only ϱ has been subverted in the case of soundness, and both ϱ and the CRS are subverted in the case of zero-knowledge? From this point on, we assume Sub-ZK QA-NIZK works in this model.
(iii) Can one obtain Sub-ZK QA-NIZKs for linear subspaces without modifying the existing constructions?

Our Contributions. We answer to the above main questions (with yes, yes, and mostly yes). It turns out that achieving Sub-ZK for state-of-the-art QA-NIZKs is considerably more complicated than for state-of-the-art SNARKs. This follows partly from the nature of QA-NIZKs (the existence of separate ϱ and pk) and from the construction of the concrete QA-NIZK. In the most relevant case ($k = 1$), it turns out that the most efficient existing QA-NIZK by Kiltz and Wee [31] is Sub-ZK (in the model described above) under a (novel) knowledge assumption given suitable algorithms that verify the correctness of both ϱ and pk. Hence, in this case, Sub-ZK comes almost for free: one only has to perform some additional computations that verify the correctness of the (language parameter and) CRS, and the proof of Sub-ZK relies on a non-falsifiable assumption.

First, we make a conceptually important observation that Sub-ZK in the CRS model, as defined in [2,4,15], is equal to *no-auxiliary-string non-black-box* zero knowledge [21] in the BPK model [9,38]. In the BPK model, the verifier (but not the prover) has a public key; and the key authority executes the functionality of an immutable bulletin board by storing the received public keys.

A zero-knowledge argument in the BPK model is either designated-verifier (the argument convinces only the designated verifier) when using the verifier's own public key or transferable (the verifier can transfer the argument to other verifiers and convince them of its validity) when using the public key pk of a third party; the latter case is essentially equivalent to the CRS model with pk being the CRS, $\mathsf{pk} = \mathsf{crs}$. The BPK model is significantly weaker than the CRS model, being arguably the weakest public key or parameter based trust model under which complicated functionalities like zero-knowledge are known to exist.

This important positive connection between no-auxiliary-string non-black-box zero knowledge and Sub-ZK was missed in the prior work on Sub-ZK; we hope it will simplify the construction and analysis of the future Sub-ZK argument systems. Because of that connection, we will usually use the abbreviation Sub-ZK to denote no-auxiliary-string non-black-box zero knowledge, but we explicitly 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 BPP [21], it follows that in the BPK model, auxiliary-string non-black-box NIZK is possible only for languages in BPP. This provides a simple proof that one can only construct non-auxiliary-string non-black-box NIZK for languages outside of BPP and thus provides an answer to the open question (i).

In Sect. 3, we define the security of QA-NIZK arguments in the BPK model; for this, we strengthen the "strong" QA-NIZK security definitions from [29] (as updated on September 2018) that consider the case of subverted ϱ but honestly generated pk. We allow for both ϱ and pk to be subverted. We model the resulting definition of *persistent zero-knowledge* after the Sub-ZK definition of SNARKs in [2], allocating a special role for the language parameter ϱ. More precisely, we require that for any efficient malicious $\mathcal{C}$ that creates the language parameter creator and the public key, there exists an efficient extractor $\mathsf{Ext}_{\mathcal{C}}$, s.t. if $\mathcal{C}$, by using random coins r, generates a language parameter ϱ and a public key pk (since there is no auxiliary input, ϱ and pk *have* to be generated by $\mathcal{C}$) then $\mathsf{Ext}_{\mathcal{C}}$, given r, outputs the secret key sk corresponding to pk.

Since we allow both ϱ and pk to be subverted, it is possible that the subverter sets $\mathsf{sk} = \mathsf{td}_\varrho$ for td_ϱ being a trapdoor for a parameter ϱ, e.g. for Kiltz-Wee QA-NIZK, $\varrho = [\boldsymbol{M}]_1$ and $\mathsf{td}_\varrho = \boldsymbol{M}$. As we show in Sect. 4, this can result in pathological QA-NIZK argument systems that are persistent zero-knowledge but not standard zero-knowledge. (This is possible since we consider an extractor that extracts the trapdoor behind ϱ and returns this as the secret key.) Hence, we say that a QA-NIZK argument system is *no-auxiliary-string non-black-box zero-knowledge (i.e., Sub-ZK)* iff it is both standard zero-knowledge and persistent zero-knowledge.

As the next main contribution, we study a variant Π_{bpk} of the most-efficient known QA-NIZK for linear subspaces Π_{kw} by Kiltz and Wee [31] (denoted there as Π'_{as}). Π_{kw} is known to be perfectly zero-knowledge and computationally sound in the CRS model under a suitable KerMDH assumption, [31] for a matrix distribution $\mathcal{D}_k$ where k is a small security-assumption-related integer; $k = 1$ in the case of asymmetric pairings. In Π_{kw}, the CRS includes a matrix $[\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k \times k}$ (assumed to be distributed according to $\mathcal{D}_k$) and the argument consists of only k group elements (thus, smaller k results in better efficiency). In the variant of Π_{kw} proposed in the current paper, pk of Π_{bpk} includes a new component $\mathsf{pk}^{\mathsf{pkv}}$ that helps to publicly check that even adversarially generated $[\bar{\boldsymbol{A}}]_2$ in pk has full rank k. In the case of many distributions $\mathcal{D}_k$ that are important in practice (we will call such distributions *efficiently verifiable*), the latter verification can be done efficiently only based on the knowledge of $[\bar{\boldsymbol{A}}]_2$ itself and thus $\mathsf{pk}^{\mathsf{pkv}}$ will be an empty string. Similarly to [2], we also define an efficient public-key verification algorithm that we denote by PKV. On top of it, we also define an efficient ϱ-verification algorithm PARV. We emphasize that we analyze Π_{kw} since it is the most efficient known QA-NIZK for linear subspaces. We leave analyzing other QA-NIZKs (that will hopefully be easier to do following our definitional framework and analysis of Π_{kw}) to the further work.

Since in the case of verifiable $\mathcal{D}_k$, we do not modify the public-key generation and the prover (thus, essentially $\Pi_{\mathsf{kw}} = \Pi_{\mathsf{bpk}}$), the (non-subversion) soundness of Π_{bpk} in the BPK model follows directly from [31]. In the non-verifiable special case $\mathcal{D}_k = \mathcal{U}_2$, we add some extra elements to pk and then prove the (non-subversion) soundness of Π_{bpk} under the SKerMDH assumption of [23]. In the subversion-case, when the language parameter could have been subverted, we prove (subverted-ϱ) soundness under KerMDH$^{\mathrm{dl}}$ or SKerMDH$^{\mathrm{dl}}$ assumption. Here, if X and Y are two assumptions, X^Y is the interactive assumption that X holds even if the adversary was given non-adaptive access to a Y oracle. See [34] for a thorough treatment of X^Y-type assumptions. Interestingly, up to now, the only non-falsifiable assumptions that have been used to construct efficient succinct NIZKs are knowledge assumptions; the use of (seemingly more standard) X^Y-type assumptions instead is one of the possibly most interesting contributions of the current paper.

As mentioned before, *knowledge-sound* QA-NIZKs are also interesting in the case when one uses them to prove the membership in the full space. We prove that Π_{bpk} is knowledge-sound by modifying a similar knowledge-soundness proof from [8] that, however, was only given in the non-subversion case, and only for $k = 1$. We use a SDL$^{\mathrm{dl}}$ (where SDL is the symmetric discrete logarithm assumption, [5]) assumption, like in the case of soundness proofs, to get knowledge-soundness even in the subversion case. We modify the proof of [8] so that it generalizes to arbitrary k. Moreover, knowledge-soundness will rely on both the SDL$^{\mathrm{dl}}$ and a hash-algebraic knowledge (HAK) assumption. In [37], Lipmaa recently defined the framework of HAK assumptions to make the algebraic group model (AGM) of Fuchsbauer *et al.* [16] more concrete and applicable. While in the AGM, it is assumed that every adversary is algebraic, a HAK assumption is defined with respect to a concrete input distribution of the adversary. I.e., a $\mathcal{D}$-HAK assumption states that if an adversary obtains an input (a vector of group elements) distributed according to a fixed distribution $\mathcal{D}$ then she knows how the group elements that she outputs depend on the input. HAK assumptions are even weaker: they allow for the case an adversary has additionally generated high min-entropy (but not necessarily uniformly random) group elements by using say elliptic-curve hashing.

Since Π_{kw} is perfectly zero-knowledge [31], we now only have to prove that it is also persistent zero-knowledge; from this, it follows that it is Sub-ZK in the BPK model. We prove that Π_{bpk} is statistically persistent zero-knowledge under either one of the two new knowledge assumptions KWKE (the *Kiltz-Wee Knowledge of Exponent* assumption) and SKWKE (the *strong* KWKE assumption)[3], assuming that its whole pk is generated by the verifier or a verifier-trusted authority—even if we are set to prove Sub-ZK that interests the prover. Intuitively, (S)KWKE guarantees that if an adversary $\mathcal{A}$ has succeeded in creating a pk accepted by PKV then one can extract corresponding $\mathsf{sk} = \boldsymbol{K}$. We prove that both assumptions hold under a *hash-algebraic knowledge* (HAK, [37]) assumption, see Theorem 1. (Here, SKWKE also relies on a computational assumption that depends on the matrix distribution $\mathcal{D}_k$ but is equal to the discrete logarithm assumption for all standard distributions $\mathcal{D}_k$.)

The proof of Theorem 1 is quite intricate. More precisely, we use a HAK assumption to extract some outputs of $\mathcal{A}$ as polynomials in indeterminates created by $\mathcal{A}$. To extract an integer sk, we use the Schwartz-Zippel lemma and let the extractor output evaluation of the polynomials at a random point. We then use the specific form of PKV to argue that such sk is correct. In the case of SKWKE, we evaluate the polynomials at two random points and use an additional reduction to a computational assumption, see Theorem 1.

Interestingly, under KWKE we only get the guarantee that the part $\mathsf{pk}^{\mathsf{zk}}$ of the pk, used either by the prover or the simulator, has been correctly computed. This, however, suffices to prove that Π_{bpk} is Sub-ZK. (Thus, Sub-ZK can be achieved even if the correctness of the whole public key cannot be verified.) Hence, in the case $\mathcal{D}_k$ is efficiently verifiable, one can get Sub-ZK essentially for free (efficiency-wise, the only added cost will be the need for a prover to verify the correctness of the public key; this can, however, be done once per public key). This is important since it means that in the case of efficiently verifiable matrix distributions, we get a stronger security property (Sub-ZK) without having to design a new, more complicated, and less efficient QA-NIZK. Arguably, in practice, one is only interested in efficiently verifiable distributions: the case $k = 1$ is the most one, and the case $k = 2$ is only needed in some applications (e.g., when one wants to rely on a weaker assumption). However, in such cases, one can usually use an efficiently verifiable distribution like $\mathcal{L}_2$ that corresponds to the 2-Lin assumption. This answers to the open questions (ii–iii).

We also show that under a stronger knowledge assumption SKWKE, one can guarantee that the whole pk has been correctly computed. However, as a drawback, the SKWKE assumption only holds if the language parameter $[\boldsymbol{M}]_1$ comes from a suitable hard distribution. The latter is, however, often the case in QA-NIZK applications, where $[\boldsymbol{M}]_1$ is a public key of a cryptographic primitive like an encryption or commitment scheme. In both cases, the soundness is guaranteed by a KerMDH assumption.

[3] It is possible to achieve the same level of security using more standard BDHKE assumption [2] by making both $[\boldsymbol{M}]_1$ and $[\boldsymbol{M}]_2$ public. Unfortunately, such a solution is less efficient; our goal was to achieve maximum efficiency.

2 Preliminaries

A random variable X has min-entropy k, $H_\infty(X) = k$, if $\max_x \Pr[X = x] = 2^{-k}$. 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 $\operatorname{im}(\mathcal{A})$ be the image of $\mathcal{A}$ (the set of valid outputs of $\mathcal{A}$), let $\mathsf{RND}_\lambda(\mathcal{A})$ denote the random tape of $\mathcal{A}$ (assuming the given value of λ), and let $r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A})$ denote the random choice of the randomizer r from $\mathsf{RND}_\lambda(\mathcal{A})$. We denote by $\mathsf{negl}(\lambda)$ an arbitrary negligible function. We write $a \approx_\lambda b$ if $|a - b| \leq \mathsf{negl}(\lambda)$. We follow Bellare *et al.* [4] 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 [4] for a discussion.

A bilinear group generator $\mathsf{PGen}(1^\lambda)$ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are additive cyclic groups of prime order $p = 2^{\Omega(\lambda)}$, $[1]_1, [1]_2$ are generators of $\mathbb{G}_1$, $\mathbb{G}_2$, resp., and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a non-degenerate PPT-computable bilinear pairing. We assume the bilinear pairing to be Type-3, 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 by now standard bracket notation, i.e., we write $[a]_\iota$ to denote $a g_\iota$ where g_ι is a fixed generator of $\mathbb{G}_\iota$. We denote $\hat{e}([a]_1, [b]_2)$ as $[a]_1[b]_2$. Thus, $[a]_1[b]_2 = [ab]_T$. We freely use the bracket notation with matrices, e.g., if $\boldsymbol{AB} = \boldsymbol{C}$ then $\boldsymbol{A}[\boldsymbol{B}]_\iota = [\boldsymbol{C}]_\iota$ and $[\boldsymbol{A}]_1[\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$.

In the Bare Public Key (BPK) model [9,38], parties have access to a public file F, a polynomial-size collection of records (id, pk_{id}), where id is a string identifying a party (e.g., a verifier), and pk_{id} is her alleged public key. In a typical zero-knowledge protocol in the BPK model, a key-owning party $\mathcal{P}_{id}$ works in two stages. In stage one (the *key-generation stage*), on input a security parameter 1^λ and randomizer r, $\mathcal{P}_{id}$ outputs a public key pk_{id} and stores the corresponding secret key sk_{id}. After that, F will include (id, pk_{id}). In stage two, each party has access to F, while $\mathcal{P}_{id}$ has possible access to 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 [38]; see also Sect. 3.

In a zero-knowledge proof or argument system, a prover convinces the verifier of 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 [6] 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. Goldreich and Oren [21] 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 $\mathscr{L}$ outside of BPP, see Lemma 1.

The *Symmetric Discrete Logarithm (SDL)* [5] *assumption* holds relative to PGen, if for any PPT $\mathcal{A}$, $\Pr\left[\mathsf{p} \leftarrow \mathsf{PGen}(1^\lambda); x \leftarrow_{\$} \mathbb{Z}_p : \mathcal{A}(\mathsf{p}, [x]_1, [x]_2) = x\right] \approx_\lambda 0$.

Kernel Matrix Diffie-Hellman Assumption (KerMDH) is a well-known assumption family formally introduced in [39]. Let $\mathcal{D}_{\ell k}$ be a probability distribution over matrices in $\mathbb{Z}_p^{\ell \times k}$, where $\ell > k$. Next, we define five commonly used distributions (see [13] for references), where $a, a_i, a_{ij} \leftarrow_{\$} \mathbb{Z}_p^*$: $\mathcal{U}_k$ (uniform), $\mathcal{L}_k$ (linear), $\mathcal{IL}_k$ (incremental linear), $\mathcal{C}_k$ (cascade), $\mathcal{SC}_k$ (symmetric cascade):

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

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

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

Assume that $\mathcal{D}_{\ell k}$ outputs matrices $\boldsymbol{A}$ where the upper $k \times k$ submatrix $\bar{\boldsymbol{A}}$ is always invertible. I.e., $\mathcal{D}_{\ell k}$ is *robust*, [28]. All the above distributions can be made robust with minimal changes. Denote the lower $(\ell - k) \times k$ submatrix of $\boldsymbol{A}$ as $\underline{\boldsymbol{A}}$. Denote $\mathcal{D}_k = \mathcal{D}_{k+1,k}$.

$\mathcal{D}_{\ell k}$-KerMDH$_{\mathbb{G}_1}$ [39] holds relative to PGen, if for any PPT $\mathcal{A}$, $\Pr\left[\mathsf{p} \leftarrow \mathsf{PGen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\ell k}; [\boldsymbol{c}]_2 \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1) : \boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_k \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell\right] \approx_\lambda 0$. $\mathcal{D}_{\ell k}$-SKerMDH [23] holds relative to PGen, if for any PPT $\mathcal{A}$, $\Pr[\mathsf{p} \leftarrow \mathsf{PGen}(1^\lambda); \boldsymbol{A} \leftarrow_{\$} \mathcal{D}_{\ell k}; ([\boldsymbol{c}_1]_1, [\boldsymbol{c}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1, [\boldsymbol{A}]_2) : \boldsymbol{A}^\top(\boldsymbol{c}_1 - \boldsymbol{c}_2) = \boldsymbol{0}_k \wedge \boldsymbol{c}_1 - \boldsymbol{c}_2 \neq \boldsymbol{0}_\ell] \approx_\lambda 0$. According to Lemma 1 of [23], if $\mathcal{D}_{\ell k}$-KerMDH holds in generic symmetric bilinear groups then $\mathcal{D}_{\ell k}$-SKerMDH holds in generic asymmetric bilinear groups. The KerMDH assumption holds also for Type-1 pairings, where $\mathbb{G}_1 = \mathbb{G}_2$, but then one needs $k \geq 2$, which affects efficiency.

Hash-Algebraic Knowledge Assumptions. The Algebraic Group Model (AGM) is a new model [16] that one can use to prove the security of a cryptographic assumption or protocol. Essentially, in AGM one assumes that each PPT algorithm (including the adversaries) is algebraic in the following sense: if the adversary $\mathcal{A}$'s input includes $[\boldsymbol{x}_\iota]_\iota$ and no other elements from the group $\mathbb{G}_\iota$ and $\mathcal{A}$ outputs group elements $[\boldsymbol{y}_\iota]_\iota$, then $\mathcal{A}$ knows matrices $\boldsymbol{N}_\iota$, such that $\boldsymbol{y}_\iota = \boldsymbol{N}_\iota \boldsymbol{x}_\iota$. Lipmaa [37] considered AGM to be as a family of algebraic knowledge assumptions. He defined the *AGM with hashing (AGMH)*, where the adversary is additionally allowed to create new group elements that have high min-entropy from the adversary's viewpoint (and in particular, without knowing their discrete logarithms). This takes into account the existence of efficient elliptic curve hashing algorithms that can be used to generate such new group elements.

Following [37], we say that a PPT algorithm $\mathcal{A}$ is *hash-algebraic (in* p*)* if there exists an efficient extractor $\mathsf{Ext}_{\mathcal{A}}$, such that for any PPT sampleable distribution $\mathcal{D}$, $\mathsf{Adv}^{\mathrm{hak}}_{\mathsf{p},\mathcal{D},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{l} \mathsf{x} = ([\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_2) \leftarrow_{\$} \mathcal{D}; r \leftarrow_{\$} \mathsf{RND}_{\lambda}(\mathcal{A}); ([\boldsymbol{y}_1]_1, [\boldsymbol{y}_2]_2) \leftarrow_{\$} \mathcal{A}(\mathsf{x}; r); \\ (\boldsymbol{N}_1, \boldsymbol{N}_2, [\boldsymbol{q}_1]_1, [\boldsymbol{q}_2]_2) \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathsf{x}; r) : \\ (\boldsymbol{y}_1 \neq \boldsymbol{N}_1 \left(\begin{smallmatrix} \boldsymbol{x}_1 \\ \boldsymbol{q}_1 \end{smallmatrix}\right) \vee \boldsymbol{y}_2 \neq \boldsymbol{N}_2 \left(\begin{smallmatrix} \boldsymbol{x}_2 \\ \boldsymbol{q}_2 \end{smallmatrix}\right)) \vee (\exists \iota, s : H_{\infty}([q_{\iota s}]_{\iota}) = O(\log \lambda)) \end{array}\right] .$$

A bilinear group p is *hash-algebraic* if every PPT algorithm $\mathcal{A}$ that obtains inputs from $\mathbb{G}_1/\mathbb{G}_2$ and outputs elements in $\mathbb{G}_1/\mathbb{G}_2$ is hash-algebraic. Clearly, a hash-algebraic adversary is less restricted than an algebraic adversary.

The requirement that $\mathcal{A}$ is hash-algebraic for a *concrete* $\mathcal{D}$ is a specific $(\mathsf{p}, \mathcal{D}, \mathcal{A})$-hash-algebraic knowledge (HAK) assumption stating that $\mathsf{Adv}^{\mathrm{hak}}_{\mathsf{p},\mathcal{D},\mathcal{A}}(\lambda) \approx_{\lambda} 0$. In AGMH, one assumes that $(\mathsf{p}, \mathcal{D}, \mathcal{A})$-HAK holds for all choices of $(\mathcal{D}, \mathcal{A})$. Alternatively, [37] calls it the p-*HAK assumption*. While proving the security of a concrete protocol in a fixed group p, it is sufficient to rely on the following assumption for a single specified distribution $\mathcal{D}$. A $(\mathsf{p}, \mathcal{D}, \mathcal{A})$-HAK assumption states that $\mathsf{Adv}^{\mathrm{hak}}_{\mathsf{p},\mathcal{D},\mathcal{A}}(\lambda) \approx_{\lambda} 0$. A $(\mathsf{p}, \mathcal{D})$-HAK assumption states that $\mathsf{Adv}^{\mathrm{hak}}_{\mathsf{p},\mathcal{D},\mathcal{A}}(\lambda) \approx_{\lambda} 0$ for all PPT $\mathcal{A}$. Analogously, the $(\mathcal{D}, \mathcal{A})$-*algebraic knowledge (AK) assumption in* p states that $\mathsf{Adv}^{\mathrm{ak}}_{\mathsf{p},\mathcal{D},\mathcal{A}}(\lambda) \approx_{\lambda} 0$.

Lipmaa [37] demonstrated the usefulness of the HAK assumption showing that Damgård's original Knowledge-of-Exponent (KE, [10]) assumption is secure under the DL and HAK assumptions. The opposite does not always hold: KE assumption (and its generalizations) cannot be used to extract unless each input group element $[z]_{\iota}$ is accompanied with a "knowledge" input $[xz]_{\iota}$ for random x. Thus, protocols that rely on HAK assumptions can, in principle, be more efficient than protocols that rely on KE assumptions only.

Intuitively, a security proof under the $(\mathsf{p}, \mathcal{D})$-HAK assumption constitutes essentially an AGMH security proof, but without one assuming that *all* PPT algorithms in the group p are (hash-)algebraic. Finally, according to the analysis of [37], it is sufficient to assume that $[\boldsymbol{q}_{\iota}]_1$ has high min-entropy while the previous approach of generic model with hashing as in [2,4,7,41] assumed that adversarially created group elements are uniformly random.

3 Defining QA-NIZK in the BPK Model

Quasi-Adaptive Non-Interactive Zero-Knowledge (QA-NIZK) argument systems [28] are quasi-adaptive in the sense that the CRS depends on a language parameter ϱ that has been sampled from a fixed distribution $\mathcal{D}_{\mathsf{p}}$. QA-NIZKs are of great interest since they are succinct and based on standard assumptions. Since QA-NIZKs have many applications, they have been a subject of intensive study, [1,23,28,30–33]. The main limitation of known QA-NIZKs is that *efficient* QA-NIZKs are only known for a restricted set of languages like the language of linear subspaces (see [12,23,24] for QA-NIZKs for other languages).

The original QA-NIZK security definitions [28] were given in the CRS model. Jutla and Roy strengthened the definitions in the full version of their paper, [29], allowing for the case when the language parameter is maliciously picked. We will lift the latter definitions to the weaker BPK model. Sometimes, the only difference compared to the definitions of [29] is in notation (a CRS will be replaced by a public key). The rest of the definitional changes are motivated by the definition of Sub-ZK zk-SNARKs in [2], e.g., a QA-NIZK in the BPK model will have a public-key verification algorithm PKV and the zero-knowledge definition mentions a subverter and an extractor. We also define a ϱ-verification algorithm PARV. Since black-box [38] and even auxiliary-input non-black-box [21] (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 [4], we will implicitly assume that the system parameters p are generated deterministically from λ; in particular, the choice of p cannot be subverted. A QA-NIZK argument system enables to prove membership in a language defined by a relation $\mathscr{R}_\varrho = \{(\mathsf{x}, \mathsf{w})\}$, which in turn is completely determined by a parameter ϱ sampled (in the honest case) from a distribution $\mathcal{D}_\mathsf{p}$. We will assume implicitly that ϱ contains p and thus not include p as an argument to algorithms that also input ϱ; recall that we assumed that p cannot be subverted. A distribution $\mathcal{D}_\mathsf{p}$ on $\mathscr{L}_\varrho$ is *witness-sampleable* [28] if there exists a PPT algorithm $\mathcal{D}'_\mathsf{p}$ that samples $(\varrho, \mathsf{td}_\varrho) \in \mathscr{R}_\mathsf{p}$ such that ϱ is distributed according to $\mathcal{D}_\mathsf{p}$.

The zero-knowledge simulator is usually required to be a single (non-black-box) PPT algorithm that works for the whole collection of relations $\mathscr{R}_\mathsf{p} = \{\mathscr{R}_\varrho\}_{\varrho \in \mathrm{im}(\mathcal{D}_\mathsf{p})}$; that is, one usually requires *uniform simulation* (see [28] for a discussion). Following [2], we accompany the universal simulator with an adversary-dependent extractor. We assume Sim also works in the case when one cannot efficiently establish whether $\varrho \in \mathrm{im}(\mathcal{D}_\mathsf{p})$. The simulator is not allowed to create new ϱ or pk but has to operate with one given to it as an input.

A tuple of PPT algorithms $\Pi = (\mathsf{PGen}, \mathsf{KGen}, \mathsf{PARV}, \mathsf{PKV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ is a *no-auxiliary-string non-black-box zero knowledge (Sub-ZK) QA-NIZK argument system* in the BPK model for a set of witness-relations $\mathscr{R}_\mathsf{p} = \{\mathscr{R}_\varrho\}_{\varrho \in \mathrm{Supp}(\mathcal{D}_\mathsf{p})}$, if the following Items i, ii, iv and v hold. Π is a *Sub-ZK QA-NIZK argument of knowledge*, if additionally Items iii holds. Here, PGen is the parameter generation algorithm, KGen is the public key generation algorithm, PARV is the ϱ-verification algorithm, PKV is the public-key verification algorithm, P is the prover, V is the verifier, and Sim is the simulator.

(i) **Perfect Completeness:** for any λ, $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$, PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{l} \varrho \leftarrow_\$ \mathcal{D}_\mathsf{p}; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\varrho); (\mathsf{x}, \mathsf{w}) \leftarrow \mathcal{A}(\mathsf{pk}); \\ \pi \leftarrow \mathsf{P}(\varrho, \mathsf{pk}, \mathsf{x}, \mathsf{w}) : \mathsf{PARV}(\varrho) = 1 \wedge \mathsf{PKV}(\varrho, \mathsf{pk}) = 1 \wedge \\ ((\mathsf{x}, \mathsf{w}) \notin \mathscr{R}_\varrho \vee \mathsf{V}(\varrho, \mathsf{pk}, \mathsf{x}, \pi) = 1) \end{array}\right] = 1 \ .$$

(ii) **Computational Quasi-Adaptive Sub-PAR Soundness:** for any $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$, and stateful PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}\varrho \leftarrow \mathcal{A}(\mathsf{p}); (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\varrho); (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathsf{pk}) : \\ \mathsf{PARV}(\varrho) = 1 \wedge \mathsf{V}(\varrho, \mathsf{pk}, \mathsf{x}, \pi) = 1 \wedge \neg(\exists \mathsf{w} : \mathscr{R}_\varrho(\mathsf{x}, \mathsf{w}) = 1))\end{array}\right] \approx_\lambda 0 \enspace .$$

(iii) **Computational Quasi-Adaptive Sub-PAR Knowledge-Soundness:** for every PPT stateful adversary adversary $\mathcal{A}$, there exist a PPT extractor $\mathsf{Ext}_\mathcal{A}$, s.t. for all $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$,

$$\Pr\left[\begin{array}{l}r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A}); \varrho \leftarrow \mathcal{A}(\mathsf{p}; r); (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\varrho); \\ (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathsf{pk}; r); \mathsf{w} \leftarrow \mathsf{Ext}_\mathcal{A}(\mathsf{p}, \mathsf{pk}; r) : \mathsf{PARV}(\varrho) = 1 \wedge \\ \mathsf{V}(\varrho, \mathsf{pk}, \mathsf{x}, \pi) = 1 \wedge \mathscr{R}_\varrho(\mathsf{x}, \mathsf{w}) = 0\end{array}\right] \approx_\lambda 0 \enspace .$$

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

(iv) **Statistical Zero Knowledge:** for any λ, $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$, and computationally unbounded adversary $\mathcal{A}$, $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| \approx_\lambda 0$, where $\varepsilon_b^{zk} :=$

$$\Pr\left[\, \varrho \leftarrow \mathcal{D}_\mathsf{p}; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\varrho) : \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\varrho, \mathsf{pk}) = 1 \quad \right] \enspace .$$

The oracle $\mathsf{O}_0(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \not\in \mathscr{R}_\varrho$, and otherwise it returns $\mathsf{P}(\varrho, \mathsf{pk}, \mathsf{x}, \mathsf{w})$. Similarly, $\mathsf{O}_1(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \not\in \mathscr{R}_\varrho$, and otherwise it returns $\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, \mathsf{x})$.

(v) **Statistical Persistent Zero Knowledge:** for any PPT subverter $\mathcal{C}$, there exists a PPT extractor $\mathsf{Ext}_\mathcal{C}$, s.t. for any λ, $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$, and computationally unbounded adversary $\mathcal{A}$, $|\varepsilon_0^{zk} - \varepsilon_1^{zk}| \approx_\lambda 0$, where

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

The oracle $\mathsf{O}_0(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \not\in \mathscr{R}_\varrho$, and otherwise it returns $\mathsf{P}(\varrho, \mathsf{pk}, \mathsf{x}, \mathsf{w})$. Similarly, $\mathsf{O}_1(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \not\in \mathscr{R}_\varrho$, and otherwise it returns $\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, \mathsf{x})$.

Π is *statistically no-auxiliary-string*[4] *non-black-box zero knowledge (Sub-ZK)* if it is both statistically zero-knowledge and statistically persistent zero-knowledge.

Knowledge-sound QA-NIZKs are useful in situations where the witness relations $\mathscr{R}_\varrho$ are trivial in the sense that for each x, there exists a w such that $(\mathsf{x}, \mathsf{w}) \in \mathscr{R}_\varrho$. In such cases, one must argue that the prover knows this w. Knowledge-sound QA-NIZK argument systems have applications in shuffles [14] and SNARKs [8, 25, 37].

In their definition of strong soundness for strong QA-NIZK, Jutla and Roy [29] made the assumption that $\mathcal{C}_\varrho$ also returns td_ϱ. This assumption reminds the AGM [16], where in the security proofs, the adversary is assumed to output

[4] Auxiliary-string non-black-box ZK [21] means that definitions hold even if any $\mathsf{aux} \in \{0,1\}^{\mathsf{poly}(\lambda)}$ is given as an additional input to $\mathcal{A}$ and $\mathcal{C}_\mathsf{pk}$ (and $\mathsf{Ext}_\mathcal{C}$).

a part of her secret state but might be stronger depending on the definition of $\mathcal{D}_{\mathsf{p}}$. Thus, one should not make such an assumption per se but prove (say, in the AGM) that it holds. In several recent reinterpretations of AGM [37], one has reworded AGM by requiring the existence of an extractor that returns the secret state. In our Sub-PAR (knowledge-)soundness definition, we require that $\mathsf{PARV}(\varrho) = 1$ (thus, $\varrho \in \mathrm{im}(\mathcal{D}_{\mathsf{p}})$ and a td_{ϱ} exists). We do not require td_{ϱ} can be extracted; we only require that w can be extracted. In our security proof, the extractor of w will first extract td_{ϱ} by using a DL oracle; we prove knowledge-soundness under a non-falsifiable assumption (more precisely, under the $\mathrm{SDL}^{\mathrm{dl}}$ assumption that states that solving SDL is intractable even if the adversary is given non-adaptive access to a DL oracle, see Fig. 6).

More precisely, in the case of the *concrete construction* of Π_{bpk}, extraction of td_{ϱ} is needed since the Π_{kw} argument system [31] (and thus also the Π_{bpk} argument system in Sect. 5) is only sound if $\mathcal{D}_{\mathsf{p}}$ is witness-sampleable. In the soundness proof in [31], one obtains td_{ϱ} from the honest ϱ-creator. In the Sub-PAR knowledge-soundness proof in Sect. 5, we extract td_{ϱ} from the malicious ϱ-creator $\mathcal{A}$ and then use td_{ϱ} to extract w. However, we use the DL oracle to extract td_{ϱ} and thus will need not have to rely on witness-sampleability of $\mathcal{D}_{\mathsf{p}}$.

We assume that a single subverter $\mathcal{C}$ produces ϱ and pk in the case of Sub-ZK, and the extractor will get access to the code of $\mathcal{C}$ and its inputs and random coins. The extractor never works with probability 1 since $\mathcal{C}$ can randomly sample (with a non-zero but negligible probability) a well-formed 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 differently from [2], the secret key extracted by $\mathsf{Ext}_{\mathcal{C}}$ is not unique in our case; see discussion in Sect. 5.)

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

For Sub-ZK, we require that both standard zero-knowledge (with trusted ϱ and pk generators) and persistent zero-knowledge (with possibly subverted ϱ and pk) generators hold. The reason behind requiring both is subtle and will be explained in Sect. 4. Very briefly, since one considers a single subverter $\mathcal{C}$ that creates both ϱ and pk, persistent zero-knowledge leaves one vulnerable against the subverter who just sets $\mathsf{sk} \leftarrow \mathsf{td}_{\varrho}$. While this attack is not possible in the case of all QA-NIZKs, as we show in Sect. 4, one can design a QA-NIZK argument system that is persistent zero-knowledge but not standard zero-knowledge. Intuitively, requiring that the same simulator Sim also works without the knowledge of td_{ϱ} makes it possible to avoid such pathological cases. However, it means that persistent zero-knowledge is not a strictly stronger notion than the standard zero-knowledge, and one requires both to obtain Sub-ZK.

Comparison to Earlier Sub-ZK Definitions. Subversion-security was defined by Bellare *et al.* [4] for the CRS model; further CRS-model subversion-security definitions were given in [2,15]. As proven in [4], one cannot achieve Sub-SND (soundness even if the CRS was generated maliciously) and non-subversion zero knowledge at the same time. Thus, subsequent efforts have concentrated on achieving either Sub-SND and witness-indistinguishability [4], subversion knowledge-soundness and witness-indistinguishability [17], or Sub-ZK (zero knowledge in the case the CRS was generated maliciously) and soundness [2,4,15]. In the latter case, the CRS is trusted by the verifier V while (following the definitions of [2]) the prover checks that the CRS is well-formed by using a publicly available algorithm. Thus, Sub-ZK in the CRS model is the same as zero-knowledge in the BPK model: the CRS has to be trusted by (or, even chosen by) V and hence can be equal to the public key of an entity trusted by V (or of V herself). Since black-box NIZK [38] and even auxiliary-string non-black-box NIZK [21] in the BPK model is impossible, one has to define no-auxiliary-string non-black-box zero knowledge (Sub-ZK) as above. Bellare *et al.* [4] motivated not incorporating auxiliary strings to the definition of Sub-ZK by known impossibility results. We will formalize this (folklore, see [42] 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* BPP.

Proof. The notions of (no-)auxiliary-string and (non-)-black-box zero-knowledge were defined by Goldreich and Oren [21] who proved that auxiliary-string (even non-black-box) zero-knowledge argument systems for languages outside of 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 BPP would contradict the impossibility result of [21]. □

Auxiliary-input zero-knowledge is usually used to achieve sequential composition of interactive zero-knowledge protocols, [21]. Sub-ZK guarantees sequential security in the case of NIZK, see [2] for a proof. In particular, the main result of [2,15], reformulated in our language, is that there exist computationally knowledge-sound Sub-ZK zk-SNARKs for NP in the BPK model.

In the case of QA-NIZKs, one has to deal with two parameters, ϱ (the language parameter) and pk (the public key). As shown in [29] (updated version from September 2018), one can achieve both soundness and zero-knowledge in the case when ϱ is subverted but pk is honestly chosen. In the persistent zero-knowledge definition, we allow for subverted pk and ϱ. Due to the impossibility result of [4], we are not aiming to achieve Sub-SND. Thus, in the definition of soundness, we assume that pk is honestly generated.

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

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 $\mathcal{D}_{\mathsf{p}}$ over $\mathbb{G}_1^{n \times m}$. We assume, following [28], that (n, m) is implicitly fixed by $\mathcal{D}_{\mathsf{p}}$. That is, a QA-NIZK for linear subspaces handles languages

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

The corresponding relation is defined as $\mathscr{R}_{[\boldsymbol{M}]_1} = \{([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathbb{G}_1^n \times \mathbb{Z}_p^m : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}\}$. This language is useful in many applications, [8,28]. As a typical application, let $[\boldsymbol{M}]_1 = [1, \mathsf{sk}]_1^\top$ be a public key of the Elgamal cryptosystem; then ciphertext $[\boldsymbol{y}]_1 \in \mathscr{L}_{[\boldsymbol{M}]_1}$ 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 [31]. In particular, they proposed a QA-NIZK Π_{kw} that assumes that the parameter $\varrho = [\boldsymbol{M}]_1 \in \mathbb{G}_1^{n \times m}$ is sampled from a witness-sampleable distribution $\mathcal{D}_{\mathsf{p}}$. Π_{kw} results in the argument that consists of k group elements, where k is the parameter ($k = 1$ being usually sufficient in the case of asymmetric pairings) related to the underlying KerMDH distribution. 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 PGen, [31]. Importantly, Π_{kw} is significantly more efficient than the Groth-Sahai NIZK for the same language. For the sake of completeness, Fig. 1 describes the Kiltz-Wee QA-NIZK argument system Π_{kw} for linear subspaces in the CRS model.

Some Applications of QA-NIZK in the BPK Model. The simplest example application is that of UC-commitments from [28], where a trusted third party generates a commitment key ϱ together with a QA-NIZK public key pk, and P opens the commitments later by disclosing a QA-NIZK argument of proper commitment under the commitment key ϱ. Here, ϱ should not be generated by P (who could then equivocate) or by V (who could then extract the message). However, pk can be generated by V. This allows one, securely generated ϱ, to be used in many applications, from UC-commitments to identity-based encryption. In each such application, a trusted authority trusted by V (e.g., V herself) can create her pk that takes the particularities of that application into account.

Another, arguably much more important application, is the use of Sub-ZK QA-NIZKs in the construction of Sub-ZK SNARKs. Several recent papers

[8,14,25,37] have used QA-NIZKs for subspace language to construct SNARKs. In these cases, one proves the membership in the trivial full vector space under knowledge assumption, resulting in a statement that (say) the argument belongs to the span of certain CRS elements only like in [37] or that two commitments that possibly use different commitment keys commit to the same vectors like in [14]. To obtain Sub-ZK SNARKs (under a knowledge assumption), in such cases also the QA-NIZK has to be Sub-ZK (under a knowledge assumption).

In many other applications, it is desirable that zero-knowledge holds even if both ϱ and pk both are chosen by V (or by possibly different parties, neither of which is trusted by P). The above Sub-ZK definitions cover this more realistic scenario; in addition, they do not require V to trust ϱ. One such application is in the LegoSNARK framework by Campanelli *et al.* [8]. LegoSNARK uses QA-NIZK for linear subspace to build Commit-and-Prove (CP) SNARKs, which can be securely and efficiently combined together, creating a complex proof system able to perform well even for heterogeneous instance representation. Unfortunately, most of the modern zk-SNARKs are not CP-SNARKs. Hence [8] proposed a QA-NIZK-based transformation that builds them using any Commit-Carrying (CC) SNARK; the latter are much more common, e.g., the most efficient zk-SNARK for QAP by Groth [27] is a CC-SNARK. Despite that, Campanelli *et al.* propose a number of CP-SNARKs that are QA-NIZK-based.

4 Persistent Zero-Knowledge $\not\Rightarrow$ Zero-Knowledge

Intuitively, it seems that persistent zero-knowledge follows from the standard zero-knowledge since the set of all possible PPT subverters $\mathcal{C}$ also includes honest algorithms. However, this intuition is wrong. We will next show that one can construct pathological QA-NIZK argument systems that achieve persistent zero-knowledge, but do not satisfy the usual definition of zero-knowledge and actually leak some information about the witness.

Let us consider a slight variation of the subspace language where $\varrho = ([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$[5] and the statement is that $[\boldsymbol{y}]_1$ belongs to the subspace spanned by the matrix $[\boldsymbol{M}]_1$. Moreover, for simplicity let us take $\boldsymbol{M} \leftarrow_{\$} \mathbb{Z}_p^{2\times 1}$. Consider the QA-NIZK argument system (*a leaky QA-NIZK*) in Fig. 2. It has secret keys from the same set $\mathbb{Z}_p^{2\times 1}$, and thus, $\boldsymbol{M}$ can pass as a secret key. *Leaky QA-NIZK* does not have a public key, the argument is simply $[\pi]_1 = [w]_1$, and the verification is done by checking that $[\pi]_1^\top [\boldsymbol{M}]_2^\top = [\boldsymbol{M}w]_1^\top [1]_2 = [\boldsymbol{y}]_1^\top [1]_2$. It is not standard zero-knowledge since the simulator only knows $[\boldsymbol{M}]_1$, $[\boldsymbol{M}]_2$, and $[\boldsymbol{y}]_1 = [M_1 w, M_2 w]_1$ and outputting $[w]_1$ breaks the following symmetric computational Diffie-Hellman (CDH) assumption: given input $([1, a, b]_1, [1, a, b]_2)$ for $a \leftarrow_{\$} \mathbb{Z}_p^*$, $b \leftarrow_{\$} \mathbb{Z}_p$, it is difficult to compute $[ab]_1$. To see this, let us suppose that the symmetric CDH challenge is $[1, a, b]_1$, $[1, a, b]_2$ for $a \leftarrow_{\$} \mathbb{Z}_p^*$, $b \leftarrow_{\$} \mathbb{Z}_p$. We denote $M_1 = 1/a$, $w = b$, $M_2 = M_2' M_1 = M_2'/a$ where $M_2' \leftarrow_{\$} \mathbb{Z}_p$. We also reset

[5] Even if ϱ is maliciously created, one can efficiently check whether it has the correct form. More precisely, given $\varrho = ([\boldsymbol{M}]_1, [\boldsymbol{M'}]_2)$, one can assure that $\boldsymbol{M} = \boldsymbol{M'}$ by checking $[\boldsymbol{M}]_1 [1]_2 = [1]_1 [\boldsymbol{M'}]_1$ and accepting only when that is the case.

```
$\mathcal{D}_{\mathsf{p}}$: $\boldsymbol{M} \leftarrow_{\$} \mathbb{Z}_p^{2\times 1}$; return $\varrho = ([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$;
$\mathsf{KGen}(\varrho)$: return $(\mathsf{pk} \leftarrow \perp, \mathsf{sk} \leftarrow \mathbb{Z}_p^{2\times 1})$;
$\mathsf{Ext}_{\mathcal{C}}(\mathsf{aux}_\varrho; r)$: Extract $\mathsf{sk} = (M_1, M_2)^\top$ by using BDHKE; return $\mathsf{sk}$;
$\mathsf{P}(\varrho, \mathsf{pk}, [\boldsymbol{y}]_1, w)$: return $[\pi]_1 \leftarrow [w]_1 \in \mathbb{G}_1^1$;
$\mathsf{Sim}(\varrho, \mathsf{pk}, \mathsf{sk}, [\boldsymbol{y}]_1)$: 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 $[\boldsymbol{y}]_1^\top [1]_2 = [\pi]_1^\top [\boldsymbol{M}]_2^\top$;
$\mathsf{PKV}(\varrho, \mathsf{pk})$: check that $\mathsf{pk} = \perp$;
```

Fig. 2. A contrived leaky subspace QA-NIZK where $\varrho = ([\boldsymbol{M}]_1, [\boldsymbol{M}]_2)$

generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ to be $[g]_1 = [a]_1$ and $[g]_2 = [a]_2$. Now if such simulator existed, we could run it with input $[M_1 g, M_2 g, M_1 wg, M_2 wg]_1 = [1, M_2', b, M_2' b]_1$, $[M_1 g, M_2 g]_2 = [1, M_2']_2$ and it would output $[wg]_1 = [ba]_1$; this would break the CDH assumption.

Surprisingly, simulation is possible (under a knowledge assumption) if we try to prove persistent zero-knowledge. We remind that the Bilinear Diffie-Hellman Knowledge of Exponent (BDHKE) [2] assumption says that if a PPT adversary $\mathcal{A}(\mathsf{p})$ outputs $([x]_1, [x]_2)$ on random coins r, then there exists an extractor that extracts x with an overwhelming probability given the same random coins r. Thus, assuming BDHKE and because $\mathsf{Ext}_{\mathcal{C}}$ is given access to the random coins of $\mathcal{C}$, $\mathsf{Ext}_{\mathcal{C}}$ can extract $\boldsymbol{M}$ and provide it to the simulator as sk. The simulator then computes $[w]_1 = M_1^{-1}[y_1]_1$.

We could divide $\mathcal{C}$ into $\mathcal{C}_\varrho$, which generates ϱ, and $\mathcal{C}_{\mathsf{pk}}$, which generates pk, such that the extractor only gets random coins of $\mathcal{C}_{\mathsf{pk}}$. This would make it impossible to extract $\boldsymbol{M}$. However, this will not work since we cannot exclude communication between $\mathcal{C}_\varrho$ and $\mathcal{C}_{\mathsf{pk}}$, e.g., $\mathcal{C}_\varrho$ can compute pk herself and send it to $\mathcal{C}_{\mathsf{pk}}$. $\mathcal{C}_{\mathsf{pk}}$ outputs pk without having any knowledge of sk, making extracting sk impossible.

Because of that, we adopted a different solution: namely, we require that *a Sub-ZK QA-NIZK argument system must satisfy both standard zero-knowledge and persistent zero-knowledge with respect to the same simulator*. This solution rules out the intuitively insecure arguments like the one in Fig. 2.

5 Construction of a QA-NIZK in the BPK Model

In this section, we will show that if the membership of $[\bar{\boldsymbol{A}}]_2$ in $\mathcal{D}_k$ can be efficiently verified, then a slight variant Π_{bpk} of the Kiltz-Wee QA-NIZK Π_{kw} for linear subspaces [31] is secure (including Sub-ZK) in the BPK model. More precisely, we say that the distribution $\mathcal{D}_k$ is *efficiently verifiable*, if there exists an algorithm $\mathsf{MATV}([\bar{\boldsymbol{A}}]_2)$ that outputs 1 if $\bar{\boldsymbol{A}}$ is invertible (recall that we assume that the matrix distribution is robust) and well-formed with respect to $\mathcal{D}_k$ and otherwise outputs 0. Clearly, the distributions $\mathcal{D}_1$, $\mathcal{L}_k$, $\mathcal{IL}_k$, $\mathcal{C}_k$, and $\mathcal{SC}_k$ (for any k) are verifiable, as can be seen in Fig. 3, while the verification whether $[\bar{\boldsymbol{A}}]_2$ is

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

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

invertible is intractable for the distribution $\mathcal{U}_k$ if $k > 1$. Indeed, if $k = 2$ then in the latter case, one needs to test if $a_{11}a_{22} - a_{12}a_{21} = 0$, given only $[\bar{\boldsymbol{A}}]_2$; the case $k > 2$ is even more complicated. Nevertheless, we show that a slightly modified version of our construction works with the distribution $\mathcal{D}_2$.

Recall that in the BPK model, the public key pk (corresponds to the CRS in Π_{kw}) belongs either to the verifier V or to a party trusted by V. One proves computational soundness in the setting where V trusts that pk is honestly generated, i.e., that the corresponding sk is secret and pk is well-formed. Since pk is not trusted by the prover P, one proves Sub-ZK in the case of a maliciously generated pk. We assume that $[\boldsymbol{M}]_1$ is sampled by a PPT subverter, and moreover, the simulator does not know the corresponding witness $\boldsymbol{M}$ or any function of $\boldsymbol{M}$ not efficiently computable from $[\boldsymbol{M}]_1$.

To modify Π_{kw} so that it would be secure in the BPK model instead of the CRS model, the most straightforward idea is to divide pk into $\mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_1$ (the part of pk that is used by P and thus intuitively needed to guarantee zero knowledge) and $\mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2$ (the part of pk is used by V and thus intuitively needed to guarantee soundness). Thus, P (resp., 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 P's public key and $\mathsf{pk}^{\mathsf{snd}}_{\mathsf{V}}$ from 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 P and V have public keys while we want to have a situation, common in the BPK model, where only V has a public key.

Instead, we assume that V's public key pk is equal to the whole CRS and then construct a public-key verification algorithm PKV. For PKV to be efficient in the case $\mathcal{D}_k$ is not efficiently verifiable, we need to add some new elements (collectively denoted as $\mathsf{pk}^{\mathsf{pkv}}$) to pk. Figure 4 describes the new QA-NIZK Π_{bpk}. The construction of PKV will be explained in Sect. 6.

We will prove that in the BPK model, Π_{bpk} is statistically persistent zero-knowledge under a novel non-falsifiable assumption, computationally quasi-adaptively Sub-PAR sound under another novel non-falsifiable assumption, and (if $\boldsymbol{M}$ has full rank) computationally quasi-adaptively Sub-PAR knowledge-sound under two non-falsifiable assumptions, one of which is novel. Some of the new non-falsifiable assumptions do not belong to the family of knowledge

```
$\mathsf{KGen}(\varrho := [\boldsymbol{M}]_1 \in \mathbb{G}_1^{n\times m})$: $\boldsymbol{A} \leftarrow_{\$} \mathcal{D}_k$; $\boldsymbol{K} \leftarrow_{\$} \mathbb{Z}_p^{n\times k}$; $[\boldsymbol{C}]_2 \leftarrow [\boldsymbol{K}\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{n\times k}$; $[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}]_1^\top \boldsymbol{K} \in \mathbb{G}_1^{m\times k}$;
    if $\mathcal{D}_k$ is efficiently verifiable then $\mathsf{pk}^{\mathsf{pkv}} \leftarrow \epsilon$; elseif $\mathcal{D}_k = \mathcal{U}_2$ then $\mathsf{pk}^{\mathsf{pkv}} \leftarrow [a_{11}, a_{12}]_1$; fi ; $\mathsf{pk}^{\mathsf{snd}} \leftarrow [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2$; $\mathsf{pk}^{\mathsf{zk}} \leftarrow [\boldsymbol{P}]_1$; $\mathsf{pk} \leftarrow (\mathsf{pk}^{\mathsf{snd}}, \mathsf{pk}^{\mathsf{zk}}, \mathsf{pk}^{\mathsf{pkv}})$; $\mathsf{sk} \leftarrow \boldsymbol{K}$;
    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 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:
        $\mathsf{pk} = (\mathsf{pk}^{\mathsf{snd}}, \mathsf{pk}^{\mathsf{zk}}, \mathsf{pk}^{\mathsf{pkv}}) \wedge \mathsf{pk}^{\mathsf{snd}} = [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2 \wedge \mathsf{pk}^{\mathsf{zk}} = [\boldsymbol{P}]_1$;
        $[\boldsymbol{P}]_1 \in \mathbb{G}_1^{m\times k} \wedge [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k\times k} \wedge [\boldsymbol{C}]_2 \in \mathbb{G}_2^{n\times k}$;
    (*) $[\boldsymbol{M}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{P}]_1 [\bar{\boldsymbol{A}}]_2$;
        if $\mathcal{D}_k$ is efficiently verifiable then $\mathsf{MATV}([\bar{\boldsymbol{A}}]_2)$;
        else check $\mathsf{pk}^{\mathsf{pkv}} = [a_{11}^*, a_{12}^*]_1 \in \mathbb{G}_1^{1\times 2} \wedge [a_{11}^*]_1[1]_2 = [1]_1[a_{11}]_2 \wedge$
          $[a_{12}^*]_1[1]_2 = [1]_1[a_{12}]_2 \wedge [a_{11}^*]_1[a_{22}]_2 - [a_{12}^*]_1[a_{21}]_2 \neq [0]_T$; fi
```

Fig. 4. Sub-ZK QA-NIZK Π_{bpk} for $[\boldsymbol{y}]_1 = [\boldsymbol{M}]_1 \boldsymbol{w}$ in the BPK model, where either (1) $\mathcal{D}_k$ is efficiently verifiable or (2) $\mathcal{D}_k = \mathcal{U}_2$.

assumptions, which is an interesting result by itself. We will study new assumptions in Sect. 6, before stating and proving the security of Π_{bpk} in Sect. 7.

6 New Non-falsifiable Assumptions

We will next motivate and define the new assumptions. We will also prove the security of KWKE and SKWKE under the HAK assumptions.

KWKE and SKWKE Assumptions. In the Sub-ZK proof, we will need two different (tautological) knowledge assumptions, KWKE (Kiltz-Wee Knowledge of Exponent), and SKWKE (Strong Kiltz-Wee Knowledge of Exponent). Similarly to Sub-ZK SNARKs [2,15], the knowledge assumption is needed to equip the simulator Sim of Π_{kw} with the correct secret key $\mathsf{sk} = \boldsymbol{K}$.

The KWKE assumption 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 Sub-ZK, it is not needed that $\mathsf{pk}^{\mathsf{snd}}$ can be computed from sk, KWKE is sufficient. To argue that KWKE is a reasonable knowledge assumption, we prove that it holds under a hash-algebraic knowledge assumption.

We also introduce a stronger knowledge assumption SKWKE that allows extracting the *unique* secret key $\boldsymbol{K}$ that was used to generate the *whole* public key pk. We prove that SKWKE holds under a HAK and a WKerMDH assumption, given that $\mathcal{D}_k$ is a WKerMDH-hard distribution. (Here, WKerMDH is a weaker variant of the well-known KerMDH distribution.) The assumption of WKerMDH-hardness often holds in practice, e.g., when ϱ corresponds to a randomly chosen public key of a cryptosystem or a commitment scheme (see Sect. 3

for an example). After that, we will prove that Π_{bpk} is Sub-ZK under either KWKE or SKWKE; in the latter case, we additionally get a guarantee that the public key is correctly formed.

We will now define the new knowledge assumptions needed in the Sub-ZK proof. In KWKE, we assume that if $\mathcal{A}$ outputs a ϱ accepted by PARV and a pk accepted by 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 $\mathsf{pk}^{\mathsf{snd}}$.

Definition 1. *Fix $k \geq 1$, $n > m \geq 1$, and a distribution $\mathcal{D}_k$. Let* PKV *be as in Fig. 4. Then $(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)$-*KWKE$_{\mathbb{G}_1}$ *(resp., $\boxed{(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)\text{-SKWKE}_{\mathbb{G}_1}}$) holds relative to* 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. $\mathsf{Adv}^{\boxed{\mathsf{s}}\mathrm{kwke}}_{\mathcal{D}_{\mathsf{p}},k,\mathcal{D}_k,\mathbb{G}_1,\mathsf{PGen},\mathcal{A},\mathsf{Ext}_{\mathcal{A}}}(\lambda) :=$*

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

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

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

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

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 $[\boldsymbol{c}]_2$. (Although, it is easy to show that $\mathcal{D}_k$-KerMDH follows from $\mathcal{D}_k$-HAK and $\mathcal{D}_k$-WKerMDH.) The Discrete Logarithm (DL) 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 say $\mathcal{SC}_k$, the non-trivial co-kernel element $\boldsymbol{c}$ has to satisfy $c_2 = -ac_1$ which enables to recover a; thus, $\mathcal{SC}_k$-WKerMDH is secure under the DL assumption. Similarly, in the case of $\mathcal{C}_k$, $c_2 = -a_1 c_1$.

Next, we will prove that KWKE (resp., SKWKE) holds under the $\mathcal{D}_k$-HAK (resp., $\mathcal{D}_k$-HAK and $\mathcal{D}_{\mathsf{p}}$-WKerMDH) assumption. Note that the use of WKerMDH, and thus of SKWKE, is questionable if $\mathcal{C}_\varrho$ is malicious; nevertheless, we consider this case for the sake of completeness.

Theorem 1 (Security of KWKE **and** SKWKE**).** *Assume that either $\mathcal{D}_k$ is efficiently verifiable or $\mathcal{D}_k = \mathcal{U}_2$. Assume $k/p \approx_\lambda 0$. Then*

*(i) $(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)$-*KWKE$_{\mathbb{G}_1}$ *holds under the $\mathcal{D}_k$-HAK assumption.*
*(ii) assuming that $\mathcal{D}_k$-HAK and $\mathcal{D}_{\mathsf{p}}$-*WKerMDH$_{\mathbb{G}_1}$ *hold (thus, $\varrho = [\boldsymbol{M}]_1$ comes from a* WKerMDH$_{\mathbb{G}_1}$*-hard distribution), $(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)$-*SKWKE$_{\mathbb{G}_1}$ *holds.*

Proof. Assume $\mathcal{A}$ is a KWKE or SKWKE adversary, s.t.: given public parameters p and randomness $r \leftarrow_{\$} \mathsf{RND}_\lambda(\mathcal{A})$, $\mathcal{A}(\mathsf{p}; r)$ outputs with probability $\varepsilon_{\mathcal{A}}$ a

```
$\mathsf{Ext}_{\mathcal{A}}(\mathsf{p}; r)$

    $([\boldsymbol{M}]_1, \mathsf{pk}) \leftarrow \mathcal{A}(\mathsf{p}; r)$; if $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 0$ then return $\perp$; fi ;
    $(\boldsymbol{N}_1, [\boldsymbol{q}_1]_2, \boldsymbol{N}_2, [\boldsymbol{q}_2]_2) \leftarrow \mathsf{Ext}^{hak}_{\mathcal{A}}(\mathsf{p}; r)$; Abort if this fails;
    Let $\bar{\boldsymbol{A}}[i]$, $\boldsymbol{C}[i]$ be such that $\bar{\boldsymbol{A}} = \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] q_{2i}$ and $\boldsymbol{C} = \sum_{i \geq 0} \boldsymbol{C}[i] q_{2i}$;
    For each $i > 0$, sample random $y_i \leftarrow_{\$} \mathbb{Z}_p$;
(♯)if $\det(\bar{\boldsymbol{A}}(\boldsymbol{y})) = 0$ then return $\perp$; fi ; // Probability $k/p$
    return $\boldsymbol{K} \leftarrow \boldsymbol{C}(\boldsymbol{y}) \bar{\boldsymbol{A}}(\boldsymbol{y})^{-1}$;

$\mathsf{Ext}^2_{\mathcal{A}}(\mathsf{p}; r)$

$([\boldsymbol{M}]_1, \mathsf{pk}) \leftarrow \mathcal{A}(\mathsf{p}; r)$; if $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 0$ then return $\perp$; fi ;
$(\boldsymbol{N}_1, [\boldsymbol{q}_1]_2, \boldsymbol{N}_2, [\boldsymbol{q}_2]_2) \leftarrow \mathsf{Ext}^{hak}_{\mathcal{A}}(\mathsf{p}; r)$; Abort if this fails;
Let $\bar{\boldsymbol{A}}[i]$, $\boldsymbol{C}[i]$ be such that $\bar{\boldsymbol{A}} = \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] q_{2i}$ and $\boldsymbol{C} = \sum_{i \geq 0} \boldsymbol{C}[i] q_{2i}$;
For each $i > 0$, 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 $\leq 2k/p$
$\boldsymbol{K} \leftarrow \boldsymbol{C}(\boldsymbol{y}) \bar{\boldsymbol{A}}(\boldsymbol{y})^{-1}$; $\boldsymbol{K}' \leftarrow \boldsymbol{C}(\boldsymbol{y}') \bar{\boldsymbol{A}}(\boldsymbol{y}')^{-1}$;
if $\boldsymbol{K} \neq \boldsymbol{K}'$ then return $\boldsymbol{K} - \boldsymbol{K}'$; else return $\boldsymbol{K}$; fi ;
```

Fig. 5. Extractors $\mathsf{Ext}_{\mathcal{A}}(\mathsf{p}; r)$ and $\mathsf{Ext}^2_{\mathcal{A}}(\mathsf{p}; r)$ in the proof of Theorem 1

language parameter $\varrho = [\boldsymbol{M}]_1$ and public key $\mathsf{pk} = ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [\boldsymbol{P}]_1, \mathsf{pk}^{\mathsf{pkv}})$, such that $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$ (in particular, $\det \bar{\boldsymbol{A}} \neq 0$ and $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$).

(i: security of KWKE**):** Assume $\mathcal{A}$ is a KWKE adversary. Let $\mathsf{Ext}^{hak}_{\mathcal{A}}$ be the extractor, existence of which is guaranteed by the $\mathcal{D}_k$-HAK assumption. Figure 5 depicts a candidate KWKE-extractor $\mathsf{Ext}_{\mathcal{A}}$, where $[q_{\iota i}]_\iota$ for $i > 0$ are group elements created by $\mathcal{A}$ (for which she does not know the discrete logarithm) in $\mathbb{G}_\iota$, and $q_{\iota 0} = 1$. Due to the $\mathcal{D}_k$-HAK assumption, $\mathsf{Ext}^{hak}_{\mathcal{A}}$ can extract $\boldsymbol{N}_\iota$ and $[\boldsymbol{q}_\iota]_\iota$, such that $\begin{bmatrix} \mathrm{vect}(\boldsymbol{M}) \\ \mathrm{vect}(\boldsymbol{P}) \end{bmatrix}_1 = \boldsymbol{N}_1 \left[\begin{smallmatrix} 1 \\ \boldsymbol{q}_1 \end{smallmatrix}\right]_1 \in \mathbb{G}_1^{mn+mk}$ and $\begin{bmatrix} \mathrm{vect}(\bar{\boldsymbol{A}}) \\ \mathrm{vect}(\boldsymbol{C}) \end{bmatrix}_2 = \boldsymbol{N}_2 [\begin{smallmatrix} 1 \\ \boldsymbol{q}_2 \end{smallmatrix}]_2 \in \mathbb{G}_2^{k^2+nk}$. Here, $\mathrm{vect}(\boldsymbol{B})$ denotes the vectorization of a matrix $\boldsymbol{B}$. Thus, e.g., $\bar{A}_{ij} = \sum_{t \geq 0} N_{k(i-1)+j,t} \boldsymbol{q}_{2t}$ and $C_{ij} = \sum_{t \geq 0} N_{k(i-1)+j+k^2,t} \boldsymbol{q}_{2t}$. Given $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$, one can efficiently compute matrices $\boldsymbol{M}[j]$, $\boldsymbol{P}[j]$, $\bar{\boldsymbol{A}}[i]$ and $\boldsymbol{C}[i]$, such that the polynomials $\boldsymbol{M}(\boldsymbol{Q_1}) := \sum_{j \geq 0} \boldsymbol{M}[j] Q_{1j}$, $\boldsymbol{P}(\boldsymbol{Q_1}) := \sum_{j \geq 0} \boldsymbol{P}[j] Q_{1j}$, $\bar{\boldsymbol{A}}(\boldsymbol{Q}_2) := \sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Q_{2i}$, and $\boldsymbol{C}(\boldsymbol{Q}_2) := \sum_{i \geq 0} \boldsymbol{C}[i] Q_{2i}$ satisfy $[\boldsymbol{M}]_1 = [\boldsymbol{M}(\boldsymbol{q}_1)]_1$, $[\boldsymbol{P}]_1 = [\boldsymbol{P}(\boldsymbol{q}_1)]_1$, $[\bar{\boldsymbol{A}}]_2 = [\bar{\boldsymbol{A}}(\boldsymbol{q}_2)]_2$, and $[\boldsymbol{C}]_2 = [\boldsymbol{C}(\boldsymbol{q}_2)]_2$.

We will now show that $\mathsf{Ext}_{\mathcal{A}}$ satisfies the requirements of the extractor in the definition of KWKE. Assume that $\mathcal{A}$ was successful with inputs $(\mathsf{p}; r)$. We execute $\mathsf{Ext}_{\mathcal{A}}(\mathsf{p}; r)$ and obtain either $\boldsymbol{K}$ or $\perp$. From (*) in PKV (i.e., $\boldsymbol{M}^\top \boldsymbol{C} = \boldsymbol{P} \bar{\boldsymbol{A}}$), $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) := (\sum_{j \geq 0} \boldsymbol{M}[j] Q_{1j})^\top \cdot (\sum_{i \geq 0} \boldsymbol{C}[i] Q_{2i}) - (\sum_{j \geq 0} \boldsymbol{P}[j] Q_{1j}) \cdot (\sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Q_{2i})$ satisfies $V(\boldsymbol{q}_1, \boldsymbol{q}_2) = \boldsymbol{0}$. We now consider the following two cases, $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = \boldsymbol{0}$ as a polynomial and $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \neq \boldsymbol{0}$ but $V(\boldsymbol{q}_1, \boldsymbol{q}_2) = \boldsymbol{0}$.

Case 1: $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = \boldsymbol{0}_{m \times k}$ as a polynomial. Since Q_{1j} and Q_{2i} are indeterminates for all $i, j > 0$, the coefficients V_{ij} of $Q_{1j} Q_{2i}$ of $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) =$

$\sum_{i \geq 0, j \geq 0} V_{ij} Q_{1j} Q_{2i}$ must be equal to $\mathbf{0}_{m \times k}$ for all $i, j \geq 0$. In particular,

$$\boldsymbol{P}[j] \cdot \bar{\boldsymbol{A}}[i] = \boldsymbol{M}[j]^\top \boldsymbol{C}[i] \ , \quad i \geq 0, j \geq 0 \ . \tag{1}$$

Let $\bar{\boldsymbol{A}}(\boldsymbol{Q}_2) = \sum \bar{\boldsymbol{A}}[i] Q_{2i} \in \mathbb{Z}_p^{k \times k}[\boldsymbol{Q}_2]$ be an affine multivariate matrix polynomial and let the polynomial $d(\boldsymbol{Q}_2) := \det(\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)) \in \mathbb{Z}_p[\boldsymbol{Q}_2]$ be its determinant. Clearly, $\deg(d(\boldsymbol{Q}_2)) \leq k$, and $\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)$ is invertible iff $d(\boldsymbol{Q}_2) \neq 0$ as a polynomial. Since $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$, $d(\boldsymbol{Q}_2) \neq 0$ and thus $\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)$ is invertible. This holds by definition for efficiently verifiable $\mathcal{D}_k$. If $\mathcal{D}_k = \mathcal{U}_2$, then $[a_{1s}]_1[1]_2 = [1]_1[a_{1s}]_2$, for $s \in \{1, 2\}$, and $[a_{11}]_1[a_{22}]_2 \neq [a_{12}]_1[a_{21}]_2$ guarantee that $d(\boldsymbol{Q}_2) \neq 0$.

By the Schwartz-Zippel lemma, $d(\boldsymbol{y}) = 0$ 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 $\boldsymbol{K}(\boldsymbol{Q}_2) := \boldsymbol{C}(\boldsymbol{Q}_2)\bar{\boldsymbol{A}}^{-1}(\boldsymbol{Q}_2) = (\sum_{i \geq 0} \boldsymbol{C}[i] Q_{2i})(\sum_{i \geq 0} \bar{\boldsymbol{A}}[i] Q_{2i})^{-1} \in \mathbb{Z}_p^{n \times k}(\boldsymbol{Q}_2)$. Let $\boldsymbol{K} := \boldsymbol{K}(\boldsymbol{y})$. Since $\bar{\boldsymbol{A}}(\boldsymbol{y})$ is invertible then from Eq. (1), $\boldsymbol{P}[j] \cdot \bar{\boldsymbol{A}}(\boldsymbol{y}) = \boldsymbol{P}[j] \cdot \left(\sum_i \bar{\boldsymbol{A}}[i] y_i\right) = \boldsymbol{M}[j]^\top \left(\sum_i \boldsymbol{C}[i] y_i\right) = \boldsymbol{M}[j]^\top \boldsymbol{C}(\boldsymbol{y})$. Thus, $\boldsymbol{P}[j] = \boldsymbol{M}[j]^\top \boldsymbol{K}$, and $\boldsymbol{P}(\boldsymbol{Q}_1) = \boldsymbol{M}(\boldsymbol{Q}_1)^\top \boldsymbol{K}$. Hence, with probability $\varepsilon_{\mathsf{Ext}_{\mathcal{A}}} \geq \varepsilon_{\mathcal{A}} - k/p$, $\boldsymbol{P}(\boldsymbol{Q}_1) = \sum_{j \geq 0} \boldsymbol{P}[j] Q_{1j} = \boldsymbol{M}(\boldsymbol{Q}_1)^\top \boldsymbol{K}$. Thus, $|\varepsilon_{\mathsf{Ext}_{\mathcal{A}}} - \varepsilon_{\mathcal{A}}| \leq k/p$ and the claim follows.

Case 2: $V(\boldsymbol{X}, \boldsymbol{Q}_1, \boldsymbol{Q}_2) \neq \mathbf{0}$ *but* $V(\boldsymbol{x}, \boldsymbol{q}_1, \boldsymbol{q}_2) = \mathbf{0}$. Following [37], we consider separately the "non-hashing" case (the adversary creates no random elements $[q_\iota]_\iota$) and the "hashing" case (the adversary creates at least one random element that has high min-entropy).

In the non-hashing case, the verification polynomial is equal to the integer matrix $V := \boldsymbol{M}[0]^\top \boldsymbol{C}[0] - \boldsymbol{P}[0] \cdot \bar{\boldsymbol{A}}[0]$. Recall that $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \neq \mathbf{0}$ but $V(\boldsymbol{q}_1, \boldsymbol{q}_2) = \mathbf{0}$. Since we are in the non-hashing case, there are no created group elements. Thus, the adversary cannot succeed in the non-hashing since the polynomial V is constant, and we need $V = 0$ and $V \neq 0$ at the same time.

Consider now the "hashing" case when $\mathcal{A}$ has created at least one random group element q_k (say, in $\mathbb{G}_1$). Clearly, $V(\boldsymbol{Q}_1, \boldsymbol{Q}_2)$ is a degree-1 polynomial in any indeterminate Q_k. Thus, by the Schwartz-Zippel lemma and since $H_\infty([q_{\iota s}]_\iota) = \omega(\log \lambda)$, the probability $1/2^{\sum_{\iota, s} H_\infty([q_{\iota s}]_\iota)}$ that $V(\boldsymbol{q}_1, \boldsymbol{q}_2) = 0$ is negligible. Hence, the probability that an adversary, who created at least one (high min-entropy) group element $[q_k]_1$, can make the verifier accept is negligible.

(ii: security of SKWKE**):** Let $\mathcal{A}$ be an SKWKE adversary that works in time $\tau(\lambda)$ and outputs $([\boldsymbol{M}]_1, \mathsf{pk})$ accepted by PKV with probability $\varepsilon_{\mathcal{A}}$. To prove that SKWKE is secure, we need to additionally show that $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$. In the process, we need to assume that $\mathcal{D}_{\mathsf{p}}$-WKerMDH is hard against $\tau(\lambda)$-time adversaries. The general proof works exactly as in the KWKE case, except one change that we discuss below. (In particular, the Case 2 is exactly the same.) We omit other details of the proof.

More precisely, the main idea is that in the proof step (i) we already established that $\boldsymbol{C}(\boldsymbol{Q}_2) = \boldsymbol{K}(\boldsymbol{Q}_2)\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)$ as polynomials. In the current step, we need to show that $\boldsymbol{C}(\boldsymbol{Q}_2) = \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)$ holds, that is, $\boldsymbol{K}(\boldsymbol{Q}_2)$ is a constant function. To guarantee the latter, we check the value of the rational function $\boldsymbol{K}(\boldsymbol{Q}_2)$

at two positions. If the two values are different, we can break $\mathcal{D}_{\mathsf{p}}$-WKerMDH. Otherwise, w.h.p., $\boldsymbol{K}(\boldsymbol{Q}_2)$ is a constant function.

More precisely, consider the extractor $\mathsf{Ext}^2_{\mathcal{A}}$ in Fig. 5. 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 $\perp$. By following similar analysis as in the case (i), $\boldsymbol{P}(\boldsymbol{Q}_1) = \boldsymbol{M}(\boldsymbol{Q}_1)^\top \boldsymbol{K}$ and $\boldsymbol{P}(\boldsymbol{Q}_1) = \boldsymbol{M}(\boldsymbol{Q}_1)^\top \boldsymbol{K}'$ which means that $\boldsymbol{M}(\boldsymbol{Q}_1)^\top(\boldsymbol{K} - \boldsymbol{K}') = \boldsymbol{0}_{m \times k}$. 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 $\mathcal{D}_{\mathsf{p}}$-WKerMDH$_{\mathbb{G}_1}$. Since breaking $\mathcal{D}_{\mathsf{p}}$-WKerMDH is hard within $\tau(\lambda)$ steps, the probability $\varepsilon_{\mathrm{WKerMDH}}$ 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 $\boldsymbol{f}(\boldsymbol{Q}_2) := \boldsymbol{C}(\boldsymbol{Q}_2)\bar{\boldsymbol{A}}^{-1}(\boldsymbol{Q}_2) - \boldsymbol{K}$. Denote the (i, j)th coefficient of the matrix $\boldsymbol{f}(\boldsymbol{Q}_2)$ by $f_{ij}(\boldsymbol{Q}_2) = \sum_s C_{is}(\boldsymbol{Q}_2)\bar{A}^{-1}_{sj}(\boldsymbol{Q}_2) - K_{ij}$. Note that $f_{ij}(\boldsymbol{Q}_2) = f'_{ij}(\boldsymbol{Q}_2)/\det(\bar{\boldsymbol{A}}(\boldsymbol{Q}_2))$, where $f'_{ij}(\boldsymbol{Q}_2)$ is some polynomial of degree $\leq k$.

At this point, we know that $\det(\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)) \neq 0$. Thus, $\boldsymbol{f}(\boldsymbol{Q}_2) \neq \boldsymbol{0}$ iff $\boldsymbol{C}(\boldsymbol{Q}_2) - \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Q}_2) \neq \boldsymbol{0}$. From this and the Schwartz-Zippel lemma it follows that if $f_{ij}(\boldsymbol{Q}_2) \neq 0$ then $\Pr_{\boldsymbol{y}}[f_{ij}(\boldsymbol{y}) = 0] \leq k/p$. If $\boldsymbol{f}(\boldsymbol{Q}_2) \neq \boldsymbol{0}$ then there exists at least one (i_0, j_0), s.t. $f_{i_0,j_0}(\boldsymbol{Q}_2) \neq 0$ and thus $\Pr_{\boldsymbol{y}}[f_{i_0,j_0}(\boldsymbol{y}) = 0] \leq k/p$. Thus, if $\boldsymbol{f}(\boldsymbol{Q}_2) \neq \boldsymbol{0}$ then $\Pr_{\boldsymbol{y}}[\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{0}] \leq k/p$.

Hence, with probability $\varepsilon_{\mathsf{Ext}^2_{\mathcal{A}}} \geq \varepsilon_{\mathcal{A}} - 3k/p - \varepsilon_{\mathrm{WKerMDH}}$, $\boldsymbol{C}(\boldsymbol{Q}_2) = \boldsymbol{K}\bar{\boldsymbol{A}}(\boldsymbol{Q}_2)$ and thus $\boldsymbol{P}(\boldsymbol{Q}_1) = \boldsymbol{M}(\boldsymbol{Q}_1)^\top \boldsymbol{K}$ and $\boldsymbol{C} = \boldsymbol{K}\bar{\boldsymbol{A}}$. Thus, $|\varepsilon_{\mathsf{Ext}^2_{\mathcal{A}}} - \varepsilon_{\mathcal{A}}| \leq 3k/p + \varepsilon_{\mathrm{WKerMDH}}$ and the security of SKWKE follows. □

In the case of SKWKE, we extract the *unique* $\boldsymbol{K}$ used to compute the CRS. Following a proof idea from [2], it is easy to show that under either the KWKE (and thus, also the SKWKE) assumption Π_{bpk} is Sub-ZK.

New Interactive Assumptions KerMDH$^{\mathrm{dl}}$ and SKerMDH$^{\mathrm{dl}}$. Since in the case of efficiently verifiable $\mathcal{D}_k$, we essentially do not modify Π_{bpk} (we only define PKV), its Sub-PAR soundness *almost* follows from that of Π_{kw} [31]. The main difference is that, due to considering the subverted language parameter, we need to change how one extracts $\boldsymbol{M}$. Namely, in [31], the KerMDH adversary $\mathcal{B}$ defined in the soundness reduction obtains $([\boldsymbol{M}]_1, \boldsymbol{M})$ sampled from $\mathcal{D}'_{\mathsf{p}}$ (this relies on the witness-sampleability). In our proof of Sub-PAR soundness (Theorem 2 in Sect. 7), $\mathcal{B}$ obtains $[\boldsymbol{M}]_1 \leftarrow \mathcal{A}(\mathsf{p})$ and then uses a non-adaptive DL oracle to extract $\boldsymbol{M}$. This means that we prove Sub-PAR soundness under a new interactive non-falsifiable KerMDH$^{\mathrm{dl}}$ assumption; however, importantly, we do not require witness-sampleability.

Since in some applications (e.g., in the setting of symmetric pairings), one uses $\mathcal{D}_2 = \mathcal{U}_2$, we prove that if $k = 2$ and $\mathcal{D}_k = \mathcal{U}_k$, then Π_{bpk} is sound under another new interactive non-falsifiable SKerMDH$^{\mathrm{dl}}$ assumption. Intuitively, in this case, $\mathsf{pk}^{\mathsf{pkv}}$ contains additional elements, needed to efficiently check that $[\bar{\boldsymbol{A}}]_2$ has full rank. If $\mathcal{D}_k$ is efficiently verifiable then by definition, $\mathsf{pk}^{\mathsf{pkv}} = \varepsilon$

(empty string) is sufficient. Since for efficiency reasons, one is interested in only small values of k, we will not consider the case of non-verifiable $\mathcal{D}_k$ with $k > 2$.

In addition, we are interested in applying the QA-NIZK in the case $\boldsymbol{M}$ has rank n (i.e., the image of $\boldsymbol{M}$ is the full space). Since then soundness holds trivially, one must prove knowledge-soundness. We show that in this case, Π_{bpk} is Sub-PAR knowledge-sound under two non-falsifiable assumptions: a HAK knowledge assumption and the new interactive SDL$^{\mathrm{dl}}$ assumption. The KerMDH$^{\mathrm{dl}}$, SKerMDH$^{\mathrm{dl}}$, and SDL$^{\mathrm{dl}}$ assumptions are X^Y-type interactive assumptions as used in [20,34], where the assumption X is assumed to hold even if the adversary is given non-adaptive access (i.e., before the X challenge is chosen) to an oracle that solves the assumption Y.

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

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{PGen}(1^\lambda); st \leftarrow \mathcal{A}^{\mathrm{dl}(\cdot)}(\mathsf{p}); x \leftarrow_{\$} \mathbb{Z}_p : \mathcal{A}(\mathsf{p}, st, [x]_1, [x]_2) = x\right] \approx_\lambda 0 \ .$$

Here, the oracle $\mathrm{dl}([y]_1)$ returns the discrete logarithm y of $[y]_1$.

The $\mathcal{D}_{\ell k}$-KerMDH$^{\mathrm{dl}}_{\mathbb{G}_1}$ *assumption* holds relative to PGen, if for any PPT $\mathcal{A}$,

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

The $\mathcal{D}_{\ell k}$-SKerMDH$^{\mathrm{dl}}$ *assumption* holds relative to PGen, if for any PPT $\mathcal{A}$,

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

Generic-model security proofs of SDL$^{\mathrm{dl}}$ and SKerMDH$^{\mathrm{dl}}$ are very similar to those of SDL and KerMDH: the field elements returned by the DL oracle are independent of the challenge and thus do not influence the rest of proof.

One could use an AK assumption instead of the SDL$^{\mathrm{dl}}$ assumption. However, the AK assumption explicitly does not allow $\mathcal{A}$ to create new group elements by using elliptic-curve hashing. The SDL$^{\mathrm{dl}}$ assumption allows the adversary to create such group elements, but allows access to *non-adaptive* DL oracle to extract their discrete logarithms. It is also not an expanding assumption, differently to many knowledge assumptions (e.g., the PKE assumption [26] that underlies many pairing-based SNARKs) that allow one to extract long "plaintext" from a short "ciphertext". Hence, the SDL$^{\mathrm{dl}}$ assumption, while still non-falsifiable, seems to be somewhat more realistic than an AK assumption. On the other hand, we need to extract $\boldsymbol{y}$ and $\boldsymbol{\pi}$ from $\mathcal{A}$'s output after the challenge is known, adaptively. In this case, a knowledge assumption (HAK) is more realistic than an adaptive DL oracle that one could also just use to break SDL directly.

7 Security of $\boldsymbol{\Pi}_{\mathsf{bpk}}$

Theorem 2. *Let Π_{bpk} be the QA-NIZK argument system for linear subspaces from Fig. 4. The following statements hold in the BPK model. Assume that $\mathcal{D}_{\mathsf{p}}$ is such that* PARV *is efficient.*

(i) $\mathit{\Pi}_{\mathsf{bpk}}$ *is perfectly complete and perfectly zero-knowledge.*

(ii) If $(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)$-KWKE$_{\mathbb{G}_1}$ *holds relative to* PGen *then* $\mathit{\Pi}_{\mathsf{bpk}}$ *is statistically persistent zero-knowledge.*

(iii) Assume $\mathcal{D}_k$ *is efficiently verifiable (resp.,* $\mathcal{D}_k = \mathcal{U}_2$*). If* $\mathcal{D}_k$-KerMDH$^{\mathrm{dl}}$ *(resp.,* $\mathcal{D}_k$-SKerMDH$^{\mathrm{dl}}$*) holds relative to* PGen *then* $\mathit{\Pi}_{\mathsf{bpk}}$ *is computationally quasi-adaptively Sub-PAR sound.*

(iv) Assume $\boldsymbol{M}$ *has rank* n *(*$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}$ *always has a solution), and that* $\mathcal{D}_k$ *is robust. If* SDL$^{\mathrm{dl}}$ *and* $\mathsf{KGen}([\boldsymbol{M}]_1)$*-HAK, for arbitrary efficiently computable* $[\boldsymbol{M}]_1$*, hold relative to* PGen *then* $\mathit{\Pi}_{\mathsf{bpk}}$ *is computationally quasi-adaptively Sub-PAR knowledge-sound.*

Proof. **(i: perfect completeness/perfect zero-knowledge):** obvious.

(ii: persistent zero-knowledge): Let $\mathcal{C}$ be a subverter that computes $([\boldsymbol{M}]_1, \mathsf{pk})$ so as to break the Sub-ZK property. That is, $\mathcal{C}(\mathsf{p}; r_{\mathcal{C}})$ outputs $([\boldsymbol{M}]_1, \mathsf{aux}_{\mathsf{pk}})$. Let $\mathcal{B}$ be the adversary from Fig. 6. Note that $\mathsf{RND}_\lambda(\mathcal{B}) = \mathsf{RND}_\lambda(\mathcal{C})$. Under the $(\mathcal{D}_{\mathsf{p}}, k, \mathcal{D}_k)$-KWKE assumption, there exists an extractor $\mathsf{Ext}^2_{\mathcal{B}}$, such that if $\mathsf{PARV}([\boldsymbol{M}]_1) = 1$ and $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$ then $\mathsf{Ext}^2_{\mathcal{B}}(\mathsf{p}; r_{\mathcal{C}})$ outputs $\boldsymbol{K}$, such that $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$. We construct a trivial extractor $\mathsf{Ext}_{\mathcal{C}}(\mathsf{p}; r_{\mathcal{C}})$ for $\mathcal{C}$, as depicted in Fig. 6. Clearly, $\mathsf{Ext}_{\mathcal{C}}$ returns $\mathsf{sk} = \boldsymbol{K}$, such that $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K}$.

$\mathcal{B}(\mathsf{p}; r_{\mathcal{C}})$	$\mathsf{Ext}_{\mathcal{C}}(\mathsf{p}; r_{\mathcal{C}})$
$([\boldsymbol{M}]_1, \mathsf{pk}, \mathsf{aux}_{\mathcal{C}}) \leftarrow \mathcal{C}(\mathsf{p}; r_{\mathcal{C}});$ **return** $\mathsf{pk};$	**return** $\mathsf{Ext}^2_{\mathcal{B}}(\mathsf{p}; r_{\mathcal{C}});$

Fig. 6. The extractor and the constructed adversary $\mathcal{B}$ from the persistent zero-knowledge proof of Theorem 2.

Fix concrete values of λ, $\mathsf{p} \in \mathrm{im}(\mathsf{PGen}(1^\lambda))$ and $r_{\mathcal{C}} \in \mathsf{RND}_\lambda(\mathcal{C})$. Let $([\boldsymbol{M}]_1, \mathsf{pk}, \mathsf{aux}_{\mathsf{pk}}) \leftarrow \mathcal{C}(\mathsf{p}; r_{\mathcal{C}})$, and run $\mathsf{Ext}_{\mathcal{C}}(\mathsf{p}; r_{\mathcal{C}})$ to obtain $\boldsymbol{K}$. Fix $([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathscr{R}_{[\boldsymbol{M}]_1}$. It clearly suffices to show that if $\mathsf{PARV}([\boldsymbol{M}]_1) = 1$, $\mathsf{PKV}([\boldsymbol{M}]_1, \mathsf{pk}) = 1$ and $([\boldsymbol{y}]_1, \boldsymbol{w}) \in \mathscr{R}_{[\boldsymbol{M}]_1}$ then $\mathsf{O}_0([\boldsymbol{y}]_1, \boldsymbol{w}) = \mathsf{P}([\boldsymbol{M}]_1, \mathsf{pk}, [\boldsymbol{y}]_1, \boldsymbol{w}) = [\boldsymbol{P}]_1^\top \boldsymbol{w}$ and $\mathsf{O}_1([\boldsymbol{y}]_1, \boldsymbol{w}) = \mathsf{Sim}([\boldsymbol{M}]_1, \mathsf{pk}, \boldsymbol{K}, [\boldsymbol{y}]_1) = \boldsymbol{K}^\top [\boldsymbol{y}]_1$ 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 $([\boldsymbol{y}]_1; \boldsymbol{w}) \in \mathscr{R}_{[\boldsymbol{M}]_1}$ it follows that $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}$. Thus, $\mathsf{O}_0([\boldsymbol{y}]_1, \boldsymbol{w}) = [\boldsymbol{P}]_1^\top \boldsymbol{w} = [\boldsymbol{K}^\top \boldsymbol{M}\boldsymbol{w}]_1 = \boldsymbol{K}^\top [\boldsymbol{y}]_1 = \mathsf{O}_1([\boldsymbol{y}]_1, \boldsymbol{w})$. Hence, O_0 and O_1 have the same distribution, and thus, $\mathit{\Pi}_{\mathsf{bpk}}$ is persistent zero-knowledge under KWKE.

(iii: $\mathcal{D}_k$ is efficiently verifiable, Sub-PAR soundness under KerMDH$^{\mathrm{dl}}$**):** follows directly from the soundness proof of $\mathit{\Pi}_{\mathsf{kw}}$ in [31]. There is only one difference: If $[\boldsymbol{M}]_1$ is not subverted (like in [31]), then one can use the witness-sampleability of $\mathcal{D}_{\mathsf{p}}$ to extract $\boldsymbol{M}$, and get a reduction to the falsifiable KerMDH assumption. In the case of Sub-PAR soundness, since the language parameter can be subverted (and thus one cannot rely on witness-sampleability), we let $\mathcal{B}$ use the DL oracle to obtain $\boldsymbol{M}$ from $[\boldsymbol{M}]_1$ and then use it in the soundness

```
$\mathcal{B}^{\mathrm{dl}(\cdot)}(\mathsf{p})$
$[\boldsymbol{M}]_1 \leftarrow \mathcal{A}(\mathsf{p})$; // $\boldsymbol{M} \in \mathbb{Z}_p^{n \times m}$
Use DL oracle $nm$ times to obtain $\boldsymbol{M}$;
return $st \leftarrow \boldsymbol{M}$;
```

```
$\mathcal{B}(\mathsf{p}, st = \boldsymbol{M}, ([\boldsymbol{A}]_1, [\boldsymbol{A}]_2))$ // $([\boldsymbol{A}]_1, [\boldsymbol{A}]_2) \in \mathbb{G}_1^{(k+1) \times k} \times \mathbb{G}_2^{(k+1) \times k}$ with $\boldsymbol{A} = (a_{ij})$
Let $\boldsymbol{M}^{\perp} \in \mathbb{Z}_p^{n \times (n-m)}$ be a basis of the kernel of $\boldsymbol{M}^{\top}$;
$\boldsymbol{K}' \leftarrow_{\$} \mathbb{Z}_p^{n \times k}$; $\boldsymbol{R} \leftarrow_{\$} \mathbb{Z}_p^{(n-m-1) \times (k+1)}$;
$[\boldsymbol{A}']_2 \leftarrow \begin{pmatrix} [\boldsymbol{A}]_2 \\ \boldsymbol{R} \cdot [\boldsymbol{A}]_2 \end{pmatrix}$; // $\boldsymbol{A}' \in \mathbb{Z}_p^{(n-m+k) \times k}$
$[\boldsymbol{C}]_2 \leftarrow (\boldsymbol{K}' || \boldsymbol{M}^{\perp})[\boldsymbol{A}']_2$;
$[\boldsymbol{P}]_1 \leftarrow [\boldsymbol{M}^{\top} \boldsymbol{K}']_1$;
$\mathsf{pk}' \leftarrow ([\bar{\boldsymbol{A}}, \boldsymbol{C}]_2, [a_{11}, a_{12}, \boldsymbol{P}]_1)$;
$([\boldsymbol{y}]_1, [\boldsymbol{\pi}]_1) \leftarrow \mathcal{A}(\mathsf{pk}')$; // $[\boldsymbol{y}]_1 \in \mathbb{G}_1^n$, $[\boldsymbol{\pi}]_1 \in \mathbb{G}_1^k$
$[\boldsymbol{c}]_1^{\top} \leftarrow [(\boldsymbol{\pi}^{\top} - \boldsymbol{y}^{\top} \boldsymbol{K}') || - \boldsymbol{y}^{\top} \boldsymbol{M}^{\perp}]_1$;
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. 7. Adversary $\mathcal{B}$ in the soundness proof of Theorem 2 (reduction to SKerMDH$^{\mathrm{dl}}$)

proof of [31] to get a reduction to the non-falsifiable KerMDH$^{\mathrm{dl}}$ assumption. Importantly, in this case, witness-sampleability is not needed.

(iii: $\mathcal{D}_k = \mathcal{U}_2$, Sub-PAR soundness under SKerMDH$^{\mathrm{dl}}$**):** In the case $\mathcal{D}_k = \mathcal{U}_2$, the proof is *similar* to the soundness proof of Π_{kw} in [31]. However, since we added $[a_{11}, a_{12}]_1$ to the public key, we reduce instead to the SKerMDH$^{\mathrm{dl}}$ assumption; this complicates the proof.

Assume that $\mathcal{A}$ breaks the soundness of Π_{bpk} with probability ε. We will build an adversary $\mathcal{B}$, see Fig. 7, that breaks SKerMDH$^{\mathrm{dl}}$ with probability $\geq \varepsilon - 1/p$. First, $\mathcal{B}$ uses the DL oracle to obtain $\boldsymbol{M}$ from $[\boldsymbol{M}]_1$; this is needed since $[\boldsymbol{M}]_1$ could be subverted. Here, witness-sampleability is not needed. As above, when the language parameter is generated honestly, the DL oracle is not needed, and one instead relies on the witness-sampleability of $\mathcal{D}_{\mathsf{p}}$ to obtain a reduction to the falsifiable SKerMDH assumption.

Note that in Fig. 7, $[\bar{\boldsymbol{A}}']_2 = [\bar{\boldsymbol{A}}]_2 \in \mathbb{G}_2^{k \times k}$. 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, $[\boldsymbol{C}]_2 = (\boldsymbol{K}' || \boldsymbol{M}^{\perp})[\boldsymbol{A}']_2 = [\boldsymbol{K}' \bar{\boldsymbol{A}}' + \boldsymbol{M}^{\perp} \underline{\boldsymbol{A}}']_2 = [(\boldsymbol{K}' + \boldsymbol{M}^{\perp} \underline{\boldsymbol{A}}' \bar{\boldsymbol{A}}^{-1}) \bar{\boldsymbol{A}}]_2 = [\boldsymbol{K} \bar{\boldsymbol{A}}]_2$ and $[\boldsymbol{P}]_1 = [\boldsymbol{M}^{\top} \boldsymbol{K}']_1 = [\boldsymbol{M}^{\top} (\boldsymbol{K} - \boldsymbol{M}^{\perp} \underline{\boldsymbol{A}}' \bar{\boldsymbol{A}}^{-1})]_1 = [\boldsymbol{M}^{\top} \boldsymbol{K}]_1$. Thus, pk' has the same distribution as the real public key.

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

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. $\boldsymbol{y}^\top \boldsymbol{C} = \boldsymbol{\pi}^\top \bar{\boldsymbol{A}}$ (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 $\boldsymbol{R} \leftarrow_{\$} \mathbb{Z}_p^{(n-m-1)\times(k+1)}$.

(Item iv: Sub-PAR knowledge-soundness): Our proof strategy is inspired by that of [8, App. F]. However, their proof is given for honestly generated language parameter $\varrho = [\boldsymbol{M}]_1$ and $\boldsymbol{M}$ is obtained by using witness-sampleability; we modify the proof by extracting $\boldsymbol{M}$ from ϱ by using a DL oracle. Thus, we need to use two different types of non-falsifiable assumptions: (1) the non-adaptive SDL^{dl} assumption to extract $\boldsymbol{M}$ from $[\boldsymbol{M}]_1$, and (2) knowledge (HAK) assumptions to extract $\boldsymbol{y}$ and $\boldsymbol{\pi}$ from $[\boldsymbol{y}]_1$ and $[\boldsymbol{\pi}]_1$; we use the fact that the verification equation holds to be able to apply HAK. Moreover, we modify the proof of [8] to work for an arbitrary k.

We construct the following SDL^{dl} adversary $\mathcal{B}$, that is given access to a non-adaptive DL oracle in the query phase and then, after that, a challenge $([x]_1, [x]_2)$, returns x. First, $\mathcal{B}$ samples r and calls $\mathcal{A}(\mathsf{p}; r)$, obtaining $[\boldsymbol{M}]_1$. $\mathcal{B}$ uses the non-adaptive DL oracle nm times, extracting the matrix $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$.

In the challenge phase, $\mathcal{B}$ obtains $([x]_1, [x]_2)$ from the challenger. After that, $\mathcal{B}$ samples random $\boldsymbol{K}_1, \boldsymbol{K}_2 \in \mathbb{Z}_p^{n\times k}$ and sets $[\boldsymbol{K}]_\iota \leftarrow [x]_\iota \boldsymbol{K}_1 + [1]_\iota \boldsymbol{K}_2$. $\mathcal{B}$ honestly generates $\mathsf{pk} = ([\boldsymbol{P}]_1, [\bar{\boldsymbol{A}}, \boldsymbol{C}]_2)$ by setting $\boldsymbol{A} \leftarrow_{\$} \mathcal{D}_k$, $[\boldsymbol{C}]_2 \leftarrow [\boldsymbol{K}]_2 \bar{\boldsymbol{A}} = \boldsymbol{K}_1 \bar{\boldsymbol{A}} [x]_2 + \boldsymbol{K}_2 \bar{\boldsymbol{A}} [1]_2 \in \mathbb{G}_2^{n\times k}$, and $[\boldsymbol{P}]_1 \leftarrow \boldsymbol{M}^\top [\boldsymbol{K}]_1 = \boldsymbol{M}^\top \boldsymbol{K}_1 [x]_1 + \boldsymbol{M}^\top \boldsymbol{K}_2 [1]_1 \in \mathbb{G}_1^{m\times k}$. Denote $\boldsymbol{P}' = \text{vect}(\boldsymbol{P}) \in \mathbb{Z}_p^{mk}$. $\mathcal{B}$ sends pk to $\mathcal{A}$ who returns $[\boldsymbol{y}, \boldsymbol{\pi}]_1$.

According to the $\mathsf{KGen}([\boldsymbol{M}]_1)$-HAK assumption for arbitrary efficiently computable $[\boldsymbol{M}]_1$, given $\mathcal{A}$ who on input $(\mathsf{p}, \mathsf{pk})$, where $\mathsf{pk} \sim \mathsf{KGen}([\boldsymbol{M}]_1)$, outputs $[\boldsymbol{y}]_1 \in \mathbb{G}_1^n$ and $[\boldsymbol{\pi}]_1 \in \mathbb{G}_1^k$, we can extract $[\boldsymbol{q}]_1 \in \mathbb{G}_1^{n_q}$, $(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3)$ and $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3)$, such that

$$\begin{aligned}
[\boldsymbol{y}]_1 =& \boldsymbol{y}_1 [1]_1 + \boldsymbol{y}_2 [\boldsymbol{P}']_1 + \boldsymbol{y}_3 [\boldsymbol{q}]_1 \ , \\
[\boldsymbol{\pi}]_1 =& \boldsymbol{\pi}_1 [1]_1 + \boldsymbol{\pi}_2 [\boldsymbol{P}']_1 + \boldsymbol{\pi}_3 [\boldsymbol{q}]_1 \ ,
\end{aligned} \tag{2}$$

Note that $\boldsymbol{y}_2 \in \mathbb{Z}_p^{n\times mk}$, $\boldsymbol{\pi}_2 \in \mathbb{Z}_p^{k\times mk}$, $\boldsymbol{y_3} \in \mathbb{Z}_p^{n\times n_q}$, and $\boldsymbol{\pi_3} \in \mathbb{Z}_p^{k\times n_q}$.

We will now write $\boldsymbol{K}' = \text{vect}(\boldsymbol{K})$, $\boldsymbol{K}_1' = \text{vect}(\boldsymbol{K}_1)$, $\boldsymbol{K}_2' = \text{vect}(\boldsymbol{K}_2)$, $\boldsymbol{P}_1 = \boldsymbol{M}^\top \boldsymbol{K}_1$, $\boldsymbol{P}_2 = \boldsymbol{M}^\top \boldsymbol{K}_2$, $\boldsymbol{P}_1' = \text{vect}(\boldsymbol{P}_1)$ and $\boldsymbol{P}_2' = \text{vect}(\boldsymbol{P}_2)$. Thus, $\boldsymbol{P} = \boldsymbol{M}^\top \boldsymbol{K} = \boldsymbol{M}^\top (x\boldsymbol{K}_1 + \boldsymbol{K}_2) = x\boldsymbol{P}_1 + \boldsymbol{P}_2$ and $\boldsymbol{P}' = x\boldsymbol{P}_1' + \boldsymbol{P}_2'$. Recall $\boldsymbol{M} \in \mathbb{Z}_p^{n\times m}$, $\boldsymbol{K} \in \mathbb{Z}_p^{n\times k}$, and $\boldsymbol{P} \in \mathbb{Z}_p^{m\times k}$.

From the verification equation $[\boldsymbol{y}]_1^\top [\boldsymbol{C}]_2 = [\boldsymbol{\pi}]_1^\top [\bar{\boldsymbol{A}}]_2$. Assuming $\bar{\boldsymbol{A}}$ is invertible, $[\boldsymbol{\pi}]_1 = [\boldsymbol{K}^\top \boldsymbol{y}]_1$. From this and Eq. (2), $\boldsymbol{\pi}_1 [1]_1 + \boldsymbol{\pi}_2 [\boldsymbol{P}']_1 + \boldsymbol{\pi}_3 [\boldsymbol{q}]_1 = [\boldsymbol{K}]_1^\top \boldsymbol{y}_1 + [\boldsymbol{K}^\top \boldsymbol{y}_2 \boldsymbol{P}']_1 + [\boldsymbol{K}^\top \boldsymbol{y}_3 \boldsymbol{q}]_1$, and thus

$$\begin{aligned}
\boldsymbol{\pi}_1 [1]_1 +& \boldsymbol{\pi}_2 [x\boldsymbol{P}_1' + \boldsymbol{P}_2']_1 + \boldsymbol{\pi}_3 [\boldsymbol{q}]_1 \\
&= [x\boldsymbol{K}_1 + \boldsymbol{K}_2]_1^\top \boldsymbol{y}_1 + [(x\boldsymbol{K}_1 + \boldsymbol{K}_2)^\top \boldsymbol{y}_2 (x\boldsymbol{P}_1' + \boldsymbol{P}_2')]_1 + [(x\boldsymbol{K}_1 + \boldsymbol{K}_2)^\top \boldsymbol{y}_3 \boldsymbol{q}]_1 \ .
\end{aligned}$$

Collecting the powers of X, we get that the verification equation states that $V(x, \boldsymbol{q}) = \boldsymbol{0}_k$, where $V(X, \boldsymbol{Q}) := \boldsymbol{a}X^2 + \boldsymbol{b}(\boldsymbol{Q})X + \boldsymbol{c}(\boldsymbol{Q})$ for

$$\begin{aligned}
\boldsymbol{a} =& \boldsymbol{K}_1^\top \boldsymbol{y}_2 \boldsymbol{P}'_1 \ , \\
\boldsymbol{b}(\boldsymbol{Q}) =& \boldsymbol{K}_1^\top \left(\boldsymbol{y}_1 + \boldsymbol{y}_2 \boldsymbol{P}'_2\right) + \left(\boldsymbol{K}_2^\top \boldsymbol{y}_2 - \boldsymbol{\pi}_2\right) \boldsymbol{P}'_1 + \boldsymbol{K}_1^\top \boldsymbol{y_3} \boldsymbol{Q} \ , \\
\boldsymbol{c}(\boldsymbol{Q}) =& \boldsymbol{K}_2^\top (\boldsymbol{y}_1 + \boldsymbol{y}_2 \boldsymbol{P}'_2) - \left(\boldsymbol{\pi}_1 + \boldsymbol{\pi}_2 \boldsymbol{P}'_2\right) + (\boldsymbol{K}_2^\top \boldsymbol{y}_3 - \boldsymbol{\pi}_3) \boldsymbol{Q} \ .
\end{aligned}$$

Since each q_i has min-entropy $\Omega(\log \lambda)$ from the adversary's viewpoint and $V(X, \boldsymbol{Q})$ is a linear polynomial in each Q_i, from $V(x, \boldsymbol{q}) = \boldsymbol{0}_k$ it follows (by the Schwartz-Zippel lemma) with an overwhelming probability $1 - \varepsilon_q$ that $V(x, \boldsymbol{Q}) = 0$ as a polynomial and thus also $V(x, \boldsymbol{0}) = \boldsymbol{a}X^2 + \boldsymbol{b}(\boldsymbol{0})X + \boldsymbol{c}(\boldsymbol{0}) = 0$, where $\boldsymbol{b} := \boldsymbol{b}(\boldsymbol{0})$ and $\boldsymbol{b} := \boldsymbol{b}(\boldsymbol{0})$. In particular, in what follows, we can assume $\boldsymbol{y}_3 = \boldsymbol{0}$ and $\boldsymbol{\pi}_3 = \boldsymbol{0}$.

Next, let $\boldsymbol{w}$ be any solution to $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{w}$; a solution exists and can be efficiently found since $\boldsymbol{M}$ has rank n. We already extracted $\boldsymbol{M}$ by using the DL oracle, while $\boldsymbol{y} = \boldsymbol{y}_1 + x\boldsymbol{d} + \boldsymbol{y}_2 \boldsymbol{P}'_2$, where $\boldsymbol{d} := \boldsymbol{y}_2 \boldsymbol{P}'_1 \in \mathbb{Z}_p^n$, can be extracted if $\boldsymbol{d} = \boldsymbol{0}_n$. Thus, if $\boldsymbol{d} = \boldsymbol{0}_n$ then we can extract and return $\boldsymbol{w}$.

To show that, w.h.p., $\boldsymbol{d} = \boldsymbol{0}_n$, consider the opposite case $\boldsymbol{d} \neq \boldsymbol{0}_n$. If $\boldsymbol{a} \neq \boldsymbol{0}_k$ (this can only happen if $\boldsymbol{d} \neq \boldsymbol{0}_n$) then we have a quadratic equation $\boldsymbol{a}[x^2]_1 + \boldsymbol{b}[x]_1 + \boldsymbol{c}[1]_1 = 0$, with $\boldsymbol{a} \neq 0$, that $\mathcal{B}$ can solve for x, and thus return x.

Assume $\boldsymbol{a} = \boldsymbol{0}_k$ but $\boldsymbol{d} \neq \boldsymbol{0}_n$. This means $\boldsymbol{d} \in \mathbb{Z}_p^n$ is a non-zero element in the kernel of $\boldsymbol{K}_1^\top \in \mathbb{Z}_p^{k \times n}$. Since for $\mathcal{A}$, $\boldsymbol{K}_1$ looks uniformly random from $\mathbb{Z}_p^{k \times n}$, the question is now what is the maximum probability that for any $\boldsymbol{d} \neq \boldsymbol{0}_k$ picked by $\mathcal{A}$, $\boldsymbol{K}_1^\top \boldsymbol{d} = \boldsymbol{0}$. Obviously, unless $\boldsymbol{d} = \boldsymbol{0}_k$, this probability is equal to $\Pr[\boldsymbol{K}_1 \leftarrow_{\$} \mathbb{Z}_p^{k \times n} : \boldsymbol{K}_1^\top \boldsymbol{d} = \boldsymbol{0}_k] = p^{-k}$.

Hence, the probability of success $\varepsilon_{\mathcal{B}}$ of $\mathcal{B}$ is at least $\varepsilon_{\mathsf{w}} - \varepsilon_q - p^{-k}$, where ε_{w} is the probability of extracting w. □

If the language parameter has been honestly generated, then one does not need the DL oracle to extract $\boldsymbol{M}$. Instead, as in [31], one relies on the witness-sampleability of $\mathcal{D}_{\mathsf{p}}$ to extract $\boldsymbol{M}$ and then finish the proof of Sub-PAR (knowledge-)soundness. Importantly, in the subverted case, we do not have to assume witness-sampleability.

We note SKerMDH is not secure when $k = 1$, [23].

Acknowledgments. We would like to thank Dario Fiore and anonymous reviewers for useful comments. Abdolmaleki, Lipmaa, and Siim were partially supported by the Estonian Research Council grant PRG49.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Disjunctions for hash proof systems: new constructions and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 69–100. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_3

2. Abdolmaleki, B., Baghery, K., Lipmaa, H., Zając, M.: A subversion-resistant SNARK. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 3–33. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_1
3. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: 45th FOCS, pp. 186–195 (2004)
4. Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: security in the face of parameter subversion. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 777–804. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_26
5. Bichsel, P., Camenisch, J., Neven, G., Smart, N.P., Warinschi, B.: Get shorty via group signatures without encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 381–398. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15317-4_24
6. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC, pp. 103–112 (2019)
7. Brown, D.R.L.: The exact security of ECDSA. Contributions to IEEE P1363a (2001). http://grouper.ieee.org/groups/1363/
8. Campanelli, M., Fiore, D., Querol, A.: LegoSNARK: modular design and composition of succinct zero-knowledge proofs. In: ACM CCS 2019, pp. 2075–2092 (2019)
9. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: 32nd ACM STOC, pp. 235–244 (2000)
10. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_36
11. Danezis, G., Fournet, C., Groth, J., Kohlweiss, M.: Square span programs with applications to succinct NIZK arguments. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 532–550. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_28
12. Daza, V., González, A., Pindado, Z., Ràfols, C., Silva, J.: Shorter quadratic QA-NIZK proofs. In: Lin, D., Sako, K. (eds.) PKC 2019, Part I. LNCS, vol. 11442, pp. 314–343. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17253-4_11
13. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An Algebraic Framework for Diffie-Hellman Assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_8
14. Fauzi, P., Lipmaa, H., Siim, J., Zając, M.: An efficient pairing-based shuffle argument. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 97–127. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_4
15. Fuchsbauer, G.: Subversion-zero-knowledge SNARKs. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part I. LNCS, vol. 10769, pp. 315–347. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76578-5_11
16. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_2
17. Fuchsbauer, G., Orrù, M.: Non-interactive zaps of knowledge. In: Preneel, B., Vercauteren, F. (eds.) ACNS 2018. LNCS, vol. 10892, pp. 44–62. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93387-0_3

18. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_37
19. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: 43rd ACM STOC, pp. 99–108 (2011)
20. Gjøsteen, K.: A new security proof for Damgård's ElGamal. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 150–158. Springer, Heidelberg (2006). https://doi.org/10.1007/11605805_10
21. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. J. Cryptol. **7**(1), 1–32 (1994)
22. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: 17th ACM STOC, pp. 291–304 (1985)
23. González, A., Hevia, A., Ràfols, C.: QA-NIZK arguments in asymmetric groups: new tools and new constructions. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 605–629. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_25
24. González, A., Ráfols, C.: New techniques for non-interactive shuffle and range arguments. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 427–444. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_23
25. González, A., Ràfols, C.: Sublinear pairing-based arguments with updatable CRS and weaker assumptions. Technical report 2019/326, IACR (2019) https://eprint.iacr.org/2019/326. Last Accessed 29 Mar 2019
26. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_19
27. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11
28. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_1
29. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. Technical report 2013/109, International Association for Cryptologic Research (2013). http://eprint.iacr.org/2013/109. Accessed 14 Sept 2018
30. Jutla, C.S., Roy, A.: Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 295–312. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_17
31. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_4
32. Libert, B., Peters, T., Joye, M., Yung, M.: Non-malleability from malleability: simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 514–532. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_29
33. Libert, B., Peters, T., Joye, M., Yung, M.: Compactly hiding linear spans. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 681–707. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_28

34. Lipmaa, H.: On the CCA1-security of Elgamal and Damgård's Elgamal. In: Lai, X., Yung, M., Lin, D. (eds.) Inscrypt 2010. LNCS, vol. 6584, pp. 18–35. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21518-6_2
35. Lipmaa, H.: Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 169–189. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_10
36. Lipmaa, H.: Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 41–60. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_3
37. Lipmaa, H.: Simulation-extractable ZK-SNARKs revisited. Technical report 2019/612, IACR (2019). https://eprint.iacr.org/2019/612. Accessed 8 Feb 2020
38. Micali, S., Reyzin, L.: Soundness in the public-key model. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 542–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_32
39. Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 729–758. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_27
40. Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: nearly practical verifiable computation. In: 2013 IEEE Symposium on Security and Privacy, pp. 238–252 (2013)
41. Stern, J., Pointcheval, D., Malone-Lee, J., Smart, N.P.: Flaws in applying proof methodologies to signature schemes. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 93–110. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_7
42. Wee, H.: Lower bounds for non-interactive zero-knowledge. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 103–117. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_6