Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces *

Charanjit S. Jutla
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
csjutla@us.ibm.com

Arnab Roy
Fujitsu Laboratories of America
Sunnyvale, CA 94058, USA
arnab@cs.stanford.edu

Abstract

We define a novel notion of quasi-adaptive non-interactive zero knowledge (NIZK) proofs for probability distributions on parametrized languages. It is quasi-adaptive in the sense that the common reference string (CRS) generator can generate the CRS depending on the language parameters. However, the simulation is required to be uniform, i.e., a single efficient simulator should work for the whole class of parametrized languages. For distributions on languages that are linear subspaces of vector spaces over bilinear groups, we give quasi-adaptive computationally sound NIZKs that are shorter and more efficient than Groth-Sahai NIZKs. For many cryptographic applications quasi-adaptive NIZKs suffice, and our constructions can lead to significant improvements in the standard model. Our construction can be based on any k-linear assumption, and in particular under the eXternal Diffie Hellman (XDH) assumption our proofs are even competitive with Random-Oracle based Σ-protocol NIZK proofs.

We also show that our system can be extended to include integer tags in the defining equations, where the tags are provided adaptively by the adversary. This leads to applicability of our system to many applications that use tags, e.g. applications using Cramer-Shoup projective hash proofs. Our techniques also lead to the shortest known (ciphertext) fully secure identity based encryption (IBE) scheme under standard static assumptions (SXDH). Further, we also get a short publicly-verifiable CCA2-secure IBE scheme.

Keywords: NIZK, Groth-Sahai, bilinear pairings, signatures, dual-system IBE, DLIN, SXDH.

*An extended abstract of this paper [JR13] appears in the proceedings of ASIACRYPT 2013. This is the full version.

Contents

1 Introduction

In [GS08] a remarkably efficient non-interactive zero-knowledge (NIZK) proof system [BFM88] was given for groups with a bilinear map, which has found many applications in design of cryptographic protocols in the standard model. All earlier NIZK proof systems (except [Gro06], which was not very efficient) were constructed by reduction to Circuit Satisfiability. Underlying this system, now commonly known as Groth-Sahai NIZKs, is a homomorphic commitment scheme. Each variable in the system of algebraic equations to be proven is committed to using this scheme. Since the commitment scheme is homomorphic, group operations in the equations are translated to corresponding operations on the commitments and new terms are constructed involving the constants in the equations and the randomness used in the commitments. It was shown that these new terms along with the commitments to variables constitute a zero-knowledge proof [GS08].

While the Groth-Sahai system is quite efficient, it still falls short in comparison to Schnorr-based Σ-protocols [Dam] turned into NIZK proofs in the Random Oracle model [BR93] using the Fiat-Shamir paradigm [FS87]. Thus, the quest remains to obtain even more efficient NIZK Proofs. In particular, in a linear system of rank t, some t of the equations already serve as commitments to t variables. Thus, the question arises if, at the very least, fresh commitments to these variables as done in Groth-Sahai NIZKs can be avoided.

Our contributions. In this paper, we show that for languages that are linear subspaces of vector spaces of the bilinear groups, one can indeed obtain more efficient computationally-sound NIZK proofs in a slightly different *quasi-adaptive* setting, which suffices for many cryptographic applications. In the quasi-adaptive setting, we consider a class of parametrized languages $\{L_\rho\}$, parametrized by ρ, and we allow the CRS generator 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. We will refer to this property as *uniform simulation*.

Many hard languages that are commonly used in cryptography are distributions on class of parametrized languages, e.g. the DDH language based on the decisional Diffie-Hellman (DDH) assumption is hard only when in the tuple $\langle \mathbf{g}, \mathbf{f}, x \cdot \mathbf{g}, x \cdot \mathbf{f} \rangle$, even $\mathbf{f}$ is chosen at random (in addition to $x \cdot \mathbf{g}$ being chosen randomly). However, applications (or trusted parties) usually set $\mathbf{f}$, once and for all, by choosing it at random, and then all parties in the application can use *multiple* instances of the above language with the same fixed $\mathbf{f}$. Thus, we can consider $\mathbf{f}$ as a parameter for a class of languages that only specify the last two components above. If NIZK proofs are required in the application for this parametrized language, then the NIZK CRS can be generated by the trusted party that chooses the language parameter $\mathbf{f}$. Hence, it can base the CRS on the language parameter[1].

We remark that adaptive NIZK proofs [BFM88] also allow the CRS to depend on the language, but without requiring uniform simulation. Such NIZK proofs that allow different efficient simulators for each particular language (from a parametrized class) are unlikely to be useful in applications. Thus, most NIZK proofs, including Groth-Sahai NIZKs, actually show that the same efficient simulator works for the whole class, i.e. they show uniform simulation. The Groth-Sahai system achieves uniform simulation without making any distinction between different classes of

[1] However, in the security definition, the efficient CRS simulator does not itself generate $\mathbf{f}$, but is given $\mathbf{f}$ as input chosen randomly.

parametrized languages, i.e. it shows a single efficient CRS simulator that works for *all* algebraic languages without taking any language parameters as input. Thus, there is potential to gain efficiency by considering quasi-adaptive NIZK proofs, i.e. by allowing the (uniform) simulator to take language parameters as input[2].

Our approach to building more efficient NIZK proofs for linear subspaces is quite different from the Groth-Sahai techniques. In fact, our system does not require any commitments to the witnesses at all. If there are t free variables in defining a subspace of the n-dimensional vector-space and assuming the subspace is full-ranked (i.e. has rank t), then t components of the vector already serve as commitment to the variables. As an example, consider the language L (over a cyclic group $\mathbb{G}$ of order q, in additive notation) to be

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

where $\mathbf{g}$, $\mathbf{f}$, $\mathbf{h}$ are parameters defining the language. Then, $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ are already binding commitments to x_1 and x_2. Thus, we only need to show that the last component $\boldsymbol{l}_3$ is consistent.

The main idea underlying our construction can be summarized as follows. 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 we use a homomorphic commitment just as in the Groth-Sahai system, but we can 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[3]. As we will see later in Section 5, a more efficient verifier is critical for obtaining short identity based encryption schemes (IBE).

In fact, the idea of using the null-space of the language is reminiscent of Waters' dual-system IBE construction [Wat09], and indeed our system is inspired by that construction[4], although the idea of using it for NIZK proofs, and in particular the proof of soundness is novel. Another contribution of the paper is in the definition of quasi-adaptive NIZK proofs.

For n equations in t variables, our quasi-adaptive computationally-sound NIZK proofs for linear subspaces require only $k(n-t)$ group elements, under the k-linear decisional assumption [Sha07, CCS09]. Thus, under the XDH assumption for bilinear groups, our proofs require only $(n-t)$ group elements. In contrast, the Groth-Sahai system requires $(n+2t)$ group elements. Similarly, under the decisional linear assumption (DLIN), our proofs require only $2(n-t)$ group elements, whereas the Groth-Sahai system requires $(2n+3t)$ group elements. These parameters are summarized in Table 1. While our CRS size grows proportional to $t(n-t)$, more importantly there is a significant comparative improvement in the number of pairings required for verification. Specifically, under XDH we require at most half the number of pairings, and under DLIN we require at most 2/3 the number of pairings. The Σ-protocol NIZK proofs based on the Random Oracle model require n group elements, t elements of $\mathbb{Z}_q$ and 1 hash value. Although our XDH based proofs require

[2]It is important to specify the information about the parameter which is supplied as input to the CRS simulator. We defer this important issue to Section 2 where we formally define quasi-adaptive NIZK proofs.

[3]Our quasi-adaptive NIZK proofs are already shorter than Groth-Sahai as they require no commitments to variables, and have to prove lesser number of equations, as mentioned earlier.

[4]In Section 5 and in the Appendix, we show that the design of our system leads to a shorter SXDH assumption based dual-system IBE.

	XDH			DLIN		
	Proof	CRS	#Pairings	Proof	CRS	#Pairings
Groth-Sahai	$n + 2t$	4	$2n(t + 2)$	$2n + 3t$	9	$3n(t + 3)$
This paper	$n - t$	$2t(n - t) + 2$	$(n - t)(t + 2)$	$2n - 2t$	$4t(n - t) + 3$	$2(n - t)(t + 2)$

Table 1: Comparison with Groth-Sahai 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.

less number of group elements, the Σ-protocol proofs do not require bilinear groups and have the advantage of being proofs of knowledge (PoK). We remark that the Groth-Sahai system is also not a PoK for witnesses that are $\mathbb{Z}_q$ elements. A recent paper by Escala et al [EHK$^+$13] has also optimized proofs of linear subspaces in a language dependent CRS setting. Their system also removes the need for commitment to witnesses but still implicitly uses Groth Sahai proofs. In comparison, our proofs are still much shorter.

Thus, for the language L above, which is just a DLIN tuple used ubiquitously for encryption, our system only requires two group elements under the DLIN assumption, whereas the Groth-Sahai system requires twelve group elements (note, $t = 2$, $n = 3$ in L above). For the Diffie-Hellman analogue of this language $\langle x \cdot \mathbf{g}, x \cdot \mathbf{f} \rangle$, our system produces a *single* element proof under the XDH assumption, which we demonstrate in Section 3 (whereas the Groth-Sahai system requires $(n + 2t =)$ 4 elements for the proof with $t = 1$ and $n = 2$).

Our NIZK proofs also satisfy some interesting new properties. Firstly, the proofs in our system are unique for each language member. This has interesting applications as we will see later in a CCA2-IBE construction. Secondly, the CRS in our system, though dependent on the language parameters, can be split into two parts. The first part is required only by the prover, and the second part is required only by the verifier, and the latter can be generated independent of the language. This is surprising since our verifier does not even take the language (parameters) as input. Only the randomization used in the verifier CRS generation is used in the prover CRS to link the two CRSes. This is in sharp contrast to Groth-Sahai NIZKs, where the verifier needs the language as input. This split-CRS property has interesting applications as we will see later.

Extension to Linear Systems with Tags. Our system does not yet extend naturally to quadratic or multi-linear equations, whereas the Groth-Sahai system does[5]. However, we can extend our system to include tags, and allow the defining equations to be polynomially dependent on tags. For example, our system can prove the following language:

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

Note that this is a non-trivial extension since the TAG is adaptively provided by the adversary after the CRS has been set.

The extension to tags is very important, as we now discuss. Many applications require that the NIZK proof also be simulation-sound. However, extending NIZK proofs for bilinear groups to be unbounded simulation-sound requires handling quadratic equations (see [CCS09] for a generic

[5]However, since commitments in Groth-Sahai NIZKs are linear, there is scope for mixing the two systems to gain efficiency.

	Public Key	Secret Key	Ciphertext	#Pairings	Anonymity
[CLL$^+$12]	$8\|\mathbb{G}_1\| + \|\mathbb{G}_T\|$	$4\|\mathbb{G}_2\|$	$4\|\mathbb{G}_1\| + \|\mathbb{G}_T\|$	4	yes
This paper	$5\|\mathbb{G}_1\| + \|\mathbb{G}_T\|$	$5\|\mathbb{G}_2\|$	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|\mathbb{Z}_q\|$	3	yes

Table 2: Comparison with the SXDH-based IBE of Chen et al. [CLL$^+$12]. The notation $|\cdot|$ denotes the bit length of an element of the given group.

construction). On the other hand, many applications just require one-time simulation soundness, and as has been shown in [JR12], this can be achieved for linear subspaces by projective hash proofs [CS02]. Projective hash proofs can be defined by linear extensions, but require use of tags. Thus, our system can handle such equations. Many applications, such as signatures, can also achieve implicit unbounded simulation soundness using projective hash proofs, and such applications can utilize our system (see Section 5).

Applications. While the cryptographic literature is replete with NIZK proofs, we will demonstrate the applicability of quasi-adaptive NIZKs, and in particular our efficient system for linear subspaces, to a few recent applications such as signature schemes [CCS09], UC commitments [FLM11], password-based key exchange [KV11, JR12], key-dependent encryption [CCS09]. For starters, based on [FLM11], our system yields an adaptive UC-secure commitment scheme (in the erasure model) that has only four group elements as commitment, and another four as opening (under the DLIN assumption; and $3 + 2$ under SXDH assumption), whereas the original scheme using Groth-Sahai NIZKs required $5 + 16$ group elements.

We also obtain one of the shortest signature schemes under a static standard assumption, i.e. SXDH, that only requires five group elements. We also show how this signature scheme can be extended to a short fully secure (and perfectly complete) dual-system IBE scheme, and indeed a scheme with ciphertexts that are only four group elements plus a tag (under the SXDH assumption). This is the shortest IBE scheme under the SXDH assumption, and is technically even shorter than a recent and independently obtained scheme of [CLL$^+$12] which requires five group elements as ciphertext. Table 2 depicts numerical differences between the parameter sizes of the two schemes. The SXDH-IBE scheme of [CLL$^+$12] uses the concept of dual pairing vector spaces (due to Okamoto and Takashima [OT08, OT09], and synthesized from Waters' dual system IBE). However, the dual vector space and its generalizations due to others [Lew12] do not capture the idea of proof verification. Thus, one of our main contributions can be viewed as showing that the dual system not only does zero-knowledge simulation but also extends to provide a computationally sound verifier for general linear systems.

Finally, using our QA-NIZKs we show a short *publicly-verifiable* CCA2-secure IBE scheme. Public verifiability is an informal but practically important notion which implies that one can publicly verify if the decryption will yield "invalid ciphertext". Thus, this can allow a network gateway to act as a filter. Our scheme only requires two additional group elements over the basic IBE scheme.

Organization of the paper. We begin the rest of the paper with the definition of quasi-adaptive NIZKs in Section 2. In Section 3 we develop quasi-adaptive NIZKs for linear subspaces under the XDH assumption. In Section 4, we extend our system to include tags, define a notion called split-CRS QA-NIZKs and extend our system to construct split-CRS NIZKs for affine spaces. Finally,

we demonstrate applications of our system in Section 5. We defer detailed proofs and descriptions to the appendix. We also describe our system based on the k-linear assumption in Appendix C.

Notations. We will be dealing with witness-relations R that are binary relations on pairs (x, w), and where w is commonly referred to as the witness. Each witness-relation defines a language $L = \{x | \exists w : R(x, w)\}$. For every witness-relation R_ρ we will use L_ρ to denote the language it defines. Thus, a NIZK proof for a witness-relation R_ρ can also be seen as a NIZK proof for its language L_ρ.

Vectors will always be row-vectors and will always be denoted by an arrow over the letter, e.g. $\vec{\mathbf{r}}$ for (row) vector of $\mathbb{Z}_q$ elements, and $\vec{\mathbf{d}}$ as (row) vector of group elements.

2 Quasi-Adaptive NIZK Proofs

Instead of considering NIZK proofs for a (witness-) relation R, we will 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 $\mathcal{D}$. We will however require, as we will see later, that the simulator generating the CRS (in the simulation world) is a single probabilistic polynomial time algorithm that works for the whole collection of relations $\mathcal{R}$.

To be more precise, we will consider ensemble of distributions on witness-relations, each distribution in the ensemble itself parametrized by a security parameter. Thus, we will 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, we will just refer to a particular relation as R_ρ, and write $\mathcal{R}_\lambda = \{R_\rho\}$.

Since in the quasi-adaptive setting the CRS could depend on the relation, we must specify what information about the relation is given to the CRS generator. Thus, we will consider an associated *parameter language* such that a member of this language is enough to characterize a particular relation, and this language member is provided to the CRS generator. For example, consider the class of parametrized relations $\mathcal{R} = \{R_\rho\}$, where parameter ρ is a tuple $\mathbf{g}, \mathbf{f}, \mathbf{h}$ of three group elements. Suppose, R_ρ (on $\langle \boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3 \rangle, \langle x_1, x_2 \rangle$) is defined as

$$R_{\langle \mathbf{g},\mathbf{f},\mathbf{h} \rangle}(\langle \boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3 \rangle, \langle x_1, x_2 \rangle) \stackrel{\text{def}}{=} \left(\begin{array}{c} x_1, x_2 \in \mathbb{Z}_q, \boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3 \in \mathbb{G} \textbf{ and} \\ \boldsymbol{l}_1 = x_1 \cdot \mathbf{g}, \boldsymbol{l}_2 = x_2 \cdot \mathbf{f}, \boldsymbol{l}_3 = (x_1 + x_2) \cdot \mathbf{h} \end{array} \right).$$

For this class of relations, one could seek a quasi-adaptive NIZK where the CRS generator is just given ρ as input. Thus in this case, the associated parameter language $\mathcal{L}_{\text{par}}$ will just be triples of group elements[6]. Moreover, the distribution $\mathcal{D}$ can just be on the parameter language $\mathcal{L}_{\text{par}}$, i.e. $\mathcal{D}$ just specifies a $\rho \in \mathcal{L}_{\text{par}}$. Again, $\mathcal{L}_{\text{par}}$ is technically an ensemble.

We call $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ 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:

[6]It is worth remarking that alternatively the parameter language could also be discrete logarithms of these group elements (w.r.t. to some base), but a NIZK proof under this associated language may not be very useful. Thus, it is critical to define the proper associated parameter language.

Quasi-Adaptive Completeness:

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

Quasi-Adaptive Soundness:

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

Quasi-Adaptive Zero-Knowledge:

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

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.

3 QA-NIZK for Linear Subspaces under the XDH Assumption

Setup. Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order q with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ chosen by a group generation algorithm. Let $\mathbf{g}_1$ and $\mathbf{g}_2$ be generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. 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. We use additive notation for the group operations in all the groups.

The bilinear pairing e naturally extends to $\mathbb{Z}_q$-vector spaces of $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same dimension n as follows: $e(\vec{\mathbf{a}}, \vec{\mathbf{b}}^\top) = \sum_{i=1}^n e(\mathbf{a}_i, \mathbf{b}_i)$. Thus, if $\vec{\mathbf{a}} = \vec{\mathrm{x}} \cdot \mathbf{g}_1$ and $\vec{\mathbf{b}} = \vec{\mathrm{y}} \cdot \mathbf{g}_2$, where $\vec{\mathrm{x}}$ and $\vec{\mathrm{y}}$ are now vectors over $\mathbb{Z}_q$, then $e(\vec{\mathbf{a}}, \vec{\mathbf{b}}^\top) = (\vec{\mathrm{x}} \cdot \vec{\mathrm{y}}^\top) \cdot e(\mathbf{g}_1, \mathbf{g}_2)$. The operator "$\top$" indicates taking the transpose.

Linear Subspace Languages. To start off with an example, a set of equations $\boldsymbol{l}_1 = x_1 \cdot \mathbf{g}, \boldsymbol{l}_2 = x_2 \cdot \mathbf{f}, \boldsymbol{l}_3 = (x_1 + x_2) \cdot \mathbf{h}$ will be expressed in the form $\vec{\boldsymbol{l}} = \vec{\mathrm{x}} \cdot \mathbf{A}$ as follows:

$$\vec{\boldsymbol{l}} = \begin{bmatrix} \boldsymbol{l}_1 & \boldsymbol{l}_2 & \boldsymbol{l}_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{g} & \mathbf{0}_1 & \mathbf{h} \\ \mathbf{0}_1 & \mathbf{f} & \mathbf{h} \end{bmatrix}$$

where $\vec{\mathrm{x}}$ is a vector of unknowns and $\mathbf{A}$ is a matrix specifying the group constants $\mathbf{g}, \mathbf{f}, \mathbf{h}$.

The scalars in this system of equations are from the field $\mathbb{Z}_q$. In general, we consider languages that are linear subspaces of vectors of $\mathbb{G}_1$ elements. These are just $\mathbb{Z}_q$-modules, and since $\mathbb{Z}_q$ is a field, they are vector spaces. In other words, the languages we are interested in can be characterized as languages parameterized by $\mathbf{A}$ as below:

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

Here $\mathbf{A}$ is an element of the associated *parameter language* $\mathcal{L}_{\mathrm{par}}$, which is all $t \times n$ matrices of $\mathbb{G}_1$ elements. The parameter language $\mathcal{L}_{\mathrm{par}}$ also has a corresponding witness relation $\mathcal{R}_{\mathrm{par}}$, where the witness is a matrix of $\mathbb{Z}_q$ elements : $\mathcal{R}_{\mathrm{par}}(\mathbf{A}, \mathsf{A})$ 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 $\mathcal{L}_{\text{par}}$. We will call the distribution $\mathcal{D}$ *robust* if with probability close to one the left-most t columns of $\mathbf{A}$ are full-ranked. We will call a distribution $\mathcal{D}$ on $\mathcal{L}_{\text{par}}$ *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 $\mathbf{A}$. As an example of a robust distribution, consider a distribution $\mathcal{D}$ on (2×3)-dimensional matrices $\begin{bmatrix} \mathbf{g} & \mathbf{0}_1 & \mathbf{h} \\ \mathbf{0}_1 & \mathbf{f} & \mathbf{h} \end{bmatrix}$ with $\mathbf{g}, \mathbf{f}$ and $\mathbf{h}$ chosen randomly from $\mathbb{G}_1$. It is easy to see that the first two columns are full-ranked if $\mathbf{g} \neq \mathbf{0}_1$ and $\mathbf{f} \neq \mathbf{0}_1$, which holds with probability $(1 - 1/q)^2$.

QA-NIZK Construction. We now describe a computationally sound quasi-adap-tive 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 $\mathcal{L}_{\text{par}}$.
Algorithm K_0: K_0 is same as the group generation algorithm for which the XDH assumption holds. $\lambda \stackrel{\text{def}}{=} (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2) \leftarrow \mathsf{K}_0(1^m)$, with $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2)$ as described above.

We will assume that the size $t \times n$ of the matrix $\mathbf{A}$ is either fixed or determined by the security parameter m. In general, t and n could also be part of the parameter language, and hence t, n could be given as part of the input to CRS generator K_1.
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 $\mathsf{D}^{t \times s}$ with all elements chosen randomly from $\mathbb{Z}_q$ and a single element b chosen randomly from $\mathbb{Z}_q$. 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.

$$\mathbf{CRS}_p^{t \times s} := \mathbf{A} \cdot \begin{bmatrix} \mathsf{D}^{t \times s} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{bmatrix} \qquad \mathbf{CRS}_v^{(n+s) \times s} := \begin{bmatrix} b \cdot \mathsf{D} \\ \mathrm{I}^{s \times s} \\ -b \cdot \mathrm{I}^{s \times s} \end{bmatrix} \cdot \mathbf{g}_2$$

Here, I denotes the identity matrix. Note that $\mathbf{CRS}_v$ is independent of the parameter.
Prover P**:** Given candidate $\vec{l} = \vec{\mathrm{x}} \cdot \mathbf{A}$ with witness vector $\vec{\mathrm{x}}$, the prover generates the following proof consisting of s elements in $\mathbb{G}_1$:

$$\vec{\mathbf{p}} := \vec{\mathrm{x}} \cdot \mathbf{CRS}_p$$

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

$$e\left(\begin{bmatrix} \vec{l} \mid \vec{\mathbf{p}} \end{bmatrix}, \mathbf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1 \times s}$$

The security of the above system depends on the DDH assumption in group $\mathbb{G}_2$. Since $\mathbb{G}_2$ is a bilinear group, this assumption is known as the XDH assumption. These assumptions are standard and are formally described in Appendix A.

Theorem 1 *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 DDH assumption holds for group* $\mathbb{G}_2$*.*

Remark. For language members, the proofs are unique as the bottom s rows of $\mathbf{CRS}_v$ are invertible.

Proof Intuition. A detailed proof of the theorem can be found in Appendix B. Here we give the main idea behind the working of the above quasi-adaptive NIZK, and in particular the soundness requirement which is the difficult part here. We first observe that completeness follows by straightforward bilinear manipulation. Zero Knowledge also follows easily: the simulator generates the same CRS as above but retains D and b as trapdoors. Now, given a language candidate $\vec{\boldsymbol{l}}$, the proof is simply $\vec{\mathbf{p}} := \vec{\boldsymbol{l}} \cdot \begin{bmatrix} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s\times s} \end{bmatrix}$. If $\vec{\boldsymbol{l}}$ is in the language, i.e., it is $\vec{\mathrm{x}} \cdot \mathbf{A}$ for some $\vec{\mathrm{x}}$, then the distribution of the simulated proof is identical to the real world proof.

We now focus on the soundness proof which we establish by transforming the system over two games. Let Game $\mathbf{G}_0$ be the original system. Since $\mathcal{D}$ is efficiently witness samplable, in Game $\mathbf{G}_1$ the challenger generates both A and $\mathbf{A} = \mathsf{A} \cdot \mathbf{g}_1$. Then it computes 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 $\mathsf{A} \cdot \begin{bmatrix} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix} = 0^{t\times s}$. Now statistically, the CRS in Game $\mathbf{G}_0$ is indistinguishable from the one where we substitute $\mathsf{D}' + b^{-1} \cdot \mathsf{W}$ for D, where D' itself is an independent random matrix. With this substitution, the $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ can be represented as

$$\mathbf{CRS}_p^{t\times s} = \mathbf{A} \cdot \begin{bmatrix} \mathsf{D}' \\ 0^{s\times s} \end{bmatrix}, \quad \mathbf{CRS}_v^{(n+s)\times s} = \begin{bmatrix} b \cdot \begin{bmatrix} \mathsf{D}' \\ 0^{s\times s} \end{bmatrix} + \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \\ -b \cdot \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathbf{g}_2$$

Now we show that if an efficient adversary can produce a "proof" $\vec{\mathbf{p}}$ for which the above pairing test holds and yet the candidate $\vec{\boldsymbol{l}}$ is not in $L_{\mathbf{A}}$, then it implies an efficient adversary that can break DDH in group $\mathbb{G}_2$. So consider a DDH game, where a challenger either provides a real DDH-tuple $\langle \mathbf{g}_2, b\cdot\mathbf{g}_2, r\cdot\mathbf{g}_2, \boldsymbol{\chi} = br\cdot\mathbf{g}_2\rangle$ or a fake DDH tuple $\langle \mathbf{g}_2, b\cdot\mathbf{g}_2, r\cdot\mathbf{g}_2, \boldsymbol{\chi} = br'\cdot\mathbf{g}_2\rangle$. 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 $\vec{\boldsymbol{l}}\,' := \left[\begin{array}{c|c} \vec{\boldsymbol{l}}_0 & -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W} \end{array}\right]$, 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 both $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ and $(\vec{\boldsymbol{l}}\,', \vec{\mathbf{p}}')$ pass the verification equation, we obtain: $\vec{\boldsymbol{l}}'_1 - \vec{\boldsymbol{l}}_1 = b(\vec{\mathbf{p}}' - \vec{\mathbf{p}})$, where $\vec{\boldsymbol{l}}'_1 = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$. In particular there exists $i \in [1, s]$, such that, $\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i} = b(\mathbf{p}'_i - \mathbf{p}_i) \neq \mathbf{0}_1$. This gives us a straightforward test for the DDH challenge: $e(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}, r\cdot\mathbf{g}_2) \stackrel{?}{=} e(\mathbf{p}'_i - \mathbf{p}_i, \boldsymbol{\chi})$. This leads to a proof of soundness of the QA-NIZK.

Remark. Observe from the proof above that the soundness can be based on the following *computational* assumption which is implied by XDH, which is a *decisional* assumption:

Definition 2 *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 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. *The assumption asserts that the following problem is hard: Given* $\mathbf{f}, \mathbf{f}^b \xleftarrow{\$} \mathbb{G}_2$, *output* $\mathbf{h}, \mathbf{h}' \in \mathbb{G}_1$, *such that* $\mathbf{h}' = \mathbf{h}^b \neq \mathbf{0}_1$.

Example: QA-NIZK for a DH tuple. In this example, we instantiate our general system to provide a NIZK for a DH tuple, that is a tuple of the form $(x \cdot \mathbf{g}, x \cdot \mathbf{f})$ for an a priori fixed base $(\mathbf{g}, \mathbf{f}) \in \mathbb{G}_1^2$. We assume DDH for the group $\mathbb{G}_2$.

As in the setup described before, we have $\mathbf{A} = \begin{bmatrix} \mathbf{g} & \mathbf{f} \end{bmatrix}$. The language is: $L = \{[x] \cdot \mathbf{A} \mid x \in \mathbb{Z}_q\}$.

Now proceeding with the framework, we generate D as $[d]$ and the element b where d and b are random elements of $\mathbb{Z}_q$. With this setting, the NIZK CRS is:

$$\mathbf{CRS}_p := \mathbf{A} \cdot \begin{bmatrix} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{1\times 1} \end{bmatrix} = \left[d \cdot \mathbf{g} + b^{-1} \cdot \mathbf{f}\right], \ \mathbf{CRS}_v := \begin{bmatrix} b \cdot \mathsf{D} \\ \mathrm{I}^{1\times 1} \\ -b \cdot \mathrm{I}^{1\times 1} \end{bmatrix} \cdot \mathbf{g}_2 = \begin{bmatrix} bd \cdot \mathbf{g}_2 \\ \mathbf{g}_2 \\ -b \cdot \mathbf{g}_2 \end{bmatrix}$$

The proof of a tuple $(\mathbf{r}, \hat{\mathbf{r}})$ with witness r, is just the *single* element $r \cdot (d \cdot \mathbf{g} + b^{-1} \cdot \mathbf{f})$. In the proof of zero knowledge, the simulator trapdoor is (d, b) and the simulated proof of $(\mathbf{r}, \hat{\mathbf{r}})$ is just $(d \cdot \mathbf{r} + b^{-1} \cdot \hat{\mathbf{r}})$.

4 Extensions

In this section we consider some useful extensions of the concepts and constructions of QA-NIZK systems. We show how the previous system can be extended to include tags. 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 define a notion called split-CRS QA-NIZK system, where the prover and verifier use distinct parts of a CRS and we construct a split-CRS system for affine systems.

Tags. 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 (the function being defined by parameters). We can handle arbitrary polynomial functions of the tags as well, but we will focus on affine functions here as most applications seem to need just affine functions. Then, the languages we handle can be characterized as

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

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 3) if with overwhelming probability the first t columns of $\mathbf{A}$ are full-ranked. Write $\mathbf{A}$ as $[\mathbf{A}_l^{t\times t} \mid \mathbf{A}_r^{t\times(n-1-t)}]$, where without loss of generality, $\mathbf{A}_l$ is non-singular. While the first $n-1-t$ components in excess of the unknowns, corresponding to $\mathbf{A}_r$, can be verified just as in Section 3, for the last component we proceed as follows.

Algorithm K_1: The CRS is generated as:

$$\mathbf{CRS}_{p,0}^{t\times 1} := \left[\ \mathbf{A}_l \ \middle| \ \vec{\mathbf{a}}_1^\top \ \right] \cdot \begin{bmatrix} \mathsf{D}_1 \\ b^{-1} \end{bmatrix} \qquad \mathbf{CRS}_{p,1}^{t\times 1} := \left[\ \mathbf{A}_l \ \middle| \ \vec{\mathbf{a}}_2^\top \ \right] \cdot \begin{bmatrix} \mathsf{D}_2 \\ b^{-1} \end{bmatrix}$$

$$\mathbf{CRS}_{v,0}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_1 \\ 1 \\ -b \end{bmatrix} \cdot \mathbf{g}_2 \qquad \mathbf{CRS}_{v,1}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_2 \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{g}_2$$

where D_1 and D_2 are random matrices of order $t \times 1$ independent of the matrix D chosen for proving the other components. The $\mathbb{Z}_q$ element b can be re-used from the other components.

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

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

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

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

The size of the proof is 1 element in the group $\mathbb{G}_1$. The proof of completeness, soundness and zero-knowledge for this quasi-adaptive system is similar to proof in Section 3 and a proof sketch can be found in Appendix D.

Split-CRS QA-NIZK Proofs. We note that the QA-NIZK described in Section 3 has an interesting split-CRS property. In a **split-CRS QA-NIZK** for a distribution of relations, the CRS generator K_1 generates two CRS-es ψ_p and ψ_v, such that the prover P *only* needs ψ_p, and the verifier V *only* needs ψ_v. In addition, the CRS ψ_v is *independent* of the particular relation R_ρ. In other words the CRS generator K_1 can be split into two PPTs K_{11} and K_{12}, such that K_{11} generates ψ_v using just λ, and K_{12} generates ψ_p using ρ and a state output by K_{11}. The key generation simulator S_1 is also split similarly. The formal definition is given in Appendix 4.

In many applications, split-CRS QA-NIZKs can lead to simpler constructions (and their proofs) and possibly shorter proofs.

Split-CRS QA-NIZK for Affine Spaces. Consider languages that are affine spaces

$$L_{\mathbf{A},\vec{\mathbf{a}}} = \{(\vec{\mathrm{x}} \cdot \mathbf{A} + \vec{\mathbf{a}}) \in \mathbb{G}_1^n \mid \vec{\mathrm{x}} \in \mathbb{Z}_q^t\}$$

The parameter language $\mathcal{L}_{\mathrm{par}}$ just specifies $\mathbf{A}$ and $\vec{\mathbf{a}}$. A distribution over $\mathcal{L}_{\mathrm{par}}$ is called robust if with overwhelming probability the left most $t \times t$ sub-matrix of $\mathbf{A}$ is non-singular (full-ranked). If $\vec{\mathbf{a}}$ is given as part of the verifier CRS, then a QA-NIZK for distributions over this class follows directly from the construction in Section 3. However, that would make the QA-NIZK non split-CRS. We now show that the techniques of Section 3 can be extended to give a split-CRS QA-NIZK for (robust and witness-samplable) distributions over affine spaces.

The common reference string (CRS) has two parts ψ_p and ψ_v which are to be used by the prover and the verifier respectively. The split-CRS generator K_{11} and K_{12} work as follows. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns.

Algorithm K_{11}: The verifier CRS generator first generates a matrix $\mathsf{D}^{t \times s}$ with all elements chosen randomly from $\mathbb{Z}_q$ and a single element b chosen randomly from $\mathbb{Z}_q$. It also generates a row vector $\vec{\mathrm{d}}^{1 \times s}$ at random from $\mathbb{Z}_q$. Next, it computes

$$\mathbf{CRS}_v^{(n+s) \times s} := \left[\begin{array}{c} b \cdot \mathsf{D} \\ \mathrm{I}^{s \times s} \\ -b \cdot \mathrm{I}^{s \times s} \end{array} \right] \cdot \mathbf{g}_2 \qquad \vec{\mathbf{f}}^{1 \times s} := e(\mathbf{g}_1, b \cdot \vec{\mathrm{d}} \cdot \mathbf{g}_2)$$

The verifier CRS ψ_v is the matrix $\mathbf{CRS}_v$ and $\vec{\mathbf{f}}$.

Algorithm K_{12}: The prover CRS generator K_{12} generates

$$\mathbf{CRS}_p^{t\times s} = \left[\begin{array}{c} \mathbf{A}^{t\times n} \\ \vec{\mathbf{a}}^{1\times n} \end{array}\right] \cdot \left[\begin{array}{c} \mathsf{D} \\ b^{-1}\cdot \mathsf{I}^{s\times s} \end{array}\right] - \left[\begin{array}{c} 0^{t\times s} \\ \vec{\mathrm{d}}^{1\times s} \end{array}\right] \cdot \mathbf{g}_1$$

The (prover) CRS ψ_p is just the matrix $\mathbf{CRS}_p$.
Prover: Given candidate $(\vec{\mathrm{x}}\cdot\mathbf{A}+\vec{\mathbf{a}})$ with witness vector $\vec{\mathrm{x}}$, the prover generates the following proof:

$$\vec{\mathbf{p}} := \left[\begin{array}{c|c} \vec{\mathrm{x}} & 1 \end{array}\right] \cdot \mathbf{CRS}_p$$

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

$$e\left(\left[\begin{array}{c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v\right) \stackrel{?}{=} \vec{\mathbf{f}}$$

We provide a proof sketch in Appendix F. The split-CRS QA-NIZK for affine spaces also naturally extends to include tags as described before in this section.

5 Applications

In this section we mention several important applications of quasi-adaptive NIZK proofs. Before we go into the details of these applications, we discuss the general applicability of quasi-adaptive NIZKs. Recall in quasi-adaptive NIZKs, the CRS is set based on the language for which proofs are required. In many applications the language is set by a trusted party, and the most obvious example of this is the trusted party that sets the CRS in some UC applications, many of which have UC realizations only with a CRS. Another obvious example is the (H)IBE trusted party that issues secret keys to various identities. In many public key applications, the party issuing the public key is also considered trusted, i.e. incorruptible, as security is defined with respect to the public key issuing party (acting as challenger). Thus, in all these settings if the language for which proofs are required is determined by a incorruptible party, then that party can also issue the QA-NIZK CRS based on that language. It stands to reason that most languages for which proofs are required are ultimately set by an incorruptible party (at least as far as the security definitions are concerned), although they may not be linear subspaces, and can indeed be multi-linear or even quadratic. For example, suppose a potentially corruptible party P wants to (NIZK) prove that $x \in L_\rho$, where L_ρ is a language that it generated. However, this proof is unlikely to be of any use unless it also proves something about L_ρ, e.g., that ρ itself is in another language, say L'. In some applications, potentially corruptible parties generate new linear languages using random tags. However, the underlying basis for these languages is set by a trusted party, and hence the NIZK CRS for the whole range of tag based languages can be generated by that trusted party based on the original basis for these languages.

Adaptive UC Commitments in the Erasure Model. The SXDH-based commitment scheme from [FLM11] requires the following quasi-adaptive NIZK proof (see Appendix G for details)

$$\{\langle R, S, T\rangle \mid \exists r : R = r\cdot\mathbf{g}, S = r\cdot\mathbf{h}, T = r\cdot(\mathbf{d}_1 + \mathrm{TAG}\cdot\mathbf{e}_1)\}$$

with parameters $\mathbf{h}, \mathbf{d}_1, \mathbf{e}_1$ (chosen randomly), which leads to a UC commitment scheme with commitment consisting of 3 $\mathbb{G}_1$ elements, and a proof consisting of two $\mathbb{G}_2$ elements. Under DLIN,

a similar scheme leads to a commitment consisting of 4 elements and an opening of another 4 elements, whereas [FLM11] stated a scheme using Groth-Sahai NIZK proofs requiring $(5 + 16)$ elements. More details can be found in Appendix G.

One-time (Relatively) Simulation-Sound NIZK for DDH and others. In [JR12] it was shown that for linear subspace languages, such as the DDH or DLIN language, or the language showing that two El-Gamal encryptions are of the same message [NY90, Sah99], the NIZK proof can be made one-time simulation sound using a projective hash proof [CS02] and proving in addition that the hash proof is correct. For the DLIN language, this one-time simulation sound proof (in Groth-Sahai system) required 15 group elements, whereas the quasi-adaptive proof in this paper leads to a proof of size only 5 group elements.

Signatures. We will now show a generic construction of existentially unforgeable signature scheme (against adaptive adversaries) from labeled CCA2-encryption schemes and split-CRS QA-NIZK proof system (as defined in Section 4) for a related language distribution. This construction is a generalization of a signature scheme from [CCS09] which used (fully) adaptive NIZK proofs and *required* constructions based on groups in which the CDH assumption holds.

Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a labeled CCA-encryption scheme on messages. Let X_m be any subset of the message space of $\mathcal{E}$ such that $1/|X_m|$ is negligible in the security parameter m. Consider the following class of (parametrized) languages $\{L_\rho\}$:

$$L_\rho = \{(c, M) \mid \exists r : \ c = \mathsf{Enc}_{\mathsf{pk}}(\mathbf{u}; r; M)\}$$

with parameter $\rho = (\mathbf{u}, \mathsf{pk})$. The notation $\mathsf{Enc}_{\mathsf{pk}}(\mathbf{u}; r; M)$ means that $\mathbf{u}$ is encrypted under public key pk with randomness r and label M. Consider the following distribution $\mathcal{D}$ on the parameters: $\mathbf{u}$ is chosen uniformly at random from X_m and pk is generated using the probabilistic algorithm KeyGen of $\mathcal{E}$ on 1^m (the secret key is discarded). Note we have an ensemble of distributions, one for each value of the security parameter, but we will suppress these details.

Let $\mathcal{Q} = (\mathsf{K}_0, \langle \mathsf{K}_{11}, \mathsf{K}_{12} \rangle, \mathsf{P}, \mathsf{V})$ be a split-CRS QA-NIZK for distribution $\mathcal{D}$ on $\{L_\rho\}$. Note that the associated parameter language $\mathcal{L}_{\text{par}}$ is just the set of pairs $(\mathbf{u}, \mathsf{pk})$, and $\mathcal{D}$ specifies a distribution on $\mathcal{L}_{\text{par}}$.

Now, consider the following signature scheme $\mathcal{S}$.

Key Generation: On input a security parameter m, run $\mathsf{K}_0(1^m)$ to get λ. Let $\mathcal{E}.\mathsf{pk}$ be generated using KeyGen of $\mathcal{E}$ on 1^m (the secret key sk is discarded). Choose $\mathbf{u}$ at random from X_m. Let $\rho = (\mathbf{u}, \mathcal{E}.\mathsf{pk})$. Generate ψ_v by running K_{11} on λ (it also generates a state s). Generate ψ_p by running K_{12} on (λ, ρ) and state s. The public key $\mathcal{S}.\mathsf{pk}$ of the signature scheme is then ψ_v. The secret key $\mathcal{S}.\mathsf{sk}$ consists of $(\mathbf{u}, \mathcal{E}.\mathsf{pk}, \psi_p)$.

Sign: The signature on M just consists of a pair $\langle c, \pi \rangle$, where c is an $\mathcal{E}$-encryption of $\mathbf{u}$ with label M (using public key $\mathcal{E}.\mathsf{pk}$ and randomness r), and π is the QA-NIZK proof generated using prover P of $\mathcal{Q}$ on input $(\psi_p, (c, M), r)$. Recall r is the witness to the language member (c, M) of L_ρ (and $\rho = (\mathbf{u}, \mathcal{E}.\mathsf{pk})$).

Verify: Given the public key $\mathcal{S}.\mathsf{pk}$ $(= \psi_v)$, and a signature $\langle c, \pi \rangle$ on message M, the verifier uses the verifier V of $\mathcal{Q}$ and outputs $\mathsf{V}(\psi_v, (c, M), \pi)$.

Theorem 3 *If $\mathcal{E}$ is a labeled CCA2-encryption scheme and $\mathcal{Q}$ is a split-CRS quasi-adaptive NIZK system for distribution $\mathcal{D}$ on class of languages $\{L_\rho\}$ described above, then the signature scheme described above is existentially unforgeable under adaptive chosen message attacks.*

The theorem is proved in Appendix G. It is worth remarking here that the reason one can use a quasi-adaptive NIZK here is because the language L_ρ for which (multiple) NIZK proof(s) is required is set (or chosen) by the (signature scheme) key generator, and hence the key generator can generate the CRS for the NIZK after it sets the language. The proof of the above theorem can be understood in terms of simulation-soundness. Suppose the above split-CRS QA-NIZK was also unbounded simulation-sound. Then, one can replace the CCA2 encryption scheme with just a CPA-encryption scheme, and still get a secure signature scheme. A proof sketch of this is as follows: an Adversary $\mathcal{B}$ is only given ψ_v (which is independent of parameters, including $\mathbf{u}$). Further, the simulator for the QA-NIZK can replace all proofs by simulated proofs (that do not use witness r used for encryption). Next, one can employ CPA-security to replace encryptions of $\mathbf{u}$ by encryptions of 1. By unbounded simulation soundness of the QA-NIZK it follows that if $\mathcal{B}$ produces a verifying signature then it must have produced an encryption of $\mathbf{u}$. However, the view of $\mathcal{B}$ is independent of $\mathbf{u}$, and hence its probability of forging a signature is negligible.

However, the best known technique for obtaining efficient unbounded simulation soundness itself requires CCA2 encryption (see [CCS09]), and in addition NIZK proofs for quadratic equations. On the other hand, if we instantiate the above theorem with Cramer-Shoup encryption scheme, we get remarkably short signatures (in fact the shortest signatures under any static and standard assumption). The Cramer-Shoup encryption scheme PK consists of $\mathbf{g}, \mathbf{f}, \mathbf{k}, \mathbf{d}, \mathbf{e}$ chosen randomly from $\mathbb{G}_1$, along with a target collision-resistant hash function $\mathcal{H}$ (with a public random key). The set X from which $\mathbf{u}$ is chosen is just the whole group $\mathbb{G}_1$. Then an encryption of $\mathbf{u}$ is obtained by picking r at random, and obtaining the tuple

$$\langle R = r \cdot \mathbf{g},\ S = r \cdot \mathbf{f},\ T = \mathbf{u} + r \cdot \mathbf{k},\ H = r \cdot (\mathbf{d} + \text{TAG} \cdot \mathbf{e}) \rangle$$

where $\text{TAG} = \mathcal{H}(R, S, T, M)$. It can be shown that it suffices to hide $\mathbf{u}$ with the hash proof H (although one has to go into the internals of the hash-proof based CCA2 encryption; see Appendix in [JR12]). Thus, we just need a (split-CRS) QA-NIZK for the tag-based *affine* system (it is affine because of the additive constant $\mathbf{u}$). There is one variable r, and three equations (four if we consider the original CCA-2 encryption) Thus, we just need $(3-1)*1\ (=2)$ proof elements, leading to a total signature size of 5 elements (i.e. $R, S, \mathbf{u} + H$, and the two proof elements) under the SXDH assumption.

Dual-System Fully Secure IBE. It is well-known that Identity Based Encryption (IBE) implies signature schemes (due to Naor), but the question arises whether the above signature scheme using Cramer-Shoup CCA2-encryption and the related QA-NIZK can be converted into an IBE scheme. To achieve this, we take a hint from Naor's IBE to Signature Scheme conversion, and let the signatures (on identities) be private keys of the various identities. The verification of the QA-NIZK from Section 3 works by checking $e\left(\left[\begin{array}{c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathsf{CRS}_v\right) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$ (or more precisely, $e\left(\left[\begin{array}{c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathsf{CRS}_v\right) \stackrel{?}{=} \vec{\mathbf{f}}$ for the affine language). However, there are two issues: (1) CRS_v needs to be randomized, (2) there are two equations to be verified (which correspond to the alternate decryption of Cramer-Shoup encryption, providing implicit simulation-soundness). Both these problems are resolved by first scaling CRS_v by a random value s, and then taking a linear combination of the two equations using a public random tag. The right hand side $s \cdot \vec{\mathbf{f}}$ can then serve as secret one-time pad for encryption. Rather than being a provable generic construction, this is more a hint to get to a really short IBE. We give the construction and a complete proof in Appendix H. It shows an

IBE scheme under the SXDH assumption where the ciphertext has only four group ($\mathbb{G}_1$) elements plus a $\mathbb{Z}_q$-tag, which is the shortest IBE known under standard static assumptions[7].

Publicly-Verifiable CCA2 Fully-Secure IBE. We can also extend our IBE scheme above to be publicly-verifiable CCA2-secure [RS92, BDPR98]. Public verifiability is an informal but practical notion: most CCA2-secure schemes have a test of well-formedness of ciphertext, and on passing the test a CPA-secure scheme style decryption suffices. However, if this test can be performed publicly, i.e. without access to the secret key, then we call the scheme publicly-verifiable. While there is a well known reduction from hierarchical IBE to make an IBE scheme CCA2-secure [BCHK07], that reduction does not make the scheme publicly-verifiable CCA2 in a useful manner. In the IBE setting, publicly-verifiable *also* requires that it be verifiable if the ciphertext is *valid for the claimed identity*. This can have interesting applications where the network can act as a filter. We show that our scheme above can be extended to be publicly-verifiable CCA2-fully-secure IBE with *only* two additional group elements in the ciphertext (and two additional group elements in the keys). We give the construction and a complete proof in Appendix I. The IBE scheme above 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. We show that if these three group elements can be QA-NIZK proven to be consistent, and given the unique proof property of our QA-NIZKs, then the above IBE scheme can be made CCA2-secure - the dual-system already has implicit simulation-soundness as explained in the signature scheme above, and we show that this QA-NIZK need not be simulation-sound. Since, there are three components, and one variable (see the appendix for details), the QA-NIZK requires only two group elements under SXDH.

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. Springer, August 2004. 5, 6, 9

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007. 5, I

[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. Springer, August 1998. 5, I

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988. 1, 1

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993. 1

[7] [CLL+12] have recently and independently obtained a short IBE under SXDH, but our IBE ciphertexts are even shorter. See Table 2 in the Introduction for detailed comparison.

[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. Springer, April 2009. 1, 1, 1, 5, 5, 10, G

[CLL+12] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. In *Pairing*, pages 122–140, 2012. 1, 2, 7, 8

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, April / May 2002. 1, 5

[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. 4

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for diffie-hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147. Springer, 2013. 1

[FLM11] Marc Fischlin, Benoît Libert, and Mark Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In *Advances in Cryptology – ASIACRYPT 2011*, Lecture Notes in Computer Science, pages 468–485. Springer, December 2011. 1, 5, G, G

[FS87] 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. Springer, August 1987. 1

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, December 2006. 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. Springer, April 2008. 1

[JR12] Charanjit S. Jutla and Arnab Roy. Relatively-sound NIZKs and password-based key-exchange. In *PKC 2012: 15th International Workshop on Theory and Practice in Public Key Cryptography*, Lecture Notes in Computer Science, pages 485–503. Springer, 2012. 1, 1, 5, 5, I

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT*, 2013. ∗

[KV11] Jonathan Katz and Vinod Vaikuntanathan. Round-optimal password-based authenticated key exchange. In *TCC 2011: 8th Theory of Cryptography Conference*, Lecture Notes in Computer Science, pages 293–310. Springer, 2011. 1, I

[Lew12] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *Advances in Cryptology – EUROCRYPT 2012*, Lecture Notes in Computer Science, pages 318–335. Springer, 2012. 1

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990. 5

[OT08] Tatsuaki Okamoto and Katsuyuki Takashima. Homomorphic encryption and signatures from vector decomposition. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 57–74. Springer, September 2008. 1

[OT09] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, December 2009. 1

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, August 1992. 5

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, October 1999. 5

[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, 10

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, August 2009. 1, H, H

A Hardness Assumptions

Definition 4 (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 $(\mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^c)$ and $(\mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^{ab})$ for $a, b, c \xleftarrow{\$} \mathbb{Z}_q$. More formally, for all PPT adversaries A there exists a negligible function $\nu()$ such that*

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

Definition 5 (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 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. 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 6 (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 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. 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 $\mathbb{G}_2$.*

B Proof of QA-NIZK for Linear Subspaces under XDH Assumption

Theorem 1 (Section 3) The 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 $\mathcal{L}_{\text{par}}$, given any group generation algorithm for which the DDH assumption holds for group $\mathbb{G}_2$.
Proof:

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

$$
\begin{aligned}
& e(\left[\begin{array}{c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_2) \quad = \quad e(\left[\begin{array}{c|c} \vec{\mathrm{x}} \cdot \mathbf{A} & \vec{\mathrm{x}} \cdot \mathbf{CRS}_1 \end{array}\right], \mathbf{CRS}_2) \\
&= e\left(\vec{\mathrm{x}} \cdot \mathbf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n \times n} & \begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array} \end{array}\right] \cdot \left[\begin{array}{c} b \cdot \mathsf{D} \\ \mathrm{I}^{s \times s} \\ -b \cdot \mathrm{I}^{s \times s} \end{array}\right], \mathbf{g}_2\right) \\
&= e\left(\vec{\mathrm{x}} \cdot \mathbf{A} \cdot \left(\left[\begin{array}{c} b \cdot \mathsf{D} \\ \mathrm{I}^{s \times s} \end{array}\right] - b \cdot \left[\begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array}\right] \right), \mathbf{g}_2\right) = \; e(\mathbf{0}_1^{1 \times s}, \mathbf{g}_2) = \mathbf{0}_T^{1 \times s}
\end{aligned}
$$

Hence completeness follows.

Zero Knowledge: The CRS is generated exactly as above. In addition, the simulator is given the trapdoor $\left[\begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array}\right]$. Now, given a language candidate $\vec{l}$, the proof is simply $\vec{\mathbf{p}} := \vec{l} \cdot \left[\begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array}\right]$. If $\vec{l}$ is in the language, i.e., it is $\vec{\mathrm{x}} \cdot \mathbf{A}$ for some $\vec{\mathrm{x}}$, 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 two games. Game $\mathbf{G}_0$ just replicates the soundness security definition. In game $\mathbf{G}_1$ the CRS is generated using witness A and its null-space, and this can be done efficiently by the challenger as the distribution is efficiently witness samplable. After this transformation, we show that a verifying proof of a non-language member implies breaking DDH in group $\mathbb{G}_2$.

Game $\mathbf{G}_0$: This is just the original system, i.e., the challenger takes a security parameter m, generates λ using K_0, then generates $\mathbf{A}$ according to $\mathcal{D}$, generates the CRS ψ using K_1, and passes $\lambda, \mathbf{A}$ and the CRS (i.e. $\mathbf{CRS}_1$, $\mathbf{CRS}_2$) to an Adversary $\mathcal{B}$. Let the $\mathcal{B}$ produce candidate $\vec{l}$ and proof $\vec{\mathbf{p}}$. We say $\mathcal{B}$ wins if $e\left(\left[\begin{array}{c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_2\right) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$ while $\vec{l}$ is not in $L_{\mathbf{A}}$. Let W_0 denote the event that $\mathcal{B}$ wins game $\mathbf{G}_0$. If we can show that $\Pr[W_0]$ is negligible (in m), then soundness follows.

Game $\mathbf{G}_1$: Since $\mathcal{D}$ is efficiently witness samplable, say using a PPT machine $\mathcal{M}$, in this game the challenger generates $\mathbf{A} = \mathsf{A} \cdot \mathbf{g}_1$ using $\mathcal{M}$, and hence the challenger also gets A (the witness to $\mathbf{A}$ in language $\mathcal{L}_{\text{par}}$). Next the challenger checks if the left most t columns of A are full-ranked. If they are not full-ranked, the Challenger declares the Adversary as winner. We will also call this event BAD. The probability of event BAD happening is negligible by definition as the distribution $\mathcal{D}$ is robust. Otherwise, it computes 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 $\mathsf{A} \cdot \left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right] = \mathsf{0}^{t\times s}$. Next, the NIZK CRS is computed as follows: The challenger generates matrix $\mathsf{D}'^{\,t\times s}$ with elements randomly chosen from $\mathbb{Z}_q$ and element b randomly chosen from $\mathbb{Z}_q$ (just as in the real CRS). Now set,

$$\left[\begin{array}{c} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{array}\right] = \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{array}\right] + b^{-1} \cdot \left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right]$$

Therefore the challenger produces,

$$\mathbf{CRS}_1^{t\times s} = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{array}\right] = \mathbf{A} \cdot \left(\left[\begin{array}{c} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{array}\right] - b^{-1} \cdot \left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right]\right) = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{array}\right]$$

$$\mathbf{CRS}_2^{(n+s)\times s} = \left[\begin{array}{c} b\cdot \mathsf{D} \\ \mathrm{I}^{s\times s} \\ -b\cdot \mathrm{I}^{s\times s} \end{array}\right] \cdot \mathbf{g}_2 = \left[\begin{array}{c} b \cdot \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{array}\right] + \left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right] \\ -b\cdot \mathrm{I}^{s\times s} \end{array}\right] \cdot \mathbf{g}_2$$

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 $\mathbf{G}_0$, conditioned on BAD not happening. Let W_1 denote the event that Adversary wins game $\mathbf{G}_1$. Since event BAD implies event W_1, it follows that $\Pr[W_1] \geq \Pr[W_0]$. Moreover,

$$\begin{aligned} \Pr[W_1] = &\ \Pr[W_1 \wedge \text{BAD}] + \Pr[W_1 \wedge \neg\text{BAD}] \\ \leq &\ \Pr[\text{BAD}] + \Pr[W_1 \wedge \neg\text{BAD}] \end{aligned}$$

Since probability of event BAD is negligible, if we can show $\Pr[W_1 \wedge \neg\text{BAD}]$ to be negligible, soundness would follow. We remark that the Challenger in game $\mathbf{G}_1$ is efficient (i.e. it can be implemented by a PPT).

Lemma 7 $\Pr[W_1 \mid \neg BAD]$ *is negligible given the DDH assumption in group* $\mathbb{G}_2$.

Proof: We will condition on the event BAD not happening in Game $\mathbf{G}_1$. We show that if adversary $\mathcal{B}$ can produce a "proof" $\vec{\mathbf{p}}$ for which the pairing test holds and yet the candidate $\vec{\boldsymbol{l}}$ is not in $L_{\mathbf{A}}$, then it implies an efficient adversary that can break DDH in group $\mathbb{G}_2$. So consider a DDH game, where a challenger either provides a real DDH-tuple $\langle \mathbf{g}_2, \hat{b} \cdot \mathbf{g}_2, r \cdot \mathbf{g}_2, \boldsymbol{\chi} = \hat{b}r \cdot \mathbf{g}_2 \rangle$ or a fake DDH tuple $\langle \mathbf{g}_2, \hat{b} \cdot \mathbf{g}_2, r \cdot \mathbf{g}_2, \boldsymbol{\chi} = \hat{b}r' \cdot \mathbf{g}_2 \rangle$.

The QA-NIZK challenger sets $b \cdot \mathbf{g}_2$ to be the same as $\hat{b} \cdot \mathbf{g}_2$ in the description of $\mathbf{G}_1$. Observe that due to our transformations, $\mathbf{CRS}_1$ does not use b at all and $\mathbf{CRS}_2$ can be constructed from $b \cdot \mathbf{g}_2$ alone. 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 $\vec{\boldsymbol{l}} := \left[\begin{array}{c|c} \vec{\boldsymbol{l}}_0 & -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W} \end{array} \right]$, 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 both $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ and $(\vec{\boldsymbol{l}}', \vec{\mathbf{p}}')$ pass the verification equation, we obtain: $\vec{\boldsymbol{l}}'_1 - \vec{\boldsymbol{l}}_1 = b(\vec{\mathbf{p}}' - \vec{\mathbf{p}})$, where $\vec{\boldsymbol{l}}_1 = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$. In particular there exists $i \in [1, s]$, such that, $\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i} = b(\mathbf{p}'_i - \mathbf{p}_i) \neq \mathbf{0}_1$. This gives us a straightforward test for the DDH challenge: $e(\boldsymbol{l}'_{1i} - \boldsymbol{l}_{1i}, r \cdot \mathbf{g}_2) \stackrel{?}{=} e(\mathbf{p}'_i - \mathbf{p}_i, \boldsymbol{\chi})$. □

This concludes our proof of soundness of the QA-NIZK. □

Remark. Observe from the proof above that the soundness can be based on the following *computational* assumption which is implied by XDH. This assumption may be potentially weaker than XDH, which is a *decisional* assumption:

Definition 8 *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 $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The assumption asserts that the following problem is hard: Given $\mathbf{f}, \mathbf{f}^b \xleftarrow{\$} \mathbb{G}_2$, output $\mathbf{h}, \mathbf{h}' \in \mathbb{G}_1$, such that $\mathbf{h}' = \mathbf{h}^b \neq \mathbf{0}_1$.*

C QA-NIZK for Linear Subspaces under the k-Linear Assumption

In this section we generalize our QA-NIZK proof system to be based on the k-linear assumption for any $k \geq 1$. We start off with defining the hardness assumption. We specially mention $DLIN$, which is the case of $k = 2$ since it's a widely used assumption.

Definition 9 (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 $(\mathbf{g}, \mathbf{f}, \mathbf{h}, \mathbf{g}^{x_1}, \mathbf{f}^{x_2}, \mathbf{h}^{x_3})$ and $(\mathbf{g}, \mathbf{f}, \mathbf{h}, \mathbf{g}^{x_1}, \mathbf{f}^{x_2}, \mathbf{h}^{x_1+x_2})$ 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*

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

Definition 10 (k-linear [Sha07, CCS09]) *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* $(\mathbf{g}_1,...,\mathbf{g}_{k+1},\mathbf{g}_1^{x_1},...,\mathbf{g}_{k+1}^{x_{k+1}})$ *and* $(\mathbf{g}_1,...,\mathbf{g}_{k+1},\mathbf{g}_1^{x_1},...,\mathbf{g}_{k+1}^{x_1+...+x_k})$ *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*

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

QA-NIZK Construction. Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order q with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let $\mathbf{g}_1$ and $\mathbf{g}_2$ be randomly chosen generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. We assume that the k-linear problem is hard in the group $\mathbb{G}_2$. The groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are in fact allowed to be the same for $k \geq 2$. In the rest of the section, we adopt the same symbols and conventions as in Section 3.

NIZK CRS: Suppose the language is $L_{\mathbf{A}} = \{\vec{x} \cdot \mathbf{A}^{t\times n} \in \mathbb{G}_1^n \mid \vec{x} \in \mathbb{Z}_q^t\}$. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns. Generate a matrix $\mathsf{D}^{t\times ks}$ with all elements chosen randomly from $\mathbb{Z}_q$ and k elements $b_1,\cdots,b_k$ chosen randomly from $\mathbb{Z}_q$. Let

$$\mathsf{E}^{ks\times ks} \stackrel{\text{def}}{=} \begin{bmatrix} b_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_k \end{bmatrix} \otimes \mathrm{I}^{s\times s}, \quad \mathsf{F}^{s\times ks} \stackrel{\text{def}}{=} \underbrace{\left[\ \mathrm{I}^{s\times s} \ \middle|\ \cdots \ \middle|\ \mathrm{I}^{s\times s}\ \right]}_{k \text{ times}} \cdot \mathsf{E}^{-1}$$

In other words, E is a diagonal matrix with s copies of each of the b_i's in the diagonal. The common reference string (CRS) has two parts $\mathbf{CRS}_1$ and $\mathbf{CRS}_2$ which are to be used by the prover and the verifier respectively.

$$\mathbf{CRS}_1^{t\times ks} = \mathbf{A} \cdot \begin{bmatrix} \mathsf{D} \\ \mathsf{F} \end{bmatrix} \qquad \mathbf{CRS}_2^{(n+ks)\times ks} = \begin{bmatrix} \begin{bmatrix} \mathsf{D} \\ \mathsf{F} \end{bmatrix} \cdot \mathsf{E} \\ -\mathsf{E} \end{bmatrix} \cdot \mathbf{g}_2$$

Prover: Given candidate $\vec{x} \cdot \mathbf{A}$ with witness vector $\vec{x}$, the prover generates the following proof:

$$\vec{\mathbf{p}} := \vec{x} \cdot \mathbf{CRS}_1$$

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

$$e\left(\left[\ \vec{l}\ \middle|\ \vec{\mathbf{p}}\ \right], \mathbf{CRS}_2\right) \stackrel{?}{=} \mathbf{0}_T^{1\times ks}$$

Theorem 11 *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 $\vec{\mathrm{x}} \cdot \mathbf{A}$ (which is a language member), the LHS of the verification equation is:

$$\begin{aligned} & e\left(\left[\begin{array}{c|c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_2\right) \\ = & e\left(\vec{\mathrm{x}} \cdot \mathbf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n \times n} & \begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array} \end{array}\right] \cdot \left[\begin{array}{c} \left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] \cdot \mathsf{E} \\ -\mathsf{E} \end{array}\right], \mathbf{g}_2\right) \\ = & e\left(\vec{\mathrm{x}} \cdot \mathbf{A} \cdot \left(\left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] \cdot \mathsf{E} - \left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] \cdot \mathsf{E}\right), \mathbf{g}_2\right) = e(\mathbf{0}_1^{1 \times ks}, \mathbf{g}_2) = \mathbf{0}_T^{1 \times ks} \end{aligned}$$

Hence completeness follows.

Zero-Knowledge: The CRS is generated exactly as above. In addition, the simulator is given the trapdoor $\left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right]$. Now, given a language candidate $\vec{l}$, the proof is simply $\vec{\mathbf{p}} := \vec{l} \cdot \left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right]$. If $\vec{l}$ is in the language, i.e., it is $\vec{\mathrm{x}} \cdot \mathbf{A}$ for some $\vec{\mathrm{x}}$, 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 two games. Game $\mathbf{G}_0$ just replicates the soundness security definition. In game $\mathbf{G}_1$ the CRS is generated using witness A and its null-space, and this can be done efficiently by the challenger as the distribution is efficiently witness samplable. After this transformation, we show that a verifying proof of a non-language member implies breaking the k-linear assumption in group $\mathbb{G}_2$.

Game $\mathbf{G}_0$: This is just the original system.

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. 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 $\mathsf{D}'^{\,t \times ks}$ with elements randomly chosen from $\mathbb{Z}_q$ and diagonal matrix $\mathsf{E}^{ks \times ks}$ as in the real CRS. Implicity set,

$$\left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] = \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s \times ks} \end{array}\right] + \underbrace{\left[\begin{array}{c|c|c} \mathsf{W} & \cdots & \mathsf{W} \\ \mathrm{I}^{s \times s} & \cdots & \mathrm{I}^{s \times s} \end{array}\right]}_{k \text{ times}} \cdot \mathsf{E}^{-1}$$

where $\mathsf{F}^{s \times ks} \stackrel{\text{def}}{=} \underbrace{\left[\begin{array}{c|c|c} \mathrm{I}^{s \times s} & \cdots & \mathrm{I}^{s \times s} \end{array}\right]}_{k \text{ times}} \cdot \mathsf{E}^{-1}$. Therefore we have,

$$\mathbf{CRS}_1^{t \times ks} = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] = \mathbf{A} \cdot \left(\left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\right] - \underbrace{\left[\begin{array}{c|c|c} \mathsf{W} & \cdots & \mathsf{W} \\ \mathrm{I}^{s \times s} & \cdots & \mathrm{I}^{s \times s} \end{array}\right]}_{k \text{ times}} \cdot \mathsf{E}^{-1}\right) = \mathbf{A} \cdot \left[\begin{array}{c} \mathsf{D}' \\ \mathsf{0}^{s \times ks} \end{array}\right]$$

$$\mathbf{CRS}_2^{(n+ks)\times ks} = \begin{bmatrix} \begin{bmatrix} \mathsf{D} \\ \mathsf{F} \end{bmatrix} \cdot \mathsf{E} \\ -\mathsf{E} \end{bmatrix} \cdot \mathbf{g}_2 = \begin{bmatrix} \begin{bmatrix} \mathsf{D}' \\ 0^{s\times ks} \end{bmatrix} \cdot \mathsf{E} + \underbrace{\left[\begin{array}{c|c|c} \mathsf{W} & \cdots & \mathsf{W} \\ \mathrm{I}^{s\times s} & \cdots & \mathrm{I}^{s\times s} \end{array}\right]}_{k \text{ times}} \\ -\mathsf{E} \end{bmatrix} \cdot \mathbf{g}_2$$

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 $\mathbf{G}_0$.

We show that if adversary $\mathcal{B}$ can produce a "proof" $\vec{\mathbf{p}}$ for which the pairing test holds and yet the candidate $\vec{\boldsymbol{l}}$ is not in $L_{\mathbf{A}}$, then it implies an efficient adversary that can break the k-linear assumption in group $\mathbb{G}_2$.

So now suppose we are given a k-linear challenge distribution

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

in the group $\mathbb{G}_2$, where $\boldsymbol{\chi}$ is either $(\sum_{i=1}^{n} r_i) \cdot \mathbf{g}_2$ or random. We generate the CRS using the challenge components $\mathbf{g}_2, b_1 \cdot \mathbf{g}_2, \cdots, b_k \cdot \mathbf{g}_2$.

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 $\vec{\boldsymbol{l}}\,' := \left[\begin{array}{c|c} \vec{\boldsymbol{l}}_0 & -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W} \end{array}\right]$, 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 both $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ and $(\vec{\boldsymbol{l}}', \vec{\mathbf{p}}')$ pass the verification equation, we obtain:

$$(\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1) \cdot \underbrace{\left[\begin{array}{c|c|c} \mathrm{I}^{s\times s} & \cdots & \mathrm{I}^{s\times s} \end{array}\right]}_{k \text{ times}} = (\vec{\mathbf{p}}' - \vec{\mathbf{p}}) \cdot \mathsf{E},$$

where $\vec{\boldsymbol{l}}_1' = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$. If we represent the ks-element vector $\vec{\mathbf{p}}$ as the sequence of k vectors: $[\vec{\mathbf{p}}_1 \cdots \vec{\mathbf{p}}_k]$, then the above equation implies:

$$\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1 = (\vec{\mathbf{p}}_1' - \vec{\mathbf{p}}_1) b_1 = \cdots = (\vec{\mathbf{p}}_k' - \vec{\mathbf{p}}_k) b_k$$

In particular there exists $i \in [1, s]$, such that,

$$\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i} = (\mathbf{p}_{1i}' - \mathbf{p}_{1i}) b_1 = \cdots = (\mathbf{p}_{ki}' - \mathbf{p}_{ki}) b_k \neq \mathbf{0}_1$$

This gives us a straightforward test for the k-linear challenge:

$$e(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}, \boldsymbol{\chi}) \stackrel{?}{=} \sum_{j=1}^{k} e(\mathbf{p}_{ji}' - \mathbf{p}_{ji}, r_j b_j \cdot \mathbf{g}_2)$$

This concludes our proof of soundness of the QA-NIZK. □

D Proof of QA-NIZK for Tag Based Linear Subspaces

Recall, the languages we handle can be characterized as

$$\left\{ \left\langle \vec{\mathrm{x}} \cdot \left[\begin{array}{c|c} \mathbf{A}^{t\times(n-1)} & (\vec{\mathbf{a}}_1^\top + \mathrm{TAG} \cdot \vec{\mathbf{a}}_2^\top) \end{array} \right], \mathrm{TAG} \right\rangle \mid \vec{\mathrm{x}} \in \mathbb{Z}_q^t, \mathrm{TAG} \in \mathbb{Z}_q \right\}$$

where $\mathbf{A}^{t\times(n-1)} \cdot \mathbf{g}_1, \vec{\mathbf{a}}_1 \cdot \mathbf{g}_1$ and $\vec{\mathbf{a}}_2 \cdot \mathbf{g}_1$ are parameters of the language. Write $\mathbf{A}$ as $[\mathbf{A}_l^{t\times t} \mid \mathbf{A}_r^{t\times(n-1-t)}]$, where w.l.o.g. $\mathbf{A}_l$ is non-singular. While the first $n-1-t$ components in excess of the unknowns, corresponding to $\mathbf{A}_r$, can be verified just as in Section 3, for the last component we proceed as follows. The CRS is generated as:

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

$$\mathbf{CRS}_{2,0}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_1 \\ 1 \\ -b \end{bmatrix} \cdot \mathbf{g}_2 \qquad \mathbf{CRS}_{2,1}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_2 \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{g}_2$$

where D_1 and D_2 are random matrices of order $t \times 1$ independent of the matrix D chosen for proving the other components. The $\mathbb{Z}_q$ element b can be re-used from the other components.

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

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

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

$$e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}}' & \vec{\mathbf{p}} \end{array} \right], \mathbf{CRS}_{2,0} + \mathrm{TAG} \cdot \mathbf{CRS}_{2,1} \right) \stackrel{?}{=} \mathbf{0}_T$$

We now prove completeness, zero-knowledge and give a sketch for soundness.

Completeness: We have,

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

and

$$\mathbf{CRS}_{2,0} + \mathrm{TAG} \cdot \mathbf{CRS}_{2,1} = \begin{bmatrix} b \cdot (\mathsf{D}_1 + \mathrm{TAG} \cdot \mathsf{D}_2) \\ 1 \\ -b \end{bmatrix} \cdot \mathbf{g}_2$$

Therefore,

$$\begin{aligned} & e\left(\left[\begin{array}{c|c} \vec{\boldsymbol{l}}' & \vec{\mathbf{p}} \end{array} \right], \mathbf{CRS}_{2,0} + \mathrm{TAG} \cdot \mathbf{CRS}_{2,1} \right) \\ = \;\; & e\left(\begin{pmatrix} \vec{\mathrm{x}} \cdot \mathbf{A}_l \cdot b \cdot (\mathsf{D}_1 + \mathrm{TAG} \cdot \mathsf{D}_2) + \\ \vec{\mathrm{x}} \cdot (\vec{\mathbf{a}}_1^\top + \mathrm{TAG} \cdot \vec{\mathbf{a}}_2^\top) - \\ \vec{\mathrm{x}} \cdot (\mathbf{A}_l \cdot \mathsf{D}_1 + \mathbf{A}_l \cdot \mathrm{TAG} \cdot \mathsf{D}_2 + (\vec{\mathbf{a}}_1^\top + \mathrm{TAG} \cdot \vec{\mathbf{a}}_2^\top) \cdot b^{-1}) \cdot b \end{pmatrix}, \mathbf{g}_2 \right) = \mathbf{0}_T \end{aligned}$$

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

Soundness: As in the proof of Theorem 1, we compute the CRS's in game $\mathbf{G}_1$ as follows. Let $\mathbf{A}_l = \mathsf{A}_l \cdot \mathbf{g}_1, \vec{\mathbf{a}}_1 = \vec{\mathsf{a}}_1 \cdot \mathbf{g}_1$ and $\vec{\mathbf{a}}_2 = \vec{\mathsf{a}}_2 \cdot \mathbf{g}_1$. Further, let $\begin{bmatrix} \mathsf{W}_1^{t\times 1} \\ 1 \end{bmatrix}$ be the null-space of $\left[\begin{array}{c|c} \mathsf{A}_l & \vec{\mathsf{a}}_1^\top \end{array}\right]$ and let $\begin{bmatrix} \mathsf{W}_2^{t\times 1} \\ 1 \end{bmatrix}$ be the null-space of $\left[\begin{array}{c|c} \mathsf{A}_l & \vec{\mathsf{a}}_2^\top \end{array}\right]$. Then the CRS's in game $\mathbf{G}_1$ are:

$$\mathbf{CRS}_{1,0} := \left[\begin{array}{c|c} \mathbf{A}_l & \vec{\mathbf{a}}_1^\top \end{array}\right] \cdot \left(\begin{bmatrix} \mathsf{D}_1' \\ 0 \end{bmatrix} + \begin{bmatrix} \mathsf{W}_1 \\ 1 \end{bmatrix} \cdot b^{-1} \right) = \left[\begin{array}{c|c} \mathbf{A}_l & \vec{\mathbf{a}}_1^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{D}_1' \\ 0 \end{bmatrix}$$

$$\mathbf{CRS}_{1,1} := \left[\begin{array}{c|c} \mathbf{A}_l & \vec{\mathbf{a}}_2^\top \end{array}\right] \cdot \left(\begin{bmatrix} \mathsf{D}_2' \\ 0 \end{bmatrix} + \begin{bmatrix} \mathsf{W}_2 \\ 1 \end{bmatrix} \cdot b^{-1} \right) = \left[\begin{array}{c|c} \mathbf{A}_l & \vec{\mathbf{a}}_2^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{D}_2' \\ 0 \end{bmatrix}$$

$$\mathbf{CRS}_{2,0}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_1' + \mathsf{W}_1 \\ 1 \\ -b \end{bmatrix} \cdot \mathbf{g}_2 \qquad \mathbf{CRS}_{2,1}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_2' + \mathsf{W}_2 \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{g}_2$$

We now claim that $\vec{\mathsf{w}}^\top \stackrel{\text{def}}{=} \begin{bmatrix} \mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2 \\ 1 \end{bmatrix}$ is the null-space of $\mathsf{A}' \stackrel{\text{def}}{=} \left[\begin{array}{c|c} \mathsf{A}_l & (\vec{\mathsf{a}}_1^\top + \text{TAG} \cdot \vec{\mathsf{a}}_2^\top) \end{array}\right]$. This is because $\vec{\mathsf{w}}^\top$ is a non-zero $t \times 1$ matrix and satisfies:

$$\mathsf{A}' \cdot \vec{\mathsf{w}}^\top = \left[\begin{array}{c|c} \mathsf{A}_l & (\vec{\mathsf{a}}_1^\top + \text{TAG} \cdot \vec{\mathsf{a}}_2^\top) \end{array}\right] \cdot \begin{bmatrix} \mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2 \\ 1 \end{bmatrix} = \mathsf{A}_l \cdot (\mathsf{W}_1 + \text{TAG} \cdot \mathsf{W}_2) + (\vec{\mathsf{a}}_1^\top + \text{TAG} \cdot \vec{\mathsf{a}}_2^\top)$$

$$= \left[\begin{array}{c|c} \mathsf{A}_l & \vec{\mathsf{a}}_1^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{W}_1 \\ 1 \end{bmatrix} + \text{TAG} \cdot \left[\begin{array}{c|c} \mathsf{A}_l & \vec{\mathsf{a}}_2^\top \end{array}\right] \cdot \begin{bmatrix} \mathsf{W}_2 \\ 1 \end{bmatrix} = 0$$

The rest of the proof is similar to the rest of the proof of soundness in Theorem 1, since A' defines the tag-based language.

E Split-CRS QA-NIZKs - Formal Definitions

Definition We call $(\mathsf{K}_0, \mathsf{K}_{11}, \mathsf{K}_{12}, \mathsf{P}, \mathsf{V})$ a **split-CRS QA-NIZK** proof system for an ensemble of distributions $\{\mathcal{D}_\lambda\}$ on collection of witness-relations $\mathcal{R}_\lambda = \{R_\rho\}$ with associated parameter language $\mathcal{L}_{\text{par}}$ if there exists a probabilistic polynomial time simulator $(\mathsf{S}_{11}, \mathsf{S}_{12}, \mathsf{S}_2)$, such that for all non-uniform PPT adversaries $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ we have
Quasi-Adaptive Completeness.

$$\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); (\psi_v, s) \leftarrow \mathsf{K}_{11}(\lambda); \rho \leftarrow \mathcal{D}_\lambda; \psi_p \leftarrow \mathsf{K}_{12}(\lambda, \rho, s); (x, w) \leftarrow \mathcal{A}_1(\lambda, \psi_v, \psi_p, \rho); \\ \pi \leftarrow \mathsf{P}(\psi_p, x, w) : \ \mathsf{V}(\psi_v, x, \pi) = 1 \text{ if } R_\rho(x, w)] = 1$$

Quasi-Adaptive Soundness.

$$\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); (\psi_v, s) \leftarrow \mathsf{K}_{11}(\lambda); \rho \leftarrow \mathcal{D}_\lambda; \psi_p \leftarrow \mathsf{K}_{12}(\lambda, \rho, s); \\ (x, \pi) \leftarrow \mathcal{A}_2(\lambda, \psi_v, \psi_p, \rho) : \ \mathsf{V}(\psi_v, x, \pi) = 1 \ \text{ and } \ \neg(\exists w : R_\rho(x, w))] \approx 0$$

Quasi-Adaptive Zero-Knowledge.

$$\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); (\psi_v, s) \leftarrow \mathsf{K}_{11}(\lambda); \rho \leftarrow \mathcal{D}_\lambda; \psi_p \leftarrow \mathsf{K}_{12}(\lambda, \rho, s) : \mathcal{A}_3^{\mathsf{P}(\psi_p,\cdot,\cdot)}(\lambda, \psi_v, \psi_p, \rho) = 1] \approx$$
$$\Pr[\lambda \leftarrow \mathsf{K}_0(1^m); (\sigma_v, s) \leftarrow \mathsf{S}_{11}(\lambda); \rho \leftarrow \mathcal{D}_\lambda; (\sigma_p, \tau) \leftarrow \mathsf{S}_{12}(\lambda, \rho, s) : \mathcal{A}_3^{\mathsf{S}(\sigma_p,\tau,\cdot,\cdot)}(\lambda, \sigma_v, \sigma_p, \rho) = 1],$$

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

F Proof of Split-CRS QA-NIZK for Affine Spaces

Theorem 12 *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_{11}, \mathsf{K}_{12}, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound split-CRS quasi-adaptive NIZK proof system for affine languages* $\{L_{\mathbf{A},\vec{\mathbf{a}}}\}$ *with parameters 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 DDH assumption holds for group* $\mathbb{G}_2$*.*

The proof of Theorem 12 is similar to that of Theorem 1. We highlight the main points in the proof sketch below.
Proof:

Completeness:

$$\begin{aligned} e\left(\left[\begin{array}{c|c|c} \vec{l} & \vec{\mathbf{p}} \end{array}\right], \mathbf{CRS}_v\right) &= \left[\begin{array}{c|c|c} \vec{\mathrm{x}} & 1 \end{array}\right] \cdot e\left(-\left[\begin{array}{c|c} \mathsf{0}^{t\times n} & \mathsf{0}^{t\times s} \\ \mathsf{0}^{1\times n} & \vec{\mathrm{d}}^{1\times s} \end{array}\right] \cdot \mathbf{g}_1, \left[\begin{array}{c} b\cdot\mathsf{D} \\ \mathsf{I}^{s\times s} \\ -b\cdot\mathsf{I}^{s\times s} \end{array}\right] \cdot \mathbf{g}_2\right) \\ &= \left[\begin{array}{c|c|c} \vec{\mathrm{x}} & 1 \end{array}\right] \cdot e\left(\mathbf{g}_1, \left[\begin{array}{c} \mathsf{0}^{t\times s} \\ b\cdot\vec{\mathrm{d}}^{1\times s} \end{array}\right] \cdot \mathbf{g}_2\right) \\ &= \vec{\mathbf{f}} \end{aligned}$$

Zero Knowledge: This is straight-forward with the simulator retaining trapdoors D, $\vec{\mathrm{d}}$, and b.

Soundness: As in the proof of Theorem 1, we compute the CRS's in game $\mathbf{G}_1$ as follows. Compute $\left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right]$ of dimension $(t+s)\times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A}\cdot\left[\begin{array}{c} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{array}\right] = \mathsf{0}^{t\times s}$.

Next, the NIZK CRS is computed as follows: The challenger generates matrix $\mathsf{D'}^{\,t\times s}$ with elements randomly chosen from $\mathbb{Z}_q$ and element b randomly chosen from $\mathbb{Z}_q$ (just as in the real CRS). Now set,

$$\left[\begin{array}{c} \mathsf{D} \\ b^{-1}\cdot\mathrm{I}^{s\times s} \end{array}\right] = \left[\begin{array}{c} \mathsf{D'} \\ \mathsf{0}^{s\times s} \end{array}\right] + b^{-1}\cdot\left[\begin{array}{c} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{array}\right]$$

Also choose $\vec{\mathrm{d}}_1$ at random and set

$$\vec{\mathrm{d}} = \vec{\mathrm{d}}_1 \; - \; \vec{\mathrm{a}}\cdot\left(\left[\begin{array}{c} \mathsf{D'} \\ 0 \end{array}\right] + b^{-1}\cdot\left[\begin{array}{c} \mathsf{W} \\ 1 \end{array}\right]\right)$$

Then, $\vec{\mathbf{f}}$ can be computed as

$$e\left(\mathbf{g}_1, b\cdot\vec{\mathrm{d}}_1\cdot\mathbf{g}_2 - \vec{\mathrm{a}}\cdot b\cdot\begin{bmatrix}\mathsf{D}'\\ 0\end{bmatrix}\cdot\mathbf{g}_2 - \vec{\mathrm{a}}\cdot\begin{bmatrix}\mathsf{W}\\ \mathrm{I}\end{bmatrix}\cdot\mathbf{g}_2\right)$$

Further $\mathbf{CRS}_p$ can be computed as

$$\mathbf{CRS}_p^{t\times s} = \begin{bmatrix}\mathbf{A}^{t\times n}\\ \vec{\mathbf{a}}^{1\times n}\end{bmatrix}\cdot\begin{bmatrix}\mathsf{D}'\\ 0^{s\times s}\end{bmatrix} - \begin{bmatrix}0^{t\times s}\\ \vec{\mathrm{d}}^{1\times s}\end{bmatrix}\cdot\mathbf{g}_1$$

Rest of the proof is as in the proof of Theorem 1, but crucially noting that in the proof of Lemma 7 while employing DDH in group $\mathbb{G}_2$, the challenge value $b\cdot\mathbf{g}_2$ suffices to simulate all occurences of b in both the CRS-es (including $\vec{\mathbf{f}}$).

□

G Application Details

Proof of the Signature Scheme Theorem.
Theorem 3 If $\mathcal{E}$ is a labeled CCA2-encryption scheme and $\mathcal{Q}$ is a split-CRS quasi-adaptive NIZK system for distribution $\mathcal{D}$ on class of languages $\{L_\rho\}$ described above, then the signature scheme described above is existentially unforgeable under adaptive chosen message attacks.
Proof: Recall the security game for a signature scheme. Once the signature scheme's public key is given to the signature-scheme adversary $\mathcal{B}$, it adaptively obtains several signatures $\langle c_i, \pi_i\rangle$ on messages M_i of its choosing. Let T denote the set of all such messages M_i. To win the game, $\mathcal{B}$ must obtain a $\langle M^*, c^*, \pi^*\rangle$ ($M^* \notin T$) which passes the public signature verification, which in this case just means that the claimed proof π^* of (c^*, M^*) being in L_ρ (where $\rho = (\mathbf{u}, \mathcal{E}.\mathsf{pk})$)) passes the QA-NIZK verifier V using the CRS ψ_v. Let W be the event that $\mathcal{B}$ wins. By soundness of the QA-NIZK, it follows that $\Pr[W]$ is at most the probability that (c, M) is in L_ρ plus a negligible amount.

To show that $\Pr[W]$ is negligible consider the following experiments:

Expt$_1$: The challenger generates the signature scheme public key $\mathcal{S}.\mathsf{pk}(=\psi_v)$ just as in the signature scheme described above, and passes it to $\mathcal{B}$. Apart from retaining the secret key $\mathcal{S}.\mathsf{sk} = (\mathbf{u}, \mathcal{E}.\mathsf{pk}, \psi_p)$, the challenger *also* retains the secret key $\mathcal{E}.\mathsf{sk}$ generated by KeyGen of $\mathcal{E}$. It then adaptively answers multiple requests for signatures on M_i by encrypting $\mathbf{u}$ with labels M_i (using $\mathcal{E}$'s encryptor Enc with key $\mathcal{E}.\mathsf{pk}$) and generating proofs π_i using ψ_p and QA-NIZK Prover P. The view of $\mathcal{B}$ is identical so far to that in the signature scheme security game. When the adversary $\mathcal{B}$ replies with a triple $\langle M^*, c^*, \pi^*\rangle$, the challenger decrypts c^* with label M^* using secret key $\mathcal{E}.\mathsf{sk}$ to get u^*. If $u^* = \mathbf{u}$ the challenger outputs WIN, otherwise it outputs LOSE. Let W_1 be the event that challenger outputs WIN. By correctness of the encryption scheme $\mathcal{E}$, the event W_1 happens whenever c^* is an encryption of $\mathbf{u}$ with label M^* under $\mathcal{E}.\mathsf{pk}$, i.e. whenever $(c^*, M^*($ are in L_ρ(where $\rho = (\mathbf{u}, \mathcal{E}.\mathsf{pk})$). Thus, $\Pr[W]$ is at most $\Pr[W_1]$ plus a negligible amount.

Expt$_2$: This is same as **Expt**$_1$ except that the Challenger generates the QA-NIZK CRS-es (and trapdoor) σ_v using S_{11} and σ_p, τ using S_{12}. Further, it generates all the proofs using $\mathsf{S}_2(\sigma_p, \tau, \cdot)$. Let W_2 be the event that challenger outputs WIN. By QA-NIZK zero-knowledge, $|\Pr[W_2] - \Pr[W_1]|$ is negligible.

Expt$_3$: This is same as **Expt**$_2$ except that the challenger now encrypts 1 instead of $\mathbf{u}$. Let W_3 be the event that challenger outputs WIN. By CCA-2 security of the encryption scheme $\mathcal{E}$, it follows that $|\Pr[W_3] - \Pr[W_2]|$ is negligible. Techincally, this requires a sequence of hybrid experiments, with each subsequent experiment replacing $\mathbf{u}$ by 1 in the next signature request of $\mathcal{B}$.

Now, note that in **Expt**$_3$, $\Pr[W_3]$ is at most $1/|X_m|$ as the view of the adversary $\mathcal{B}$ is independent of $\mathbf{u}$. Thus, by hypothesis about X_m , $\Pr[W_3]$ is negligible. It follows that $\Pr[W]$ is negligible as well. □

Couple of remarks are in order here. If we did not have a split-CRS QA-NIZK, but a QA-NIZK where the verifier also needed a CRS that depended on ρ, then in **Expt**$_3$ above the view of the Adversary $\mathcal{B}$ would depend on $\mathbf{u}$. In such a case, one can still get a signature scheme (as in [CCS09]) but one has to encrypt a hard to compute challenge such as $x \cdot \mathbf{u}$ (given $\mathbf{u}$, $\mathbf{g}$ and $x \cdot \mathbf{g}$). However, the size of the QA-NIZK proof and hence the signature would not increase as although the number of equations to prove would go up by one, but so would the number of variables (note the additional variable x).

UC Adaptive Commitments in the Erasure Model. Here we instantiate the scheme due to Fischlin, Libert and Manulis [FLM11] in our QA-NIZK tag-based linear subspace proof system. The following consturction is under the SXDH assumption.

Consider the tag-based language L_ρ, with tag t,

$$\exists r. \left(\begin{array}{c} R = r \cdot \mathbf{g}, S = r \cdot \mathbf{h}, \\ T = r \cdot K_1, H = r \cdot (\mathbf{d}_1 + t \cdot \mathbf{e}_1) \end{array} \right)$$

with parameter ρ being $\mathbf{h}, \mathbf{d}_1, \mathbf{e}_1$, and with the distribution on the parameters being that they are chosen randomly and uniformly (as in the Cramer-Shoup Key Generation). We can assume that $\mathbf{g}$ is part of the Group description, and is chosen randomly as part of group generation. Consider a QA-NIZK $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ for the above distribution of (tag-based) linear languages.

UC CRS-Gen(λ)**:**

$$\mathbf{g}, \mathbf{h}, K_1, \mathbf{d}_1, \mathbf{e}_1, \mathsf{K}_0(\mathbf{h}, \mathbf{d}_1, \mathbf{e}_1)$$

Commit$(crs, M, sid, cid, P_i, P_j)$**:** to commit to message $M \in \mathcal{G}$ for party P_j upon receiving a command $(commit, sid, cid, P_i, P_j, M)$, party P_i proceeds as follows:

1. Generate $r \xleftarrow{\$} \mathbb{Z}_q$. Compute a Cramer-Shoup Encryption of M as follows:

   $$R = r \cdot \mathbf{g}, S = r \cdot \mathbf{h}, T = M + r \cdot K_1, H = r \cdot (\mathbf{d}_1 + t \cdot \mathbf{e}_1)$$

   where t is the tag generated using a collision-resitant hash function just as in Cramer-Shoup encryption.

2. Generate QA-NIZK proof (using P) π of:

   $$\exists r. \left(\begin{array}{c} R = r \cdot \mathbf{g}, S = r \cdot \mathbf{h}, \\ T - M = r \cdot K_1, H = r \cdot (\mathbf{d}_1 + t \cdot \mathbf{e}_1) \end{array} \right)$$

   with witness r.

3. Keep π and erase r.
4. Commitment is $c = (R, S, T, H)$: $\boxed{\text{4 group elements}}$

Open$(crs, M, sid, cid, P_i, P_j)$**:** Reveal M and π, which is $(4-1)*1 = \boxed{\text{3 group elements}}$.

As the proof is for $(T - M)$ it can be shown that it suffices to hide M with the hash key itself (see a similar remark for the signature scheme), which leads to a commitment consisting of three elements, and a proof (opening) consisting of another two elements. A similar scheme using QA-NIZKs, and under the DLIN assumption leads to a commitment consisting of 4 elements and an opening of another 4 elements, whereas [FLM11] stated a scheme using Groth-Sahai NIZK proofs requiring $(5 + 16)$ elements.

H Dual System IBE under SXDH Assumption

We first consider the QA-NIZK for the affine language (incorporating tags)

$$\langle R = r \cdot \mathbf{g}_2,\ S = r \cdot \mathbf{f}\,,\ T = \mathbf{u} + r \cdot (\mathbf{d} + i \cdot \mathbf{e})\rangle$$

where i is an identity, and can be viewed as a tag. More precisely, the affine-system is given by

$$L_\rho = \{r \cdot (\begin{bmatrix} \mathbf{g}_2 & \mathbf{f} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \mathbf{d} \end{bmatrix} + i \cdot \begin{bmatrix} 0 & 0 & \mathbf{e} \end{bmatrix}) + \begin{bmatrix} 0 & 0 & \mathbf{u} \end{bmatrix} \mid r \in \mathbb{Z}_q\}$$

where ρ consists of the matrices $\begin{bmatrix} \mathbf{g}_2 & \mathbf{f} \end{bmatrix}$ and $\begin{bmatrix} 0 & 0 & \mathbf{u} \end{bmatrix}$ (affine shift), and group elements $\mathbf{d}$ and $\mathbf{e}$ (for defining the tag based last component). Note that T corresponds to the language component that depends on a tag. So, let's focus on the components $\langle R, S\rangle$ first. In the notation of Section 3, this is a language with rank one, and two dimensions, i.e. $n = 2$, $t = 1$ and $s = (n - t) = 1$. Let $\mathbf{f} = \mathbf{g}_2^c$ for some $c \in \mathbb{Z}_q$. Then the matrix A is $\begin{bmatrix} 1 & c \end{bmatrix}$. Further its null-space is generated by $\begin{bmatrix} -c & 1 \end{bmatrix}$.

For the IBE scheme, instead of generating the CRS as in Section 3 for the above language, we will generate the CRS as in game $\mathbf{G}_1$ in the proof of soundness of QA-NIZK (see Appendix B), as this will be more in line with the original construction of Waters, and hence possibly easier to relate. Thus, the two CRS-es are generated by choosing a matrix D' of dimension $t \times s$, which in this case is just one element. This single element in D' will be called Δ_3 in the IBE scheme below. The $\mathbf{CRS}_1$ (prover CRS) is then specified by $\mathsf{A} \cdot \mathbf{g}_2$ and $\Delta_3 \cdot \mathbf{g}_2$. Recall, the prover CRS is to be used in KeyGen in IBE.

The verifier CRS, i.e. $\mathbf{CRS}_1$ is specified by $\mathbf{g}_1$, $b \cdot \mathbf{g}_1$ and $(b \cdot \Delta_3 - c) \cdot \mathbf{g}_1$. Similarly, the CRS-es for the tag based element T, and the affine shift $\mathbf{u}$ can be obtained from Sections 4 and 4 resp. The element T will require single element matrices D'_1 and D'_2 (for $\mathbf{d}$ and $\mathbf{e}$ resp.), which will be called Δ_1 and Δ_2 respectively (see Appendix D). Similarly, using Appendix 4, we derive the CRS element required for the affine shift, which will be $e(\mathbf{g}_1, (b \cdot \Delta_4 - u) \cdot \mathbf{g}_2)$ (see the vector $\vec{\mathbf{f}}$ in Appendix 4, and note we want the representation corresponding to the simulation of game $\mathbf{G}_1$ in the soundness proof). That completes the description of how we intend to setup the CRS-es in the IBE using the QA-NIZK for the above language.

Now, the verifier CRS needs to be randomized to represent IBE ciphertexts, and hence each ciphertext is a scaling of the verifier CRS by a $\mathbb{Z}_q$ scalar s (as in game $\mathbf{G}_2$ of the soundness proof in section 3). Also, there is one variable r, and two equations in excess of the variables, and hence

the verification requires testing two pairing product equations – which is a problem as mentioned in Section 5. The two pairing product equation tests can be converted into one by taking a linear combination with a random public tag, and this gives us the final form of the ciphertext. The (fully secure) IBE scheme so obtained is described below, along with a proof of security. For a security definition of fully secure IBE we refer the reader to [Wat09].

For ease of reading, we switch to multiplicative group notation in the following.

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, b, d, e, u from $\mathbb{Z}_q$, and publishes the following public key **PK**:
$\mathbf{g}_1$, $\mathbf{g}_1^b$, $\mathbf{v}_1 = \mathbf{g}_1^{-\Delta_1 \cdot b + d}$, $\mathbf{v}_2 = \mathbf{g}_1^{-\Delta_2 \cdot b + e}$, $\mathbf{v}_3 = \mathbf{g}_1^{-\Delta_3 \cdot b + c}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta_4 \cdot b + u}$.
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, d, e, u.

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

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

and the ciphertext is $C = \langle C_0, C_1, C_2, C_3, \text{TAG} \rangle$.

KeyGen(MSK, i): The authority chooses r at random from $\mathbb{Z}_q$ and creates

$$R = \mathbf{g}_2^r, S = \mathbf{g}_2^{r \cdot c}, T = \mathbf{g}_2^{u + r \cdot (d + i \cdot e)},\ W_1 = \mathbf{g}_2^{-\Delta_4 - r \cdot (\Delta_1 + i \cdot \Delta_2)}, W_2 = \mathbf{g}_2^{-r \cdot \Delta_3}$$

as the secret key K_i for identity i.

Decrypt(K_i, C): Let TAG be the tag in C. Obtain

$$\kappa = \frac{e(C_1, S^{\text{TAG}} \cdot T) \cdot e(C_2, W_1 \cdot W_2^{\text{TAG}})}{e(C_3, R)}$$

and output C_0 / κ.

Theorem 13 *Under the SXDH Assumption, the above scheme is a fully-secure IBE scheme.*

Proof: We will just show that $\mathbf{k}^s$ (as used in blinding the plaintext M) is distributed randomly in the view of an adaptive Adversary, who after obtaining the public key, adaptively obtains secret keys for multiple identities $i_1, i_2, ..., i_n$, and a ciphertext for identity i (where all the identities are chosen adaptively by the Adversary, and i is different from the secret key identities). The ciphertext can be obtained by the Adversary at any stage.

We will consider a sequence of games, and show that the Adversary's view is either statistically or computationally indistinguishable between any two consecutive games. Game G_0 is same as the actual adaptive security IBE game above.

Game G_1: In this game the challenger behaves exactly like the authority while publishing the **PK**, and while generating the secret keys. However, it picks another random value s' from $\mathbb{Z}_q$, and outputs the following as ciphertext (for identity i):

$$\begin{aligned} C_0 &= M \cdot \mathbf{k}^s \cdot e(\mathbf{g}_1, \mathbf{g}_2)^{u \cdot s'}, \\ C_1 &= \mathbf{g}_1^{s+s'}, C_2 = \mathbf{g}_1^{b \cdot s}, \\ C_3 &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{g}_1^{(d + i \cdot e + \text{TAG} \cdot c) s'} \end{aligned} \tag{1}$$

The tag TAG is chosen randomly as in game G_0. This simulation of the ciphertext is called *semi-functional ciphertext* in [Wat09]. Intuitively, from the point of view of QA-NIZK proofs, the semi-functional ciphertext provides simulation-soundness as the null-space of the language is reflected as a factor (linear combination in additive notation)"shifted" by s'.

The view of the Adversary in games G_0 and G_1 is computationally indistinguishable by employing the DDH assumption in group $\mathbb{G}_1$ on the tuples $\langle \mathbf{g}_1, \mathbf{g}_1^b, \mathbf{g}_1^{bs}, \mathbf{g}_1^s \rangle$, and $\langle \mathbf{g}_1, \mathbf{g}_1^b, \mathbf{g}_1^{bs}, \mathbf{g}_1^{s+s'} \rangle$. The former tuple is used in game G_0 and the latter in game G_1. Note that the order of the last two components in the DDH tuples is switched from usual formulation of DDH; however, it is easy to see that this formulation is equivalent to the usual DDH.

Game G_2: In this game the challenger chooses Δ_1', Δ_2', Δ_3', Δ_4' at random and sets $\Delta_1 = (\Delta_1' + d)/b$, $\Delta_2 = (\Delta_2' + e)/b$, $\Delta_3 = (\Delta_3' + c)/b$, $\Delta_4 = (\Delta_4' + u)/b$. Thus, the **PK** is now output as $\mathbf{g}_1$, $\mathbf{g}_1^b$, $\mathbf{v}_1 = \mathbf{g}_1^{-\Delta_1'}$, $\mathbf{v}_2 = \mathbf{g}_1^{-\Delta_2'}$, $\mathbf{v}_3 = \mathbf{g}_1^{-\Delta_3'}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta_4'}$.
Further, the secret keys are output as

$$\begin{aligned} R &= \mathbf{g}_2^r, S = \mathbf{g}_2^{r \cdot c}, T = \mathbf{g}_2^{u + r \cdot (d + i \cdot e)}, \\ W_1 &= \mathbf{g}_2^{[-\Delta_4' - u - r \cdot (\Delta_1' + d + i \cdot (\Delta_2' + e))]/b}, \\ W_2 &= \mathbf{g}_2^{-r \cdot (\Delta_3' + c)/b}. \end{aligned} \tag{2}$$

The view of the Adversary in games G_2 and G_1 is statistically identical.

Game G_3: This game is actually a sequence of several hybrid games, with the j-th hybrid game $G_{3,j}$ changing the simulation of the j-th secret key generation. Game $G_{3,0}$ is just the same as game G_2.

In game $G_{3,j}$ the challenger modifies the output of the j-th secret key as follows (assume that the identity requested by the Adversary is i_j): it chooses r_j, r_j' and r_j'' at random and sets

$$\begin{aligned} R &= \mathbf{g}_2^{r_j}, S = \mathbf{g}_2^{r_j \cdot c + r_j'}, \\ T &= \mathbf{g}_2^{r_j'' + r_j \cdot (d + i_j \cdot e)}, \\ W_1 &= \mathbf{g}_2^{[-\Delta_4' - r_j'' - r_j \cdot (\Delta_1' + d + i_j \cdot (\Delta_2' + e))]/b}, \\ W_2 &= \mathbf{g}_2^{(-r_j' - r_j \cdot (\Delta_3' + c))/b}. \end{aligned}$$

Note that u has completely vanished from the j-th (and earlier) secret key simulation. This simulation of the secret key is called *semi-functional key.*

Lemma 14 *The view of the Adversary in game $G_{3,j}$ is computationally indistinguishable from the view of the Adversary in game $G_{3,j-1}$.*

Proof:

Let H_0 be same as the game $G_{3,j-1}$. In game H_1, the challenger chooses $d = d_1 + c \cdot d_2$, and $e = e_1 + c \cdot e_2$, and tag TAG in the ciphertext as $-(d_2 + i \cdot e_2)$. where d_1, d_2, e_1 and e_2 are random and independent values from $\mathbb{Z}_q$. It is easy to see that d, e and TAG are random and independent, and hence the view of the Adversary in games H_0 and H_1 is statistically identical. Note that with this value of TAG, C_3 (in the ciphertext) can be generated by the challenger as

$$\begin{aligned} C_3 &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{g}_1^{(d_1 + i \cdot e_1 + (d_2 + i \cdot e_2) \cdot c + \text{TAG} \cdot c) s'} \\ &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{g}_1^{(d_1 + i \cdot e_1) s'} \end{aligned}$$

As a consequence c is not used at all in the simulation of the ciphertext (whose elements are all in group $\mathbb{G}_1$). The simulation of PK (without using c) is unchanged from game G_2.

In game H_2, the challenger generates the j-th secret-key by choosing r_j and r'_j uniformly and independently and setting

$$\begin{aligned} &R = \mathbf{g}_2^{r_j}, S = \mathbf{g}_2^{r_j \cdot c + r'_j}, \\ &T = \mathbf{g}_2^{u + r_j \cdot (d_1 + c \cdot d_2 + i_j \cdot (e_1 + c \cdot e_2)) + r'_j \cdot (d_2 + i_j e_2)} \\ &W_1 = \mathbf{g}_2^{[-\Delta'_4 - u - r_j \cdot (\Delta'_1 + d_1 + c d_2 + i_j \cdot (\Delta'_2 + e_1 + c e_2)) - r'_j (d_2 + i_j e_2)]/b}, \\ &W_2 = \mathbf{g}_2^{(-r_j \cdot (\Delta'_3 + c) - r'_j)/b}. \end{aligned}$$

Recall that in game H_1, the secret key is being generated as in Equation (2), with $d = d_1 + cd_2$ and $e = e_1 + ce_2$. The view of the Adversary in games H_2 and H_1 is computationally indistinguishable, and this is shown by employing the DDH assumption on the two tuples $\langle \mathbf{g}_2, \mathbf{g}_2^c, \mathbf{g}_2^{r_j}, \mathbf{g}_2^{cr_j} \rangle$ and $\langle \mathbf{g}_2, \mathbf{g}_2^c, \mathbf{g}_2^{r_j}, \mathbf{g}_2^{cr_j + r'_j} \rangle$, where the first tuple is employed in simulating game H_1 and the second tuple is used in simulating game H_2.

In game H_3, the challenger generates the j-th secret key as

$$\begin{aligned} &R = \mathbf{g}_2^{r_j}, S = \mathbf{g}_2^{r_j \cdot c + r'_j}, \\ &T = \mathbf{g}_2^{u + r_j \cdot (d_1 + c \cdot d_2 + i_j \cdot (e_1 + c \cdot e_2)) + r'_j \cdot r''_j} \\ &W_1 = \mathbf{g}_2^{[-\Delta'_4 - u - r_j \cdot (\Delta'_1 + d_1 + c d_2 + i_j \cdot (\Delta'_2 + e_1 + c e_2)) - r'_j \cdot r''_j]/b}, \\ &W_2 = \mathbf{g}_2^{(-r_j \cdot (\Delta'_3 + c) - r'_j)/b}. \end{aligned}$$

where r_j, r'_j and r''_j are chosen randomly and independently (and independently from all other variables). Note that d and e are also chosen independently and randomly (back as in game H_0). Moreover, TAG is also chosen at random, and C_3 output just as in game H_0, i.e. $\mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{g}_1^{(d + i \cdot e + \text{TAG} \cdot c) s'}$.

The view of the Adversary in game H_3 and H_2 is statistically identical by noting that $d = d_1 + c \cdot d_2$, and $e = e_1 + c \cdot e_2$, $\text{TAG} = -(d_2 + i \cdot e_2)$ and $r''_j = d_2 + i_j e_2$ are all random and independent (since $i \neq i_j$). This can be seen by noting that the four by four matrix of coefficients of d, e, TAG, r''_j in their linear representation in terms of d_1, d_2, e_1, e_2 is non-singular.

In game H_4, the challenger generates d, e and TAG at random (instead of $d_1 + cd_2$ etc.), and also chooses r'''_j at random (and independent of r_j, r'_j and other variables) and outputs the ciphertext as

$$\begin{aligned} R &= \mathbf{g}_2^{r_j}, S = \mathbf{g}_2^{r_j \cdot c + r'_j}, \\ T &= \mathbf{g}_2^{r'''_j} \\ W_1 &= \mathbf{g}_2^{[-\Delta'_4 - r'''_j - r_j \cdot (\Delta'_1 + i_j \cdot \Delta'_2)]/b}, \\ W_2 &= \mathbf{g}_2^{(-r_j \cdot (\Delta'_3 + c) - r'_j)/b}. \end{aligned}$$

Game H_4 is statistically identical to game H_3, as $(u + r'_j \cdot r''_j + r_j \cdot (d + i_j \cdot e))$ in game H_3 is random and independent of r'_j, and hence is distributed same as a random r'''_j as in game H_4. Now note that game H_4 is identical to the game $G_{3,j}$ as described above the lemma 14 statement. □

We now continue with the proof of the theorem. Game G_4 is just the game $G_{3,n}$ (where n is the number of secret key requests). Note that in game G_4 the only place that u is used is in the ciphertext component C_0 which is simulated by the challenger as $C_0 = M \cdot \mathbf{k}^s \cdot e(\mathbf{g}_1, \mathbf{g}_2)^{us'}$ (see equation (1)). Hence, C_0 is completely random and independent of M in the view of the Adversary in game G_4 (note u is non-zero with high probability). That completes the proof.

We also claim that the ciphertext is anonymity preserving[8]. This is because in game H_4, the component C_3 is randomized by d and e which do not appear elsewhere and hence the ciphertext is independent of the identity i. □

I Publicly Verifiable CCA2-IBE under SXDH Assumption

The definition of CCA2-secure encryption [BDPR98] naturally extends to the Identity-based encryption setting [BCHK07]. We stress that we prove fully adaptive security, i.e. the Adversary can choose the identity for which it invokes the encryption oracle adaptively. Our scheme is also publicly-verifiable CCA2 secure. This is an informal property, and should not be confused with plaintext-awareness. In practical terms, most CCA2-secure schemes have a decryptor that first does a consistency test on the ciphertext. Only if the ciphertext passes the consistency test, does it actually decrypt the ciphertext. If this consistency test can be done publicly, i.e. without the secret key, then the scheme is called *publicly-verifiable* CCA2-secure. In the IBE setting, the consistency test also checks if the ciphertext is valid for the particular claimed identity. This property can be useful in filtering inconsistent ciphertexts at a network gateway itself. Publicly-verifiable CCA2 encryption schemes were also used to obtain password-based key exchange schemes in [KV11, JR12]. Similar applications may be possible for publicly-verifiable CCA2-IBE schemes.

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 $\mathbf{g}_2$ and $\mathbf{g}_1$ 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 $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta_5 \cdot b + u}$.

[8] While our IBE scheme was obtained independently of [CLL+12] in 2012, we observed the anonymity property only after someone pointed us to the anonymity property of the latter. Thus the credit for the first anonymous IBE under standard static assumptions goes to [CLL+12] alone.

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 $\langle C_0, C_1, C_2, C_3, i, \text{TAG}, h\rangle \in L$.

KeyGen(MSK, i): The authority chooses r at random from $\mathbb{Z}_q$ and creates

$$R = \mathbf{g}_2^r, S_1 = \mathbf{g}_2^{r \cdot c}, S_2 = \mathbf{g}_2^{r \cdot z}, T = \mathbf{g}_2^{u + r \cdot (d + i \cdot e)},$$
$$W_1 = \mathbf{g}_2^{-\Delta_5 - r \cdot (\Delta_1 + i \cdot \Delta_2)}, W_2 = \mathbf{g}_2^{-r \cdot \Delta_3}, W_3 = \mathbf{g}_2^{-r \cdot \Delta_4}$$

as the secret key K_i for identity i.

Decrypt(K_i, C): Let TAG be the tag in C. Let $h = \mathcal{H}(C_0, C_1, C_2, \text{TAG}, i)$. First (publicly) verify that the ciphertext satisfies the QA-NIZK for the language above. Then, obtain

$$\kappa = \frac{e(C_1, S_1^{\text{TAG}} \cdot S_2^h \cdot T) \cdot e(C_2, W_1 \cdot W_2^{\text{TAG}} \cdot W_3^h)}{e(C_3, R)}$$

and output C_0/κ. If the QA-NIZK does not verify, output $\perp$.

This public-verifiability of the consistency test is informally called the publicly-verifiable CCA2 security.

Theorem 15 *Under the SXDH Assumption, the above scheme is a CCA2 fully-secure IBE scheme.*

Proof: We will just show that $\mathbf{k}^s$ (as used in blinding the plaintext M) is distributed randomly in the view of an adaptive Adversary, who after obtaining the public key, adaptively obtains secret keys for multiple identities $i_1, i_2, ..., i_n$, and a ciphertext for identity i (where all the identities are chosen adaptively by the Adversary, and i is different from the secret key identities). The ciphertext can be obtained by the Adversary at any stage.

We will consider a sequence of games, and show that the Adversary's view is either statistically or computationally indistinguishable between any two consecutive games. Game G_0 is same as the actual adaptive security IBE game above.

Game G_1: In this game the challenger behaves exactly like the authority while publishing the **PK**, and while generating the secret keys, as well as generating the ciphertext (for identity i). It also behaves the same for serving decryption requests, except that if the QA-NIZK verification fails then the the challenger wins.

The probability of the Adversary winning in game G_0' is no less than the probability of the Adversary winning in game G_0 since the Adversary can itself check that a proof is not going to verify, and hence just not make such a query. Moreover, in game G_0 the adversary gets no additional information from the challenger when the verification (and hence decryption) fails. Thus, the view of an Adversary which does not make such calls is identical to the view of an adversary that makes such a call in game G_0.

Game G_2: Recall that in the real world (and game G_1), the challenger wins (outright) if the Adversary supplies a ciphertext for decryption which is identical to the ciphertext output by the challenger, and if the identity is also the same. In game G_2 the challenger wins if the hash h computed (using $\mathcal{H}$ as above) on the the Adversary supplied ciphertext is same as the hash computed on the ciphertext output by the challenger, and the idenity is same. The probability of the Adversary winning in game G_2 is no less than the probability of the Adversary winning in game G_1 since if the hash is same, and the idenitiy is same, and the QA-NIZK verifies, then it implies that C_3 is also identical in the two ciphertexts. This further implies that the proofs are identical, as the proof is uniquely determined once the language components are set[9].

Game G_3: Recall that the decryption requests for identity j are served by obtaining

$$\kappa = \frac{e(C_1, S_1^{\text{TAG}} \cdot S_2^h \cdot T) \cdot e(C_2, W_1 \cdot W_2^{\text{TAG}} \cdot W_3^h)}{e(C_3, R)}$$

where

$$R = \mathbf{g}_2^r, S_1 = \mathbf{g}_2^{r \cdot c}, S_2 = \mathbf{g}_2^{r \cdot z}, T = \mathbf{g}_2^{u + r \cdot (d + i \cdot e)},\ W_1 = \mathbf{g}_2^{-\Delta_5 - r \cdot (\Delta_1 + i \cdot \Delta_2)}, W_2 = \mathbf{g}_2^{-r \cdot \Delta_3}, W_3 = \mathbf{g}_2^{-r \cdot \Delta_4}$$

is fixed for identity j by choosing r at random. However, in game G_3, each decryption request is served by choosing this r freshly at random. This is identical to the real world game, since the decryption oracle first verifies the QA-NIZK, which guarantees that C_1, C_2, C_3 are of the correct form. This ensures that κ is independent of the value of r, and hence a fresh value r can be chosen for each decryption request. Thus, the view of the Adversary in games G_2 and G_3 is identical.

Game G_4: In this game the challenger behaves exactly like in game G_3, except that it picks another random value s' from $\mathbb{Z}_q$, and outputs the following as ciphertext (for identity i):

$$\begin{aligned} C_0 &= M \cdot \mathbf{k}^s \cdot e(\mathbf{g}_1, \mathbf{g}_2)^{u \cdot s'}, \\ C_1 &= \mathbf{g}_1^{s+s'}, C_2 = \mathbf{g}_1^{b \cdot s}, \\ C_3 &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{v}_4^{h \cdot s} \cdot \mathbf{g}_1^{(d + i \cdot e + \text{TAG} \cdot c + h \cdot z) s'} \end{aligned} \tag{3}$$

The tag TAG is chosen randomly as in game G_0 (and G_1).

The view of the Adversary in games G_3 and G_4 is computationally indistinguishable by employing the DDH assumption in group $\mathbb{G}_2$ on the tuples $\langle \mathbf{g}_1, \mathbf{g}_1^b, \mathbf{g}_1^{bs}, \mathbf{g}_1^s \rangle$, and $\langle \mathbf{g}_1, \mathbf{g}_1^b, \mathbf{g}_1^{bs}, \mathbf{g}_1^{s+s'} \rangle$. The former tuple is used in game G_3 and the latter in game G_4.

Game G_5: In this game the challenger chooses Δ_1', Δ_2', Δ_3', Δ_4',Δ_5' at random and sets $\Delta_1 = (\Delta_1' + d)/b$, $\Delta_2 = (\Delta_2' + e)/b$, $\Delta_3 = (\Delta_3' + c)/b$, $\Delta_4 = (\Delta_4' + z)/b$, $\Delta_5 = (\Delta_5' + u)/b$. Thus, the **PK** is now output as

[9]This property can be proven by checking the structure of the CRS for prover and verifier in the tag based system also.

$\mathbf{g}_1$, $\mathbf{g}_1^b$, $\mathbf{v}_1 = \mathbf{g}_1^{-\Delta'_1}$, $\mathbf{v}_2 = \mathbf{g}_1^{-\Delta'_2}$, $\mathbf{v}_3 = \mathbf{g}_1^{-\Delta'_3}$, $\mathbf{v}_4 = \mathbf{g}_1^{-\Delta'_4}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta'_5}$.
Further, the secret keys are output as

$$\begin{aligned}
&R = \mathbf{g}_2^r, S_1 = \mathbf{g}_2^{r\cdot c}, S_2 = \mathbf{g}_2^{r\cdot z}, T = \mathbf{g}_2^{u+r\cdot(d+i\cdot e)},\\
&W_1 = \mathbf{g}_2^{[-\Delta'_5-u-r\cdot(\Delta'_1+d+i\cdot(\Delta'_2+e))]/b},\\
&W_2 = \mathbf{g}_2^{-r\cdot(\Delta'_3+c)/b}, W_3 = \mathbf{g}_2^{-r\cdot(\Delta'_4+z)/b}.
\end{aligned} \tag{4}$$

The computation of κ in decryption requests is similarly changed.

The view of the Adversary in games G_5 and G_4 is statistically identical.

Game G_6: This game is actually a sequence of several hybrid games, with the j-th hybrid game $G_{6,j}$ changing the simulation of the j-th secret key generation. Game $G_{6,0}$ is just the same as game G_5.

In game $G_{6,j}$ the challenger modifies the output of the j-th secret key as follows (assume that the identity requested by the Adversary is i_j): it chooses r_j, r'_j and r''_j at random and sets

$$\begin{aligned}
&R = \mathbf{g}_2^{r_j}, S_1 = \mathbf{g}_2^{r_j\cdot c}\mathbf{g}_2^{r'_j}, S_2 = \mathbf{g}_2^{r_j\cdot z}\\
&T = \mathbf{g}_2^{r''_j+r_j\cdot(d+i_j\cdot e)},\\
&W_1 = \mathbf{g}_2^{[-\Delta'_5-r''_j-r_j\cdot(\Delta'_1+d+i_j\cdot(\Delta'_2+e))]/b},\\
&W_2 = \mathbf{g}_2^{(-r'_j-r_j\cdot(\Delta'_3+c))/b}, W_3 = \mathbf{g}_2^{(-r_j\cdot(\Delta'_4+z))/b}.
\end{aligned}$$

Note that u has completely vanished from the j-th (and earlier) secret key simulation.

Lemma 16 *The view of the Adversary in game $G_{6,j}$ is computationally indistinguishable from the view of the Adversary in game $G_{6,j-1}$.*

Proof of this lemma is identical to the proof of the corresponding lemma (Lemma 14) in the plain IBE proof.

Game G_7: This game is again a sequence of several hybrid games, with the j-th hybrid game $G_{7,j}$ changing the simulation of the j-th decryption request. Game $G_{7,0}$ is just the game $G_{6,n}$ (where n is the number of secret key requests).

In game $G_{7,j}$ the challenger chooses r_j, r'_j, r''_j at random and uses the following in computation of κ (w.l.o.g.[10] let the identity for the decryption request be same as i. Let TAG_j be the tag supplied and h_j be the hash computed on the given ciphertext):

$$\begin{aligned}
&R = \mathbf{g}_2^{r_j},\\
&S_1^{\text{TAG}_j}\cdot T\cdot S_2^{h_j} = \mathbf{g}_2^{r_j\cdot(\text{TAG}_j\cdot c+h_j\cdot z+d+i\cdot e)+r'_j+r''_j},
\end{aligned}$$

[10] Although the Adversary might as well request the keys for identities different from i, it may not want to do that before the identity i is chosen. Thus strictly speaking, we should allow decryption requests for different identities, but our proof extends as we have already shown earlier in game G_6 how to handle giving keys to the Adversary for other identities.

and

$$W_1 \cdot W_2^{\text{TAG}_j} \cdot W_3^{h_j} = \mathbf{g}_2^{[-\Delta_5' - r_j'' - r_j \cdot (\Delta_1' + d + i \cdot (\Delta_2' + e))]/b} \cdot \mathbf{g}_2^{(-r_j \cdot \text{TAG}_j \cdot (\Delta_3' + c) - \text{TAG}_j \cdot r_j')/b} \cdot \mathbf{g}_2^{(-r_j \cdot h_j \cdot (\Delta_4' + z))/b}$$

Lemma 17 *The view of the Adversary in game $G_{7,j}$ is computationally indistinguishable from the view of the Adversary in game $G_{7,j-1}$.*

Proof: Let H_0 be same as the game $G_{6,j-1}$. In game H_1, the challenger chooses $z = z_1 + c \cdot z_2$, $d = d_1 + c \cdot d_2$, and tag TAG in the ciphertext as $-(d_2 + h \cdot z_2)$. where c, z_1, z_2, d_1 and d_2 are random and independent values from $\mathbb{Z}_q$. It is easy to see that c, z, d, and TAG are random and independent, and hence the view of the Adversary in games H_0 and H_1 is statistically identical. Note that with this value of TAG, C_3 (in the ciphertext) can be generated by the challenger as

$$\begin{aligned} C_3 &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{v}_4^{h \cdot z} \cdot \mathbf{g}_1^{(d_1 + i \cdot e + h \cdot z_1 + (d_2 + h \cdot z_2) \cdot c + \text{TAG} \cdot c) s'} \\ &= \mathbf{v}_1^s \cdot \mathbf{v}_2^{i \cdot s} \cdot \mathbf{v}_3^{\text{TAG} \cdot s} \cdot \mathbf{v}_4^{h \cdot z} \cdot \mathbf{g}_1^{(d_1 + i \cdot e + h \cdot z_1) s'} \end{aligned}$$

As a consequence c is not used at all in the simulation of the ciphertext (whose elements are all in group $\mathbb{G}_2$). The simulation of PK (without using c) is unchanged from game G_5.

In game H_2, the challenger generates the components in j-th decryption request by choosing r_j and r_j' uniformly and independently and setting

$$R = \mathbf{g}_2^{r_j}, \quad S_1^{\text{TAG}_j} \cdot T \cdot S_2^{h_j} = \mathbf{g}_2^{u + r_j \cdot ((\text{TAG}_j + h_j \cdot z_2) \cdot c + h_j \cdot z_1 + d_1 + c \cdot d_2 + i \cdot e) + r_j' (d_2 + \text{TAG}_j + h_j \cdot z_2)},$$

and

$$W_1 \cdot W_2^{\text{TAG}_j} \cdot W_3^{h_j} = \mathbf{g}_2^{[-\Delta_5' - u - r_j \cdot (\Delta_1' + d_1 + d_2 \cdot c + i \cdot (\Delta_2' + e))]/b} \cdot \mathbf{g}_2^{(-r_j \cdot \text{TAG}_j \cdot (\Delta_3' + c) - \text{TAG}_j \cdot r_j')/b} \cdot \mathbf{g}_2^{(-r_j \cdot h_j \cdot (\Delta_4' + z) - h_j \cdot z_2 \cdot r_j')/b} \cdot \mathbf{g}_2^{-r_j' (d_2 + \text{TAG}_j + h_j \cdot z_2)/b}$$

Recall that in game H_1, the secret key is being generated as in Equation (4), with $d = d_1 + c \cdot d_2$. The view of the Adversary in games H_2 and H_1 is computationally indistinguishable, and this is shown by employing the DDH assumption on the two tuples $\langle \mathbf{g}_2, \mathbf{g}_2^c, \mathbf{g}_2^{r_j}, \mathbf{g}_2^{c r_j} \rangle$ and $\langle \mathbf{g}_2, \mathbf{g}_2^c, \mathbf{g}_2^{r_j}, \mathbf{g}_2^{c r_j + r_j'} \rangle$, where the first tuple is employed in simulating game H_1 and the second tuple is used in simulating game H_2.

In game H_3, the challenger generates the components in the j-th decryption as

$$R = \mathbf{g}_2^{r_j}, \quad S_1^{\text{TAG}_j} \cdot T \cdot S_2^{h_j} = \mathbf{g}_2^{u + r_j \cdot (\text{TAG}_j \cdot c + h_j \cdot z + d + i \cdot e) + r_j' \cdot r_j''},$$

and

$$W_1 \cdot W_2^{\text{TAG}_j} \cdot W_3^{h_j} = \mathbf{g}_2^{[-\Delta'_5 - u - r_j \cdot (\Delta'_1 + d + i \cdot (\Delta'_2 + e))]/b} \cdot \mathbf{g}_2^{(-r_j \cdot \text{TAG}_j \cdot (\Delta'_3 + c))/b} \cdot \mathbf{g}_2^{(-r_j \cdot h_j \cdot (\Delta'_4 + z))/b} \cdot \mathbf{g}_2^{-r'_j \cdot r''_j/b}$$

where r_j, r'_j and r''_j are chosen randomly and independently (and independently from all other variables). Note that d and z are also chosen independently and randomly (back as in game H_0). Moreover, TAG is also chosen at random, and C_3 output just as in game H_0.

The view of the Adversary in game H_3 and H_2 is statistically identical by noting that $d = d_1 + c \cdot d_2$, $z = z_1 + c \cdot z_2$, $\text{TAG} = -(d_2 + h \cdot z_2)$ and $r''_j = d_2 + \text{TAG}_j + h_j \cdot z_2$ are all random and independent (since $h_j \neq h$). This can be seen by noting that the four by four matrix of coefficients of d, z, TAG, r''_j in their linear representation in terms of d_1, d_2, z_1, z_2 is non-singular.

In game H_4, the challenger generates d, z and TAG at random (instead of $d_1 + c \cdot d_2$ etc.), and also chooses r'''_j at random (and independent of r_j, r'_j and other variables) and uses the following in decryption

$$R = \mathbf{g}_2^{r_j}, \;\; S_1^{\text{TAG}_j} \cdot T \cdot S_2^{h_j} = \mathbf{g}_2^{r'''_j + r_j \cdot ((\text{TAG}_j + h_j \cdot z_2) \cdot c + h_j \cdot z_1 + d_1 + c \cdot d_2 + i \cdot e)},$$

and

$$W_1 \cdot W_2^{\text{TAG}_j} \cdot W_3^{h_j} = \mathbf{g}_2^{[-\Delta'_5 - r'''_j - r_j \cdot (\Delta'_1 + d_1 + d_2 \cdot c + i \cdot (\Delta'_2 + e))]/b} \cdot \mathbf{g}_2^{(-r_j \cdot \text{TAG}_j \cdot (\Delta'_3 + c))/b} \cdot \mathbf{g}_2^{(-r_j \cdot h_j \cdot (\Delta'_4 + z))/b}$$

Game H_4 is statistically identical to game H_3, as ($= u + r'_j \cdot r''_j$) in game H_3 is random and independent of r'_j, and hence is distributed same as a random r'''_j as in game H_4. Now note that game H_4 is identical to the game $G_{7,j}$ as described above the Lemma 16 statement. □

Game G_8 is just the game $G_{7,n}$ where n is the number of decryption queries. Note that in game G_8 the only place that u is used is in the ciphertext component C_0 which is simulated by the challenger as $C_0 = M \cdot \mathbf{k}^s \cdot e(\mathbf{g}_1, \mathbf{g}_2)^{us'}$ (see equation (1)). Hence, C_0 is completely random and independent of M in the view of the Adversary in game G_7 (note u is non-zero with high probability). That completes the proof. □