Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces

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

Arnab Roy
Fujitsu Laboratories of America
Sunnyvale, CA 94058, USA

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 parameters defining the language. 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 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 Symmetric eXternal Diffie Hellman (SXDH) 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).

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

Contents

1 Introduction

In [12] a remarkably efficient non-interactive zero-knowledge (NIZK) proof system [2] 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 [11], 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 [12].

While the Groth-Sahai system is quite efficient, it still falls short in comparison to Schnorr-based Σ-protocols [7] turned into NIZK proofs in the Random Oracle model [1] using the Fiat-Shamir paradigm [9]. Thus, the quest remains to obtain even more efficient NIZK Proofs.

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 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 [2] also allow the CRS to depend on the language, but such a NIZK that allows different efficient simulators for each particular language (from a parametrized class) is 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 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.

[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.

by allowing the (uniform) simulator to take language parameters as input/footnoteIt 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..

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

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

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, as the null-space of $L_\rho^\perp$ is just L. 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. This allows for a more efficient verifier, but requires a more sophisticated proof[2]. As we will see later in Section 6, 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 [18], and indeed our system is inspired by that construction[3], 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 NIZK proofs for linear subspaces require only $k(n-t)$ group elements, under the k-linear decisional assumption [17, 4]. Thus, under the SXDH 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 Figure 1, where we also compare with Σ-protocol based NIZK proofs in the random oracle model. Our SXDH based proofs are actually shorter than the Σ-protocol NIZK proofs, although the latter 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.

Thus, for the language L above, which is just a DLIN tuple used ubiquitously for encryption, our system only requires two additional group elements under the DLIN assumption, whereas the

[2]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.

[3]In Section 6 and in the Appendix, we show that the design of our system leads to a shorter SXDH assumption based dual-system IBE.

	SXDH		DLIN		Discrete Log + Random Oracle	
	Proof	CRS	Proof	CRS	Proof	CRS
Groth-Sahai	$n+2t$	4	$2n+3t$	9	-	-
This paper	$n-t$	$2t(n-t)+2$	$2n-2t$	$4t(n-t)+3$	-	-
Σ-Protocol	-	-	-	-	$(n:G)+(t:\mathbb{Z}_q)+(1:Hash)$	RO

Table 1: Comparison with existing techniques for 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. The CRS in this paper's system is usually much smaller, as many of the constants in the linear system can be zero.

Groth-Sahai system requires twelve additional 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 SXDH assumption, which we demonstrate in Section 4 (whereas the Groth-Sahai system requires $(n + 2t =)4$ elements for the proof; note $t = 1$ and $n = 2$).

Our system does not yet extend naturally to quadratic or multi-linear equations, whereas the Groth-Sahai system does. 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:

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

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 [4] for a generic construction). On the other hand, many applications just require one-time simulation soundness, and as has been shown in [13], this can be achieved for linear subspaces by projective hash proofs [6]. Projective hash proofs can be defined by linear extensions but using 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 6).

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 [4], UC commitments [10] and password-based key exchange [14, 13]. In particular, based on [10], 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 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 [5] which requires five group elements as ciphertext.

Organization of the paper. We begin the rest of the paper with the definition of quasi-adaptive NIZKs in Section 2 followed by definitions of the hardness assumptions that we use in Section 3. In Section 4 we develop quasi-adaptive NIZKs for linear subspaces under the SXDH assumption. In Section 5, we extend our system to include tags. Finally, we demonstrate applications of our system in Section 6. We defer detailed proofs and descriptions to the appendix. In Appendix B, we describe our system based on the k-linear assumption.

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. $\vec{\mathrm{r}}$ 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

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

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[4]. 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.

[4]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 for this associated language may not be very useful. Thus, it is critical to define the proper associated parameter language.

We call $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ a *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 $(\mathsf{S}_1, \mathsf{S}_2)$, such that for all non-uniform PPT adversaries $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ we have:

Quasi-Adaptive Completeness:

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

Quasi-Adaptive Soundness:

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

Quasi-Adaptive Zero-Knowledge:

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

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

To summarize, a quasi-adaptive NIZK

1. allows to split a hard language, or more precisely a distribution of languages, into two languages: (a) a language of parameters $\mathcal{L}_{\text{par}}$ (and the projected distribution $\mathcal{D}$ on $\mathcal{L}_{\text{par}}$), and (b) the projection L_ρ of the original language once the parameter ρ (in $\mathcal{L}_{\text{par}}$) has been fixed,
2. requires uniform simulation, i.e. a single efficient simulator for the whole distribution of parameters,
3. allows to pass the parameter as an input to the CRS generator and CRS generator simulator,
4. allows the Adversary to adaptively generate L_ρ tuples for the prover/simulator in the zero-knowledge definition (this is same as that in usual adaptive NIZK definitions).

Remark. One can consider a stronger definition of soundness where the Adversary gets more information (than ρ) about the relation R_ρ. Indeed, our quasi-adaptive NIZK for linear subspaces will have the property that the Adversary can even be given the discrete logarithms of the parameters. However, we do not yet see an application for such a stronger definition. Such a definition could however be useful in situations where the QA-NIZK's CRS can be generated with an under-specification of the relation R_ρ.

3 Hardness Assumptions

Definition 1 (DDH [8]) *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 2 (SXDH [3]) *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 $\mathbb{G}_2$.*

4 NIZK for Linear Subspaces under the SXDH Assumption

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.

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.

A set of equations $\boldsymbol{l}_1 = x_1 \cdot \mathbf{g}_1, \boldsymbol{l}_2 = x_2 \cdot \mathbf{f}_1, \boldsymbol{l}_3 = (x_1 + x_2) \cdot \mathbf{h}_1$ will be expressed in the form $\vec{\boldsymbol{l}} = \vec{\mathrm{x}} \cdot \mathsf{A} \cdot \mathbf{g}_1$ as follows:

$$\left[\begin{array}{ccc} \boldsymbol{l}_1 & \boldsymbol{l}_2 & \boldsymbol{l}_3 \end{array} \right] = \left[\begin{array}{cc} x_1 & x_2 \end{array} \right] \cdot \left[\begin{array}{ccc} 1 & 0 & c_2 \\ 0 & c_1 & c_2 \end{array} \right] \cdot \mathbf{g}_1$$

where $\mathbf{f}_1 = c_1 \cdot \mathbf{g}_1, \mathbf{h}_1 = c_2 \cdot \mathbf{g}_1$ are constants and $\vec{\mathrm{x}}$ is a vector of unknowns.

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:

$$L_\rho = \{\vec{\mathrm{x}} \cdot \mathsf{A} \cdot \mathbf{g}_1 \in \mathbb{G}_1^n \mid \vec{\mathrm{x}} \in \mathbb{Z}_q{}^t\}$$

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

Let the $t \times n$ dimensional matrix ρ be chosen according to a distribution $\mathcal{D}$ on L_ρ. We will call the distribution $\mathcal{D}$ *robust* if with probability close to one the left-most t columns of ρ 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 (ρ, A) that satisfy the relation $\mathcal{R}_{\text{par}}$ (i.e. $\mathcal{R}_{\text{par}}(\rho, \mathsf{A})$ holds), and further the resulting distribution of the output ρ 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 ρ. As an example of a robust distribution, consider a distribution $\mathcal{D}$ on (2×3)-dimensional matrices $\begin{bmatrix} 1 & 0 & c_2 \\ 0 & c_1 & c_2 \end{bmatrix}$ with c_1 and c_2 chosen randomly from $\mathbb{Z}_q$. It is easy to see that the first two columns are full-ranked with probability $(1 - 1/q)$.

We now describe a quasi-adaptive NIZK $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ for robust and efficiently-samplable distributions over linear subspaces $\{L_\rho\}$ with associated parameter language $\mathcal{L}_{\text{par}}$.

Algorithm K_0: K_0 is same as the group generation algorithm for which the SXDH assumption holds. It takes 1^m as input, where m is the security parameter and generates λ that includes q, the three groups, the generators $\mathbf{g}_1$ and $\mathbf{g}_2$ of $\mathbb{G}_1$ and $\mathbb{G}_2$, and the bilinear pairing e.

We will assume that the size $t \times n$ of the matrix 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 $\mathsf{A}^{t \times n} \cdot \mathbf{g}_1$ 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 $\mathbf{CRS}_1$ and $\mathbf{CRS}_2$ which are to be used by the prover and the verifier respectively.

$$\mathbf{CRS}_1^{t \times (n+s)} = \mathsf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n \times n} & \begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array} \end{array} \right] \cdot \mathbf{g}_1 \qquad \mathbf{CRS}_2^{(n+s) \times s} = \left[\begin{array}{c} b \cdot \mathsf{D} \\ \mathrm{I}^{s \times s} \\ -b \cdot \mathrm{I}^{s \times s} \end{array} \right] \cdot \mathbf{g}_2$$

Here, I denotes the identity matrix. Note that $\mathbf{CRS}_1$ can be generated from the parameter $\mathsf{A} \cdot \mathbf{g}_1$, since K_1 knows D and b. Also, note that $\mathbf{CRS}_2$ is independent of the parameter.

Prover P**:** Given candidate $\vec{\mathrm{x}} \cdot \mathsf{A} \cdot \mathbf{g}_1$ with witness vector $\vec{\mathrm{x}}$, the prover generates the following proof:

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

Note that the first n elements of the proof are exactly the candidate. We can assume that the language candidate is just given as the first part (i.e. first n elements) of the proof.

Verifier V**:** Given $\vec{\boldsymbol{l}}$, and a proof $\vec{\mathbf{p}}$, the verifier first checks that the first n elements of $\vec{\mathbf{p}}$ form the candidate $\vec{\boldsymbol{l}}$ and then checks the following:

$$e(\vec{\mathbf{p}}, \mathbf{CRS}_2) \stackrel{?}{=} \mathbf{0}_T^{1 \times s}$$

Theorem 3 *Let $\mathcal{D}$ be a robust and efficiently witness-samplable distribution over $\mathcal{L}_{par}$. For any group generation algorithm for which the DDH assumption holds for group $\mathbb{G}_2$, the above algorithms K_0, K_1, the Prover P, and the Verifier V constitute a quasi-adaptive NIZK for distribution $\mathcal{D}$ over the class of languages $\{L_\rho\}$ with associated parameter language $\mathcal{L}_{par}$.*

A detailed proof of the theorem can be found in Appendix A. 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 (again, see Appendix A for details). 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 $\vec{\mathbf{p}}' := \vec{\boldsymbol{l}} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{array}{c}\mathsf{D}\\ b^{-1}\cdot\mathrm{I}^{s\times s}\end{array}\end{array}\right]$. If $\vec{\boldsymbol{l}}$ is in the language, i.e., it is $\vec{\mathrm{x}} \cdot \mathsf{A} \cdot \mathbf{g}_1$ 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. For simplicity, assume that $\mathsf{A}^{t\times n}$ is a full-ranked matrix with its left-most t columns being full-ranked. Given a language L_ρ with parameter $\rho = \mathsf{A}^{t\times n} \cdot \mathbf{g}_1$, Let the null-space of vector space generated by rows of A be generated by columns of matrix $\mathsf{U}^{n\times s}$ ($s = (n-t)$). Then by basic linear algebra, the null-space of U is just A, and hence $\vec{l}$ is in span of A iff $\vec{l} \cdot \mathsf{U} = \vec{0}^{1\times s}$. Hence, one could verify a candidate $\vec{\boldsymbol{l}}$ (a vector in group $\mathbb{G}_1$) to be in L_ρ by checking the following bilinear pairing equation

$$e(\vec{\boldsymbol{l}},\ \mathsf{U}\cdot\mathbf{g}_2) \stackrel{?}{=} \mathbf{0}_T^{1\times s}.$$

However, we can not expect the CRS generator K_1 to produce $\mathsf{U}\cdot\mathbf{g}_2$ as CRS, as it is only given $\rho = \mathsf{A}\cdot\mathbf{g}_1$ as input. For *hard languages* L_ρ, efficiently computing $\mathsf{U}\cdot\mathbf{g}_2$ from ρ implies breaking the hardness of L_ρ (using the bilinear pairing).

However, one can envision the CRS being a hiding commitment of $\mathsf{U}\cdot\mathbf{g}_2$, which to a computationally bounded adversary is also a binding commitment. W.l.o.g. assume that U has its bottom s rows as the identity matrix (recall, U has rank $s = (n-t)$), and let the top t rows of U be called W. Thus, a perfectly *binding* commitment to $\mathsf{W}\cdot\mathbf{g}_2$ can be $\langle \mathbf{g}_1, \mathsf{D}\cdot\mathbf{g}_1, b\cdot\mathbf{g}_2, (\mathsf{W}+b\cdot\mathsf{D})\cdot\mathbf{g}_2\rangle$, for any scalar b and any matrix $\mathsf{D}^{t\times s}$ over $\mathbb{Z}_q$. Note that the first two elements are in $\mathbb{G}_1$, while the last two are in $\mathbb{G}_2$. In fact, the following is also a perfectly binding commitment to $\mathsf{W}\cdot\mathbf{g}_2$: $\langle \mathsf{A}_1\cdot\mathbf{g}_1, \mathsf{A}_1\mathsf{D}\cdot\mathbf{g}_1, b\cdot\mathbf{g}_2, (\mathsf{W}+b\cdot\mathsf{D})\cdot\mathbf{g}_2\rangle$, where A_1 is the left most t columns of A (recall, A_1 is non-singular).

With this in mind, we let Algorithm K_1 set the CRS as described in the specification before (which is the hiding form of the commitment), and that can be computed from just ρ. Now statistically, this CRS 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 (and noting that $\mathsf{A}\cdot\mathsf{U} = 0^{t\times s}$), the $\mathbf{CRS}_1$ and $\mathbf{CRS}_2$ can be represented as

$$\mathbf{CRS}_1^{t\times(n+s)} = \mathsf{A}\cdot\left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{array}{c}\mathsf{D}'\\ 0^{s\times s}\end{array}\end{array}\right]\cdot\mathbf{g}_1,\ \ \mathbf{CRS}_2^{(n+s)\times s} = \left[\begin{array}{c} b\cdot\left[\begin{array}{c}\mathsf{D}'\\ 0^{s\times s}\end{array}\right] + \left[\begin{array}{c}\mathsf{W}\\ \mathrm{I}^{s\times s}\end{array}\right] \\ -b\cdot\mathrm{I}^{s\times s}\end{array}\right]\cdot\mathbf{g}_2$$

However, we do not expect the pairing test of the verifier with this representation of the CRS, i.e. $e(\vec{\mathbf{p}}, \mathbf{CRS}_2) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$ to be statistically binding for $\vec{\mathbf{p}}$ to be a proof of a L_ρ member (i.e. of the form $\vec{\mathrm{x}}\cdot\mathsf{A}\cdot\mathbf{g}_1$). One can hope that the CRS can be made computationally binding by employing DDH on $\langle b\cdot\mathbf{g}_2, \mathsf{D}'\cdot\mathbf{g}_2, b\cdot\mathsf{D}'\cdot\mathbf{g}_2\rangle$. However, $\mathbf{CRS}_1$ requires $\mathsf{D}'\cdot\mathbf{g}_1$ and not $\mathsf{D}'\cdot\mathbf{g}_2$, and so this approach is untenable.

However, there is an alternate way to show that if an efficient Adversary $\mathcal{B}$ can produce a "proof" $\vec{\mathbf{p}}$ for which the above pairing test holds and yet the candidate (contained as first part of

$\vec{\mathbf{p}}$) is not in L_ρ, 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 $\langle \mathbf{g}_2, b\cdot\mathbf{g}_2, r\cdot b\cdot\mathbf{g}_2, \boldsymbol{\chi} = r\cdot\mathbf{g}_2\rangle$ or a fake DDH tuple $\langle \mathbf{g}_2, b\cdot\mathbf{g}_2, r\cdot b\cdot\mathbf{g}_2, \boldsymbol{\chi} = (r+r')\cdot\mathbf{g}_2\rangle$ (let's denote the last component in the tuple provided by the challenger as $\boldsymbol{\chi}$). We build an adversary $\mathcal{S}$ (for the DDH challenge) that first simulates the challenger in the QA-NIZK soundness experiment by using the DDH-challenge components $\mathbf{g}_2, b\cdot\mathbf{g}_2$ to build the above representation of the CRS (in particular CRS_2). After $\mathcal{B}$ replies with "proof" $\vec{\mathbf{p}}$, the DDH Adversary $\mathcal{S}$ checks if

$$e\left(\vec{\mathbf{p}}, \begin{bmatrix} r\cdot b\cdot \mathsf{D}'\cdot\mathbf{g}_2 + \mathsf{W}\cdot\boldsymbol{\chi} \\ \mathrm{I}^{s\times s}\cdot\boldsymbol{\chi} \\ -r\cdot b\cdot\mathrm{I}^{s\times s}\cdot\mathbf{g}_2 \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$$

When the DDH challenge is a real DDH tuple, this test has (at least) the same probability of passing as the verifier test in the QA-NIZK (i.e. $e(\vec{\mathbf{p}}, \mathsf{CRS}_2) \stackrel{?}{=} \mathbf{0}_T$). When the DDH challenge is a fake DDH tuple, one can show that this test *fails* with high probability (given that $\vec{\mathbf{p}}$'s first part is not in L_ρ), especially since $\mathcal{B}$'s response is independent of r'. Further, when $\vec{\mathbf{p}}$'s first part is in L_ρ, the above test has identical probability of passing in both the real and fake DDH setting. This leads to a proof of soundness of the QA-NIZK. Details of the proof can be found in Appendix A.

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 SXDH for the groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

As in the setup described before, we have $\mathsf{A} = \left[\begin{array}{c|c} c_1 & c_2 \end{array}\right]$, where c_1 and c_2 are the discrete logs of $\mathbf{g}$ and $\mathbf{f}$ respectively with respect to $\mathbf{g}_1$. The language is: $L = \{[x]\cdot\mathsf{A}\cdot\mathbf{g}_1 \mid x\in\mathbb{Z}_q\}$.

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:

$$\mathsf{CRS}_1 := \mathsf{A}\cdot\left[\begin{array}{c|c} \mathrm{I}^{2\times 2} & \begin{array}{c}\mathsf{D}\\ b^{-1}\cdot\mathrm{I}^{1\times 1}\end{array}\end{array}\right]\cdot\mathbf{g}_1 = \left[\begin{array}{c|c|c} \mathbf{g} & \mathbf{f} & (d\cdot\mathbf{g}+b^{-1}\cdot\mathbf{f}) \end{array}\right]$$

$$\mathsf{CRS}_2 := \begin{bmatrix} b\cdot\mathsf{D} \\ \mathrm{I}^{1\times 1} \\ -b\cdot\mathrm{I}^{1\times 1}\end{bmatrix}\cdot\mathbf{g}_2 = \begin{bmatrix} bd\cdot\mathbf{g}_2 \\ \mathbf{g}_2 \\ -b\cdot\mathbf{g}_2\end{bmatrix}$$

The proof of a tuple $(\mathbf{r},\hat{\mathbf{r}})$ with witness r, is just the *single* element $r\cdot(d\cdot\mathbf{g}+b^{-1}\cdot\mathbf{f})$. Along with the tuple this forms: $\vec{\mathbf{p}} := \left[\begin{array}{c|c|c} \mathbf{r} & \hat{\mathbf{r}} & r\cdot(d\cdot\mathbf{g}+b^{-1}\cdot\mathbf{f})\end{array}\right]$. In the proof of zero knowledge, the simulator trapdoor is (d,b) and the simulated proof of $(\mathbf{r},\hat{\mathbf{r}})$ is just $(d\cdot\mathbf{r}+b^{-1}\cdot\hat{\mathbf{r}})$.

5 Introducing Tags

In this section we show how the previous system can be extended to include tags which are elements of $\mathbb{Z}_q$ and are included as part of the proof. The tags are used as part of the defining equations of the language.

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

$$\left\{ \left\langle \vec{x} \cdot \left[\ \mathsf{A}^{t\times(n-1)}\ \middle|\ (\vec{a}_1^\top + \text{TAG} \cdot \vec{a}_2^\top)\ \right] \cdot \mathbf{g}_1, \text{TAG} \right\rangle \ \middle|\ \vec{x} \in {\mathbb{Z}_q}^t, \text{TAG} \in \mathbb{Z}_q \right\}$$

where $\mathsf{A}^{t\times(n-1)} \cdot \mathbf{g}_1, \vec{a}_1 \cdot \mathbf{g}_1$ and $\vec{a}_2 \cdot \mathbf{g}_1$ are parameters of the language. A distribution is still called robust (as in Section 4) if with overwhelming probability the first t columns of A are full-ranked. Write A as $[\mathsf{A}_l^{t\times t} \mid \mathsf{A}_r^{t\times(n-1-t)}]$, where w.l.o.g. A_l is non-singular. While the first $n-1-t$ components in excess of the unknowns, corresponding to A_r, can be verified just as in Section 4, for the last component we proceed as follows. The CRS is generated as:

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

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

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 $\vec{l'} \stackrel{\text{def}}{=} \vec{x} \cdot \left[\ \mathsf{A}_l\ \middle|\ (\vec{a}_1^\top + \text{TAG} \cdot \vec{a}_2^\top)\ \right] \cdot \mathbf{g}_1$. The prover generates the following proof for the last component:

$$\vec{\mathbf{p}} := \left[\ \vec{l'}\ \middle|\ \vec{x} \cdot (\mathsf{CRS}_{1,0} + \text{TAG} \cdot \mathsf{CRS}_{1,1})\ \right]$$

Verifier: Given a proof $\vec{\mathbf{p}}$, which includes the candidate for membership, the verifier checks the following:

$$e(\vec{\mathbf{p}}, \mathsf{CRS}_{2,0} + \text{TAG} \cdot \mathsf{CRS}_{2,1}) \stackrel{?}{=} \mathbf{0}_T$$

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 4 and a proof sketch can be found in Appendix C.

6 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 [10] requires the following quasi-adaptive NIZK proof (see Appendix F for details)

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

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 [10] stated a scheme using Groth-Sahai NIZK proofs requiring $(5 + 16)$ elements. More details can be found in Appendix F.

One-time (Relatively) Simulation-Sound NIZK for DDH and other languages. In [13] 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 [15, 16], the NIZK proof can be made one-time simulation sound using a projective hash proof [6] 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 QA-NIZK proof system for a related language distribution. This construction is a generalization of a signature scheme from [4] which used (fully) adaptive NIZK proofs and *required* constructions based on groups in which the CDH assumption holds.

We first need to define a special form of QA-NIZK proof systems which we will call *split-CRS QA-NIZK proofs*. 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. A formal definition can be found in Section D.

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 $\{L_\rho\}$.

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 4 *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 F. 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 [4]), 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

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

where $\textsc{tag} = \mathcal{H}(R, S, T, M)$. 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 [13]). 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 4 works by checking $e(\vec{\mathbf{p}}, \mathsf{CRS}_v) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$ (or more precisely, $e(\vec{\mathbf{p}}, \mathsf{CRS}_v) \stackrel{?}{=} \vec{\mathbf{f}}$ 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 (as in the soundness proof game $\mathbf{G}_2$), 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 a complete (stand-alone) proof in Appendix G. 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[5]. Moreover, this scheme has perfect completeness.

References

[1] 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

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

[3] 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. 2, 7

[4] 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, 6, 6, 8, F

[5] 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, 5

[6] 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 Cryptol-*

[5] [5] have recently and independently obtained a short IBE under SXDH, but our IBE ciphertexts are even shorter.

ogy – EUROCRYPT 2002, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, April / May 2002. 1, 6

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

[8] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 1

[9] 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

[10] 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, 6, F, F

[11] 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

[12] 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

[13] 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, 6, 6

[14] 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

[15] 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. 6

[16] 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. 6

[17] 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, 8

[18] 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, G, G

A Proof of QA-NIZK for Linear Subspaces under SXDH Assumption

Theorem 3 (Section 4) Let $\mathcal{D}$ be a robust and efficiently witness-samplable distribution over $\mathcal{L}_{\text{par}}$. For any group generation algorithm for which the DDH assumption holds for group G_2, the algorithms K_0, K_1, the Prover P, and the Verifier V constitute a quasi-adaptive NIZK for distribution $\mathcal{D}$ over the class of languages $\{L_\rho\}$ with associated parameter language $\mathcal{L}_{\text{par}}$.
Proof:

Completeness: For a candidate $\vec{x} \cdot \mathsf{A} \cdot \mathbf{g}_1$ (which is a language member), the left-hand-side of the verification equation is:

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

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 $\vec{\mathbf{p}}' := \vec{l} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n \times n} & \begin{array}{c} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s \times s} \end{array} \end{array} \right]$. If $\vec{l}$ is in the language, i.e., it is $\vec{x} \cdot \mathsf{A} \cdot \mathbf{g}_1$ for some $\vec{x}$, then the distribution of the simulated proof is identical to the real world proof. Therefore, the simulated NIZK CRS and simulated proofs of language members are identically distributed as the real world. Hence the system is perfect Zero Knowledge.

Soundness: We prove soundness by transforming the system over three 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. In game $\mathbf{G}_2$ the challenger uses a real DDH challenge distribution to generate the CRS and execute the verifier, which is then transformed in the third game to a fake DDH challenge distribution. After this transformation, we show that a verifying proof implies that the given candidate is a member of the language.

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 ρ according to $\mathcal{D}$, generates the CRS ψ using K_1, and passes λ, ρ and the CRS (i.e. CRS_1, CRS_2) to an Adversary $\mathcal{B}$. Let the $\mathcal{B}$ produce $\vec{\mathbf{p}}$. We say $\mathcal{B}$ wins if $e(\vec{\mathbf{p}}, \mathsf{CRS}_2) \stackrel{?}{=} \mathbf{0}_T^{1 \times s}$ while the first part of $\vec{\mathbf{p}}$ is not in L_ρ. Let W_0 denote the event that $\mathcal{B}$ wins game $\mathbf{G}_0$. Note that $\Pr[W_0]$ is exactly the probability of an Adversary producing a proof $\vec{\mathbf{p}}$ that passes the QA-NIZK verifier V while the proof's first part is not in L_ρ, and hence 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 M, in this game the challenger generates $\rho = \mathsf{A} \cdot \mathbf{g}_1$ using M, and hence the challenger also gets A (the witness to ρ 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 $\begin{bmatrix} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix}$ of dimension $(t+s)\times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A} \cdot \begin{bmatrix} \mathsf{W}^{t\times s} \\ \mathrm{I}^{s\times s} \end{bmatrix} = \mathsf{0}^{t\times s}$. 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). Implicitly set,

$$\begin{bmatrix} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{bmatrix} = \begin{bmatrix} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{bmatrix} + b^{-1} \cdot \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix}$$

Therefore the challenger produces,

$$\begin{aligned}
\mathbf{CRS}_1^{t\times(n+s)} &= \mathsf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{matrix} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{matrix} \end{array} \right] \cdot \mathbf{g}_1 = \mathsf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{bmatrix} \mathsf{D} \\ b^{-1}\cdot \mathrm{I}^{s\times s} \end{bmatrix} - b^{-1} \cdot \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \end{array} \right] \cdot \mathbf{g}_1 \\
&= \mathsf{A} \cdot \left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{matrix} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{matrix} \end{array} \right] \cdot \mathbf{g}_1 \\
\mathbf{CRS}_2^{(n+s)\times s} &= \begin{bmatrix} b\cdot \mathsf{D} \\ \mathrm{I}^{s\times s} \\ -b\cdot \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathbf{g}_2 = \begin{bmatrix} b \cdot \begin{bmatrix} \mathsf{D}' \\ \mathsf{0}^{s\times s} \end{bmatrix} + \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \\ -b\cdot \mathrm{I}^{s\times s} \end{bmatrix} \cdot \mathbf{g}_2
\end{aligned}$$

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).

Game $\mathbf{G}_2$: This game is identical to the previous game, except that the final bilinear pairings test is modified from $e(\vec{\mathbf{p}}, \mathbf{CRS}_2) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$ to being

$$e\left(\vec{\mathbf{p}}, \begin{bmatrix} r\cdot b\cdot \mathsf{D}'\cdot \mathbf{g}_2 + r\cdot \mathsf{W}\cdot \mathbf{g}_2 \\ r\cdot \mathrm{I}^{s\times s}\cdot \mathbf{g}_2 \\ -r\cdot b\cdot \mathrm{I}^{s\times s}\cdot \mathbf{g}_2 \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1\times s}$$

where r is chosen at random from $\mathbb{Z}_q$ by the challenger. It is easy to see that if the test in game $\mathbf{G}_1$ passes, then this test passes as well, as $\mathbf{CRS}_2$ has just been scaled by a scalar r. Let W_2 be the event that adversary $\mathcal{B}$ wins game $\mathbf{G}_2$. It follows that $\Pr[W_2 \mid \neg BAD] \geq \Pr[W_1 \mid \neg BAD]$, and hence $\Pr[W_2 \wedge \neg BAD] \geq \Pr[W_1 \wedge \neg BAD]$.

The Challenger in game $\mathbf{G}_2$ continues to be efficient (i.e. can be implemented by a PPT).

Game $\mathbf{G}_3$: This game is identical to the previous game, except that the final bilinear pairings test is now the following:

$$e\left(\vec{\mathbf{p}}, \begin{bmatrix} r \cdot b \cdot \mathsf{D}' \cdot \mathbf{g}_2 + (r+r') \cdot \mathsf{W} \cdot \mathbf{g}_2 \\ (r+r') \cdot \mathrm{I}^{s \times s} \cdot \mathbf{g}_2 \\ -r \cdot b \cdot \mathrm{I}^{s \times s} \cdot \mathbf{g}_2 \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1 \times s} \tag{1}$$

where r *and* r' are chosen at random from $\mathbb{Z}_q$ by the challenger. Let W_3 be the event that adversary $\mathcal{B}$ wins game $\mathbf{G}_3$. The Challenger in game $\mathbf{G}_3$ also continues to be efficient.

Lemma 5 below shows that $|\Pr[W_2 \wedge \neg\mathrm{BAD}] - \Pr[W_3 \wedge \neg\mathrm{BAD}]|$ is negligible. Lemma 6 below shows that $\Pr[W_3 \mid \neg\mathrm{BAD}]$ itself is negligible. From the two lemmas and the fact the $\Pr[\mathrm{BAD}]$ is negligible, it follows that $\Pr[W_2 \wedge \neg\mathrm{BAD}]$ is negligible. But, $\Pr[W_2 \wedge \neg BAD] \geq \Pr[W_1 \wedge \neg BAD]$, and hence soundness of the QA-NIZK follows. □

Lemma 5 *$|\Pr[W_2 \wedge \neg BAD] - \Pr[W_3 \wedge \neg BAD]|$ is negligible under the DDH assumption for group $\mathbb{G}_2$.*

Proof: Consider a challenger that uses a group generation algorithm to produce the bilinear groups $\mathbb{G}_1$ and $\mathbb{G}_2$ including their generators, say $\mathbf{g}_1$ and $\mathbf{g}_2$, such that it is hard to distinguish between a real DDH tuple $\langle \mathbf{g}_2, b \cdot \mathbf{g}_2, r \cdot b \cdot \mathbf{g}_2, \boldsymbol{\chi} = r \cdot \mathbf{g}_2 \rangle$ or a fake DDH tuple $\langle \mathbf{g}_2, b \cdot \mathbf{g}_2, r \cdot b \cdot \mathbf{g}_2, \boldsymbol{\chi} = (r+r') \cdot \mathbf{g}_2 \rangle$ (we will denote by $\boldsymbol{\chi}$ the last component of the tuple provided as DDH challenge). While this may look slightly different from the usual formalization of DDH, it is easy to show that this is equivalent. We will now show that if $p = |\Pr[W_2 \mid \neg\mathrm{BAD}] - \Pr[W_3 \mid \neg\mathrm{BAD}]|$, then there is an adversary $\mathcal{S}$ that can distinguish between the real and fake DDH tuple with probability p. Thus, by the DDH assumption for $\mathbb{G}_2$, the lemma would follow.

The Adversary $\mathcal{S}$ will use the DDH challenge to simulate the challengers from games $\mathbf{G}_2$ and $\mathbf{G}_3$ to adversary $\mathcal{B}$ in those games, and use $\mathcal{B}$'s response and the trapdoors kept by the challengers (in games $\mathbf{G}_2$ and $\mathbf{G}_3$) to respond with a bit to the DDH Challenger.

As remarked earlier, the Challenger in games $\mathbf{G}_2$ and $\mathbf{G}_3$ is efficient, so $\mathcal{B}$ just emulates the challenger, which is identical in both games except for performing the final pairing test. The CRS_2 generation in both games can be done using $\mathbf{g}_2$ and $b \cdot \mathbf{g}_2$ supplied by the DDH challenger (which is same in both real and fake DDH tuples), and the matrices D' (that the challenger chooses randomly) and W that the challenger efficiently computes. If the matrix A's left most t columns were not full-ranked then $\mathcal{B}$ just quits and responds with bit 0 to the DDH challenger (recall, this is same as event BAD).

After adversary $\mathcal{B}$ responds with $\vec{\mathbf{p}}$, the adversary $\mathcal{A}$ performs the following pairing test

$$e\left(\vec{\mathbf{p}}, \begin{bmatrix} \mathsf{D}' \cdot r \cdot b \cdot \mathbf{g}_2 + \mathsf{W} \cdot \boldsymbol{\chi} \\ \mathrm{I}^{s \times s} \boldsymbol{\chi} \\ -\mathrm{I}^{s \times s} \cdot r \cdot b \cdot \mathbf{g}_2 \end{bmatrix}\right) \stackrel{?}{=} \mathbf{0}_T^{1 \times s} \tag{2}$$

Note that depending on whether $\boldsymbol{\chi}$ came from the real DDH tuple or the fake DDH tuple, this test emulates the test in game $\mathbf{G}_2$ or game $\mathbf{G}_3$ resp. If the answer to this test is false, adversary $\mathcal{S}$ responds with bit 0 to the DDH challenger. Otherwise, i.e. if the test is true, $\mathcal{S}$ checks if the first part of $\vec{\mathbf{p}}$ is in the language L_ρ, which it can efficiently test as it has W, the null-space of A.

More precisely, it can just do the following bilinear pairing test $e(\vec{\mathbf{p}}_{1..n}, \mathsf{w} \cdot \mathbf{g}_2) \stackrel{?}{=} 0_T^{1\times s}$. If this test *passes* then $\mathcal{S}$ outputs 0 to the DDH challenger (i.e. the response of $\mathcal{B}$ was not in L_ρ). Otherwise, if the test fails it outputs 1 to the DDH challenger. Thus, it outputs 1 to the DDH challenger iff the pairing test in Equation (2) passes and the candidate was not in the language. In other words, when the DDH challenger produces the real DDH tuple, adversary $\mathcal{S}$ outputs 1 iff Adversary $\mathcal{B}$ wins game $\mathbf{G}_2$ and event BAD does not happen. Similarly, when the DDH challenger produces the fake DDH tuple, adversary $\mathcal{S}$ outputs 1 iff Adversary $\mathcal{B}$ wins game $\mathbf{G}_3$ and event BAD does not happen. This completes the proof. □

Lemma 6 $\Pr[W_3 \mid \neg BAD]$ *is negligible.*

Proof: We will condition on the event BAD not happening in Game $\mathbf{G}_3$. Recall the first part in $\vec{\mathbf{p}}$ is just $\vec{\boldsymbol{l}}$. The verification equation in game $\mathbf{G}_3$)(i.e. Equation (1)) can be re-written as:

$$e(\vec{\mathbf{p}}, \mathbf{CRS}_2 \cdot r) + e\left(\vec{\boldsymbol{l}}, \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot r' \cdot \mathbf{g}_2\right) = \mathbf{0}_T^{1\times s}$$

Since r' is not used in the generation of the NIZK CRS, it follows that $\vec{\boldsymbol{l}}$ and the proof $\vec{\mathbf{p}}$ provided by the adversary are independent of r'. Thus, the probability of event W_3 remains same if r' was chosen after *all* other random coins have been chosen by the challenger and the adversary.

For the sake of contradiction, suppose that for any fixed choice of *all* the coins (except r') the probability over random choice of r' of the verification equation passing in game $\mathbf{G}_3$ (while $\vec{\boldsymbol{l}} \in L_\rho$) is *strictly* more than $1/q$. This implies there are at least two choices of r' for this fixed choice of all other coins such that the pairing test passes the (fixed) tuple and the (fixed) proof while the (fixed) tuple is not in the language. But, if all other coins are same, except for r', then the pairing test passing means that

$$e\left(\vec{\boldsymbol{l}}, \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot (r'_1 - r'_2) \cdot \mathbf{g}_2\right) = \mathbf{0}_T^{1\times s}$$

where r'_1 and r'_2 are any two such distinct values of r'. But, this would imply that $\vec{\boldsymbol{l}}$ is in the language, as it is in the null-space of $\begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix}$, which is a contradiction. Thus, for any fixed choice of all the coins (except r') the probability over r' that the pairing test passes and the tuple is not in the language is at most $1/q$. Thus, $\Pr[W_3 \mid \neg\mathrm{BAD}]$ is itself at most $1/q$, which is negligible. □

B 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 7 (DLIN [3]) *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 8 (k-linear [17, 4]) *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*

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

Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order q with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. 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 4.

NIZK CRS: Suppose the language is $L = \{\vec{x} \cdot \mathsf{A}^{t\times n} \cdot \mathbf{g}_1 \in \mathbb{G}_1^n \mid \vec{x} \in {\mathbb{Z}_q}^t\}$. 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 a diagonal matrix $\mathsf{E}^{ks\times ks}$ with diagonal elements chosen randomly from $\mathbb{Z}_q$. Let $\mathsf{F}^{s\times ks} \stackrel{\text{def}}{=} \underbrace{\left[\ \mathrm{I}^{s\times s} \mid \cdots \mid \mathrm{I}^{s\times s}\ \right]}_{k \text{ times}} \cdot \mathsf{E}^{-1}$. 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.

$$\mathbf{CRS}_1^{t\times(n+ks)} = \mathsf{A} \cdot \left[\ \mathrm{I}^{n\times n} \ \middle|\ \begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array}\ \right] \cdot \mathbf{g}_1 \qquad\qquad \mathbf{CRS}_2^{(n+ks)\times ks} = \left[\begin{array}{c} \left[\begin{array}{c} \mathsf{D} \\ \mathsf{F} \end{array} \right] \cdot \mathsf{E} \\ -\mathsf{E} \end{array} \right] \cdot \mathbf{g}_2$$

Note that $\mathbf{CRS}_1$ can be generated from $\mathsf{A} \cdot \mathbf{g}_1$ - the knowledge of A is not necessary.

Prover: Given candidate $\vec{x} \cdot \mathsf{A} \cdot \mathbf{g}_1$ with witness vector $\vec{x}$, the prover generates the following proof:

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

Verifier: The verifier generates a vector $\vec{s}$ with ks entries randomly chosen from $\mathbb{Z}_q$. Given a proof $\vec{\mathbf{p}}$, the verifier first checks that the first n elements form the candidate $\vec{\boldsymbol{l}}$ and then checks the following:

$$e(\vec{\mathbf{p}}, \mathbf{CRS}_2 \cdot \vec{s}^\top) \stackrel{?}{=} \mathbf{0}_T$$

Theorem 9 *Let* $\mathcal{D}$ *be a robust and efficiently witness-samplable distribution over* $\mathcal{L}_{par}$*. For any group generation algorithm for which the* k*-linear assumption holds for group* $\mathbb{G}_2$*, the above algorithms* K_0*,* K_1*, the Prover* P*, and the Verifier* V *constitute a quasi-adaptive NIZK for distribution* $\mathcal{D}$ *over the class of languages* $\{L_\rho\}$ *with associated parameter language* $\mathcal{L}_{par}$*.*

Proof:

Completeness: For a candidate $\vec{x} \cdot \mathsf{A} \cdot \mathbf{g}_1$ (which is a language member), the LHS of the verification equation is:

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

Hence completeness follows.

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

Soundness: We prove soundness by transforming the system over three games. The first transformation involves generating the public keys in a different way, which is still statistically identical to the original system. In the second transformation, we use a real k-linear challenge distribution to generate the CRS and execute the verifier, which is then transformed in the third step to a fake k-linear challenge distribution. After this transformation, we show that a verifiable proof implies that the given candidate is a member of the language.

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 $\left[\begin{matrix} \mathsf{W}^{t \times s} \\ \mathrm{I}^{s \times s} \end{matrix} \right]$ of dimension $(t+s) \times s$ whose columns form a complete basis for the null-space of A, which means $\mathsf{A} \cdot \left[\begin{matrix} \mathsf{W}^{t \times s} \\ \mathrm{I}^{s \times s} \end{matrix} \right] = 0^{t \times s}$. 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,

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

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

$$\mathbf{CRS}_1^{t\times(n+ks)} = \mathsf{A}\cdot\left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{array}{c}\mathsf{D}\\ \mathsf{F}\end{array}\end{array}\right]\cdot \mathbf{g}_1 = \mathsf{A}\cdot\left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \left[\begin{array}{c}\mathsf{D}\\ \mathsf{F}\end{array}\right] - \underbrace{\left[\begin{array}{c|c|c} \mathsf{W} & \cdots & \mathsf{W}\\ \mathrm{I}^{s\times s} & \cdots & \mathrm{I}^{s\times s}\end{array}\right]}_{k \text{ times}}\cdot \mathsf{E}^{-1}\end{array}\right]\cdot \mathbf{g}_1$$

$$= \mathsf{A}\cdot\left[\begin{array}{c|c} \mathrm{I}^{n\times n} & \begin{array}{c}\mathsf{D}'\\ 0^{s\times ks}\end{array}\end{array}\right]\cdot \mathbf{g}_1$$

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

Observe that D has identical distribution as in game G_0 and the rest of the computations were same. So game G_1 is statistically indistinguishable from game G_0.

$\mathbf{G}_2$: In this game, we use a vector of real k-linear challenge distributions to generate the CRS and to execute the verification algorithm. Let B be a $(ks \times ks)$ diagonal matrix with diagonal entries chosen randomly from $\mathbb{Z}_q$ and $\vec{\mathrm{r}}$ be a ks element vector also chosen randomly from $\mathbb{Z}_q$. Let $\vec{\mathrm{r}}$ be also represented as $[\vec{\mathrm{r}}_1 \ \cdots \ \vec{\mathrm{r}}_k]$, where each of the $\vec{\mathrm{r}}_i$'s is an s element vector. The following lemma is actually s different instances of a k-linear problem and thus can be proved by a standard hybrid argument:

Lemma 10 *Given $B\cdot\mathbf{g}, B\cdot\vec{r}^\top\cdot\mathbf{g}$ and $\mathbf{g}$, it is computationally infeasible to distinguish between $(\vec{r}_1+\cdots+\vec{r}_k)^\top\cdot\mathbf{g}$ and $(\vec{r}_1+\cdots+\vec{r}_k+\vec{r}_{k+1})^\top\cdot\mathbf{g}$ under the k-linear assumption.*

In fact, we can take B to be a set of k independently random diagonal elements repeated s times.

So now suppose we are given a real challenge distribution $\mathsf{B}\cdot\mathbf{g}, \mathsf{B}\cdot\vec{\mathrm{r}}^\top\cdot\mathbf{g}, \mathbf{g}$ and $(\vec{\mathrm{r}}_1+\cdots+\vec{\mathrm{r}}_k)^\top\cdot\mathbf{g}$ in the group $\mathbb{G}_2$. In this game we generate the CRS as follows: $\mathbf{CRS}_1$ is generated as in $\mathbf{G}_1$.

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

We also change the term $\mathbf{CRS}_2\cdot\vec{\mathrm{s}}^\top$ in the verification equation $e(\vec{\mathbf{p}}, \mathbf{CRS}_2\cdot\vec{\mathrm{s}}^\top) \stackrel{?}{=} \mathbf{0}_T$ to the following:

$$\left[\begin{array}{c} \mathsf{D}'\cdot\mathsf{B}\cdot\vec{\mathrm{r}}^\top\cdot\mathbf{g} + \mathsf{W}\cdot(\vec{\mathrm{r}}_1+\cdots+\vec{\mathrm{r}}_k)^\top\cdot\mathbf{g}\\ (\vec{\mathrm{r}}_1+\cdots+\vec{\mathrm{r}}_k)^\top\cdot\mathbf{g}\\ -\mathsf{B}\cdot\vec{\mathrm{r}}^\top\cdot\mathbf{g}\end{array}\right]$$

Since the distributions of $\mathsf{B}, \vec{\mathrm{r}}$ and $\mathbf{g}$ are identical to $\mathsf{E}, \vec{\mathrm{s}}$ and $\mathbf{g}_2$, game $\mathbf{G}_2$ is statistically indistinguishable from game $\mathbf{G}_1$.

Game $\mathbf{G}_3$: In this game, we use a vector of *fake* k-linear challenge distributions to generate the CRS and to execute the verification algorithm. Suppose we are given a fake challenge distribution

$\mathsf{B} \cdot \mathbf{g}, \mathsf{B} \cdot \vec{\mathrm{r}}^\top \cdot \mathbf{g}, \mathbf{g}$ and $(\vec{\mathrm{r}}_1 + \cdots + \vec{\mathrm{r}}_k + \vec{\mathrm{r}}_{k+1})^\top \cdot \mathbf{g}$ in the group $\mathbb{G}_2$, where $\vec{x}_{k+1}$ is another vector of s elements chosen randomly from $\mathbb{Z}_q$. In this game we generate the CRS in the same way as game $\mathbf{G}_2$: CRS_1 is generated as in game $\mathbf{G}_1$.

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

The term $\mathsf{CRS}_2 \cdot \vec{\mathrm{s}}^\top$ in the verification equation $e(\vec{\mathbf{p}}, \mathsf{CRS}_2 \cdot \vec{\mathrm{s}}^\top) \stackrel{?}{=} \mathbf{0}_T$ also changes according to the fake challenge distribution:

$$\begin{bmatrix} \mathsf{D}' \cdot \mathsf{B} \cdot \vec{\mathrm{r}}^\top \cdot \mathbf{g} + \mathsf{W} \cdot (\vec{\mathrm{r}}_1 + \cdots + \vec{\mathrm{r}}_k + \vec{\mathrm{r}}_{k+1})^\top \cdot \mathbf{g} \\ (\vec{\mathrm{r}}_1 + \cdots + \vec{\mathrm{r}}_k + \vec{\mathrm{r}}_{k+1})^\top \cdot \mathbf{g} \\ -\mathsf{B} \cdot \vec{\mathrm{r}}^\top \cdot \mathbf{g} \end{bmatrix}$$

Let us call this term T and note that:

$$e(\vec{\mathbf{p}}, \mathsf{T}) = e(\vec{\mathbf{p}}, \mathsf{CRS}_2 \cdot \vec{\mathrm{r}}^\top) + e\left(\vec{\mathbf{p}}, \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \\ 0^{ks \times s} \end{bmatrix} \cdot \vec{\mathrm{r}}_{k+1}^\top \cdot \mathbf{g}\right)$$

$$= e(\vec{\mathbf{p}}, \mathsf{CRS}_2 \cdot \vec{\mathrm{r}}^\top) + e\left(\vec{\boldsymbol{l}}, \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot \vec{\mathrm{r}}_{k+1}^\top \cdot \mathbf{g}\right)$$

Now, game $\mathbf{G}_3$ is indistinguishable from game $\mathbf{G}_2$ by Lemma 10. We now show that this implies soundness based on indistinguishability of the results of the verification steps in both games. Let the verification equation in game $\mathbf{G}_3$ pass some candidate $\vec{\boldsymbol{l}}$ provided by the adversary which is not in L, with non-negligible probability. That is, with non-negligible probability:

$$e(\vec{\mathbf{p}}, \mathsf{CRS}_2 \cdot \vec{\mathrm{r}}^\top) + e\left(\vec{\boldsymbol{l}}, \begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix} \cdot \vec{\mathrm{r}}_{k+1}^\top \cdot \mathbf{g}\right) = \mathbf{0}_T$$

Now we proceed as in the proof of soundness of Theorem 3. We provide a sketch here, but the arguments can be made rigorous as in Appendix A. Since the vector $\vec{\mathrm{r}}_{k+1}^\top \cdot \mathbf{g}$ is not used in the generation of the NIZK CRS, it follows that $\vec{\boldsymbol{l}}$ and the proof $\vec{\mathbf{p}}$ provided by the adversary are independent of $\vec{\mathrm{r}}_{k+1}$. Let $\vec{\mathrm{w}}_i^\top$ denote the i-th column in $\begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix}$. Let p_i (for $i \in [1..s]$) be the probability that $e(\vec{\mathbf{p}}, \mathsf{T}) = 0$ **and** $\vec{\boldsymbol{l}} \cdot \vec{\mathrm{w}}_i^\top \neq 0$. Then by fixing $\vec{\mathrm{r}}$, and all other random variables including $\vec{\mathrm{r}}_{k+1}$, except the i-th variable in $\vec{\mathrm{r}}_{k+1}$, it follows that p_i is negligible. Then by union bound, the probability that $e(\vec{\mathbf{p}}, \mathsf{T}) = 0$ **and** there exists an $i \in [1..s]$ such that $\vec{\boldsymbol{l}} \cdot \vec{\mathrm{w}}_i^\top \neq 0$ is negligible. Since the vectors $\vec{\mathrm{w}}_i$ generate the null-space of A, this implies that the probability that $e(\vec{\mathbf{p}}, \mathsf{T}) = 0$ **and** $\vec{\boldsymbol{l}} \notin \mathrm{span}(\mathsf{A})$ is negligible. Note that the second conjunct is equivalent to $\vec{\boldsymbol{l}} \notin L$.

To complete the argument, we also have to prove that proofs for *language members* behave the same way in both games $\mathbf{G}_2$ and $\mathbf{G}_3$. This is straightforward because a language member is already in the nullspace of $\begin{bmatrix} \mathsf{W} \\ \mathrm{I}^{s\times s} \end{bmatrix}$. Therefore, the extra term in T just vanishes when we do the pairing product. $\square$

C Proof of QA-NIZK for Tag Based Linear Subspaces

Recall, the languages we handle can be characterized as

$$\left\{ \left\langle \vec{x} \cdot \left[\ \mathsf{A}^{t \times (n-1)} \ \middle| \ (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) \ \right] \cdot \mathbf{g}_1, \textsc{tag} \right\rangle \ | \ \vec{x} \in \mathbb{Z}_q{}^t, \textsc{tag} \in \mathbb{Z}_q \right\}$$

where $\mathsf{A}^{t \times (n-1)} \cdot \mathbf{g}_1, \vec{a}_1 \cdot \mathbf{g}_1$ and $\vec{a}_2 \cdot \mathbf{g}_1$ are parameters of the language. Write A as $[\mathsf{A}_l^{t \times t} \mid \mathsf{A}_r^{t \times (n-1-t)}]$, where w.l.o.g. A_l is non-singular. While the first $n-1-t$ components in excess of the unknowns, corresponding to A_r, can be verified just as in Section 4, for the last component we proceed as follows. The CRS is generated as:

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

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

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 $\vec{\boldsymbol{l}}' \stackrel{\text{def}}{=} \vec{x} \cdot \left[\ \mathsf{A}_l \ \middle| \ (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) \ \right] \cdot \mathbf{g}_1$. The prover generates the following proof for the last component:

$$\vec{\mathbf{p}} := \left[\ \vec{\boldsymbol{l}}' \ \middle| \ \vec{x} \cdot (\mathbf{CRS}_{1,0} + \textsc{tag} \cdot \mathbf{CRS}_{1,1}) \ \right]$$

Verifier: Given a proof $\vec{\mathbf{p}}$, which includes the candidate for membership, the verifier checks the following:

$$e(\vec{\mathbf{p}}, \mathbf{CRS}_{2,0} + \textsc{tag} \cdot \mathbf{CRS}_{2,1}) \stackrel{?}{=} \mathbf{0}_T$$

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

Completeness: We have,

$$\vec{\mathbf{p}} = \left[\ \vec{x} \cdot \mathsf{A}_l \ \middle| \ \vec{x} \cdot (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) \ \middle| \ \vec{x} \cdot (\mathsf{A}_l \cdot \mathsf{D}_1 + \mathsf{A}_l \cdot \textsc{tag} \cdot \mathsf{D}_2 + (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) \cdot b^{-1}) \ \right] \cdot \mathbf{g}_1$$

and

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

Therefore,

$$\begin{aligned} & e(\vec{\mathbf{p}}, \mathbf{CRS}_{2,0} + \textsc{tag} \cdot \mathbf{CRS}_{2,1}) \\ = & \begin{pmatrix} \vec{x} \cdot \mathsf{A}_l \cdot b \cdot (\mathsf{D}_1 + \textsc{tag} \cdot \mathsf{D}_2) + \\ \vec{x} \cdot (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) - \\ \vec{x} \cdot (\mathsf{A}_l \cdot \mathsf{D}_1 + \mathsf{A}_l \cdot \textsc{tag} \cdot \mathsf{D}_2 + (\vec{a}_1^\top + \textsc{tag} \cdot \vec{a}_2^\top) \cdot b^{-1}) \cdot b \end{pmatrix} \cdot e(\mathbf{g}_1, \mathbf{g}_2) = \mathbf{0}_T \end{aligned}$$

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 3, we compute the CRS's in game G_1 as follows. Let $\begin{bmatrix} \mathsf{W}_1^{t\times 1} \\ 1 \end{bmatrix}$ be the null-space of $\left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_1^\top\ \right]$ and let $\begin{bmatrix} \mathsf{W}_2^{t\times 1} \\ 1 \end{bmatrix}$ be the null-space of $\left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_2^\top\ \right]$. Then the CRS's in game G_1 are:

$$\begin{aligned}
\mathsf{CRS}_{1,0} &:= \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_1^\top\ \right] \cdot \left(\begin{bmatrix} \mathsf{D}_1' \\ 0 \end{bmatrix} + \begin{bmatrix} \mathsf{W}_1 \\ 1 \end{bmatrix} \cdot b^{-1} \right) \cdot \mathbf{g}_1 = \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_1^\top\ \right] \cdot \begin{bmatrix} \mathsf{D}_1' \\ 0 \end{bmatrix} \cdot \mathbf{g}_1 \\
\mathsf{CRS}_{1,1} &:= \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_2^\top\ \right] \cdot \left(\begin{bmatrix} \mathsf{D}_2' \\ 0 \end{bmatrix} + \begin{bmatrix} \mathsf{W}_2 \\ 1 \end{bmatrix} \cdot b^{-1} \right) \cdot \mathbf{g}_1 = \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_2^\top\ \right] \cdot \begin{bmatrix} \mathsf{D}_2' \\ 0 \end{bmatrix} \cdot \mathbf{g}_1 \\
\mathsf{CRS}_{2,0}^{(t+2)\times 1} &:= \begin{bmatrix} b \cdot \mathsf{D}_1' + \mathsf{W}_1 \\ 1 \\ -b \end{bmatrix} \cdot \mathbf{g}_2 \qquad \mathsf{CRS}_{2,1}^{(t+2)\times 1} := \begin{bmatrix} b \cdot \mathsf{D}_2' + \mathsf{W}_2 \\ 0 \\ 0 \end{bmatrix} \cdot \mathbf{g}_2
\end{aligned}$$

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

$$\begin{aligned}
\mathsf{A}' \cdot \vec{\mathrm{w}}^\top = \left[\ \mathsf{A}_l \mid (\vec{\mathrm{a}}_1^\top + \textsc{tag} \cdot \vec{\mathrm{a}}_2^\top)\ \right] \cdot \begin{bmatrix} \mathsf{W}_1 + \textsc{tag} \cdot \mathsf{W}_2 \\ 1 \end{bmatrix} &= \mathsf{A}_l \cdot (\mathsf{W}_1 + \textsc{tag} \cdot \mathsf{W}_2) + (\vec{\mathrm{a}}_1^\top + \textsc{tag} \cdot \vec{\mathrm{a}}_2^\top) \\
&= \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_1^\top\ \right] \cdot \begin{bmatrix} \mathsf{W}_1 \\ 1 \end{bmatrix} + \textsc{tag} \cdot \left[\ \mathsf{A}_l \mid \vec{\mathrm{a}}_2^\top\ \right] \cdot \begin{bmatrix} \mathsf{W}_2 \\ 1 \end{bmatrix} = 0
\end{aligned}$$

The rest of the proof is similar to the rest of the proof of soundness in Theorem 3, since A' defines the tag-based language.

D Split-CRS QA-NIZK Proofs

We note that the QA-NIZK described in Section 4 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, and a formal definition follows.

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

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 $(x, w) \notin R_\rho$.

E Split-CRS QA-NIZK for Affine Spaces

Consider languages that are affine spaces

$$L_\rho = \{(\vec{\mathrm{x}} \cdot \mathsf{A} + \vec{\mathrm{a}}) \cdot \mathbf{g}_1 \in G_1^n \mid \vec{\mathrm{x}} \in {\mathbb{Z}_q}^t\}$$

with parameter ρ being $\mathsf{A}^{t \times n} \cdot \mathbf{g}_1$ and $\vec{\mathrm{a}}^{1 \times n} \cdot \mathbf{g}_1$. The parameter language $\mathcal{L}_{\text{par}}$ just specifies A and $\vec{\mathrm{a}}$. A distribution over $\mathcal{L}_{\text{par}}$ is called robust if with overwhelming probability the left most $t \times t$ sub-matrix of A is non-singular (full-ranked). If $\vec{\mathrm{a}} \cdot \mathbf{g}_1$ is given as part of the verifier CRS, then a QA-NIZK for distributions over this class follows directly from the construction in Section 4. However, that would make the QA-NIZK non split-CRS. We now show that the techniques of Section 4 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 $s \stackrel{\triangle}{=} n - t$: this is the number of equations in excess of the unknowns. K_{11} starts by generating 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 generate a row vector $\vec{\mathrm{d}}^{1 \times s}$ at random from $\mathbb{Z}_q$. Next, it generates

$$\mathbf{CRS}_v^{(n+s) \times s} = \begin{bmatrix} b \cdot \mathsf{D} \\ \mathsf{I}^{s \times s} \\ -b \cdot \mathsf{I}^{s \times s} \end{bmatrix} \cdot \mathbf{g}_2$$

The verifier CRS ψ_v is the matrix $\mathbf{CRS}_v$ and an additional (row) vector in the target group $\vec{\mathbf{f}}^{1 \times s} = e(\mathbf{g}_1, b \cdot \vec{\mathrm{d}} \cdot \mathbf{g}_2)$.

The prover CRS generator K_{12} generates

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

The (prover) CRS ψ_p is just the matrix $\mathbf{CRS}_p$.

Prover: Given candidate $(\vec{x} \cdot \mathsf{A} + \vec{a}) \cdot \mathbf{g}_1$ with witness vector $\vec{x}$, the prover generates the following proof:

$$\vec{\mathbf{p}} := [\vec{x} \mid 1] \cdot \mathsf{CRS}_p$$

Note that the proof includes the candidate.

Verifier: Given a proof $\vec{\mathbf{p}}$, the verifier first checks that the first n elements form the candidate $\vec{\mathbf{l}}$ and then checks the following:

$$e(\vec{\mathbf{p}}, \mathsf{CRS}_v) \stackrel{?}{=} \vec{\mathbf{f}}$$

The split-CRS QA-NIZK for affine spaces also naturally extends to include tags as described in Section 5.

Theorem 11 *Let $\mathcal{D}$ be a robust and efficiently witness-samplable distribution over $\mathcal{L}_{par}$ as defined above. For any group generation algorithm for which the DDH assumption holds for group G_2, the above K_0, K_1, the Prover P, and the Verifier V constitute a split-CRS quasi-adaptive NIZK for distribution $\mathcal{D}$ over the above class of languages $\{L_\rho\}$ with associated parameter language $\mathcal{L}_{par}$.*

The proof of this theorem is similar to that of theorem 3. We higlight the main points in the proof sketch below.
Proof:

Completeness:

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

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

Soundness: As in the proof of Theorem 3, we compute the CRS's in game 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). Implicitly 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 implicity set

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

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 $\mathsf{\textbf{CRS}}_p$ can be computed as just

$$\left[\begin{array}{c|c} \mathsf{A} & \mathsf{A} \cdot \begin{bmatrix} \mathsf{D}' \\ 0 \end{bmatrix} \\ \vec{\mathrm{a}} & -\vec{\mathrm{d}}_1 \end{array} \right]$$

Rest of the proof is as in the proof of Theorem 3, but crucially noting that in the proof of lemma 5 while employing DDH in group $\mathbb{G}_2$, the challenge values $b \cdot \mathbf{g}_2$ and $r \cdot b \cdot \mathbf{g}_2$ suffice to simulate all occurences of b in both the CRS-es (including $\vec{\mathbf{f}}$).

□

F Application Details

Proof of the Signature Scheme Theorem.

Theorem 4 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. $\square$

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 [4]) 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 [10] 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,

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

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 [10] stated a scheme using Groth-Sahai NIZK proofs requiring $(5+16)$ elements.

G Dual System IBE under SXDH Assumption

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

$$\langle R = r \cdot \mathbf{g}_1,\ S = r \cdot \mathbf{f}\,,\ T = \mathbf{u} + r \cdot (\mathbf{d} + i \cdot \mathbf{e}) \rangle$$

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

$$L_\rho = \{ r \cdot (\begin{bmatrix} \mathbf{g}_1 & \mathbf{f} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \mathbf{d} \end{bmatrix} + i \cdot \begin{bmatrix} 0 & 0 & \mathbf{e} \end{bmatrix}) + \begin{bmatrix} 0 & 0 & \mathbf{u} \end{bmatrix} \mid r \in \mathbb{Z}_q \}$$

where ρ consists of the matrices $\begin{bmatrix} \mathbf{g}_1 & \mathbf{f} \end{bmatrix}$ 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 4, this is a language with rank one, and two dimensions, i.e. $n = 2$, $t = 1$ and $s = (n - t) = 1$. Let $\mathbf{f} = \mathbf{g}_1^c$ 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 4 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 A), 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 CRS_1 (prover CRS) is then specified by $\mathsf{A} \cdot \mathbf{g}_1$ and $\Delta_3 \cdot \mathbf{g}_1$. Recall, the prover CRS is to be used in KeyGen in IBE.

The verifier CRS, i.e. CRS_1 is specified by $\mathbf{g}_2$, $b \cdot \mathbf{g}_2$ and $(b \cdot \Delta_3 - c) \cdot \mathbf{g}_2$. Similarly, the CRS-es for the tag based element T, and the affine shift $\mathbf{u}$ can be obtained from Sections 5 and E 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 C). Similarly, using Appendix E, we derive the CRS element required for the affine shift, which will be $e(\mathbf{g}_1, (b \cdot \Delta_4 - u)\mathbf{g}_2)$ (see the vector $\vec{\mathbf{f}}$ in Appendix E, 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 4). 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 6. 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 [18]. 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 $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_T)$ with $\mathbf{g}_1$ and $\mathbf{g}_2$ as generators of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively . Assume that $\mathcal{G}_1$ and $\mathcal{G}_2$ are of order q, and let e be a bilinear pairing on $\mathcal{G}_1 \times \mathcal{G}_2$. Then it picks c at random from $\mathbb{Z}_q$, and sets $\mathbf{f} = \mathbf{g}_1^c$. 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 $\mathbf{PK}$:
$\mathbf{g}_2$, $\mathbf{g}_2^b$, $\mathbf{v}_1 = \mathbf{g}_2^{-\Delta_1 \cdot b + d}$, $\mathbf{v}_2 = \mathbf{g}_2^{-\Delta_2 \cdot b + e}$, $\mathbf{v}_3 = \mathbf{g}_2^{-\Delta_3 \cdot b + c}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\Delta_4 \cdot b + u}$.
The authority retains the following master secret key $\mathbf{MSK}$: $\mathbf{g}_1$, $\mathbf{f}$, 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

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

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

$$R = \mathbf{g}_1^r, S = \mathbf{f}^r, T = \mathbf{g}_1^{u + r \cdot (d + i \cdot e)},\ W_1 = \mathbf{g}_1^{-\Delta_4 - r \cdot (\Delta_1 + i \cdot \Delta_2)}, W_2 = \mathbf{g}_1^{-r \cdot \Delta_3}$$

as the secret key K_i for identity i.

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

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

and output C_0/κ.

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):

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

The tag TAG is chosen randomly as in game G_0. This simulation of the ciphertext is called *semi-functional ciphertext* in [18]. 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 $\mathcal{G}_1$ on the tuples $\langle \mathbf{g}_2, \mathbf{g}_2^b, \mathbf{g}_2^{bs}, \mathbf{g}_2^s \rangle$, and $\langle \mathbf{g}_2, \mathbf{g}_2^b, \mathbf{g}_2^{bs}, \mathbf{g}_2^{s+s'} \rangle$. The former tuple is used in game G_0 and the latter in game G_1.

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

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

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

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

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 12 *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

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

As a consequence c is not used at all in the simulation of the ciphertext (whose elements are all in group $\mathcal{G}_1$). 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

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

Recall that in game H_1, the secret key is being generated as in Equation (4), with $d = d_1 + c d_2$ and $e = e_1 + c e_2$. 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 $\langle \mathbf{g}_1, \mathbf{f} = \mathbf{g}_1^c, \mathbf{g}_1^{r_j}, \mathbf{g}_1^{c r_j} \rangle$ and $\langle \mathbf{g}_1, \mathbf{f} = \mathbf{g}_1^c, \mathbf{g}_1^{r_j}, \mathbf{g}_1^{c r_j + r'_j} \rangle$, 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 j-th secret key as

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

where r_j, r'_j and r''_j are chosen randomly and independently (and independently from all other variables).

The view of the Adversary in game H_3 and H_2 is statistically identical by noting that $d = d_1 + c \cdot d_3$, 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 $d_1 + c d_2$ etc.), and also chooses r'''_j at random (and independent of r_j, r'_j and other variables) and outputs the ciphertext as

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

Game H_4 is statistically identical to game H_3, as $(= \ u + r'_j \cdot r''_j)$ 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 12 statement. $\square$

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 (3)). 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. $\square$