Switching Lemma for Bilinear Tests and Constant-size NIZK Proofs for Linear Subspaces

Charanjit S. Jutla
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA

Arnab Roy
Fujitsu Laboratories of America
Sunnyvale, CA 94085, USA

Abstract

We state a switching lemma for tests on adversarial inputs involving bilinear pairings in hard groups, where the tester can effectively switch the randomness used in the test from being given to the adversary at the outset to being chosen after the adversary commits its input. The switching lemma can be based on any k-linear hardness assumptions on one of the groups. In particular, this enables convenient information theoretic arguments in the construction of sequence of games proving security of cryptographic schemes, mimicking proofs and constructions in the random oracle model.

As an immediate application, we show that the quasi-adaptive NIZK proofs of Jutla and Roy [JR13] for linear subspaces can be further shortened to *constant*-size proofs, independent of the number of witnesses and equations. In particular, under the XDH assumption, a length n vector of group elements can be proven to belong to a subspace of rank t with a quasi-adaptive NIZK proof consisting of just a single group element. Similar quasi-adaptive aggregation of proofs is also shown for Groth-Sahai NIZK proofs of linear multi-scalar multiplication equations, as well as linear pairing-product equations (equations without any quadratic terms).

Keywords: NIZK, bilinear pairings, quasi-adaptive, Groth-Sahai, Random Oracle, IBE, CCA2.

1 Introduction

Testing pairing equations in bilinear groups is a fundamental component of numerous cryptographic schemes spanning public key encryption schemes, signatures, zero knowledge proofs and so on. We state and prove a *switching lemma* for testing pairing equations in bilinear groups, where an adversary is given some random group elements from one of the groups, and the pairing test (of equality and/or inequality) is performed on adversary's output and the same random group elements. We show that the tester can replace the random group elements in the test with a new set of fresh random group elements, effectively mimicking the behavior of a random oracle. This switching lemma can be based on any k-linear hardness assumptions on one of the groups. This not only enables convenient information theoretic arguments in the construction of sequence of games proving security of cryptographic schemes, but also allows more efficient protocols reminiscent of the Fiat-Shamir paradigm using random oracles [FS86].

Fiat-Shamir paradigm is best illustrated by the conversion of 3-round sigma protocol [Dam] for proof of knowledge (PoK) of discrete logarithms to a random oracle based NIZK. Consider an example where the prover is trying to prove possession of the discrete logarithm x of a public value

g^x. In the first round the prover commits to a random value r by sending g^r. In response, the verifier generates a fresh random value c and sends to the prover. The prover then responds with $r + cx$. This constitutes an honest verifier zero-knowledge PoK. In transforming this to a NIZK, a public random oracle H is used and the prover just transmits $(g^r, r + H(g^r, g^x) \cdot x)$. Essentially the random oracle induces the effect of a 'fresh' randomness that can be used for verification and is not under any effective control of the prover. In this paper we create an analogous effect in the standard model using the computational hardness of k-linear problems (such as DDH and DLIN) in bilinear groups. We show that even if the testing parameters are given to the prover, while verifying one can switch to freshly generated testing parameters with negligible change in probability of success of the verification.

As an immediate application, we show that the quasi-adaptive NIZK proofs of Jutla and Roy [JR13] for linear subspaces can be further shortened to *constant*-size proofs, independent of the number of variables and equations. In [JR13], it was shown that for languages that are linear subspaces of vector spaces of the bilinear groups, one can obtain more efficient computationally-sound NIZK proofs compared to [GS08] in a slightly different *quasi-adaptive* setting, which suffices for many cryptographic applications. In the quasi-adaptive setting, a class of parametrized languages $\{L_\rho\}$ is considered, parametrized by ρ, and CRS generator is allowed to generate the CRS based on the language parameter ρ. However, the CRS simulator in the zero-knowledge setting is required to be a single efficient algorithm that works for the whole parametrized class or probability distributions of languages, by taking the parameter as input. This property was referred to as *uniform simulation*.

The main idea underlying construction in [JR13] can be summarized as follows. Consider the language L (over a cyclic group $\mathbb{G}$ of order q, in additive notation) defined as

$$L = \left\{ \langle \boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3 \rangle \in \mathbb{G}^3 \mid \exists x_1, x_2 \in \mathbb{Z}_q : \ \boldsymbol{l}_1 = x_1 \cdot \mathbf{g},\ \boldsymbol{l}_2 = x_2 \cdot \mathbf{f},\ \boldsymbol{l}_3 = (x_1 + x_2) \cdot \mathbf{h} \right\}$$

where $\mathbf{g}$, $\mathbf{f}$, $\mathbf{h}$ are parameters defining the language. Suppose the CRS can be set to be a basis for the null-space $L_\rho^\perp$ of the language L_ρ. Then, just pairing a potential language candidate with $L_\rho^\perp$ and testing for all-zero suffices to prove that the candidate is in L_ρ, as the null-space of $L_\rho^\perp$ is just L_ρ. However, efficiently computing null-spaces in hard bilinear groups is itself hard. Thus, an efficient CRS simulator cannot generate $L_\rho^\perp$, but can give a (hiding) commitment that is computationally indistinguishable from a binding commitment to $L_\rho^\perp$. To achieve this the authors use a homomorphic commitment just as in the Groth-Sahai system, but use the simpler El-Gamal encryption style commitment as opposed to the more involved Groth-Sahai commitments, and this allows for a more efficient verifier.

Our contributions. For n equations in t variables, our quasi-adaptive computationally-sound NIZK proofs for linear subspaces require only k group elements, under the k-linear decisional assumption [HK07, Sha07]. Thus, under the XDH [1] assumption for bilinear groups, our proofs require only *one* group element. In contrast, the Groth-Sahai system requires $(n+2t)$ group elements and the Jutla-Roy system requires $(n - t)$ group elements. Similarly, under the decisional linear assumption (DLIN), our proofs require only 2 group elements, whereas the Groth-Sahai system requires $(2n + 3t)$ group elements and the Jutla-Roy system requires $(2n - 2t)$ group elements.

[1] XDH is the assumption that DDH is hard in one of the pairing groups. Also note that DDH is same as the k-linear assumption for $k = 1$. See Appendix A.

	XDH			DLIN		
	Proof	CRS	#Pairings	Proof	CRS	#Pairings
Groth-Sahai	$n+2t$	4	$2n(t+2)$	$2n+3t$	9	$3n(t+3)$
Jutla-Roy	$n-t$	$2t(n-t)+2$	$(n-t)(t+2)$	$2n-2t$	$4t(n-t)+3$	$2(n-t)(t+2)$
Schnorr-NIZKs (RO)	$t+2$	$-$	$-$	$-$	$-$	$-$
This paper	1	$n+t+1$	$n+1$	2	$2(n+t+2)$	$2(n+2)$

Table 1: Comparison with Groth-Sahai, Jutla-Roy and Schnorr-NIZKs for Linear Subspaces. Parameter t is the number of unknowns or witnesses and n is the dimension of the vector space, or in other words, the number of equations.

	DLIN Linear Multi-Scalar and Linear Pairing-Product		
	Proof	CRS	#Pairings
Groth-Sahai	$3(s+t)+9n$	9	$9n(s+t+3)+n$
This paper	$3(s+t)+18$	$9+4n$	$18(s+t+3)+n$

Table 2: Comparison with (1) Groth-Sahai for n number of linear Scalar Multiplication Equations: $\vec{y}\cdot\vec{\mathbf{a}}_j + \vec{b}_j\cdot\vec{\mathbf{x}} = \mathbf{u}_j$, with $j \in [1,n]$, $\vec{y} \in \mathbb{Z}_q{}^s, \vec{\mathbf{x}} \in \mathbb{G}^t$ and $\mathbf{u}_j \in \mathbb{G}$. and (2) Groth-Sahai for n number of linear Pairing Product Equations: $e(\vec{\mathbf{y}}, \vec{\mathbf{a}}_j) \;+\; e(\vec{\mathbf{b}}_j, \vec{\mathbf{x}}) = \mathbf{u}_j$, with $j \in [1,n]$, $\vec{\mathbf{y}} \in \mathbb{G}^s$, $\vec{\mathbf{x}} \in \mathbb{G}^t$ and $\mathbf{u}_j \in \mathbb{G}_T$.

These parameters are summarized in Table 1. While our CRS size grows linearly with n, the number of pairing operations is competitive and could be significantly less compared to earlier schemes for appropriate n and t.

Note that Schnorr proofs of multiple equations in the random oracle can also be combined into a proof consisting of only two group elements (by taking random linear combinations employing the random oracle), but it still requires commitments to all the variables. The commitments lie in $\mathbb{Z}_q$, where q is the group order. Thus, our proofs are even shorter than Schnorr proofs. On the other hand, Schnorr proofs are proof of knowledge (as opposed to ours or Groth-Sahai), and can be slightly faster to verify as they only use exponentiation instead of pairings.

We also show that proofs of multiple linear scalar-multiplication equations, as well as multiple *linear* pairing product equations (i.e. without any bilinear terms) can be aggregated into a single proof in the Groth-Sahai system. This can lead to significant shortening of proofs of multiple linear pairing product equations. The comparisons are tabulated in Table 2. We remark that this is in contrast to the batching of Groth-Sahai proof verification [BFI+10], where the proofs were not aggregated, but multiple pairing equations were batched together during the verification step. We can use similar batching techniques to improve the verification step; therefore, we skip taking these optimizations into consideration.

While the cryptographic literature is replete with applications using NIZK proofs of algebraic languages over bilinear groups, and many examples were given in [JR13] involving NIZK proofs of linear subspaces, we focus on two particular cases where aggregation of proofs of linear subspaces lead to interesting results. We consider a construction of [CCS09] to convert key-dependent message (KDM) CPA secure encryption scheme [BHHO08] into a KDM-CCA2 secure scheme which involved proving $O(N)$ linear equations, where N is the security parameter. With our aggregation of proofs, the size of this proof (in the quasi-adaptive setting) is reduced to just 2 group elements (under the

DLIN assumption) from the earlier $O(N)$ sized quasi-adaptive proofs and Groth-Sahai proofs. It is also easy to see that the quasi-adaptive setting for proving the NIZK suffices, as is the case for most applications. As another application we reduce the size of the publicly-verifiable CCA2-IBE scheme obtained in [JR13] by another group element to just five group elements plus a tag. This makes it shorter than the CCA2-IBE scheme obtained using the [CHK04] paradigm from hierarchical-IBE (HIBE) and in addition is publicly-verifiable.

Organization of the paper. We begin the rest of the paper with the switching lemma for bilinear tests in hard groups in Section 2. We recall the quasi-adaptive NIZK definitions in Section 3 and develop constant-size quasi-adaptive NIZKs for linear subspaces in Section 4. In Section 5, we apply our switching lemma to aggregate Groth-Sahai NIZKs. Finally, we provide application examples in Section 6. The hardness assumptions we use are standard and are summarized in Appendix A.

2 Switching Lemma for Bilinear Tests in Hard Groups

Notations. Consider bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with pairing e to a target group $\mathbb{G}_T$, all of prime order q, and random generators $\mathbf{g}_1 \in \mathbb{G}_1$ and $\mathbf{g}_2 \in \mathbb{G}_2$. Let $\mathbf{0}_1$, $\mathbf{0}_2$ and $\mathbf{0}_T$ be the identity elements in the three groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ respectively. The pairing operation naturally extends to vectors of elements (by summation) and correspondingly to matrices of elements. In this section and the next two, vectors will always be row-vectors and will always be denoted by an arrow over the letter, e.g. $\vec{\mathrm{r}}$ for (row) vector of $\mathbb{Z}_q$ elements, and $\vec{\mathbf{d}}$ as (row) vector of group elements. We will use additive notation for group operations, with $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ viewed as $\mathbb{Z}_q$ vector spaces and the scalar product operation also naturally extends to vectors and matrices of group and $\mathbb{Z}_q$ elements.

Intuition. We first explain the intuition behind the Switching Lemma by way of a toy example. Suppose we are given three elements $\mathbf{g}, \mathbf{f}(= a \cdot \mathbf{g}), \mathbf{h}(= b \cdot \mathbf{g})$ in the group $\mathbb{G}_1$ and we need a proof system, not necessarily ZK, for tuples of the form $(x \cdot \mathbf{g}, x \cdot \mathbf{f}, x \cdot \mathbf{h})$. Towards that end, suppose the following CRS is published: $((ar_1 + br_2) \cdot \mathbf{g}_2, -r_1 \cdot \mathbf{g}_2, -r_2 \cdot \mathbf{g}_2)$. So the pairing test $e(x \cdot \mathbf{g}, (ar_1 + br_2) \cdot \mathbf{g}_2) + e(x \cdot \mathbf{f}, -r_1 \cdot \mathbf{g}_2) + e(x \cdot \mathbf{h}, -r_2 \cdot \mathbf{g}_2) = \mathbf{0}_T$, satisfies completeness, i.e., it holds for valid tuples.

However, how do we know that it's sound? A look at the pairing equation shows that there is a fair degree of freedom to satisfy it, without being a valid tuple. So we definitely have to resort to a computational assumption to argue soundness. This is where the switching lemma comes in. It says that, even though we publish the CRS using r_1, r_2, while verifying we can switch them with fresh r_1', r_2' chosen randomly and independently from $\mathbb{Z}_q$.

This means if a candidate tuple $(\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3)$ satisfies the original test with a certain probability, then it also satisfies the switched test: $e(\boldsymbol{l}_1, (ar_1' + br_2') \cdot \mathbf{g}_2) + e(\boldsymbol{l}_2, -r_1' \cdot \mathbf{g}_2) + e(\boldsymbol{l}_3, -r_2' \cdot \mathbf{g}_2) = \mathbf{0}_T$ with almost the same probability. Rearranging, we get: $r_1' \cdot e(a \cdot \boldsymbol{l}_1 - \boldsymbol{l}_2, \mathbf{g}_2) + r_2' \cdot e(b \cdot \boldsymbol{l}_1 - \boldsymbol{l}_3, \mathbf{g}_2) = \mathbf{0}_T$. Now, observe that the r_1', r_2' were chosen *after* the tuple was given. So with high probability, both of $e(a \cdot \boldsymbol{l}_1 - \boldsymbol{l}_2, \mathbf{g}_2)$ and $e(b \cdot \boldsymbol{l}_1 - \boldsymbol{l}_3, \mathbf{g}_2)$ must be $\mathbf{0}_T$. Therefore, $\boldsymbol{l}_2 = a \cdot \boldsymbol{l}_1$ and $\boldsymbol{l}_3 = b \cdot \boldsymbol{l}_1$, thus proving soundness.

Another way to look at this is that we produced a single CRS by random linear combination of CRS'es to prove the individual languages $\{(x \cdot \mathbf{g}, x \cdot \mathbf{f}) \mid x \in \mathbb{Z}_q\}$ and $\{(x \cdot \mathbf{g}, x \cdot \mathbf{h}) \mid x \in \mathbb{Z}_q\}$. Since the combined CRS is given to the adversary, we cannot resort to information theoretic arguments in order to separate the individual equations. However with the switching lemma in play, once we switch to fresh independently random linear combination for verification, it becomes straightforward to apply information theoretic arguments.

We first state and prove a simple switching lemma, as it's proof is easier to understand.

Lemma 1 (Simple Switching Lemma) *Let $\mathcal{D}$ be an arbitrary efficiently samplable distribution over $n \times (k+1)$ matrices from $\mathbb{Z}_q$. For any PPT adversary $\mathcal{A}$ that produces a length n vector of $\mathbb{G}_1$ elements, let $\Delta_{\mathcal{A}}$ be the following probability*

$$\Pr\left[\begin{array}{c} \vec{\mathbf{r}} \stackrel{\$}{\leftarrow} \mathbb{G}_2^k,\ \mathsf{R} \stackrel{\$}{\leftarrow} \mathbb{G}_2^{k\times k},\ C^{n\times(k+1)} \leftarrow \mathcal{D},\ \vec{\mathbf{f}}^{1\times n} \leftarrow \mathcal{A}(\mathbf{g}_1, \mathbf{g}_2, \vec{\mathbf{r}}, \mathsf{R}, C): \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times n} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, C \cdot \left[\begin{array}{c} \mathsf{R} \\ \vec{\mathbf{r}} \end{array}\right]\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right]$$

Then, under the k-linear assumption for group $\mathbb{G}_2$, the following probability is negligibly close to $\Delta_{\mathcal{A}}$:

$$\Pr\left[\begin{array}{c} \vec{\mathbf{r}} \stackrel{\$}{\leftarrow} \mathbb{G}_2^k,\ \mathsf{R} \stackrel{\$}{\leftarrow} \mathbb{G}_2^{k\times k},\ C^{n\times(k+1)} \leftarrow \mathcal{D},\ \vec{\mathbf{f}}^{1\times n} \leftarrow \mathcal{A}(\mathbf{g}_1, \mathbf{g}_2, \vec{\mathbf{r}}, \mathsf{R}, C),\ \vec{\mathbf{r}}' \stackrel{\$}{\leftarrow} \mathbb{G}_2^k: \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times n} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, C \cdot \left[\begin{array}{c} \mathsf{R} \\ \vec{\mathbf{r}}' \end{array}\right]\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right]$$

The absolute value of the difference in the probabilities is bounded by ADV*(klin), which is the maximum probability among all PPT adversaries of winning a k-linear challenge in $\mathbb{G}_2$.*

The proof of this lemma goes by showing that if the second probability differs from the first by δ then one can construct an adversary $\mathcal{B}$ that has advantage close to δ in the k-linear challenge. This is achieved by embedding the k-linear challenge in a different bilinear test which turns out to be a random linear combination of the above k bilinear tests. Details can be found in Appendix B.

We now state the switching lemma which is actually used in all applications in this paper.

Lemma 2 (Switching Lemma) *Let $\mathcal{D}$ be an arbitrary efficiently samplable distribution over $n \times m$ matrices from $\mathbb{Z}_q$. For any PPT adversary $\mathcal{A}$ producing a vector of n elements from group $\mathbb{G}_1$, let $\Delta_{\mathcal{A}}$ be the following probability*

$$\Pr\left[\begin{array}{c} \mathsf{R} \stackrel{\$}{\leftarrow} \mathbb{G}_2^{m\times k},\ C^{n\times m} \leftarrow \mathcal{D},\ \vec{\mathbf{f}}^{1\times n} \leftarrow \mathcal{A}(\mathbf{g}_1, \mathbf{g}_2, \mathsf{R}, C): \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times n} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, C \cdot \mathsf{R}\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right]$$

Then, under the k-linear assumption for group $\mathbb{G}_2$, the following probability is negligibly close to $\Delta_{\mathcal{A}}$.

$$\Pr\left[\begin{array}{c} \mathsf{R} \stackrel{\$}{\leftarrow} \mathbb{G}_2^{m\times k},\ C^{n\times m} \leftarrow \mathcal{D},\ \vec{\mathbf{f}}^{1\times n} \leftarrow \mathcal{A}(\mathbf{g}_1, \mathbf{g}_2, \mathsf{R}, C),\ \mathsf{R}' \stackrel{\$}{\leftarrow} \mathbb{G}_2^{m\times k}: \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times n} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, C \cdot \mathsf{R}'\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right]$$

The absolute value of the difference in the probabilities is bounded by $m \cdot$ ADV$(klin)$.

The proof of this lemma is also given in Appendix B. In the case $m \leq k$, we can show that the result follows information-theoretically. When $m > k$, we show that the result follows by a hybrid argument, employing the k-linear hardness assumption inductively.

3 Quasi-Adaptive NIZK Proofs

We recall here the definitions from [JR13] and provide a summary. Instead of considering NIZK proofs for a (witness-) relation R, the authors consider Quasi-Adaptive NIZK proofs for a probability distribution $\mathcal{D}$ on a collection of (witness-) relations $\mathcal{R} = \{R_\rho\}$. The quasi-adaptiveness allows for the common reference string (CRS) to be set based on R_ρ after the latter has been chosen according to $\mathcal{D}$. However the simulator generating the CRS (in the simulation world) is required to be a single probabilistic polynomial time algorithm that works for the whole collection of relations $\mathcal{R}$.

To be more precise, they consider ensemble $\{\mathcal{D}_\lambda\}$ of distributions on collection of relations $\mathcal{R}_\lambda$, where each $\mathcal{D}_\lambda$ specifies a probability distribution on $\mathcal{R}_\lambda = \{R_{\lambda,\rho}\}$. When λ is clear from context it can be dropped. Since in the quasi-adaptive setting the CRS could depend on the relation, an associated *parameter language* $\mathcal{L}_{\text{par}}$ is considered such that a member of this language is enough to characterize a particular relation, and this language member is provided to the CRS generator.

A tuple of algorithms $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is called a *QA-NIZK* proof system for witness-relations $\mathcal{R}_\lambda = \{R_\rho\}$ with parameters sampled from a distribution $\mathcal{D}$ over associated parameter language $\mathcal{L}_{\text{par}}$, if there exists a probabilistic polynomial time simulator $(\mathsf{S}_1, \mathsf{S}_2)$, such that for all non-uniform PPT adversaries $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ we have:

Quasi-Adaptive Completeness:

$$\begin{aligned}\Pr[&\lambda \leftarrow \mathsf{K}_0(1^m); \rho \leftarrow \mathcal{D}_\lambda; \psi \leftarrow \mathsf{K}_1(\lambda, \rho); (x, w) \leftarrow \mathcal{A}_1(\lambda, \psi, \rho);\\ &\pi \leftarrow \mathsf{P}(\psi, x, w) : \;\; \mathsf{V}(\psi, x, \pi) = 1 \text{ if } R_\rho(x, w)] = 1\end{aligned}$$

Quasi-Adaptive Soundness:

$$\begin{aligned}\Pr[&\lambda \leftarrow \mathsf{K}_0(1^m); \rho \leftarrow \mathcal{D}_\lambda; \psi \leftarrow \mathsf{K}_1(\lambda, \rho);\\ &(x, \pi) \leftarrow \mathcal{A}_2(\lambda, \psi, \rho) : \;\; \mathsf{V}(\psi, x, \pi) = 1 \;\text{ and }\; \neg(\exists w : R_\rho(x, w))] \approx 0\end{aligned}$$

Quasi-Adaptive Zero-Knowledge:

$$\begin{aligned}&\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); \rho \leftarrow \mathcal{D}_\lambda; \psi \leftarrow \mathsf{K}_1(\lambda, \rho) : \mathcal{A}_3^{\mathsf{P}(\psi,\cdot,\cdot)}(\lambda, \psi, \rho) = 1] \approx\\ &\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); \rho \leftarrow \mathcal{D}_\lambda; (\psi, \tau) \leftarrow \mathsf{S}_1(\lambda, \rho) : \mathcal{A}_3^{\mathsf{S}(\psi,\tau,\cdot,\cdot)}(\lambda, \psi, \rho) = 1],\end{aligned}$$

where $\mathsf{S}(\psi, \tau, x, w) = \mathsf{S}_2(\psi, \tau, x)$ for $(x, w) \in R_\rho$ and both oracles (i.e. P and S) output failure if $(x, w) \notin R_\rho$.

Note that ψ is the CRS in the above definitions.

4 Aggregating Quasi-Adaptive Proofs of Linear Subspaces

We summarize the linear-subspace QA-NIZK setting of [JR13] here and refer the reader to that paper for details.

Linear Subspace Languages. We consider languages that are linear subspaces of vectors of $\mathbb{G}_1$ elements. In other words, the languages we are interested in can be characterized as languages parametrized by $\mathbf{A}$ as below:

$$L_{\mathbf{A}} = \{\vec{x} \cdot \mathbf{A} \in \mathbb{G}_1^n \mid \vec{x} \in \mathbb{Z}_q{}^t\}, \text{ where } \mathbf{A} \text{ is a } t \times n \text{ matrix of } \mathbb{G}_1 \text{ elements.}$$

Here $\mathbf{A}$ is an element of the associated *parameter language* $\mathcal{L}_{\text{par}}$, which is all $t \times n$ matrices of $\mathbb{G}_1$ elements. The parameter language $\mathcal{L}_{\text{par}}$ also has a corresponding witness relation $\mathcal{R}_{\text{par}}$, where the witness is a matrix of $\mathbb{Z}_q$ elements : $\mathcal{R}_{\text{par}}(\mathbf{A}, \mathsf{A})$ iff $\mathbf{A} = \mathsf{A} \cdot \mathbf{g}_1$.

Robust and Efficiently Witness-Samplable Distributions. Let the $t \times n$ dimensional matrix $\mathbf{A}$ be chosen according to a distribution $\mathcal{D}$ on $\mathcal{L}_{\text{par}}$. The distribution $\mathcal{D}$ is called *robust* if with probability close to one the left-most t columns of $\mathbf{A}$ are full-ranked. A distribution $\mathcal{D}$ on $\mathcal{L}_{\text{par}}$ is called *efficiently witness-samplable* if there is a probabilistic polynomial time algorithm such that it outputs a pair of matrices $(\mathbf{A}, \mathsf{A})$ that satisfy the relation $\mathcal{R}_{\text{par}}$ (i.e., $\mathcal{R}_{\text{par}}(\mathbf{A}, \mathsf{A})$ holds), and further the resulting distribution of the output $\mathbf{A}$ is same as $\mathcal{D}$. For example, the uniform distribution on $\mathcal{L}_{\text{par}}$ is efficiently witness-samplable, by first picking A at random, and then computing $\mathbf{A}$.

QA-NIZK Construction. We now describe a computationally sound quasi-adaptive NIZK $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ for linear subspace languages $\{L_{\mathbf{A}}\}$ with parameters sampled from a robust and efficiently witness-samplable distribution $\mathcal{D}$ over the associated parameter language $\mathcal{L}_{\text{par}}$. As a conceptual starting point, we first develop a construction which has k^2 element proofs, demonstrating a single application of the Switching Lemma. Later, we give a k element construction which linearly combines the first construction proofs and uses an additional layer of Switching Lemma application. Our description here is self sufficient and relates to the scheme in [JR13] in that we linearly combine proofs of multiple elements yielding constant-size proofs.

Algorithm K_1: The algorithm K_1 generates the CRS as follows. Let $\mathbf{A}^{t \times n}$ be the parameter supplied to K_1. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns. It generates a matrix $\mathsf{D}^{t \times k^2}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $\{b_v\}_{v \in [1,k]}$ and sk elements $\{r_{iu}\}_{i \in [1,s], u \in [1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Define matrices $\mathsf{R}^{s \times k^2}$ and $\mathsf{B}^{k^2 \times k^2}$ component-wise as follows:

$$(\mathsf{R})_{i,k(u-1)+v} = r_{iu}, \text{ with } i \in [1, s], u, v \in [1, k].$$

$$(\mathsf{B})_{ij} = \begin{cases} b_v & \text{if } i = j = k(u-1) + v, \text{ with } u, v \in [1, k] \\ 0 & \text{if } i \neq j, \text{ with } i, j \in [1, k^2] \end{cases}$$

Intuitively, the matrix R is a k times column-wise repetition of the r_{ij}'s, and if we denote $\{b_v\}_{v \in [1,k]}$ by $\vec{b}$, then the diagonal matrix B is just the vector $\vec{b}$ repeated k times along the diagonal (i.e. $\mathsf{B}_{k(u-1)+v, k(u-1)+v}$ is b_v and not b_u).

The common reference string (CRS) has two parts $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ which are to be used by the prover and the verifier respectively.

$$\mathbf{CRS}_p^{t \times k^2} := \mathbf{A} \cdot \begin{bmatrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{bmatrix} \qquad \mathbf{CRS}_v^{(n+k^2) \times k^2} = \begin{bmatrix} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{bmatrix} \cdot \mathbf{g}_2$$

Prover P**:** Given candidate $\vec{\boldsymbol{l}} = \vec{\mathrm{x}} \cdot \mathbf{A}$ with witness vector $\vec{\mathrm{x}}^{1\times t}$, the prover generates the following proof consisting of k^2 elements in $\mathbb{G}_1$: $\vec{\mathbf{p}}^{1\times k^2} := \vec{\mathrm{x}} \cdot \mathbf{CRS}_p$

Verifier V**:** Given candidate $\vec{\boldsymbol{l}}$, and proof $\vec{\mathbf{p}}$, the verifier checks the following (k^2 equations) :

$$e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k^2}$$

Theorem 3 *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound quasi-adaptive NIZK proof system for linear subspace languages* $\{L_{\mathbf{A}}\}$ *with parameters* $\mathbf{A}$ *sampled from a robust and efficiently witness-samplable distribution* $\mathcal{D}$ *over the associated parameter language* $\mathcal{L}_{par}$*, given any group generation algorithm for which the k-linear assumption holds for group* $\mathbb{G}_2$*.*

Proof Intuition. We now give a proof sketch for soundness and defer the full proof, including completeness and zero knowledge, to Appendix C.

Soundness: We prove soundness by transforming the system over a sequence of games. Consider an adversary $\mathcal{A}$ that *wins* if it can produce a "proof" $\vec{\mathbf{p}}$ for a candidate $\vec{\boldsymbol{l}}$ that is not in $L_{\mathbf{A}}$ and yet the pairing test $e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k^2}$ holds. Game $\mathbf{G}_0$ just replicates the soundness security definition. In Game $\mathbf{G}_1$ the CRS is generated using parameter witness A and its null-space, and this can be done efficiently by the challenger as the parameter distribution is efficiently witness samplable. After this transformation, we show that in the case of a certain event, a verifying proof of a non-language member implies breaking the k-linear assumption in group $\mathbb{G}_2$, while in the case of the event not occurring we can apply the Switching Lemma to bound the probability of the adversary winning.

In Game $\mathbf{G}_1$, the challenger efficiently samples $\mathbf{A}$ according to distribution $\mathcal{D}$, along with witness A (since $\mathcal{D}$ is an efficiently witness samplable distribution). Since A is a $t \times (t+s)$ dimensional rank t matrix, there is a rank s matrix $\left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right]$ of dimension $(t+s) \times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A} \cdot \left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right] = \mathsf{0}^{t\times s}$. In this game, the NIZK CRS is computed as follows: Generate matrix $\mathsf{D}'^{\,t\times k^2}$ with elements randomly chosen from $\mathbb{Z}_q$ and the matrices $\mathsf{R}^{s\times k^2}$ and $\mathsf{B}^{k^2\times k^2}$ as in the real CRS. Implicitly set: $\mathsf{D} = \mathsf{D}' + \mathsf{W}\ \mathsf{R}\ \mathsf{B}^{-1}$. Therefore we have,

$$\mathbf{CRS}_p^{t\times k^2} = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{array}\right] = \mathbf{A} \cdot \left(\left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times k^2} \end{array}\right] + \left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right] \cdot \mathsf{R}\ \mathsf{B}^{-1} \right) = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times k^2} \end{array}\right]$$

$$\mathbf{CRS}_v^{(n+k^2)\times k^2} = \left[\begin{array}{c} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{array}\right] \cdot \mathbf{g}_2 = \left[\begin{array}{c} \mathsf{D}'\ \mathsf{B} + \mathsf{W}\ \mathsf{R} \\ \mathsf{R} \\ -\mathsf{B} \end{array}\right] \cdot \mathbf{g}_2$$

Suppose that $\mathcal{A}$ wins $\mathbf{G}_1$. Now, let us partition the $\mathbb{Z}_q$ matrix A as $\left[\begin{array}{c|c} \mathsf{A}_0^{t\times t} & \mathsf{A}_1^{t\times s} \end{array}\right]$ and the candidate vector $\vec{\boldsymbol{l}}$ as $\left[\begin{array}{c|c} \vec{\boldsymbol{l}}_0^{1\times t} & \vec{\boldsymbol{l}}_1^{1\times s} \end{array}\right]$. Note that, since A_0 has rank t, the elements of $\vec{\boldsymbol{l}}_0$ are 'free' elements and $\vec{\boldsymbol{l}}_0$ can be extended to a unique n element vector $\vec{\boldsymbol{l}}\,'$, which is a member of $L_{\mathbf{A}}$. This member vector $\vec{\boldsymbol{l}}\,'$ can be computed as $\vec{\boldsymbol{l}}\,' := \left[\ \vec{\boldsymbol{l}}_0 \mid -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}\ \right]$, where $\mathsf{W} = -\mathsf{A}_0^{-1}\mathsf{A}_1$. The proof of $\vec{\boldsymbol{l}}\,'$ is computed as $\vec{\mathbf{p}}\,' := \vec{\boldsymbol{l}}_0 \cdot \mathsf{D}'$. Since $\mathcal{A}$ wins $\mathbf{G}_1$, then $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ passes the verification test, and further by design $(\vec{\boldsymbol{l}}', \vec{\mathbf{p}}')$ passes the verification test. Thus, we obtain: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \cdot \mathsf{R} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}}) \cdot \mathsf{B}$, where $\vec{\boldsymbol{l}}_1' = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$. This gives us a set of equalities, for all $u \in [1, k]$:

$$\sum_{i=1}^{s} (\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot r_{iu} = (\mathbf{p}'_{k(u-1)+1} - \mathbf{p}_{k(u-1)+1}) \cdot b_1 = \cdots = (\mathbf{p}'_{k(u-1)+k} - \mathbf{p}_{k(u-1)+k}) \cdot b_k \tag{1}$$

Note that since $\vec{\boldsymbol{l}}$ is not in the language, there exists an $i \in [1, s]$, such that $\vec{\boldsymbol{l}}_{1i}' - \vec{\boldsymbol{l}}_{1i} \neq 0$. Now consider the event E defined as follows:

$$\text{Event } E \equiv \text{ For some } u \in [1, k] : \sum_{i=1}^{s} (\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot r_{iu} \neq \mathbf{0}_1 \tag{2}$$

Our strategy now is to show that the probability of $\mathcal{A}$ winning in both the events E and $\neg E$ is negligible. Under the event $\neg E$, we apply the Switching Lemma to switch the r_{iu}'s to a fresh set of random values r'_{iu}'s while verifying. After that, we argue information theoretically that the probability of winning the switched game is negligible. Under the event E, we show that one can build a k-linear challenge adversary using $\mathcal{A}$, such that if $\mathcal{A}$ wins then this new adversary can efficiently compute the (least) u in Event E, and using the multiple equalities in Equation 1 it can break the k-linear challenge. □

We now show that the proof system described above with k^2 group elements can be further shortened to just k group elements. The main idea is to observe that Equation 1 is again several sets of equations, and we can carefully set up the system so that the prover only shows random linear combinations of Equation 1. Then resorting to Switching Lemma we conclude that the individual equations must be true. We now describe this optimized Quasi-Adaptive NIZK proof system in detail.

QA-NIZK construction with k elements. In this construction the **Algorithm** K_1 generates the CRS as follows. It generates a matrix $\mathsf{D}^{t\times k}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $\{b_v\}_{v\in[1,k]}$ and k^3 elements $\{t_{uvw}\}_{u,v,w\in[1,k]}$ and sk elements $\{r_{iu}\}_{i\in[1,s],u\in[1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Define matrices $\mathsf{R}^{s\times k}$ and $\mathsf{B}^{k\times k}$ component-wise as follows:

$$(\mathsf{R})_{iw} = \sum_{u=1}^{k}\sum_{v=1}^{k} r_{iu} t_{uvw}, \text{ with } i \in [1, s], w \in [1, k].$$

$$(\mathsf{B})_{vw} = \sum_{u=1}^{k} b_v t_{uvw}, \text{ with } v, w \in [1, k].$$

The construction of CRS_p and CRS_v remain algebraically the same, although now they use lesser elements. The prover and verifier also retain the same algebraic form. The set of equalities for this construction corresponding to the equation $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \cdot \mathsf{R} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}}) \cdot \mathsf{B}$, is for all $w \in [1, k]$:

$$\sum_{i=1}^{s} \left[(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot \left(\sum_{u=1}^{k}\sum_{v=1}^{k} r_{iu} t_{uvw} \right) \right] - \sum_{v=1}^{k} \left[(\mathbf{p}'_v - \mathbf{p}_v) \cdot \left(\sum_{u=1}^{k} b_v t_{uvw} \right) \right] = \mathbf{0}_1 \tag{3}$$

Rearranging, we get for all $w \in [1, k]$:

$$\sum_{u=1}^{k} \sum_{v=1}^{k} \left[t_{uvw} \left(\sum_{i=1}^{s} \left[(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot r_{iu} \right] - (\mathbf{p}'_v - \mathbf{p}_v) \cdot b_v \right) \right] = \mathbf{0}_1 \tag{4}$$

Now, using the Switching Lemma and after applying information theoretic arguments, we transition to a game where the adversary wins if it wins the original game and the following event occurs:

$$\text{For all } u \in [1, k] : \sum_{i=1}^{s} (\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot r_{iu} = (\mathbf{p}'_1 - \mathbf{p}_1) \cdot b_1 = \cdots = (\mathbf{p}'_k - \mathbf{p}_k) \cdot b_k \tag{5}$$

After this point, the proof is analogous to the previous QA-NIZK construction. Detailed proof is given in Appendix D. We also give a more optimized construction in Appendix E which uses less randomness and enjoys a better security reduction.

5 Aggregating Groth-Sahai Proofs

We show that proofs of multiple linear scalar-multiplication equations, as well as multiple *linear* pairing product equations can be aggregated into a single proof in the Groth-Sahai system. We will focus on describing the aggregation for the scalar-multiplication equations, as the results for the linear pairing product equations are obtained in almost an identical manner.

Consider bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with pairing e into a third group $\mathbb{G}_T$. Consider equations of the type

$$\sum_{i=1}^{n} y_i \cdot \mathbf{a}_i \; + \; \sum_{i=1}^{m} b_i \cdot \mathbf{x}_i = \mathbf{t}_1 \tag{6}$$

where the variables y_i are to take values in $\mathbb{Z}_q$, the variables $\mathbf{x}_i$ are to take values in $\mathbb{G}_1$. The constants $\mathbf{a}_i$ are in $\mathbb{G}_1$, and scalar constants y_i are in $\mathbb{Z}_q$. Moreover, $\mathbf{t}_1$ is in $\mathbb{G}_1$.

When the bilinear group is symmetric, i.e. $\mathbb{G}_1 = \mathbb{G}_2$, and under the DLIN assumption, the Groth-Sahai NIZK proof of the above equation requires commitments to the variables, each commitment being of size *three* group elements (for both y_i or $\mathbf{x}_i$). In addition it requires a proof of *nine* group elements. When there are multiple equations of the above kind in the same variables, the commitments to the variables remain the same, but each equation requires nine group elements. In other words, if there are $m + n$ variables and k equations, the full proof of the k equations has size $3 \cdot (m + n) + 9k$ group elements.

We will now show that in the quasi-adaptive setting, the full proof of the k equations can be obtained with size $3 \cdot (m + n) + 9$ group elements. We first describe how the proof is done in the Groth-Sahai system, and then we will point out the relevant changes. The proofs and commitments actually belong to the $\mathbb{Z}_q$-module $\mathbb{G}^3$ (where $\mathbb{G} = \mathbb{G}_1 = \mathbb{G}_2$).

We will write these groups in additive notation, and the bilinear pairing operation $e(A, B)$ written in infix notation as $A \otimes B$, with the pairing operation defining a tensor product $\mathbb{G} \otimes \mathbb{G}$ over $\mathbb{Z}_q$. Without loss of generality (see e.g. A2.2 in [Eis95]), we can assume that $\mathbb{G}_T = \mathbb{G} \otimes \mathbb{G}$. Further, this naturally extends to a tensor product $\mathbb{G}^3 \otimes \mathbb{G}^3$. One can also define a tensor product $\mathbb{Z}_q \otimes \mathbb{G}$, but since $\mathbb{G}$ is a $\mathbb{Z}_q$-module, this tensor product is just $\mathbb{G}$.

Let $\iota_1 : \mathbb{Z}_q \to \mathbb{G}^3$, $\iota_2 : \mathbb{G} \to \mathbb{G}^3$, $p_1 : \mathbb{G}^3 \to \mathbb{Z}_q$, $p_2 : \mathbb{G}^3 \to \mathbb{G}$ be group homomorphisms s.t. $\iota_1 \circ p_1$, and $\iota_2 \circ p_2$ are identity maps in $\mathbb{Z}_q$ and $\mathbb{G}$ resp. Note that the maps ι_1 and ι_2 naturally define a group homomorphism ι_T from $\mathbb{Z}_q \otimes \mathbb{G}$ $(= \mathbb{G})$ to $\mathbb{G}^3 \otimes \mathbb{G}^3$, and similarly p_1 and p_2 define a group homomorphism p_T from $\mathbb{G}^3 \otimes \mathbb{G}^3$ to $\mathbb{Z}_q \otimes \mathbb{G}$ $(= \mathbb{G})$.

The NIZK common reference string (CRS) consists of three elements from $\mathbb{G}^3$, i.e. $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3 \in \mathbb{G}^3$. They are chosen as follows: $\boldsymbol{u}_1 = (\alpha \cdot \mathbf{g}, \mathcal{O}, \mathbf{g})$, and $\boldsymbol{u}_2 = (\mathcal{O}, \beta \cdot \mathbf{g}, \mathbf{g})$, and $\boldsymbol{u}_3 = r\boldsymbol{u}_1 + s\boldsymbol{u}_2$, for random $\alpha, \beta, r, s \in \mathbb{Z}_q$, and random $\mathbf{g} \in \mathbb{G} \backslash \mathcal{O}$. This real-world CRS $\vec{\boldsymbol{u}}$ is sometimes also referred to as the *binding* CRS.

The map $\iota_2(\mathcal{Z})$ is just $(\mathcal{O}, \mathcal{O}, \mathcal{Z})$, and $p_2(\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3) = \mathcal{Z}_3 - \alpha^{-1}\mathcal{Z}_1 - \beta^{-1}\mathcal{Z}_2$, which shows that $\iota_2 \circ p_2$ is an identity map. It also shows that $p_2(\boldsymbol{u}_1) = p_2(\boldsymbol{u}_2) = p_2(\boldsymbol{u}_3) = \mathcal{O}$. Now, the **commitments** to elements $\mathcal{Z}$ in $\mathbb{G}$ are made by picking r_1, r_2, r_3 at random from $\mathbb{Z}_q$, and setting $c_2(\mathcal{Z}) = \iota_2(\mathcal{Z}) + r_1\boldsymbol{u}_1 + r_2\boldsymbol{u}_2 + r_3\boldsymbol{u}_3$. Thus, $p_2(c_2(\mathcal{Z})) = \mathcal{Z}$, and hence the name binding CRS.

The map $\iota_1(z)$ is $\iota_2(z \cdot \mathbf{g})$, and hence commitment to $z \in \mathbb{Z}_q$ is $c_1(z) = c_2(z \cdot \mathbf{g})$.

For equations of the form (6)[2], i.e. $\vec{y} \cdot \vec{\mathbf{a}} + \vec{b} \cdot \vec{\mathbf{x}} = \mathbf{t}_1$, a proof $\vec{\boldsymbol{\pi}}$ (along with commitments to variables) is obtained by setting $\vec{\boldsymbol{\pi}} = S^\top \iota_2(\vec{\mathbf{a}}) + R^\top \iota_1(\vec{b}) + \vec{\boldsymbol{\theta}}$, where R is the matrix of rows (r_1, r_2, r_3), coming from $c_2(\mathbf{x}_i)$, one for each committed variable $\mathbf{x}_i$, and S is the matrix of rows (r_1, r_2, r_3), coming from $c_1(y_i)$, one for each committed variable y_i. Note, $\vec{\boldsymbol{\pi}}$ is vector of three $\mathbb{G}^3$ elements. The vector $\vec{\boldsymbol{\theta}}$ is set to be a random linear combination of $H_i\vec{\boldsymbol{u}}$, where H_i are finitely many matrices, and form a basis for the solutions to $\vec{\boldsymbol{u}} \bullet H\vec{\boldsymbol{u}} = 0$. It turns out that these matrices H_i are independent of the ZK simulator trapdoors α and β.

Let "$\bullet$" denote the dot product of vectors of elements from $\mathbb{G}^3$ and $\mathbb{G}^3$ w.r.t. product $\otimes$. The commitments $\vec{c_1}, \vec{c_2}$ and the proof are **verified** by the following equation:

$$\iota_1(\vec{b}) \bullet \vec{c_2} + \vec{c_1} \bullet \iota_2(\vec{\mathbf{a}}) = \iota_1(1) \bullet \iota_2(\mathbf{t}_1) + \vec{\boldsymbol{u}} \bullet \vec{\boldsymbol{\pi}}.$$

Quasi-Adaptive Aggregation. In the quasi-adaptive setting [JR13], the NIZK CRS is allowed to depend on the language parameters, but with the further requirement that the ZK simulation be uniform. In the above context, the language parameters are $\vec{\mathbf{a}}$ and $\vec{\mathrm{b}}$. Note $\mathbf{t}_1$ is *not* a language parameter, as it is a quantity produced by the prover.

So, let there be k equations in the same variables, with the j-th equation being

$$\vec{y} \cdot \vec{\mathbf{a}}^j + \vec{b}^j \cdot \vec{\mathbf{x}} = \mathbf{t}_1^j \tag{7}$$

In the above setting the prover produces k proofs, $\vec{\boldsymbol{\pi}}^j$. We would like the prover to give a random linear combination of these proofs, where the randomness is fixed in the CRS setup. In the DLIN setting, we need two different linear combinations. Thus, let the CRS generator choose two random $\mathbb{Z}_q$-vectors $\vec{\rho}$ and $\vec{\psi}$. The prover is required to produce $\vec{\boldsymbol{\pi}}_\rho = \sum_{j \in [1,k]} \rho_j \cdot \vec{\boldsymbol{\pi}}^j$ and $\vec{\boldsymbol{\pi}}_\psi = \sum_{j \in [1,k]} \psi_j \cdot \vec{\boldsymbol{\pi}}^j$. To be able to do so, the prover needs $\sum_j \rho_j \cdot \iota_2(\vec{\mathbf{a}}^j)$, $\sum_j \rho_j \cdot \iota_1(\vec{b}^j)$ (and similar terms using ψ_j). The $\vec{\boldsymbol{\theta}}$ terms in the proofs need not be linearly combined, and the prover can just add one such term to each of $\vec{\boldsymbol{\pi}}_\rho$ and $\vec{\boldsymbol{\pi}}_\psi$, as its purpose is only to allow zero-knowledge simulation (i.e. witness hiding). The CRS generator can certainly produce these elements and give them as part of the CRS. The CRS generator also needs to give as part of the verification CRS the terms $\langle \iota_1(\rho_j) \rangle_j$ and $\langle \iota_1(\psi_j) \rangle_j$. In order to apply the switching lemma, we show in the proof of the theorem below that if $\vec{\mathbf{a}}^j$ are

[2] In this section we will use the vector notation to represent vectors as columns.

efficiently witness samplable, then the CRS generator can also simulate this verification CRS given $\rho_j \cdot \mathbf{g}$ and $\psi_j \cdot \mathbf{g}$.

The verification is now done as follows:

$$(\sum_j \rho_j \cdot \iota_1(\vec{b^j})) \bullet \vec{c_2} + \vec{c_1} \bullet (\sum_j \rho_j \cdot \iota_2(\vec{\mathbf{a}}^j)) = \sum_j (\iota_1(\rho_j) \bullet \iota_2(\mathbf{t}_1^j)) + \vec{\boldsymbol{u}} \bullet \vec{\boldsymbol{\pi}}_\rho \tag{8}$$

$$(\sum_j \psi_j \cdot \iota_1(\vec{b^j})) \bullet \vec{c_2} + \vec{c_1} \bullet (\sum_j \psi_j \cdot \iota_2(\vec{\mathbf{a}}^j)) = \sum_j (\iota_1(\psi_j) \bullet \iota_2(\mathbf{t}_1^j)) + \vec{\boldsymbol{u}} \bullet \vec{\boldsymbol{\pi}}_\psi \tag{9}$$

Theorem 4 *The above system constitutes a computationally-sound quasi-adaptive NIZK proof system for equations (7) with parameters $\langle \vec{\mathbf{a}}^j \rangle_j$, $\langle \vec{b^j} \rangle_j$, whenever $\langle \vec{\mathbf{a}}^j \rangle_j$ are chosen according to an efficiently witness-samplable distribution, and given any group generation algorithm for which the DLIN assumption holds.*

Proof of the theorem can be found in Appendix F. Since Groth-Sahai proofs of more general equations (involving quadratic terms) require pairing of adversarially supplied commitments with each other, the switching lemma is not directly applicable. It remains an open problem to aggregate such NIZK proofs.

6 Extensions and Applications

Tags. We extend the system of Section 4 to include tags mirroring [JR13]. The tags are elements of $\mathbb{Z}_q$, are included as part of the proof and are used as part of the defining equations of the language. We still get k element proofs based on the k-linear assumption. Details are in Appendix G.

KDM-CCA2 Encryption [CCS09]. In the paper [CCS09], the authors construct a public key encryption scheme simultaneously secure against key dependent chosen plaintext (KDM) and adaptive chosen ciphertext attacks (CCA2). They apply a Naor-Yung "double encryption" paradigm to combine any KDM-CPA secure scheme with any IND-CCA2 secure scheme along with an appropriate NIZK proof, to obtain a KDM-CCA2 secure scheme. In a particular construction, they obtain short ciphertexts by combining the KDM-CPA secure scheme of [BHHO08] with the IND-CCA2 scheme of [CS98], along with a Groth-Sahai NIZK proof. We show that the NIZK proof required in this construction can be considerably shortened. We defer the reader to [CCS09] for details of the scheme, and just describe the equations to be proved here. Consider bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ in which the K-linear and L-linear assumptions hold, respectively.

Let $\vec{\mathbf{g}}_1, \cdots, \vec{\mathbf{g}}_K, \mathbf{h}_1, \cdots, \mathbf{h}_K$ be part of the public key of the KDM-CPA secure encryption scheme and let $\vec{\mathbf{f}}_1, \cdots, \vec{\mathbf{f}}_K, \mathbf{c}_1, \cdots, \mathbf{c}_K, \mathbf{d}_1, \cdots, \mathbf{d}_K, \mathbf{e}_1, \cdots, \mathbf{e}_K$ be part of the public key of the IND-CCA2 secure encryption scheme. Let $(\vec{\mathbf{g}}, \mathbf{h}) \in \mathbb{G}_1^N \times \mathbb{G}_1$ be a ciphertext from the KDM-CPA secure encryption scheme and $(\vec{\mathbf{f}}, \mathbf{a}, \mathbf{b}) \in \mathbb{G}_1^{K+1} \times \mathbb{G}_1 \times \mathbb{G}_1$ be a ciphertext from the IND-CCA2 secure encryption scheme, with label l. Let $t = H(\vec{\mathbf{f}}, \mathbf{a}, l)$, where H is a collision resistant hash. The purpose of the NIZK proof is to establish that they encrypt the same plaintext. This translates to the following statement:

$$\exists r_1, \cdots, r_K, w_1, \cdots, w_K : \begin{pmatrix} \vec{\mathbf{g}} = \sum_{i=1}^K r_i \cdot \vec{\mathbf{g}}_i \ \wedge\ \vec{\mathbf{f}} = \sum_{i=1}^K w_i \cdot \vec{\mathbf{f}}_i \wedge \\ \mathbf{b} = \sum_{i=1}^K w_i \cdot (\mathbf{d}_i + t \cdot \mathbf{e}_i) \wedge \\ \mathbf{h} - \mathbf{a} = \sum_{i=1}^K r_i \cdot \mathbf{h}_i - \sum_{i=1}^K w_i \cdot \mathbf{c}_i \end{pmatrix}$$

This translates into $N + K + 3$ equations in $2K$ variables. Using the Groth-Sahai NIZK scheme, this requires $(2K)(L + 1)$ elements of $\mathbb{G}_2$ and $(N + K + 3)L$ elements of $\mathbb{G}_1$. In our scheme this requires L elements of $\mathbb{G}_1$ in the proof - **1** under DDH and **2** under DLIN assumptions in $\mathbb{G}_2$.

CCA2-IBE Scheme [JR13]. The definition of CCA2-secure encryption [BDPR98] naturally extends to the Identity-Based Encryption setting [CHK04]. In [JR13], the authors construct a fully adaptive CCA2-secure IBE, which also allows public verification of the assertion that a ciphertext is valid for the particular claimed identity. The IBE scheme has four group elements (and a tag), where one group element serves as one-time pad for encrypting the plaintext. The remaining three group elements form a linear subspace with one variable as witness and three integer tags corresponding to: (a) the identity, (b) the tag needed in the IBE scheme, and (c) a 1-1 (or universal one-way) hash of some of the elements. It was shown that if these three group elements can be QA-NIZK proven to be consistent, and given the unique proof property of the QA-NIZKs, then the IBE scheme can be made CCA2-secure. Since, there are three components, and one variable the QA-NIZK required only two group elements under SXDH. We slightly shorten the proof to one element under SXDH. We defer the reader to [JR13] for details, and just describe the Key Generation and Encryption steps in Appendix H.

References

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Berlin, Germany. 6, 7, 8

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Berlin, Germany. 6

[BFI+10] Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert, and Damien Vergnaud. Batch Groth-Sahai. In Jianying Zhou and Moti Yung, editors, *ACNS 10: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in Computer Science*, pages 218–235, Beijing, China, June 22–25, 2010. Springer, Berlin, Germany. 1

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. 1, 6

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 351–368, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany. 1, 6

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany. 1, 6

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Berlin, Germany. 6

[Dam] Ivan Damgård. On Σ protocols. `http://www.daimi.au.dk/~ivan/Sigma.pdf`. 1

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 5

[Eis95] David Eisenbud. *Commutative algebra with a view toward algebraic geometry*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, 1995. 5

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1986. Springer, Berlin, Germany. 1

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany. 1

[HK07] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 553–571, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Berlin, Germany. 1, 9

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 1–20, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany. (document), 1, 1, 3, 4, 4, 5, 6, 6, G, H

[Sha07] Hovav Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. `http://eprint.iacr.org/`. 1, 9

A Hardness Assumptions

Definition 5 (DDH [DH76]) *Assuming a generation algorithm $\mathcal{G}$ that outputs a tuple $(q, \mathbb{G}, \mathbf{g})$ such that $\mathbb{G}$ is of prime order q and has generator g, the DDH assumption asserts that it is computationally infeasible to distinguish between $(\mathbf{g}, a\cdot\mathbf{g}, b\cdot\mathbf{g}, c\cdot\mathbf{g})$ and $(\mathbf{g}, a\cdot\mathbf{g}, b\cdot\mathbf{g}, ab\cdot\mathbf{g})$ for $a, b, c \xleftarrow{\$} \mathbb{Z}_q$. More formally, for all PPT adversaries A there exists a negligible function $\nu()$ such that*

$$\left| \begin{array}{c} Pr[(q, \mathbb{G}, \mathbf{g}) \leftarrow \mathcal{G}(1^m); a, b, c \leftarrow \mathbb{Z}_q : A(\mathbf{g}, a\cdot\mathbf{g}, b\cdot\mathbf{g}, c\cdot\mathbf{g}) = 1] - \\ Pr[(q, \mathbb{G}, \mathbf{g}) \leftarrow \mathcal{G}(1^m); a, b \leftarrow \mathbb{Z}_q : A(\mathbf{g}, a\cdot\mathbf{g}, b\cdot\mathbf{g}, ab\cdot\mathbf{g}) = 1] \end{array} \right| < \nu(m)$$

Definition 6 (XDH [BBS04]) *Consider a generation algorithm $\mathcal{G}$ taking the security parameter as input, that outputs a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2)$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q with generators $\mathbf{g}_1, \mathbf{g}_2$ and $e(\mathbf{g}_1, \mathbf{g}_2)$ respectively and which allow an efficiently computable $\mathbb{Z}_q$-bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The eXternal decisional Diffie-Hellman (XDH) assumption asserts that the Decisional Diffie-Hellman (DDH) problem is hard in* one *of the groups $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Definition 7 (SXDH [BBS04]) *Consider a generation algorithm $\mathcal{G}$ taking the security parameter as input, that outputs a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2)$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q with generators $\mathbf{g}_1, \mathbf{g}_2$ and $e(\mathbf{g}_1, \mathbf{g}_2)$ respectively and which allow an efficiently computable $\mathbb{Z}_q$-bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The Symmetric eXternal decisional Diffie-Hellman (SXDH) assumption asserts that the Decisional Diffie-Hellman (DDH) problem is hard in* both *the groups $\mathbb{G}_1$ and $\mathbb{G}_2$.*

Definition 8 (DLIN [BBS04]) *Assuming a generation algorithm $\mathcal{G}$ that outputs a tuple $(q, \mathbb{G})$ such that $\mathbb{G}$ is of prime order q and has generators $\mathbf{g}, \mathbf{f}, \mathbf{h} \xleftarrow{\$} \mathbb{G}$, the $DLIN$ assumption asserts that it is computationally infeasible to distinguish between $(\mathbf{g}, \mathbf{f}, \mathbf{h}, x_1\cdot\mathbf{g}, x_2\cdot\mathbf{f}, x_3\cdot\mathbf{h})$ and $(\mathbf{g}, \mathbf{f}, \mathbf{h}, x_1\cdot\mathbf{g}, x_2\cdot\mathbf{f}, (x_1+x_2)\cdot\mathbf{h})$ for $x_1, x_2, x_3 \xleftarrow{\$} \mathbb{Z}_q$. More formally, for all PPT adversaries A there exists a negligible function $\nu()$ such that*

$$\left| \begin{array}{c} Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g}, \mathbf{f}, \mathbf{h} \xleftarrow{\$} \mathbb{G}; x_1, x_2, x_3 \xleftarrow{\$} \mathbb{Z}_q : A(\mathbf{g}, \mathbf{f}, \mathbf{h}, x_1\cdot\mathbf{g}, x_2\cdot\mathbf{f}, x_3\cdot\mathbf{h}) = 1] - \\ Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g}, \mathbf{f}, \mathbf{h} \xleftarrow{\$} \mathbb{G}; x_1, x_2 \xleftarrow{\$} \mathbb{Z}_q : A(\mathbf{g}, \mathbf{f}, \mathbf{h}, x_1\cdot\mathbf{g}, x_2\cdot\mathbf{f}, (x_1+x_2)\cdot\mathbf{h}) = 1] \end{array} \right| < \nu(m)$$

Definition 9 (k-linear [Sha07, HK07]) *For a constant $k \geq 1$, assuming a generation algorithm $\mathcal{G}$ that outputs a tuple $(q, \mathbb{G})$ such that $\mathbb{G}$ is of prime order q and has generators $\mathbf{g}_1, \cdots, \mathbf{g}_{k+1} \xleftarrow{\$} \mathbb{G}$, the k-linear assumption asserts that it is computationally infeasible to distinguish between $(\mathbf{g}_1, ..., \mathbf{g}_{k+1}, x_1\cdot\mathbf{g}_1, ..., x_{k+1}\cdot\mathbf{g}_{k+1})$ and $(\mathbf{g}_1, ..., \mathbf{g}_{k+1}, x_1\cdot\mathbf{g}_1, ..., (x_1+...+x_k)\cdot\mathbf{g}_{k+1})$ for $x_1, ..., x_{k+1} \xleftarrow{\$} \mathbb{Z}_q$. More formally, for all PPT adversaries A there exists a negligible function $\nu()$ such that*

$$\left| \begin{array}{c} Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g}_1, \cdots, \mathbf{g}_{k+1} \xleftarrow{\$} \mathbb{G}; x_1, ..., x_{k+1} \xleftarrow{\$} \mathbb{Z}_q : A(\mathbf{g}_1, ..., \mathbf{g}_{k+1}, x_1\cdot\mathbf{g}_1, ..., x_{k+1}\cdot\mathbf{g}_{k+1}) = 1] - \\ Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g}_1, \cdots, \mathbf{g}_{k+1} \xleftarrow{\$} \mathbb{G}; x_1, ..., x_k \xleftarrow{\$} \mathbb{Z}_q : A(\mathbf{g}_1, ..., \mathbf{g}_{k+1}, x_1\cdot\mathbf{g}_1, ..., (x_1+...+x_k)\cdot\mathbf{g}_{k+1}) = 1] \end{array} \right| < \nu(m)$$

B Proof of the Switching Lemma

Instead of using the k-linear assumption directly, we use a related assumption we call the *k-lifted linear assumption* which, as we show, is implied by the k-linear assumption with a perfect reduction.

Definition 10 (k-lifted linear assumption) *For a constant $k \geq 1$, assuming a generation algorithm $\mathcal{G}$ that outputs a tuple $(q, \mathbb{G})$ such that $\mathbb{G}$ is of prime order q and has a generator $\mathbf{g} \xleftarrow{\$} \mathbb{G}$, the k-lifted linear assumption asserts that it is computationally infeasible to distinguish between*

$$\textsc{Tuple}_0 = \left(b_1 \cdot \mathbf{g}, \cdots, b_k \cdot \mathbf{g}, r_1 \cdot \mathbf{g}, \cdots, r_k \cdot \mathbf{g},\ b_1 s_1 \cdot \mathbf{g}, \cdots, b_k s_k \cdot \mathbf{g}, \left(\sum_{i=1}^{n} r_i s_i\right) \cdot \mathbf{g}\right)$$

and

$$\textsc{Tuple}_1 = (b_1 \cdot \mathbf{g}, \cdots, b_k \cdot \mathbf{g}, r_1 \cdot \mathbf{g}, \cdots, r_k \cdot \mathbf{g},\ b_1 s_1 \cdot \mathbf{g}, \cdots, b_k s_k \cdot \mathbf{g}, s_{k+1} \cdot \mathbf{g}) \,.$$

More formally, for all PPT adversaries A there exists a negligible function $\nu()$ such that

$$\left| \begin{array}{c} Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g} \xleftarrow{\$} \mathbb{G}; b_1, ..., b_k, r_1, ..., r_k, s_1, ..., s_{k+1} \xleftarrow{\$} \mathbb{Z}_q : A(\textsc{Tuple}_0) = 1] - \\ Pr[(q, \mathbb{G}) \leftarrow \mathcal{G}(1^m); \mathbf{g} \xleftarrow{\$} \mathbb{G}; b_1, ..., b_k, r_1, ..., r_k, s_1, ..., s_{k+1} \xleftarrow{\$} \mathbb{Z}_q : A(\textsc{Tuple}_1) = 1] \end{array} \right| < \nu(m)$$

It is worth noting that the k-linear assumption is a variant of the k-lifted linear assumption with all $r_1, ..., r_k$ equal to one. But the following shows that the k-lifted linear assumption is weaker.

Theorem 11 (k-linear assumption ⇒ k-lifted linear assumption) *The k-linear assumption implies the k-lifted linear assumption. Moreover, the advantage of a k-lifted linear adversary is upper bounded by the maximum advantage of a k-linear adversary.*

Proof: Suppose we are given a k-linear instance $(b_1 \cdot \mathbf{g}, ..., b_k \cdot \mathbf{g},\quad \mathbf{g},\quad b_1 s_1 \cdot \mathbf{g}, ..., b_k s_k \cdot \mathbf{g},\quad \boldsymbol{\chi})$, where $\boldsymbol{\chi}$ is either $(s_1 + ... + s_k) \cdot \mathbf{g}$ or random. We will now efficiently construct a k-lifted linear instance from this k-linear instance, thus showing a randomized reduction from the k-linear challenge problem to the k-lifted linear challenge problem.

Choose $u_1, ..., u_k \xleftarrow{\$} \mathbb{Z}_q$. Implicitly set $r_i = b_i u_i + 1$. Thus $r_i \cdot \mathbf{g} = u_i \cdot (b_i \cdot \mathbf{g}) + \mathbf{g}$. The $r_i \cdot \mathbf{g}$'s are independently random wrt the $b_i \cdot \mathbf{g}$'s due to the u_i's. Now consider $\boldsymbol{\chi}' = \sum_{i=1}^{k} [u_i \cdot (b_i s_i \cdot \mathbf{g})] + \boldsymbol{\chi}$. In the case where $\boldsymbol{\chi} = (s_1 + ... + s_k) \cdot \mathbf{g}$, this is equal to $\sum_{i=1}^{k} [u_i \cdot (b_i s_i \cdot \mathbf{g})] + (s_1 + ... + s_k) \cdot \mathbf{g} = \sum_{i=1}^{k} [u_i \cdot (b_i s_i \cdot \mathbf{g}) + s_i \cdot \mathbf{g}] = \sum_{i=1}^{k} [r_i s_i \cdot \mathbf{g}]$. In the case where $\boldsymbol{\chi}$ is random, $\boldsymbol{\chi}'$ is random. Hence the following tuple is a k-lifted linear $\textsc{Tuple}_0$ if $\boldsymbol{\chi}$ is from a real k-linear distribution and is a k-lifted linear $\textsc{Tuple}_1$ if $\boldsymbol{\chi}$ is from a fake k-linear distribution:

$$\left(b_1 \cdot \mathbf{g}, \cdots, b_k \cdot \mathbf{g}, r_1 \cdot \mathbf{g}, \cdots, r_k \cdot \mathbf{g},\ b_1 s_1 \cdot \mathbf{g}, \cdots, b_k s_k \cdot \mathbf{g}, \boldsymbol{\chi}'\right)$$

□

Now we proceed to prove the Simple Switching Lemma and the Switching Lemma.

Proof: (of Lemma 1) Let the latter probability be $\Delta'_{\mathcal{A}}$, and suppose it differs from the former by δ. Then, we will show that there is an adversary $\mathcal{S}$, that can use $\mathcal{A}$ to distinguish a real k-lifted linear $\textsc{Tuple}_0$ from a fake k-lifted linear $\textsc{Tuple}_1$ (in $\mathbb{G}_2$) with probability negligibly close to δ. So, let a *k-lifted linear* challenger produce:

$$(b_1 \cdot \mathbf{g}_2, \cdots, b_k \cdot \mathbf{g}_2, r_1 \cdot \mathbf{g}_2, \cdots, r_k \cdot \mathbf{g}_2,\ b_1 s_1 \cdot \mathbf{g}_2, \cdots, b_k s_k \cdot \mathbf{g}_2, \boldsymbol{\chi})$$

in the group $\mathbb{G}_2$, where $\boldsymbol{\chi}$ is either $(\sum_{i=1}^{k} r_i s_i) \cdot \mathbf{g}_2$ (in the real k-lifted challenge challenge case) or random (in the fake challenge case). Note that b_i, r_i and s_i are chosen randomly and independently by the challenger.

Let vectors $\vec{\mathbf{r}}$ and $\vec{\mathrm{s}}$ be defined component-wise as $(\vec{\mathbf{r}})_i = r_i \cdot \mathbf{g}_2$ and $(\vec{\mathrm{s}})_i = s_i$, respectively. Define the k by k matrix B as the diagonal matrix with the i-th diagonal element set to b_i. Further, let $\mathbf{B} = \mathsf{B} \cdot \mathbf{g}_2$.

$\mathcal{S}$ samples $\mathsf{C}^{n\times(k+1)} \leftarrow \mathcal{D}$, and chooses $\mathbf{g}_1$ at random. It also samples a k by k matrix R of $\mathbb{Z}_q$ elements. $\mathcal{S}$ sets $\mathbf{R} = \mathsf{R} \cdot \mathbf{B}$. Observe that $\mathbf{R}$'s entries are distributed uniformly random independent of $\mathbf{B}$. $\mathcal{S}$ then gives $\mathbf{g}_1$, $\mathbf{g}_2$, $\vec{\mathbf{r}}$, $\mathbf{R}$ and C to adversary $\mathcal{A}$. The adversary $\mathcal{A}$ in response produces $\vec{\mathbf{f}}^{1\times n}$. Now, $\mathcal{S}$ checks that $\vec{\mathbf{f}}$ is non-zero and noting that $\mathcal{S}$ has access to $\mathsf{B} \cdot \vec{\mathrm{s}}^\top \cdot \mathbf{g}_2$, it (efficiently) performs the following bilinear test:

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathsf{R} \cdot \mathsf{B} \cdot \vec{\mathrm{s}}^\top \cdot \mathbf{g}_2 \\ \boldsymbol{\chi} \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T \tag{10}$$

$\mathcal{S}$ outputs 1 if the test succeeds, and otherwise it outputs 0.

Note that the above experiment has two games, one corresponding to real k-lifted linear challenge TUPLE_0 choice, and one corresponding to fake k-lifted linear TUPLE_1 challenge choice. We will call these games $\mathbf{G}_0$ (the real game) and $\mathbf{G}_0'$ (the fake game). Our aim is to show that the probability of $\mathcal{S}$ outputting 1 in the real game differs from the probability of its outputting 1 in the fake game by (negligibly close to) δ. In other words we aim to show,

$$|Pr_{\mathbf{G}_0}[\mathcal{S}(\cdot) = 1] \; - \; Pr_{\mathbf{G}_0{}'}[\mathcal{S}(\cdot) = 1]| \geq \delta - \epsilon$$

where ϵ is a negligible function of the security parameters.

To prove this, we first modify the above two games and argue that the probabilities for both the real and the fake games remain the same respectively. In the modified games $\mathbf{G}_1$ and $\mathbf{G}_1'$, $\mathcal{S}$ itself chooses the k-lifted linear challenges according to the same distribution as in $\mathbf{G}_0$ and $\mathbf{G}_0'$. However, it defers the choice of $\vec{\mathrm{s}}$ to after $\mathcal{A}$ has responded (noting that $\mathcal{A}$ is not given anything related to $\vec{\mathrm{s}}$). After $\mathcal{A}$ responds, $\mathcal{S}$ chooses $\vec{\mathrm{s}}$ at random, and also sets $\boldsymbol{\chi}$ as either $\vec{\mathbf{r}} \cdot \vec{\mathrm{s}}^\top$ (in $\mathbf{G}_1$) and as $\vec{\mathbf{r}}' \cdot \vec{\mathrm{s}}^\top$ (in $\mathbf{G}_1'$), where $\vec{\mathbf{r}}'$ is another random k-tuple independent of $\vec{\mathbf{r}}$. $\mathcal{S}$ then performs the same test (10) as above, and outputs 1 if the test succeeds, and otherwise it outputs 0. Since the distributions in games $\mathbf{G}_1$ and $\mathbf{G}_1'$ are identical to the distributions in $\mathbf{G}_0$ and $\mathbf{G}_0'$ (resp.), the probabilities of $\mathcal{S}$ outputting 1 remains the same in the respective games.

Now, note that in the (real) game $\mathbf{G}_1$ the above test (10) is equivalent to testing

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}} \end{bmatrix} \cdot \vec{\mathrm{s}}^\top\right) \stackrel{?}{=} \mathbf{0}_T \,, \tag{11}$$

and in the (fake) game $\mathbf{G}_1'$ the test (10) is equivalent to testing

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}}' \end{bmatrix} \cdot \vec{\mathrm{s}}^\top\right) \stackrel{?}{=} \mathbf{0}_T \,. \tag{12}$$

We now define games $\mathbf{G}_2$ and $\mathbf{G}_2'$ which are identical to games $\mathbf{G}_1$ and $\mathbf{G}_1'$ (resp.) except that instead of (10) the final test performed by $\mathcal{S}$ in $\mathbf{G}_2$ is

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}} \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}, \tag{13}$$

and the final test performed by $\mathcal{S}$ in $\mathbf{G}_2'$ is

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}}' \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}. \tag{14}$$

Going through the details of games $\mathbf{G}_2$ and $\mathbf{G}_2{}'$, and comparing them with definition of $\Delta_{\mathcal{A}}$ and $\Delta'_{\mathcal{A}}$ (resp.), it is clear that

$$Pr_{\mathbf{G}_2}[\mathcal{S}(\cdot) = 1] = \Delta_{\mathcal{A}} \text{ and } Pr_{\mathbf{G}_2{}'}[\mathcal{S}(\cdot) = 1] = \Delta'_{\mathcal{A}} \tag{15}$$

Now we focus back on game $\mathbf{G}_1$. Let us consider a particular choice of random coins, say COINS* that are used in obtaining all the random variables *except* $\vec{\mathrm{s}}$. These coins also include the internal coins used by $\mathcal{A}$ in producing $\vec{\mathbf{f}}$. Thus, COINS* along with $\mathcal{A}$'s strategy fixes $\vec{\mathbf{f}}, \mathsf{C}, \mathbf{R}$ and $\vec{\mathbf{r}}$. Now consider those values of $\vec{\mathbf{f}}, \mathsf{C}, \mathbf{R}, \vec{\mathbf{r}}$ such that

$$e\left(\vec{\mathbf{f}}, \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}} \end{bmatrix}\right) \neq \mathbf{0}_T^{1\times k}. \tag{16}$$

In particular if $\vec{\mathrm{f}}^{1\times n}$ stands for the discrete log of $\vec{\mathbf{f}}$ w.r.t. $\mathbf{g}_1$, then the above is equivalent by bi-linearity of e to

$$\vec{\mathrm{f}} \cdot \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}} \end{bmatrix} \neq \mathbf{0}_2^{1\times k}.$$

For any such value of COINS*, we consider the probability over random choice of $\vec{\mathrm{s}}$ that test (11) holds. Again, since by bi-linearity of e the test (11) is equivalent to

$$\left(\vec{\mathrm{f}} \cdot \mathsf{C} \cdot \begin{bmatrix} \mathbf{R} \\ \vec{\mathbf{r}} \end{bmatrix}\right) \cdot \vec{\mathrm{s}}^{\top} = \mathbf{0}_2,$$

this probability is at most $1/q$ for COINS* satisfying (16). For coins *not* satisfying (16), i.e. where the pairing e yields an all zero vector, this probability (over choice of $\vec{\mathrm{s}}$) is trivially one. Now, by definition of $\mathbf{G}_2$, $Pr_{\mathbf{G}_2}[\mathcal{S}(\cdot) = 1]$ is exactly the fraction of coins *not* satisfying (16) in $\mathbf{G}_2$, and also in $\mathbf{G}_1$ (since the distributions are identical in the two games). Since by (15), $Pr_{\mathbf{G}_2}[\mathcal{S}(\cdot) = 1] = \Delta_{\mathcal{A}}$, we have

$$\begin{aligned}
Pr_{\mathbf{G}_1}[\mathcal{S}(\cdot) = 1] &= \sum_{\text{COINS}^*} Pr_{\mathbf{G}_1}[\mathcal{S}(\cdot) = 1 \mid \text{coins} = \text{COINS}^*] \cdot Pr_{\mathbf{G}_1}[\text{coins} = \text{COINS}^*] \\
&= \sum_{\text{COINS}^*:\ (16) \text{ holds}} Pr_{\mathbf{G}_1}[\mathcal{S}(\cdot) = 1 \mid \text{coins} = \text{COINS}^*] \cdot Pr_{\mathbf{G}_1}[\text{coins} = \text{COINS}^*] \\
&+ \sum_{\text{COINS}^*:\ \neg\ (16) \text{ holds}} 1 \cdot Pr_{\mathbf{G}_1}[\text{coins} = \text{COINS}^*] \\
&\leq 1/q + \Delta_{\mathcal{A}} \text{ (and also } \geq \Delta_{\mathcal{A}})
\end{aligned}$$

A similar argument shows that the probability of equation (12) holding (i.e. $\mathcal{S}$ outputting 1 in $\mathbf{G}_1{}'$) differs from $\Delta'_{\mathcal{A}}$ by at most $1/q$. Since, by hypothesis $|\Delta'_{\mathcal{A}} - \Delta_{\mathcal{A}}| = \delta$, it follows that the probability of $\mathcal{S}$ outputting 1 in $\mathbf{G}_1{}'$ differs from $\mathcal{S}$ outputting 1 in $\mathbf{G}_1$ by at least $\delta - 2/q$. Since the probability of $\mathcal{S}$ outputting 1 is same in game $\mathbf{G}_1$ ($\mathbf{G}_1{}'$) as in game $\mathbf{G}_0$ ($\mathbf{G}_0{}'$ resp.), that completes the proof. □

It is instructive to go through the details of proof of Lemma 1, as the following proof that uses similar ideas but in an inductive argument, skips detailed description of games for sake of exposition.

Proof: (of Lemma 2) When $m \leq k$, the lemma follows information-theoretically as follows (although the proof for $m > k$ also works for this case). Let $\vec{f}^{1\times n}$ be the discrete log of $\vec{\mathbf{f}}$ w.r.t. $\mathbf{g}_1$. Similarly, let R be the matrix of discrete logs of $\mathbf{R}$ w.r.t. $\mathbf{g}_2$. Then, $e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\mathbf{R}\right) = \vec{\mathbf{0}}_T^{1\times k}$ is equivalent to $\vec{f}\cdot\mathsf{C}\cdot R = \vec{0}^{1\times k}$. Since R is randomly and uniformly chosen and $m \leq k$, this is equivalent to $\vec{f}\cdot\mathsf{C} = \vec{0}^{1\times m}$ with high probability (over choice of R). Thus the first probability being close to $\Delta_{\mathcal{A}}$ is equivalent to probability of the following

$$\vec{f} \neq \vec{0} \text{ and } \ \vec{f}\cdot\mathsf{C} = \vec{0}^{1\times m}$$

being negligibly close to $\Delta_{\mathcal{A}}$. But this implies that the second probability in the lemma is also negligibly close to $\Delta_{\mathcal{A}}$.

Now we focus on the case that $m > k$. Consider the following inductive hypothesis (over j):

$$\Pr\left[\begin{array}{c}\mathbf{R} \xleftarrow{\$} \mathbb{G}_2^{m\times k},\ \mathsf{C}^{n\times m} \leftarrow \mathcal{D},\ \vec{\mathbf{f}}^{1\times n} \leftarrow \mathcal{A}(\mathbf{g}_1, \mathbf{g}_2, \mathbf{R}, \mathsf{C}),\ \mathbf{R}' \xleftarrow{\$} \mathbb{G}_2^{m\times k}: \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}^{1\times n} \textbf{ and } \ e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\mathbf{R}''\right) = \vec{\mathbf{0}}_T^{1\times k}\end{array}\right]$$

is negligibly close to $\Delta_{\mathcal{A}}$, where $\mathbf{R}''$ has the first $m-j$ rows same as first $m-j$ rows of $\mathbf{R}$ and the last j rows same as the last j rows of $\mathbf{R}'$. In the base case, i.e., when $j = 0$, this is same as the hypothesis (antecedent) in the lemma, and when $j = m$, this induction hypothesis is same as the claim (consequent) in the lemma. Thus, we just need to prove the induction step.

For an adversary $\mathcal{A}$, suppose the difference in the two probabilities corresponding to (induction hypothesis for) $j = t$ and $j = t+1$ be δ. More precisely, let the probability for adversary $\mathcal{A}$ corresponding to $j = t$ by $\Delta^t_{\mathcal{A}}$. Thus, we are supposing that $|\Delta^t_{\mathcal{A}} - \Delta^{t+1}_{\mathcal{A}}| \geq \delta$. Using $\mathcal{A}$ as a black box we wlll demonstrate an adversary $\mathcal{S}$ that will have advantage at least (negligibly close to) δ to break the k-lifted linear assumption. The proof is similar to the proof of Lemma 1.

So, let a *k-lifted linear* challenger produce:

$$(b_1\cdot\mathbf{g}_2, \cdots, b_k\cdot\mathbf{g}_2, r_1\cdot\mathbf{g}_2, \cdots, r_k\cdot\mathbf{g}_2,\ b_1 s_1\cdot\mathbf{g}_2, \cdots, b_k s_k\cdot\mathbf{g}_2, \boldsymbol{\chi})$$

in the group $\mathbb{G}_2$, where $\boldsymbol{\chi}$ is either $(\sum_{i=1}^n r_i s_i)\cdot\mathbf{g}_2$ or random. Note that b_i, r_i and s_i are chosen randomly and independently by the challenger.

Let vectors $\vec{\mathbf{r}}$ and $\vec{\mathsf{s}}$ be defined component-wise as $(\vec{\mathbf{r}})_i = r_i\cdot\mathbf{g}_2$ and $(\vec{\mathsf{s}})_i = s_i$, respectively. Define the k by k matrix B as the diagonal matrix with the i-th diagonal element set to b_i. Further, let $\mathbf{B} = \mathsf{B}\cdot\mathbf{g}_2$.

$\mathcal{S}$ samples $\mathsf{C}^{n\times m} \leftarrow \mathcal{D}$, and chooses $\mathbf{g}_1$ at random. It next samples an $(m-t-1)$ by k matrix R_1 at random from $\mathbb{Z}_q$ (i.e. all elements of the matrix chosen randomly and independently from $\mathbb{Z}_q$). It sets $\mathbf{R}_1 = \mathsf{R}_1\cdot\mathbf{B}$. It further samples a t by k matrix $\mathbf{R}_2$ at random from $\mathbb{G}_2$ (i.e. all elements chosen randomly and independently from $\mathbb{G}_2$). Finally $\mathcal{S}$ sets $\mathbf{R}$ to be the rows of $\mathbf{R}_1$, the row $\vec{\mathbf{r}}$ and the rows of $\mathbf{R}_2$ combined (in that order) to form an m by k matrix. Observe that all of $\mathbf{R}$'s entries are independently random. The adversary $\mathcal{A}$ is then given $\mathbf{g}_1$, $\mathbf{g}_2$, $\mathbf{R}$ and C. The adversary $\mathcal{A}$ in response produces $\vec{\mathbf{f}}$. Now, $\mathcal{S}$ first checks that $\vec{\mathbf{f}}$ is non-zero. It then chooses another t by k matrix R_2 at random from $\mathbb{Z}_q$ and sets $\mathbf{R}_2 = \mathsf{R}_2\cdot\mathbf{B}$. Noting that $\mathcal{S}$ has access to $\mathsf{B}\cdot\vec{\mathsf{s}}^\top\cdot\mathbf{g}_2$, $\mathcal{S}$ performs the following bilinear test

$$e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\left[\begin{array}{c}\mathsf{R}_1\cdot\mathsf{B}\cdot\vec{\mathsf{s}}^\top\cdot\mathbf{g}_2 \\ \boldsymbol{\chi} \\ \mathsf{R}_2\cdot\mathsf{B}\cdot\vec{\mathsf{s}}^\top\cdot\mathbf{g}_2\end{array}\right]\right) \stackrel{?}{=} \mathbf{0}_T$$

When the k-lifted linear tuple is a $\textsc{Tuple}_0$ this is equivalent to testing

$$e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\begin{bmatrix}\mathbf{R}_1\\ \vec{\mathbf{r}}\\ \mathbf{R}_2\end{bmatrix}\cdot\vec{\mathrm{s}}^\top\right) \stackrel{?}{=} \mathbf{0}_T$$

and is otherwise equivalent to testing

$$e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\begin{bmatrix}\mathbf{R}_1\\ \vec{\mathbf{r}}'\\ \mathbf{R}_2\end{bmatrix}\cdot\vec{\mathrm{s}}^\top\right) \stackrel{?}{=} \mathbf{0}_T,$$

where $\vec{\mathbf{r}}'$ is another random k-tuple independent of $\vec{\mathbf{r}}$ (and all the other randomness mentioned so far). Using the same detailed argument given in proof of Lemma 1, since $\mathcal{A}$'s response is independent of the s_i's, the probability of the former test holding (i.e., for a k-lifted linear $\textsc{Tuple}_0$) is just $\Delta^t_{\mathcal{A}}$ by induction hypothesis for $j = t$, except for the negligible difference due to the following event:

$$e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\begin{bmatrix}\mathbf{R}_1\\ \vec{\mathbf{r}}\\ \mathbf{R}_2\end{bmatrix}\cdot\vec{\mathrm{s}}^\top\right) = \mathbf{0}_T \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\begin{bmatrix}\mathbf{R}_1\\ \vec{\mathbf{r}}\\ \mathbf{R}_2\end{bmatrix}\right) \neq \mathbf{0}_T^{1\times k}$$

The probability for the latter test holding (i.e., for a k-lifted linear $\textsc{Tuple}_1$) is $\Delta^{t+1}_{\mathcal{A}}$ (again, except for the negligible difference due to an outcome similar to above) – note that the row $\vec{\mathbf{r}}'$ and the rows of $\mathbf{R}_2$ would serve as the $t+1$ last rows of $\mathbf{R}'$ in the induction hypothesis. That completes the proof. □

C Aggregating Quasi-Adaptive Proofs of Linear Subspaces

We repeat the construction here for ease of reading and provide a detailed proof.

Algorithm K_1: The algorithm K_1 generates the CRS as follows. Let $\mathbf{A}^{t\times n}$ be the parameter supplied to K_1. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns. It generates a matrix $\mathsf{D}^{t\times k^2}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $\{b_v\}_{v\in[1,k]}$ and sk elements $\{r_{ij}\}_{i\in[1,s],j\in[1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Define matrices $\mathsf{R}^{s\times k^2}$ and $\mathsf{B}^{k^2\times k^2}$ component-wise as follows:

$$(\mathsf{R})_{i,k(u-1)+v} = r_{iu}, \ \text{with } i \in [1,s], u, v \in [1,k].$$

$$(\mathsf{B})_{ij} = \begin{cases} b_v & \text{if } i = j = k(u-1)+v, \ \text{with } u, v \in [1,k] \\ 0 & \text{if } i \neq j, \ \text{with } i, j \in [1,k^2] \end{cases}$$

Intuitively, the matrix R is a k times column-wise repetition of the r_{ij}'s, and if we denote $\{b_v\}_{v\in[1,k]}$ by $\vec{b}$, then the diagonal matrix B is just the vector $\vec{b}$ repeated k times along the diagonal (i.e. $\mathsf{B}_{k(u-1)+v,k(u-1)+v}$ is b_v and not b_u).

The common reference string (CRS) has two parts $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ which are to be used by the prover and the verifier respectively.

$$\mathbf{CRS}_p^{t\times k^2} := \mathbf{A}\cdot\begin{bmatrix}\mathsf{D}\\ \mathsf{R}\ \mathsf{B}^{-1}\end{bmatrix} \qquad\qquad \mathbf{CRS}_v^{(n+k^2)\times k^2} = \begin{bmatrix}\mathsf{D}\ \mathsf{B}\\ \mathsf{R}\\ -\mathsf{B}\end{bmatrix}\cdot\mathbf{g}_2$$

Prover P**:** Given candidate $\vec{l} = \vec{x} \cdot \mathbf{A}$ with witness vector $\vec{x}^{1\times t}$, the prover generates the following proof consisting of k^2 elements in $\mathbb{G}_1$: $\quad \vec{\mathbf{p}}^{1\times k^2} := \vec{x} \cdot \mathbf{CRS}_p$

Verifier V**:** Given candidate $\vec{l}$, and proof $\vec{\mathbf{p}}$, the verifier checks the following (k^2 equations) :

$$e\left(\left[\ \vec{l} \ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k^2}$$

Theorem 12 (Theorem 3 repeated) *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound quasi-adaptive NIZK proof system for linear subspace languages* $\{L_{\mathbf{A}}\}$ *with parameters* $\mathbf{A}$ *sampled from a robust and efficiently witness-samplable distribution* $\mathcal{D}$ *over the associated parameter language* $\mathcal{L}_{par}$*, given any group generation algorithm for which the k-linear assumption holds for group* $\mathbb{G}_2$*.*

Proof:

Completeness: For a candidate $\vec{x} \cdot \mathbf{A}$ (which is a language member), the left-hand-side of the verification equation is:

$$e(\left[\ \vec{l} \ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_2) \quad = \quad e(\left[\ \vec{x} \cdot \mathbf{A} \ \middle|\ \vec{x} \cdot \mathbf{CRS}_1\ \right], \mathbf{CRS}_2)$$

$$= e\left(\vec{x} \cdot \mathbf{A} \cdot \left[\ \mathrm{I}^{n\times n} \ \middle|\ \begin{matrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{matrix}\ \right] \cdot \left[\begin{matrix} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{matrix} \right], \mathbf{g}_2 \right)$$

$$= e\left(\vec{x} \cdot \mathbf{A} \cdot \left(\left[\begin{matrix} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \end{matrix} \right] - \left[\begin{matrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{matrix} \right] \cdot \mathsf{B} \right), \mathbf{g}_2 \right) = \ e(\mathbf{0}_1^{1\times k^2}, \mathbf{g}_2) = \mathbf{0}_T^{1\times k^2}$$

Hence completeness follows.

Zero Knowledge: The CRS is generated exactly as in the original system. In addition, the simulator is given the trapdoor $\left[\begin{matrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{matrix} \right]$. Now, given a language candidate $\vec{l}$, the proof is simply $\vec{\mathbf{p}} := \vec{l} \cdot \left[\begin{matrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{matrix} \right]$. If $\vec{l}$ is in the language, i.e., it is $\vec{x} \cdot \mathbf{A}$ for some $\vec{x}$, then the distribution of the simulated proof is identical to the real world proof. Therefore, the simulated NIZK CRS and simulated proofs of language members are identically distributed as the real world. Hence the system is perfect Zero Knowledge.

Soundness: We prove soundness by transforming the system over a sequence of games. Game $\mathbf{G}_0$ just replicates the soundness security definition. In Game $\mathbf{G}_1$ the CRS is generated using parameter witness A and its null-space, and this can be done efficiently by the challenger as the parameter distribution is efficiently witness samplable. After this transformation, we show that in the case of a certain event, a verifying proof of a non-language member implies breaking the k-linear assumption in group $\mathbb{G}_2$, while in the case of the event not occurring we can apply the Switching Lemma to bound the probability of the adversary winning.

Game $\mathbf{G}_0$**:** This is just the original system. Consider an adversary $\mathcal{A}$ that *wins* if it can produce a "proof" $\vec{\mathbf{p}}$ for which the pairing test $e\left(\left[\ \vec{l} \ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k^2}$ holds and yet the candidate $\vec{l}$ is not in $L_{\mathbf{A}}$. Let the advantage of adversary $\mathcal{A}$ in Game $\mathbf{G}_0$ be $\Delta_{\mathcal{A}}$.

Game $\mathbf{G}_1$: In this game, the challenger efficiently samples $\mathbf{A}$ according to distribution $\mathcal{D}$, along with witness A (since $\mathcal{D}$ is an efficiently witness samplable distribution). If the left most t columns of A are not full ranked, the adversary wins. Otherwise, since A is a $t \times (t+s)$ dimensional rank t matrix, there is a rank s matrix $\left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right]$ of dimension $(t+s)\times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A}\cdot\left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right] = \mathsf{0}^{t\times s}$. In this game, the NIZK CRS is computed as follows: Generate matrix $\mathsf{D}'^{\,t\times k^2}$ with elements randomly chosen from $\mathbb{Z}_q$ and the matrices $\mathsf{R}^{s\times k^2}$ and $\mathsf{B}^{k^2\times k^2}$ as in the real CRS. Implicitly set: $\mathsf{D} = \mathsf{D}' + \mathsf{W}\ \mathsf{R}\ \mathsf{B}^{-1}$. Therefore we have,

$$\mathbf{CRS}_p^{t\times k^2} = \mathbf{A}\cdot\left[\begin{array}{c} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{array}\right] = \mathbf{A}\cdot\left(\left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times k^2} \end{array}\right] + \left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right]\cdot \mathsf{R}\ \mathsf{B}^{-1}\right) = \mathbf{A}\cdot\left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times k^2} \end{array}\right]$$

$$\mathbf{CRS}_v^{(n+k^2)\times k^2} = \left[\begin{array}{c} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{array}\right]\cdot \mathbf{g}_2 = \left[\begin{array}{c} \mathsf{D}'\ \mathsf{B} + \mathsf{W}\ \mathsf{R} \\ \mathsf{R} \\ -\mathsf{B} \end{array}\right]\cdot \mathbf{g}_2$$

Observe that D has identical distribution as in game $\mathbf{G}_0$ and the rest of the computations were same. So Game $\mathbf{G}_1$ is statistically indistinguishable from Game $\mathbf{G}_0$ and the advantage of $\mathcal{A}$ in Game $\mathbf{G}_1$ remains $\Delta_{\mathcal{A}}$.

Suppose that $\mathcal{A}$ wins $\mathbf{G}_1$. Now, let us partition the $\mathbb{Z}_q$ matrix A as $\left[\ \mathsf{A}_0^{t\times t} \mid \mathsf{A}_1^{t\times s}\ \right]$ and the candidate vector $\vec{\boldsymbol{l}}$ as $\left[\ \vec{\boldsymbol{l}}_0^{1\times t} \mid \vec{\boldsymbol{l}}_1^{1\times s}\ \right]$. Note that, since A_0 has rank t, the elements of $\vec{\boldsymbol{l}}_0$ are 'free' elements and $\vec{\boldsymbol{l}}_0$ can be extended to a unique n element vector $\vec{\boldsymbol{l}}\,'$, which is a member of $L_{\mathbf{A}}$. This member vector $\vec{\boldsymbol{l}}\,'$ can be computed as $\vec{\boldsymbol{l}}\,' := \left[\ \vec{\boldsymbol{l}}_0 \mid -\vec{\boldsymbol{l}}_0\cdot\mathsf{W}\ \right]$, where $\mathsf{W} = -\mathsf{A}_0^{-1}\mathsf{A}_1$. The proof of $\vec{\boldsymbol{l}}\,'$ is computed as $\vec{\mathbf{p}}\,' := \vec{\boldsymbol{l}}_0\cdot\mathsf{D}'$. Since $\mathcal{A}$ wins $\mathbf{G}_1$, then $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ passes the verification test, and further by design $(\vec{\boldsymbol{l}}', \vec{\mathbf{p}}')$ passes the verification test. Thus, we obtain: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)\cdot\mathsf{R} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}})\cdot\mathsf{B}$, where $\vec{\boldsymbol{l}}_1' = -\vec{\boldsymbol{l}}_0\cdot\mathsf{W}$.

This gives us a set of equalities, for all $u \in [1,k]$:

$$\sum_{i=1}^{s}(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i})\cdot r_{iu} = (\mathbf{p}'_{k(u-1)+1} - \mathbf{p}_{k(u-1)+1})\cdot b_1 = \cdots = (\mathbf{p}'_{k(u-1)+k} - \mathbf{p}_{k(u-1)+k})\cdot b_k \qquad (17)$$

Also note that since $\vec{\boldsymbol{l}}$ is not in the language, there exists an $i \in [1,s]$, such that $\vec{\boldsymbol{l}}_{1i}' - \vec{\boldsymbol{l}}_{1i} \neq 0$.

Game $\mathbf{G}_2$: Game $\mathbf{G}_2$ is exactly set up as Game $\mathbf{G}_1$, except that we restrict $\mathcal{A}$ to win in Game $\mathbf{G}_2$ only if it wins Game $\mathbf{G}_1$ **and** Event E does not occur, where E is defined as follows:

$$\text{Event } E \equiv \text{ For some } u \in [1,k] : \sum_{i=1}^{s}(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i})\cdot r_{iu} \neq \mathbf{0}_1 \qquad (18)$$

Therefore, we have: $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1] = \Pr[\mathcal{A} \text{ wins } \mathbf{G}_1 \wedge E] + \Pr[\mathcal{A} \text{ wins } \mathbf{G}_2]$. We show now that the first term is upper bounded by $\textsc{adv}(klin)$. To do that we construct a k-linear adversary $\mathcal{B}$ from $\mathcal{A}$ and part of the Game $\mathbf{G}_1$ challenger, such that if $\mathcal{A}$ wins $\mathbf{G}_1$ and Event E occurs, $\mathcal{B}$ is able to win the k-linear challenge.

So suppose, $\mathcal{B}$ is given a k-linear instance $(b_1 \cdot \mathbf{g}_2, ..., b_k \cdot \mathbf{g}_2, \quad \mathbf{g}_2, \quad b_1 s_1 \cdot \mathbf{g}_2, ..., b_k s_k \cdot \mathbf{g}_2, \quad \boldsymbol{\chi})$ in the group $\mathbb{G}_2$, where $\boldsymbol{\chi}$ is either $(s_1 + ... + s_k) \cdot \mathbf{g}_2$ or random. $\mathcal{B}$ then computes the matrix $\mathsf{B} \cdot \mathbf{g}_2$ by setting $(\mathsf{B} \cdot \mathbf{g}_2)_{k(u-1)+v,k(u-1)+v} = b_v \cdot \mathbf{g}_2$ and setting all the non-diagonal elements to $\mathbf{0}_2$. The adversary $\mathcal{B}$ then generates the matrices A, R and D' and computes $\mathbf{A} := \mathsf{A} \cdot \mathbf{g}_1$ and the CRS'es $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ from these matrices and the matrix $\mathsf{B} \cdot \mathbf{g}_2$, similar to Game $\mathbf{G}_1$. After that, adversary $\mathcal{A}$ is given $\mathbf{A}, \mathbf{CRS}_p$ and $\mathbf{CRS}_v$. Adversary $\mathcal{A}$ returns $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$, from which $\mathcal{B}$ computes $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$, identically as in the description of Game $\mathbf{G}_1$. Now we condition on the event that $\mathcal{A}$ wins $\mathbf{G}_1$ and event E occurs. Then, $\sum_{i=1}^{s} (\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu}$ is non-zero for some $u \in [1, k]$, say for $u = w$. This w can be efficiently computed by $\mathcal{B}$ since it has all the r_{iu}'s. Now $\mathcal{B}$ performs the following test:

$$e\left(\sum_{i=1}^{s} (\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iw}, \boldsymbol{\chi}\right) \stackrel{?}{=} \sum_{j=1}^{k} e(\mathbf{p}_{k(w-1)+j}' - \mathbf{p}_{k(w-1)+j}, b_j s_j \cdot \mathbf{g}_2).$$

In the case $\boldsymbol{\chi}$ is equal to $(s_1 + ... + s_k) \cdot \mathbf{g}$, the equality will hold by virtue of Equation 17. In the case $\boldsymbol{\chi}$ is random, the equality will not hold. Thus we have: $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1 \wedge E] \leq \textsc{adv}(klin)$.

Game $\mathbf{G}_3$: We now prepare to employ the **Switching Lemma**. To do that, in Game $\mathbf{G}_3$, we replace R in the verification test by R', which we define as follows: Generate sk elements $\{r_{ij}'\}_{i \in [1,s], j \in [1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Now set:

$$(\mathsf{R}')_{i,k(u-1)+v} = r_{iu}', \text{ with } i \in [1, s], u, v \in [1, k].$$

Note that $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ remain the same as Game $\mathbf{G}_2$, i.e. use R, and it is only the verification test which changes to $e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v'\right) \stackrel{?}{=} \mathbf{0}_T^{1 \times k^2}$ where $\mathbf{CRS}_v'$ uses R' instead of R. We claim now that the probability of $\mathcal{A}$ winning Game $\mathbf{G}_3$ is negligibly close to the probability of $\mathcal{A}$ winning Game $\mathbf{G}_2$. This follows by employing Switching Lemma 2 on a composite adversary built from $\mathcal{A}$ and part of Game $\mathbf{G}_2$ challenger. Details follow.

Let us define the matrices $\hat{\mathbf{R}}^{s \times k}$ and $\hat{\mathbf{R}}'^{\, s \times k}$, which are obtained by removing the repetitive columns in $\mathsf{R} \cdot \mathbf{g}_2$ and $\mathsf{R}' \cdot \mathbf{g}_2$: $\quad (\hat{\mathbf{R}})_{iu} = r_{iu} \cdot \mathbf{g}_2$ and $(\hat{\mathbf{R}}')_{iu} = r_{iu}' \cdot \mathbf{g}_2$, with $i \in [1, s], u \in [1, k]$. Note that all the elements of $\hat{\mathbf{R}}$ and $\hat{\mathbf{R}}'$ are independently random elements from $\mathbb{G}_2$. Now, the condition for ($\mathcal{A}$ wins $\mathbf{G}_2$) can be rewritten as: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \mathbf{0}_1^{1 \times s} \wedge e(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \hat{\mathbf{R}}) = \mathbf{0}_T^{1 \times k}$. Similarly, the condition for ($\mathcal{A}$ wins $\mathbf{G}_3$) can be rewritten as: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \mathbf{0}_1^{1 \times s} \wedge e(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \hat{\mathbf{R}}') = \mathbf{0}_T^{1 \times k}$.

We now construct the composite (switching lemma) adversary $\mathcal{B}$ from $\mathcal{A}$ and mimicking part of the Game $\mathbf{G}_2$ challenger. The switching lemma challenger just generates matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{R}}'$ randomly from $\mathbb{G}_2^{s \times k}$ and gives just $\hat{\mathbf{R}}$ to the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ then generates the matrices A, B and D' and computes $\mathbf{A} := \mathsf{A} \cdot \mathbf{g}_1$ and the CRS'es $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$, from these matrices and the given matrix $\hat{\mathbf{R}}$, just as in Game $\mathbf{G}_1$ (and Game $\mathbf{G}_2$). After that it gives adversary $\mathcal{A}$ the matrices $\mathbf{A}, \mathbf{CRS}_p$ and $\mathbf{CRS}_v$. Adversary $\mathcal{A}$ returns $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$, from which $\mathcal{B}$ computes $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$, identically as in the description of Game $\mathbf{G}_1$. Finally, $\mathcal{B}$ returns $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$ to the switching lemma challenger. At this point, if the switching lemma challenger uses the matrix $\hat{\mathbf{R}}$ for the pairing test, then we have exactly the setting of Game $\mathbf{G}_2$. On the other hand, if it uses the matrix $\hat{\mathbf{R}}'$ for the pairing test, then we have exactly the setting of Game $\mathbf{G}_3$. By Switching Lemma 2 (note C is set to the identity matrix below), we have

$$\Pr[\mathcal{A} \text{ wins } \mathbf{G}_2] = \Pr\left[\begin{array}{c} \hat{\mathsf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathsf{R}}, \mathsf{C}) : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \ \textbf{and} \ \ e\left(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \mathsf{C}\cdot\hat{\mathsf{R}}\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right] \tag{19}$$

$$\leq \Pr\left[\begin{array}{c} \hat{\mathsf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathsf{R}}, \mathsf{C}),\ \hat{\mathsf{R}}' \xleftarrow{\$} \mathbb{G}_2^{s\times k} : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \ \textbf{and} \ \ e\left(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \mathsf{C}\cdot\hat{\mathsf{R}}'\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right] + s \cdot \textsc{adv}(klin) \tag{20}$$

$$= \Pr\left[\mathcal{A} \text{ wins } \mathbf{G}_3\right] + s \cdot \textsc{adv}(klin) \tag{21}$$

Finally, we claim that $\Pr\left[\mathcal{A} \text{ wins } \mathbf{G}_3\right]$ is information-theoretically negligible. We have:

$$\Pr\left[\mathcal{A} \text{ wins } \mathbf{G}_3\right] = \Pr\left[\begin{array}{c} \hat{\mathsf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathsf{R}}, \mathsf{C}),\ \hat{\mathsf{R}}' \xleftarrow{\$} \mathbb{G}_2^{s\times k} : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \ \textbf{and} \ \ \forall u \in [1,k] : \sum_{i=1}^{s} (\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu}' = \mathbf{0}_1 \end{array}\right] \leq 1/|\mathbb{G}_1|$$

The last inequality holds since the r_{iu}''s were chosen after the adversary responded and $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$ is a non-zero vector, i.e., at least one of the quantities $(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i})$'s is non-zero. Therefore, $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_3] \leq 1/|\mathbb{G}_1|$. Combining all results, we have:

$$\Delta_{\mathcal{A}} \leq \textsc{adv}(klin) + s \cdot \textsc{adv}(klin) + 1/|\mathbb{G}_1| = (s+1) \cdot \textsc{adv}(klin) + 1/|\mathbb{G}_1|.$$

□

D Quasi-Adaptive k Element Proofs of Linear Subspaces

We repeat the construction here for ease of reading and provide a detailed proof.

Algorithm K_1: The algorithm K_1 generates the CRS as follows. Let $\mathbf{A}^{t\times n}$ be the parameter supplied to K_1. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns. It generates a matrix $\mathsf{D}^{t\times k}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $\{b_v\}_{v\in[1,k]}$ and k^3 elements $\{t_{uvw}\}_{u,v,w\in[1,k]}$ and sk elements $\{r_{iu}\}_{i\in[1,s],u\in[1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Define matrices $\mathsf{R}^{s\times k}$ and $\mathsf{B}^{k\times k}$ component-wise as follows:

$$(\mathsf{R})_{iw} = \sum_{u=1}^{k}\sum_{v=1}^{k} r_{iu} t_{uvw}, \text{ with } i \in [1,s], w \in [1,k].$$

$$(\mathsf{B})_{vw} = \sum_{u=1}^{k} b_v t_{uvw}, \text{ with } v, w \in [1,k].$$

The common reference string (CRS) has two parts $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ which are to be used by the prover and the verifier respectively.

$$\mathbf{CRS}_p^{t\times k} := \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{array}\right] \qquad \mathbf{CRS}_v^{(n+k)\times k} = \left[\begin{array}{c} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{array}\right] \cdot \mathbf{g}_2$$

Prover P**:** Given candidate $\vec{l} = \vec{x} \cdot \mathbf{A}$ with witness vector $\vec{x}^{1\times t}$, the prover generates the following proof consisting of k elements in $\mathbb{G}_1$: $\quad \vec{\mathbf{p}}^{1\times k} := \vec{x} \cdot \mathbf{CRS}_p$

Verifier V**:** Given candidate $\vec{l}$, and proof $\vec{\mathbf{p}}$, the verifier checks the following (k equations) :

$$e\left(\left[\ \vec{l}\ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}$$

Theorem 13 *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound quasi-adaptive NIZK proof system for linear subspace languages* $\{L_{\mathbf{A}}\}$ *with parameters* $\mathbf{A}$ *sampled from a robust and efficiently witness-samplable distribution* $\mathcal{D}$ *over the associated parameter language* $\mathcal{L}_{par}$*, given any group generation algorithm for which the k-linear assumption holds for group* $\mathbb{G}_2$*.*

Proof: Completeness and Zero-Knowledge are are same as the previous QA-NIZK.

Soundness: We prove soundness by transforming the system over a sequence of games. Game $\mathbf{G}_0$ just replicates the soundness security definition. In Game $\mathbf{G}_1$ the CRS is generated using parameter witness A and its null-space, and this can be done efficiently by the challenger as the parameter distribution is efficiently witness samplable. After this transformation, we show that in the case of a certain event, a verifying proof of a non-language member implies breaking the k-linear assumption in group $\mathbb{G}_2$, while in the case of the event not occurring we can apply the Switching Lemma to bound the probability of the adversary winning.

Game $\mathbf{G}_0$: This is just the original system. Consider an adversary $\mathcal{A}$ which wins if it can produce a "proof" $\vec{\mathbf{p}}$ for which the pairing test $e\left(\left[\ \vec{l}\ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}$ holds and yet the candidate $\vec{l}$ is not in $L_{\mathbf{A}}$. Let the advantage of adversary $\mathcal{A}$ in Game $\mathbf{G}_0$ be $\Delta_{\mathcal{A}}$.

Game $\mathbf{G}_1$: In this game, the discrete logarithms of the defining constants of the language L are given to the CRS generator, or in other words A is given (by the efficiently witness samplable property). Since A is a $t \times (t+s)$ dimensional rank t matrix, there is a rank s matrix $\begin{bmatrix} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix}$ of dimension $(t+s) \times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A} \cdot \begin{bmatrix} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix} = 0^{t\times s}$. In this game, the NIZK CRS is computed as follows: Generate matrix $\mathsf{D}'^{\ t\times k}$ with elements randomly chosen from $\mathbb{Z}_q$ and the matrices $\mathsf{R}^{s\times k}$ and $\mathsf{B}^{k\times k}$ as in the real CRS. Implicitly set: $\mathsf{D} = \mathsf{D}' + \mathsf{W}\ \mathsf{R}\ \mathsf{B}^{-1}$. Therefore we have,

$$\mathbf{CRS}_p^{t\times k} = \mathbf{A} \cdot \begin{bmatrix} \mathsf{D} \\ \mathsf{R}\ \mathsf{B}^{-1} \end{bmatrix} = \mathbf{A} \cdot \left(\begin{bmatrix} \mathsf{D}' \\ 0^{s\times k} \end{bmatrix} + \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathsf{R}\ \mathsf{B}^{-1} \right) = \mathbf{A} \cdot \begin{bmatrix} \mathsf{D}' \\ 0^{s\times k} \end{bmatrix}$$

$$\mathbf{CRS}_v^{(n+k)\times k} = \begin{bmatrix} \mathsf{D}\ \mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{bmatrix} \cdot \mathbf{g}_2 = \begin{bmatrix} \mathsf{D}'\ \mathsf{B} + \mathsf{W}\ \mathsf{R} \\ \mathsf{R} \\ -\mathsf{B} \end{bmatrix} \cdot \mathbf{g}_2$$

Observe that D has identical distribution as in game $\mathbf{G}_0$ and the rest of the computations were same. So Game $\mathbf{G}_1$ is statistically indistinguishable from Game $\mathbf{G}_0$ and the advantage of $\mathcal{A}$ in Game $\mathbf{G}_1$ remains $\Delta_{\mathcal{A}}$.

Suppose that $\mathcal{A}$ wins $\mathbf{G}_1$. Now, let us partition the $\mathbb{Z}_q$ matrix A as $\left[\ \mathsf{A}_0^{t\times t}\ \middle|\ \mathsf{A}_1^{t\times s}\ \right]$ and the candidate vector $\vec{l}$ as $\left[\ \vec{l}_0^{1\times t}\ \middle|\ \vec{l}_1^{1\times s}\ \right]$. Note that, since A_0 has rank t, the elements of $\vec{l}_0$ are 'free' elements and $\vec{\boldsymbol{l}}_0$ can be extended to a unique n element vector $\vec{\boldsymbol{l}}\,'$, which is a member of $L_{\mathbf{A}}$. This member vector $\vec{\boldsymbol{l}}\,'$ can be computed as $\vec{\boldsymbol{l}}\,' := \left[\ \vec{\boldsymbol{l}}_0 \mid -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}\ \right]$, where $\mathsf{W} = -\mathsf{A}_0^{-1}\mathsf{A}_1$. The proof of $\vec{\boldsymbol{l}}\,'$ is computed as $\vec{\mathbf{p}}\,' := \vec{\boldsymbol{l}}_0 \cdot \mathsf{D}'$. Since $\mathcal{A}$ wins $\mathbf{G}_1$, then $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ passes the verification test, and further by design $(\vec{\boldsymbol{l}}', \vec{\mathbf{p}}')$ passes the verification test. Thus, we obtain: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \cdot \mathsf{R} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}}) \cdot \mathsf{B}$, where $\vec{\boldsymbol{l}}_1' = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$.

This gives us a set of equalities:

$$\text{For all } w \in [1,k] : \sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot \left(\sum_{u=1}^{k} \sum_{v=1}^{k} r_{iu} t_{uvw} \right) \right] - \sum_{v=1}^{k} \left[(\mathbf{p}_v' - \mathbf{p}_v) \cdot \left(\sum_{u=1}^{k} b_v t_{uvw} \right) \right] = \mathbf{0}_1 \tag{22}$$

Rearranging, we get:

$$\text{For all } w \in [1,k] : \sum_{u=1}^{k} \sum_{v=1}^{k} \left[t_{uvw} \left(\sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu} \right] - (\mathbf{p}_v' - \mathbf{p}_v) \cdot b_v \right) \right] = \mathbf{0}_1 \tag{23}$$

Also note that since $\vec{\boldsymbol{l}}$ is not in the language, there exists an $i \in [1,s]$, such that $\vec{\boldsymbol{l}}_{1i}' - \vec{\boldsymbol{l}}_{1i} \neq 0$.

Game $\mathbf{G}_2$: Let us now define the following event F:

$$F \quad \equiv \quad \text{For some } u,v \in [1,k] : \left(\sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu} \right] - (\mathbf{p}_v' - \mathbf{p}_v) \cdot b_v \right) \neq \mathbf{0}_1$$

We define that $\mathcal{A}$ wins in Game $\mathbf{G}_2$ if $\mathcal{A}$ wins in Game $\mathbf{G}_1$ *and* event F does not occur. Now we show that $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_2]$ is negligibly close to $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1]$. We establish this by showing that $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1 \wedge F]$ is negligible.

We now prepare to employ the **Switching Lemma**. To do that, Generate k^3 elements $\{t'_{uvw}\}_{u,v,w \in [1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Now we replace T in the verification test by T' in a Game ${\mathbf{G}_1}'$, related to Game $\mathbf{G}_1$, which we define as follows:

$$(\mathsf{T})_{k(u-1)+v,w} = t_{uvw}, \ \text{with } u,v,w \in [1,k].$$
$$(\mathsf{T}')_{k(u-1)+v,w} = t'_{uvw}, \ \text{with } u,v,w \in [1,k].$$

Note that $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ remain the same as Game $\mathbf{G}_1$, i.e. use T, and it is only the verification test which changes to $e\left(\left[\ \vec{\boldsymbol{l}} \mid \vec{\mathbf{p}}\ \right], \mathbf{CRS}_v'\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}$ where $\mathbf{CRS}_v'$ uses T' instead of T. We claim now that the probability of $\mathcal{A}$ winning Game ${\mathbf{G}_1}'$ is negligibly close to the probability of $\mathcal{A}$ winning Game $\mathbf{G}_1$ and the event F occurring.

The claim is established by constructing a switching lemma adversary, such that winning Game $\mathbf{G}_1$ corresponds to one scenario and winning Game ${\mathbf{G}_1}'$ corresponds to the other scenario. Once that is done, the switching lemma lets us reason that the probabilities of winning are negligibly close under the k-linear assumption.

We construct a switching lemma attacker $\mathcal{B}$ from $\mathcal{A}$ and part of the Game $\mathbf{G}_1$ challenger. The switching lemma challenger just generates matrices $\mathbf{T}$ and $\mathbf{T}'$ randomly from $\mathbb{G}_2^{k^2\times k}$ and gives just $\mathbf{T}$ to the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ then generates the $\mathsf{A}, \{r_{iu}\}_{i\in[1,s],u\in[1,k]}, \{b_v\}_{v\in[1,k]}$ and D' and computes $\mathbf{A} := \mathsf{A} \cdot \mathbf{g}_1$ and the CRS'es $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$, from these matrices and the given matrix $\mathbf{T}$, just as in Game $\mathbf{G}_1$. After that, adversary $\mathcal{A}$ is given $\mathbf{A}, \mathbf{CRS}_p$ and $\mathbf{CRS}_v$. Adversary $\mathcal{A}$ returns $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$, from which $\mathcal{B}$ computes $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$, identically as in the description of Game $\mathbf{G}_1$. Then $\mathcal{B}$ computes the vector $\vec{\mathbf{f}}^{1\times k^2}$ with the $[k(u-1)+v]$-th component equal to:

$$\mathbf{f}_{k(u-1)+v} = \sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu} \right] - (\mathbf{p}_v' - \mathbf{p}_v) \cdot b_v$$

Finally, $\mathcal{B}$ returns $\vec{\mathbf{f}}$ to the switching lemma challenger. At this point, if the switching lemma challenger uses the matrix $\mathbf{T}$ for the pairing test, then we exactly have the setting of Game $\mathbf{G}_1$. On the other hand, if it uses the matrix $\mathbf{T}'$ for the pairing test, then we exactly have the setting of Game ${\mathbf{G}_1}'$.

$$\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1 \wedge F] = \Pr\left[\begin{array}{c} \mathbf{T} \xleftarrow{\$} \mathbb{G}_2^{k^2\times k},\ \mathsf{C} := \mathrm{I}^{k\times k},\ \vec{\mathbf{f}} \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \mathbf{T}, \mathsf{C}) : \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times k^2} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\mathbf{T}\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array} \right] \tag{24}$$

$$\leq \Pr\left[\begin{array}{c} \mathbf{T} \xleftarrow{\$} \mathbb{G}_2^{k^2\times k},\ \mathsf{C} := \mathrm{I}^{k\times k},\ \vec{\mathbf{f}} \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \mathbf{T}, \mathsf{C}),\ \mathbf{T}' \xleftarrow{\$} \mathbb{G}_2^{k^2\times k} : \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times k^2} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\mathbf{T}'\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array} \right] + k^2 \cdot \textsc{adv}(klin) \tag{25}$$

$$= \Pr\left[\mathcal{A} \text{ wins } {\mathbf{G}_1}'\right] + k^2 \cdot \textsc{adv}(klin) \tag{26}$$

Finally, we claim that $\Pr\left[\mathcal{A} \text{ wins } {\mathbf{G}_1}'\right]$ is information-theoretically negligible. We have:

$$\Pr\left[\mathcal{A} \text{ wins } {\mathbf{G}_1}'\right] = \Pr\left[\begin{array}{c} \mathbf{T} \xleftarrow{\$} \mathbb{G}_2^{k^2\times k},\ \mathsf{C} := \mathrm{I}^{k\times k},\ \vec{\mathbf{f}} \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \mathbf{T}, \mathsf{C}),\ \mathbf{T}' \xleftarrow{\$} \mathbb{G}_2^{k^2\times k} : \\ \vec{\mathbf{f}} \neq \vec{\mathbf{0}}_1^{1\times k^2} \ \textbf{and} \ \ e\left(\vec{\mathbf{f}}, \mathsf{C}\cdot\mathbf{T}'\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array} \right] \leq 1/|\mathbb{G}_1|$$

The last inequality holds since the t'_{uvw}'s were chosen after the adversary responded and $\vec{\mathbf{f}}$ is a non-zero vector, i.e., at least one of the quantities $\mathbf{f}_i$'s is non-zero. Therefore, $\Pr[\mathcal{A} \text{ wins } {\mathbf{G}_1}'] \leq 1/|\mathbb{G}_1|$.

Game $\mathbf{G}_3$: Game $\mathbf{G}_3$ is exactly set up as Game $\mathbf{G}_2$, except that we restrict $\mathcal{A}$ to only win in Game $\mathbf{G}_3$ if it wins Game $\mathbf{G}_1$ **and** Event E does not occur, where E is defined as follows:

$$\text{Event } E \equiv \text{ For some } u \in [1,k] : \sum_{i=1}^{s} (\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu} \neq \mathbf{0}_1 \tag{27}$$

Recall that the event F defined in Game $\mathbf{G}_1$ does not occur in both Games $\mathbf{G}_2$ and $\mathbf{G}_3$.

Therefore, we have: $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_2] = \Pr[\mathcal{A} \text{ wins } \mathbf{G}_2 \wedge E] + \Pr[\mathcal{A} \text{ wins } \mathbf{G}_3]$. We show now that the first term is upper bounded by $\textsc{adv}(klin)$. To do that we construct a k-linear adversary $\mathcal{B}$ from $\mathcal{A}$ and part of the Game $\mathbf{G}_2$ challenger, such that if $\mathcal{A}$ wins $\mathbf{G}_2$ and Event E occurs, $\mathcal{B}$ is able to win the k-linear challenge.

So suppose, $\mathcal{B}$ is given a k-linear instance $(b_1 \cdot \mathbf{g}_2, ..., b_k \cdot \mathbf{g}_2, \quad \mathbf{g}_2, \quad b_1 s_1 \cdot \mathbf{g}_2, ..., b_k s_k \cdot \mathbf{g}_2, \quad \boldsymbol{\chi})$ in the group $\mathbb{G}_2$, where $\boldsymbol{\chi}$ is either $(s_1 + ... + s_k) \cdot \mathbf{g}_2$ or random. $\mathcal{B}$ then computes the matrix $\mathsf{B} \cdot \mathbf{g}_2$ by choosing $\{t_{uvw}\}_{u,v,w\in[1,k]}$ randomly from $\mathbb{Z}_q$ and setting $(\mathsf{B} \cdot \mathbf{g}_2)_{vw} = \sum_{u=1}^{k} [t_{uvw}(b_v \cdot \mathbf{g}_2)]$. The adversary $\mathcal{B}$ then generates the matrices A, R and D' and computes $\mathbf{A} := \mathsf{A} \cdot \mathbf{g}_1$ and the CRS'es $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$, from these matrices and the matrix $\mathsf{B} \cdot \mathbf{g}_2$, similar to Game $\mathbf{G}_1$. After that, adversary $\mathcal{A}$ is given $\mathbf{A}, \mathbf{CRS}_p$ and $\mathbf{CRS}_v$. Adversary $\mathcal{A}$ returns $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$, from which $\mathcal{B}$ computes

$(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$, identically as in the description of Game $\mathbf{G}_1$. Now, since $\mathcal{A}$ wins $\mathbf{G}_1$ and event E occurs, $\sum_{i=1}^{s}(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu}$ is non-zero for some $u \in [1, k]$, say for $u = z$, which can be computed efficiently. Now $\mathcal{B}$ performs the following test:

$$e\left(\sum_{i=1}^{s}(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iz}, \boldsymbol{\chi}\right) \stackrel{?}{=} \sum_{j=1}^{k} e(\mathbf{p}_j' - \mathbf{p}_j, b_j s_j \cdot \mathbf{g}_2).$$

In the case $\boldsymbol{\chi}$ is equal to $(s_1 + ... + s_k) \cdot \mathbf{g}$, the equality will hold by virtue of Equation 17. In the case $\boldsymbol{\chi}$ is random, the equality will not hold. Thus we have: $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_1 \wedge E] \leq \text{ADV}(klin)$.

Game $\mathbf{G}_4$: We now prepare to employ the **Switching Lemma**. To do that, in Game $\mathbf{G}_4$, we replace r_{iu}s in the verification test by r_{iu}'s. Note that $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ remain the same as Game $\mathbf{G}_3$, i.e. use r_{iu}s, and it is only the verification test which changes to $e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v'\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}$ where $\mathbf{CRS}_v'$ uses r_{iu}'s instead of r_{iu}s. We claim now that the probability of $\mathcal{A}$ winning Game $\mathbf{G}_4$ is negligibly close to the probability of $\mathcal{A}$ winning Game $\mathbf{G}_3$.

The claim is established by constructing a switching lemma adversary, such that winning Game $\mathbf{G}_3$ corresponds to one scenario and winning Game $\mathbf{G}_4$ corresponds to the other scenario. Once that is done, the switching lemma lets us reason that the probabilities of winning are negligibly close under the k-linear assumption.

Let us define the matrices $\hat{\mathbf{R}}^{s\times k}$ and $\hat{\mathbf{R}}'^{\,s\times k}$ component-wise as follows:

$$(\hat{\mathbf{R}})_{iu} = r_{iu} \cdot \mathbf{g}_2 \text{ and } (\hat{\mathbf{R}}')_{iu} = r_{iu}' \cdot \mathbf{g}_2, \text{ with } i \in [1, s], u \in [1, k].$$

Note that all the elements of $\hat{\mathbf{R}}$ and $\hat{\mathbf{R}}'$ are independently random elements from $\mathbb{G}_2$. Now, the condition for ($\mathcal{A}$ wins $\mathbf{G}_3$) can be rewritten as: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \mathbf{0}_1^{1\times s} \wedge e(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \hat{\mathbf{R}}) = \mathbf{0}_T^{1\times k}$. Similarly, the condition for ($\mathcal{A}$ wins $\mathbf{G}_4$) can be rewritten as: $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \mathbf{0}_1^{1\times s} \wedge e(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \hat{\mathbf{R}}') = \mathbf{0}_T^{1\times k}$.

We construct a switching lemma attacker $\mathcal{B}$ from $\mathcal{A}$ and part of the Game $\mathbf{G}_3$ challenger. The switching lemma challenger just generates matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{R}}'$ randomly from $\mathbb{G}_2^{s\times k}$ and gives just $\hat{\mathbf{R}}$ to the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ then generates $\mathsf{A}, \{t_{uvw}\}_{u,v,w\in[1,k]}$ and D' and computes $\mathbf{A} := \mathsf{A} \cdot \mathbf{g}_1$ and the matrix B and the CRS'es $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$, from these matrices and the given matrix $\hat{\mathbf{R}}$, just as in Game $\mathbf{G}_1$. After that, adversary $\mathcal{A}$ is given $\mathbf{A}, \mathbf{CRS}_p$ and $\mathbf{CRS}_v$. Adversary $\mathcal{A}$ returns $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$, from which $\mathcal{B}$ computes $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$, identically as in the description of Game $\mathbf{G}_1$. Finally, $\mathcal{B}$ returns $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$ to the switching lemma challenger. At this point, if the switching lemma challenger uses the matrix $\hat{\mathbf{R}}$ for the pairing test, then we exactly have the setting of Game $\mathbf{G}_3$. On the other hand, if it uses the matrix $\hat{\mathbf{R}}'$ for the pairing test, then we exactly have the setting of Game $\mathbf{G}_4$.

$$\Pr[\mathcal{A} \text{ wins } \mathbf{G}_3] = \Pr\left[\begin{array}{c} \hat{\mathbf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathbf{R}}, \mathsf{C}) : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \textbf{ and } e\left(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \mathsf{C} \cdot \hat{\mathbf{R}}\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right] \tag{28}$$

$$\leq \Pr\left[\begin{array}{c} \hat{\mathbf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathbf{R}}, \mathsf{C}),\ \hat{\mathbf{R}}' \xleftarrow{\$} \mathbb{G}_2^{s\times k} : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \textbf{ and } e\left(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1, \mathsf{C} \cdot \hat{\mathbf{R}}'\right) = \vec{\mathbf{0}}_T^{1\times k} \end{array}\right] + s \cdot \text{ADV}(klin) \tag{29}$$

$$= \Pr\left[\mathcal{A} \text{ wins } \mathbf{G}_4\right] + s \cdot \text{ADV}(klin) \tag{30}$$

Finally, we claim that $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_4]$ is information-theoretically negligible. We have:

$$\Pr[\mathcal{A} \text{ wins } \mathbf{G}_4] = \Pr\left[\begin{array}{c} \hat{\mathsf{R}} \xleftarrow{\$} \mathbb{G}_2^{s\times k},\ \mathsf{C} := \mathrm{I}^{s\times s},\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \leftarrow \mathcal{B}(\mathbf{g}_1, \mathbf{g}_2, \hat{\mathsf{R}}, \mathsf{C}),\ \hat{\mathsf{R}}' \xleftarrow{\$} \mathbb{G}_2^{s\times k} : \\ (\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \neq \vec{\mathbf{0}}_1^{1\times s} \textbf{ and } \forall u \in [1,k] : \sum_{i=1}^{s} (\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu}' = \mathbf{0}_1 \end{array}\right] \leq 1/|\mathbb{G}_1|$$

The last inequality holds since the r_{iu}''s were chosen after the adversary responded and $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1)$ is a non-zero vector, i.e., at least one of the quantities $(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i})$'s is non-zero. Therefore, $\Pr[\mathcal{A} \text{ wins } \mathbf{G}_4] \leq 1/|\mathbb{G}_1|$. Combining all results, we have:

$$\Delta_{\mathcal{A}} \leq k^2 \textsc{adv}(klin) + \textsc{adv}(klin) + s \cdot \textsc{adv}(klin) + O(1)/q = (k^2 + s + 1) \cdot \textsc{adv}(klin) + O(1)/q.$$

□

E More Optimized k Element QA-NIZK Proofs

In this construction the **Algorithm** K_1 generates the CRS as follows. It generates a matrix $\mathsf{D}^{t\times k}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $\{b_v\}_{v\in[1,k]}$ and $k(2k-1)$ elements $\{t_{uw}\}_{u\in[1,2k-1],w\in[1,k]}$ and sk elements $\{r_{iu}\}_{i\in[1,s],u\in[1,k]}$ all chosen randomly from $\mathbb{Z}_q$. Define matrices $\mathsf{R}^{s\times k}$ and $\mathsf{B}^{k\times k}$ component-wise as follows:

$$(\mathsf{R})_{iw} = \sum_{u=1}^{k} r_{iu} t_{uw}, \text{ with } i \in [1,s], w \in [1,k].$$

$$(\mathsf{B})_{vw} = \begin{cases} b_1(t_{1w} - t_{k+1,w}) & \text{if } v = 1, w \in [1,k] \\ b_v(t_{vw} - t_{k+v,w} + t_{k+v-1,w}) & \text{if } v \in [2, k-1], w \in [1,k] \\ b_k(t_{kw} + t_{2k-1,w}) & \text{if } v = k, w \in [1,k] \end{cases}$$

The construction of $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ remain algebraically the same. The prover and verifier also retain the same algebraic form. The set of equalities for this construction corresponding to the equation $(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \cdot \mathsf{R} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}}) \cdot \mathsf{B}$, is for all $w \in [1,k]$:

$$\sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot \left(\sum_{u=1}^{k} r_{iu} t_{uw}\right)\right] - \sum_{v=2}^{k-1} \left[(\mathbf{p}_v' - \mathbf{p}_v) \cdot b_v \left(t_{vw} - t_{k+v,w} + t_{k+v-1,w}\right)\right]$$
$$- (\mathbf{p}_1' - \mathbf{p}_1) \cdot b_1(t_{1w} - t_{k+1,w}) - (\mathbf{p}_k' - \mathbf{p}_k) \cdot b_k(t_{kw} + t_{2k-1,w}) = \mathbf{0}_1 \tag{31}$$

Rearranging, we get for all $w \in [1,k]$:

$$\sum_{u=1}^{k} \left[t_{uw} \left(\sum_{i=1}^{s} \left[(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}) \cdot r_{iu}\right] - (\mathbf{p}_u' - \mathbf{p}_u) \cdot b_u\right)\right]$$
$$+ \sum_{u=1}^{k-1} t_{u+k,w} \left[(\mathbf{p}_u' - \mathbf{p}_u) \cdot b_u - (\mathbf{p}_{u+1}' - \mathbf{p}_{u+1}) \cdot b_{u+1}\right] = \mathbf{0}_1 \tag{32}$$

Now, using the Switching Lemma and after applying information theoretic arguments, we transition to a game where the adversary wins if it wins the original game and the following event occurs:

$$\text{For all } u \in [1,k] : \sum_{i=1}^{s} (\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}) \cdot r_{iu} = (\mathbf{p}'_1 - \mathbf{p}_1) \cdot b_1 = \cdots = (\mathbf{p}'_k - \mathbf{p}_k) \cdot b_k \tag{33}$$

After this point, the proof is analogous to the previous QA-NIZK construction. Advantage against soundness is now upper bounded by $(2k + s) \cdot \text{ADV}(klin) + O(1)/q$.

F Proof of Groth-Sahai Aggregation

Proof: [of Theorem 4] Completeness of the above system is obvious by design. In the quasi-adaptive setting the zero-knowledge simulation is required to be uniform, i.e. a single efficient TM is required to simulate the CRS given the language parameters. In the above system, the vectors of randomness $\vec{\rho}$ and $\vec{\psi}$ are chosen by the CRS simulator randomly as well. Then, given the language parameters, it can generate the CRS using $\vec{\rho}$ and $\vec{\psi}$. The zero-knowledge proof simulation is similar to the Groth-Sahai zero-knowledge proof simulation and we skip the details.

Focusing on the soundness proof, we define a sequence of games, starting with game $\mathbf{G}_0$ which is just the (soundness) security definition game. The Adversary wins $\mathbf{G}_0$ if it can produce $\langle \mathbf{t}_1^j \rangle_{j \in [1,k]}$, commitments to $\vec{y}$ and $\vec{\mathbf{x}}$, as well as proofs $\vec{\boldsymbol{\pi}}_\rho$ and $\vec{\boldsymbol{\pi}}_\psi$, such that $\langle \mathbf{t}_1^j \rangle_j$ do not satisfy the above equations (7) and yet the verification tests (8) and (9) pass.

In game $\mathbf{G}_1$, the challenger (efficiently) samples $\vec{\mathbf{a}}$ along with witnesses $\vec{a}$ (s.t. $\vec{\mathbf{a}} = \vec{a} \cdot \mathbf{g}$). The component $\sum_j \rho_j \cdot \iota_2(\vec{\mathbf{a}}^j)$ of the CRS is now generated as $\sum_j \rho_j \cdot \iota_1(\vec{a}^j)$. The probability of the adversary winning $\mathbf{G}_1$ remains the same as winning $\mathbf{G}_0$.

In game $\mathbf{G}_2$, the challenger generates $\sum_j \rho_j \cdot \iota_1(\vec{a}^j)$ as $\sum_j \iota_2(\rho_j \cdot \mathbf{g}) \cdot \vec{a}^j$, and $\sum_j \rho_j \cdot \iota_1(\vec{b}^j)$ as $\sum_j \iota_1(\rho_j) \cdot \vec{b}^j$ (and similarly for ψ terms). Note that each of $\vec{a}^j$ is a vector of length n, and hence the first term is a length n vector of $\mathbb{G}^3$ elements, and similarly the second term is a length m vector of $\mathbb{G}^3$ elements. Similar change occurs in the verification tests (8) and (9). Again, the probability of the adversary winning $\mathbf{G}_2$ is same as winning $\mathbf{G}_1$.

Applying projection p_T to both sides of these versions of (8) and (9), and using the commutativity properties mentioned above, we get that if the adversary wins $\mathbf{G}_2$ then

$$(\sum_j \rho_j \cdot \vec{b}^j) \cdot p_2(\vec{c_2}) + p_1(\vec{c_1}) \cdot (\sum_j \rho_j \mathbf{g} \cdot \vec{\mathrm{a}}^j) = \sum_j (\rho_j \cdot \mathbf{t}_1^j) \tag{34}$$

$$(\sum_j \psi_j \cdot \vec{b}^j) \cdot p_2(\vec{c_2}) + p_1(\vec{c_1}) \cdot (\sum_j \psi_j \mathbf{g} \cdot \vec{\mathrm{a}}^j) = \sum_j (\psi_j \cdot \mathbf{t}_1^j) \tag{35}$$

Thus, the probability of adversary winning $\mathbf{G}_2$ is at most the probability that it produces $\langle \mathbf{t}_1^j \rangle_j$, $\vec{c_1}$, $\vec{c_2}$ (not all zero, otherwise they are in the language), such that Equations (34) and (35) hold.

In game $\mathbf{G}_3$, the challenger generates the CRS as in $\mathbf{G}_2$ but instead of a public verification of the proofs and commitments, it does the following: It first generates fresh random vectors $\vec{\rho}'$ and $\vec{\psi}'$, and performs the following bilinear pairing test (using trapdoors α and β to efficiently compute

p_2):

$$\sum_j (\rho'_j \mathbf{g}) \otimes ((\vec{b}^j)^\top p_2(\vec{c_2}) + (\vec{a}^j)^\top p_2(\vec{c_1}) - \mathbf{t}_1^j) = \mathbf{0}_T \tag{36}$$

$$\sum_j (\psi'_j \mathbf{g}) \otimes ((\vec{b}^j)^\top p_2(\vec{c_2}) + (\vec{a}^j)^\top p_2(\vec{c_1}) - \mathbf{t}_1^j) = \mathbf{0}_T \tag{37}$$

The adversary wins game $\mathbf{G}_3$ if the k quantities $\langle (\vec{b}^j)^\top p_2(\vec{c_2}) + (\vec{a}^j)^\top p_2(\vec{c_1}) - \mathbf{t}_1^j \rangle$ are not all zero and the above test passes. The k quantities above being not all zero is equivalent to $\langle \mathbf{t}_1^j \rangle$ not being in the language. Note $p_1(\vec{c_1})$ has been replaced by $p_2(\vec{c_1})$. While map p_2 in Groth-Sahai system is an efficient map (given trapdoors α and β), the map p_1 is not efficient. However, it is the case that $p_1(\mathbf{f}) \cdot \mathbf{g} = p_2(\mathbf{f})$ for any $\mathbf{f}$ in $\mathbb{G}^3$. Thus, the equations (34) and (35) hold iff the tests (36) and (37) hold with $\vec{\rho'}$ and $\vec{\psi'}$ replaced with original $\vec{\rho}$ and $\vec{\psi}$.

If the probability of adversary winning game $\mathbf{G}_0$ is Δ, then by the Switching Lemma Δ is upper bounded by the probability of adversary winning $\mathbf{G}_3$ plus $(k+1) \cdot \text{ADV}(\text{DLIN})$. Now, the probability of adversary winning game $\mathbf{G}_3$ is at most $1/|q|^2$, and that completes the proof. □

G QA-NIZK Extension with Tags

In this section we show how the system of Section 4 can be extended to include tags mirroring [JR13]. The tags are elements of $\mathbb{Z}_q$, are included as part of the proof and are used as part of the defining equations of the language.

While our system works for any number of components in the tuple (except the first t) being dependent on any number of tags, to simplify the presentation we will focus on only one dependent element and only one tag. Also for simplicity, we will assume that this element is an affine function of the tag. These languages can be characterized as

$$L_{\mathbf{A},\vec{\mathbf{a}}_1,\vec{\mathbf{a}}_2} = \left\{ \left\langle \vec{x} \cdot \left[\begin{array}{c|c} \mathbf{A} & (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) \end{array} \right], \text{TAG} \right\rangle \mid \vec{x} \in {\mathbb{Z}_q}^t, \text{TAG} \in \mathbb{Z}_q \right\}$$

where $\mathbf{A}^{t \times (n-1)}, \vec{\mathbf{a}}_1^{1 \times t}$ and $\vec{\mathbf{a}}_2^{1 \times t}$ are parameters of the language. A distribution is still called robust (as in Section 4) if with overwhelming probability the first t columns of $\mathbf{A}$ are full-ranked.

Algorithm K_1: The CRS is generated as:

$$\mathbf{CRS}_{p,0}^{t \times k} := \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_1^\top \end{array} \right] \cdot \left[\begin{array}{c} \mathsf{D}_1 \\ \mathsf{R}\ \mathsf{B}^{-1} \end{array} \right] \qquad \mathbf{CRS}_{p,1}^{t \times k} := \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_2^\top \end{array} \right] \cdot \left[\begin{array}{c} \mathsf{D}_2 \\ \mathsf{R}\ \mathsf{B}^{-1} \end{array} \right]$$

$$\mathbf{CRS}_{v,0}^{(n+k) \times k} := \left[\begin{array}{c} \mathsf{D}_1\mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{array} \right] \cdot \mathbf{g}_2 \qquad \mathbf{CRS}_{v,1}^{(n+k) \times k} := \left[\begin{array}{c} \mathsf{D}_2\mathsf{B} \\ 0^{s \times k} \\ 0^{k \times k} \end{array} \right] \cdot \mathbf{g}_2$$

where D_1 and D_2 are matrices of order $t \times k$ with elements randomly chosen from $\mathbb{Z}_q$. The matrices $\mathsf{R}^{s \times k}$ and $\mathsf{B}^{k \times k}$ are generated just as in Section 4.

Prover: Let $\vec{\boldsymbol{l}} \stackrel{\text{def}}{=} \vec{x} \cdot \left[\begin{array}{c|c} \mathbf{A} & (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) \end{array} \right]$. The prover generates the following proof:

$$\vec{\mathbf{p}}^{1 \times k} := \vec{x} \cdot (\mathbf{CRS}_{p,0} + \text{TAG} \cdot \mathbf{CRS}_{p,1})$$

Verifier: Given a proof $\vec{\mathbf{p}}$ for candidate $\vec{\boldsymbol{l}}$ the verifier checks the following:

$$e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_{v,0} + \text{TAG} \cdot \mathbf{CRS}_{v,1}\right) \stackrel{?}{=} \mathbf{0}_T^{1\times k}$$

The size of the proof is k elements in the group $\mathbb{G}_1$. The proof of completeness, soundness and zero-knowledge for this quasi-adaptive system is similar to the QA-NIZKs of Section 4 and a proof sketch follows.

Theorem 14 *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound quasi-adaptive NIZK proof system for the tag-based linear subspace languages* $L_{\mathbf{A},\vec{\mathbf{a}}_1,\vec{\mathbf{a}}_2}$ *with parameters* $\mathbf{A}, \vec{\mathbf{a}}_1, \vec{\mathbf{a}}_2$ *sampled from a robust and efficiently witness-samplable distribution* $\mathcal{D}$*, given any group generation algorithm for which the k-linear assumption holds for group* $\mathbb{G}_2$*.*

Proof:

Completeness: We have,

$$\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right] = \left[\begin{array}{c|c|c} \vec{\mathrm{x}} \cdot \mathbf{A} & \vec{\mathrm{x}} \cdot (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) & \vec{\mathrm{x}} \cdot (\mathbf{A} \cdot \mathsf{D}_1 + \mathbf{A} \cdot \text{TAG} \cdot \mathsf{D}_2 + (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) \cdot \mathsf{R}\ \mathsf{B}^{-1}) \end{array}\right]$$

and

$$\mathbf{CRS}_{v,0} + \text{TAG} \cdot \mathbf{CRS}_{v,1} = \begin{bmatrix} (\mathsf{D}_1 + \text{TAG} \cdot \mathsf{D}_2)\mathsf{B} \\ \mathsf{R} \\ -\mathsf{B} \end{bmatrix} \cdot \mathbf{g}_2$$

Therefore,

$$\begin{aligned} & e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_{v,0} + \text{TAG} \cdot \mathbf{CRS}_{v,1}\right) \\ = \quad & e\left(\left(\begin{array}{c} \vec{\mathrm{x}} \cdot \mathbf{A} \cdot (\mathsf{D}_1 + \text{TAG} \cdot \mathsf{D}_2)\mathsf{B} + \\ \vec{\mathrm{x}} \cdot (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) \cdot \mathsf{R} - \\ \vec{\mathrm{x}} \cdot (\mathbf{A} \cdot \mathsf{D}_1 + \mathbf{A} \cdot \text{TAG} \cdot \mathsf{D}_2 + (\vec{\mathbf{a}}_1^\top + \text{TAG} \cdot \vec{\mathbf{a}}_2^\top) \cdot \mathsf{R}\ \mathsf{B}^{-1}) \cdot \mathsf{B} \end{array}\right), \mathbf{g}_2\right) = \mathbf{0}_T^{1\times k} \end{aligned}$$

Zero Knowledge: This is straight-forward with the simulator being given trapdoors $\mathsf{D}_1, \mathsf{D}_2$ and $\mathsf{R}\ \mathsf{B}^{-1}$.

Soundness: As in the proof of Theorem 3, we compute the CRS's in game $\mathbf{G}_1$ as follows. Let $\mathbf{A} = \mathsf{A} \cdot \mathbf{g}_1, \vec{\mathbf{a}}_1 = \vec{\mathrm{a}}_1 \cdot \mathbf{g}_1$ and $\vec{\mathbf{a}}_2 = \vec{\mathrm{a}}_2 \cdot \mathbf{g}_1$. Further, let $\begin{bmatrix} \mathsf{W}_1^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix}$ be the null-space of $\left[\begin{array}{c|c} \mathsf{A} & \vec{\mathrm{a}}_1^\top \end{array}\right]$ and let $\begin{bmatrix} \mathsf{W}_2^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix}$ be the null-space of $\left[\begin{array}{c|c} \mathsf{A} & \vec{\mathrm{a}}_2^\top \end{array}\right]$. Then the CRS's in game $\mathbf{G}_1$ are:

$$\mathbf{CRS}_{p,0} := \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_1^\top \end{array}\right] \cdot \left(\begin{bmatrix} \mathsf{D}'_1 \\ 0^{s\times k} \end{bmatrix} + \begin{bmatrix} \mathsf{W}_1 \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathsf{R}\ \mathsf{B}^{-1}\right) = \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_1^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{D}'_1 \\ 0^{s\times k} \end{bmatrix}$$

$$\mathbf{CRS}_{p,1} := \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_2^\top \end{array}\right] \cdot \left(\begin{bmatrix} \mathsf{D}'_2 \\ 0^{s\times k} \end{bmatrix} + \begin{bmatrix} \mathsf{W}_2 \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathsf{R}\ \mathsf{B}^{-1}\right) = \left[\begin{array}{c|c} \mathbf{A} & \vec{\mathbf{a}}_2^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{D}'_2 \\ 0^{s\times k} \end{bmatrix}$$

$$\mathbf{CRS}_{v,0}^{(n+k)\times k} := \begin{bmatrix} \mathsf{D}'_1\mathsf{B} + \mathsf{W}_1\mathsf{R} \\ \mathsf{R} \\ -\mathsf{B} \end{bmatrix} \cdot \mathbf{g}_2 \qquad \mathbf{CRS}_{v,1}^{(n+k)\times k} := \begin{bmatrix} \mathsf{D}'_2\mathsf{B} + \mathsf{W}_2\mathsf{R} \\ 0^{s\times k} \\ 0^{k\times k} \end{bmatrix} \cdot \mathbf{g}_2$$

We now claim that $\vec{w}^\top \stackrel{\text{def}}{=} \begin{bmatrix} \mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2 \\ \mathrm{I}^{s\times s} \end{bmatrix}$ is the null-space of $\mathsf{A}' \stackrel{\text{def}}{=} \left[\ \mathsf{A} \mid (\vec{a}_1^\top + \text{TAG} \cdot \vec{a}_2^\top)\ \right]$. This is because $\vec{w}^\top$ is a non-zero $t \times s$ matrix and satisfies:

$$\mathsf{A}'\vec{w}^\top = \left[\ \mathsf{A} \mid (\vec{a}_1^\top + \text{TAG} \cdot \vec{a}_2^\top)\ \right] \begin{bmatrix} \mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2 \\ \mathrm{I}^{s\times s} \end{bmatrix} = \mathsf{A}(\mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2) + (\vec{a}_1^\top + \text{TAG} \cdot \vec{a}_2^\top)$$

$$= \left[\ \mathsf{A} \mid \vec{a}_1^\top\ \right] \cdot \begin{bmatrix} \mathsf{W}_1 \\ \mathrm{I}^{s\times s} \end{bmatrix} + \text{TAG} \cdot \left[\ \mathsf{A} \mid \vec{a}_2^\top\ \right] \cdot \begin{bmatrix} \mathsf{W}_2 \\ \mathrm{I}^{s\times s} \end{bmatrix} = 0^{t\times s}$$

The rest of the proof is similar to the rest of the proof of soundness in Theorem 13, since A' defines the tag-based language. □

H Publicly Verifiable CCA2 Secure IBE

We defer the reader to [JR13] for details, and just describe the Key Generation and Encryption steps here. Group operations are denoted multiplicatively for consistency with the paper.

Setup: The authority uses a group generation algorithm for which the SXDH assumption holds to generate a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $\mathbf{g}_1$ and $\mathbf{g}_2$ as generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are of order q, and let e be a bilinear pairing on $\mathbb{G}_1 \times \mathbb{G}_2$. Then it picks c at random from $\mathbb{Z}_q$, and sets $\mathbf{f} = \mathbf{g}_2^c$. It further picks Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, b, d, e, u, z from $\mathbb{Z}_q$, and publishes the following public key **PK**:
$\mathbf{g}_1$, $\mathbf{g}_1^b$, $\mathbf{v}_1 = \mathbf{g}_1^{-\Delta_1 \cdot b + d}$, $\mathbf{v}_2 = \mathbf{g}_1^{-\Delta_2 \cdot b + e}$, $\mathbf{v}_3 = \mathbf{g}_1^{-\Delta_3 \cdot b + c}$, $\mathbf{v}_4 = \mathbf{g}_1^{-\Delta_4 \cdot b + z}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta_5 \cdot b + u}$.

Consider the language:

$$L = \{\langle C_1, C_2, C_3, i, \text{TAG}, h\rangle \mid \exists s\ :\ C_1 = \mathbf{g}_1^s, C_2 = \mathbf{g}_1^{bs}, C_3 = \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{v}_4^{h \cdot s}\}$$

It also publishes the QA-NIZK CRS for the language L (which uses tags i, TAG and h). It also publishes a 1-1, or Universal One-Way Hash function (UOWHF) $\mathcal{H}$. The authority retains the following master secret key **MSK**: $\mathbf{g}_2$, $\mathbf{f}$ $(= \mathbf{g}_2^c)$, and Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, d, e, u, z.

Encrypt(PK, i, M): the encryption algorithm chooses s and TAG at random from $\mathbb{Z}_q$. It then blinds M as $C_0 = M \cdot \mathbf{k}^s$, and also creates

$$C_1 = \mathbf{g}_1^s, C_2 = \mathbf{g}_1^{b \cdot s}, C_3 = \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{v}_4^{h \cdot s},$$

where $h = \mathcal{H}(C_0, C_1, C_2, \text{TAG}, i)$. The ciphertext is then $C = \langle C_0, C_1, C_2, C_3,\ \text{TAG},\ \mathbf{p}_1, \mathbf{p}_2\rangle$, where $\langle \mathbf{p}_1, \mathbf{p}_2\rangle$ is a QA-NIZK proof that $\langle C_0, C_1, C_2, C_3, i, \text{TAG}, h\rangle \in L$.

With the scheme in this paper, the ciphertext needs just **1** element of proof $\mathbf{p}$, instead of the two presented by the authors.