Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces

Charanjit S. Jutla[1] and Arnab Roy[2]

[1] IBM T. J. Watson Research Center
Yorktown Heights, NY 10598, USA
csjutla@us.ibm.com
[2] Fujitsu Laboratories of America
Sunnyvale, CA 94085, USA
arnab@cs.stanford.edu

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

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

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

1 Introduction

In [13] a remarkably efficient non-interactive zero-knowledge (NIZK) proof system [3] 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 [12], which was not very efficient) were constructed

K. Sako and P. Sarkar (Eds.) ASIACRYPT 2013 Part I, LNCS 8269, pp. 1–20, 2013.

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 [13].

While the Groth-Sahai system is quite efficient, it still falls short in comparison to Schnorr-based Σ-protocols [8] turned into NIZK proofs in the Random Oracle model [2] using the Fiat-Shamir paradigm [10]. Thus, the quest remains to obtain even more efficient NIZK Proofs. In particular, in a linear system of rank t, some t of the equations already serve as commitments to t variables. Thus, the question arises if, at the very least, fresh commitments to these variables as done in Groth-Sahai NIZKs can be avoided.

Our Contributions. In this paper, we show that for languages that are linear subspaces of vector spaces of the bilinear groups, one can indeed obtain more efficient computationally-sound NIZK proofs in a slightly different *quasi-adaptive* setting, which suffices for many cryptographic applications. In the quasi-adaptive setting, we consider a class of parametrized languages $\{L_\rho\}$, parametrized by ρ, and we allow the CRS generator to generate the CRS based on the language parameter ρ. However, the CRS simulator in the zero-knowledge setting is required to be a single efficient algorithm that works for the whole parametrized class or probability distributions of languages, by taking the parameter as input. We will refer to this property as *uniform simulation.*

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

We remark that adaptive NIZK proofs [3] also allow the CRS to depend on the language, but without requiring uniform simulation. Such NIZK proofs that allow different efficient simulators for each particular language (from a parametrized class) are unlikely to be useful in applications. Thus, most NIZK proofs, including Groth-Sahai NIZKs, actually show that the same efficient

[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.

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. by allowing the (uniform) simulator to take language parameters as input[2].

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

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

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

The main idea underlying our construction can be summarized as follows. Suppose the CRS can be set to be a basis for the null-space $L_\rho^\perp$ of the language L_ρ. Then, just pairing a potential language candidate with $L_\rho^\perp$ and testing for all-zero suffices to prove that the candidate is in L_ρ, as the null-space of $L_\rho^\perp$ is just L_ρ. However, efficiently computing null-spaces in hard bilinear groups is itself hard. Thus, an efficient CRS simulator cannot generate $L_\rho^\perp$, but can give a (hiding) commitment that is computationally indistinguishable from a binding commitment to $L_\rho^\perp$. To achieve this we use a homomorphic commitment just as in the Groth-Sahai system, but we can use the simpler El-Gamal encryption style commitment as opposed to the more involved Groth-Sahai commitments, and this allows for a more efficient verifier[3]. As we will see later in Section 5, a more efficient verifier is critical for obtaining short identity based encryption schemes (IBE).

In fact, the idea of using the null-space of the language is reminiscent of Waters' dual-system IBE construction [24], and indeed our system is inspired by that construction[4], although the idea of using it for NIZK proofs, and in particular the proof of soundness is novel. Another contribution of the paper is in the definition of quasi-adaptive NIZK proofs.

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

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

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

For n equations in t variables, our quasi-adaptive computationally-sound NIZK proofs for linear subspaces require only $k(n-t)$ group elements, under the k-linear decisional assumption [23,5]. Thus, under the XDH assumption for bilinear groups, our proofs require only $(n-t)$ group elements. In contrast, the Groth-Sahai system requires $(n+2t)$ group elements. Similarly, under the decisional linear assumption (DLIN), our proofs require only $2(n-t)$ group elements, whereas the Groth-Sahai system requires $(2n+3t)$ group elements. These parameters are summarized in Table 1. While our CRS size grows proportional to $t(n-t)$, more importantly there is a significant comparative improvement in the number of pairings required for verification. Specifically, under XDH we require at most half the number of pairings, and under DLIN we require at most 2/3 the number of pairings. The Σ-protocol NIZK proofs based on the Random Oracle model require n group elements, t elements of $\mathbb{Z}_q$ and 1 hash value. Although our XDH based proofs require less number of group elements, the Σ-protocol proofs do not require bilinear groups and have the advantage of being proofs of knowledge (PoK). We remark that the Groth-Sahai system is also not a PoK for witnesses that are $\mathbb{Z}_q$ elements. A recent paper by Escala et al [9] has also optimized proofs of linear subspaces in a language dependent CRS setting. Their system also removes the need for commitment to witnesses but still implicitly uses Groth Sahai proofs. In comparison, our proofs are still much shorter.

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

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

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

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

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

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

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

The extension to tags is very important, as we now discuss. Many applications require that the NIZK proof also be simulation-sound. However, extending NIZK proofs for bilinear groups to be unbounded simulation-sound requires handling quadratic equations (see [5] for a generic construction). On the other hand, many applications just require one-time simulation soundness, and as has been shown in [14], this can be achieved for linear subspaces by projective hash proofs [7]. Projective hash proofs can be defined by linear extensions, but require use of tags. Thus, our system can handle such equations. Many applications, such as signatures, can also achieve implicit unbounded simulation soundness using projective hash proofs, and such applications can utilize our system (see Section 5).

Applications. While the cryptographic literature is replete with NIZK proofs, we will demonstrate the applicability of quasi-adaptive NIZKs, and in particular our efficient system for linear subspaces, to a few recent applications such as signature schemes [5], UC commitments [11], password-based key exchange [16,14], key-dependent encryption [5]. For starters, based on [11], our system yields an adaptive UC-secure commitment scheme (in the erasure model) that has only four group elements as commitment, and another four as opening (under the DLIN assumption; and $3 + 2$ under SXDH assumption), whereas the original scheme using Groth-Sahai NIZKs required $5 + 16$ group elements.

We also obtain one of the shortest signature schemes under a static standard assumption, i.e. SXDH, that only requires five group elements. We also show how this signature scheme can be extended to a short fully secure (and perfectly complete) dual-system IBE scheme, and indeed a scheme with ciphertexts that are only four group elements plus a tag (under the SXDH assumption). This is the shortest IBE scheme under the SXDH assumption, and is technically even shorter than a recent and independently obtained scheme of [6] which requires five group elements as ciphertext. Table 2 depicts numerical differences between the parameter sizes of the two schemes. The SXDH-IBE scheme of [6] uses the concept of dual pairing vector spaces (due to Okamoto and Takashima [19,20],

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

and synthesized from Waters' dual system IBE). However, the dual vector space and its generalizations due to others [17] do not capture the idea of proof verification. Thus, one of our main contributions can be viewed as showing that the dual system not only does zero-knowledge simulation but also extends to provide a computationally sound verifier for general linear systems.

Table 2. Comparison with the SXDH-based IBE of Chen et al. [6]. The notation $|\cdot|$ denotes the bit length of an element of the given group.

	Public Key	Secret Key	Ciphertext	#Pairings	Anonymity
CLLWW12 [6]	$8\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert$	$4\lvert\mathbb{G}_2\rvert$	$4\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert$	4	yes
This paper	$5\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert$	$5\lvert\mathbb{G}_2\rvert$	$3\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_T\rvert + \lvert\mathbb{Z}_q\rvert$	3	yes

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

Organization of the Paper. We begin the rest of the paper with the definition of quasi-adaptive NIZKs in Section 2. In Section 3 we develop quasi-adaptive NIZKs for linear subspaces under the XDH assumption. In Section 4, we extend our system to include tags, define a notion called split-CRS QA-NIZKs and extend our system to construct split-CRS NIZKs for affine spaces. Finally, we demonstrate applications of our system in Section 5. We defer detailed proofs and descriptions to the full paper [15]. We also describe our system based on the k-linear assumption in [15].

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 $L = \{x \mid \exists w : R(x, w)\}$. 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

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

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

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

Quasi-Adaptive Completeness:

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

Quasi-Adaptive Soundness:

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

Quasi-Adaptive Zero-Knowledge:

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

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

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

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

Note that ψ is the CRS in the above definitions.

3 QA-NIZK for Linear Subspaces under the XDH Assumption

Setup. Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order q with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ chosen by a group generation algorithm. Let $\mathbf{g}_1$ and $\mathbf{g}_2$ be generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Let $\mathbf{0}_1$, $\mathbf{0}_2$ and $\mathbf{0}_T$ be the identity elements in the three groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ respectively. We use additive notation for the group operations in all the groups.

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

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

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

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

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

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

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

Robust and Efficiently Witness-Samplable Distributions. Let the $t \times n$ dimensional matrix $\mathbf{A}$ be chosen according to a distribution $\mathcal{D}$ on $\mathcal{L}_{\mathrm{par}}$. We will call the distribution $\mathcal{D}$ *robust* if with probability close to one the left-most t columns of $\mathbf{A}$ are full-ranked. We will call a distribution $\mathcal{D}$ on $\mathcal{L}_{\mathrm{par}}$ *efficiently witness-samplable* if there is a probabilistic polynomial time algorithm such that it outputs a pair of matrices $(\mathbf{A}, \mathsf{A})$ that satisfy the relation $\mathcal{R}_{\mathrm{par}}$ (i.e., $\mathcal{R}_{\mathrm{par}}(\mathbf{A}, \mathsf{A})$ holds), and further the resulting distribution of the output $\mathbf{A}$ is same as $\mathcal{D}$. For example, the uniform distribution on $\mathcal{L}_{\text{par}}$ is efficiently witness-samplable, by first picking A at random, and then computing $\mathbf{A}$. As an example of a robust distribution, consider a distribution $\mathcal{D}$ on (2×3)-dimensional matrices $\begin{bmatrix} \mathbf{g} & \mathbf{0}_1 & \mathbf{h} \\ \mathbf{0}_1 & \mathbf{f} & \mathbf{h} \end{bmatrix}$ with $\mathbf{g}, \mathbf{f}$ and $\mathbf{h}$ chosen randomly from $\mathbb{G}_1$. It is easy to see that the first two columns are full-ranked if $\mathbf{g} \neq \mathbf{0}_1$ and $\mathbf{f} \neq \mathbf{0}_1$, which holds with probability $(1 - 1/q)^2$.

QA-NIZK Construction. We now describe a computationally sound quasi-adaptive NIZK $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ for linear subspace languages $\{L_{\mathbf{A}}\}$ with parameters sampled from a robust and efficiently witness-samplable distribution $\mathcal{D}$ over the associated parameter language $\mathcal{L}_{\text{par}}$.

Algorithm K_0. K_0 is same as the group generation algorithm for which the XDH assumption holds. $\lambda \stackrel{\text{def}}{=} (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2) \leftarrow \mathsf{K}_0(1^m)$, with $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2)$ as described above.

We will assume that the size $t \times n$ of the matrix $\mathbf{A}$ is either fixed or determined by the security parameter m. In general, t and n could also be part of the parameter language, and hence t, n could be given as part of the input to CRS generator K_1.

Algorithm K_1. The algorithm K_1 generates the CRS as follows. Let $\mathbf{A}^{t \times n}$ be the parameter supplied to K_1. Let $s \stackrel{\text{def}}{=} n - t$: this is the number of equations in excess of the unknowns. It generates a matrix $\mathsf{D}^{t \times s}$ with all elements chosen randomly from $\mathbb{Z}_q$ and a single element b chosen randomly from $\mathbb{Z}_q$. The common reference string (CRS) has two parts $\mathbf{CRS}_p$ and $\mathbf{CRS}_v$ which are to be used by the prover and the verifier respectively.

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

Here, I denotes the identity matrix. Note that $\mathbf{CRS}_v$ is independent of the parameter.

Prover P. Given candidate $\vec{\boldsymbol{l}} = \vec{\mathrm{x}} \cdot \mathbf{A}$ with witness vector $\vec{\mathrm{x}}$, the prover generates the following proof consisting of s elements in $\mathbb{G}_1$:

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

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

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

The security of the above system depends on the DDH assumption in group $\mathbb{G}_2$. Since $\mathbb{G}_2$ is a bilinear group, this assumption is known as the XDH assumption. These assumptions are standard and are formally described in [15].

Theorem 1. *The above algorithms* $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ *constitute a computationally sound quasi-adaptive NIZK proof system for linear subspace languages* $\{L_{\mathbf{A}}\}$ *with* *parameters* $\mathbf{\mathsf{A}}$ *sampled from a robust and efficiently witness-samplable distribution* $\mathcal{D}$ *over the associated parameter language* $\mathcal{L}_{par}$*, given any group generation algorithm for which the DDH assumption holds for group* $\mathbb{G}_2$*.*

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

Proof Intuition. A detailed proof of the theorem can be found in [15]. Here we give the main idea behind the working of the above quasi-adaptive NIZK, and in particular the soundness requirement which is the difficult part here. We first observe that completeness follows by straightforward bilinear manipulation. Zero Knowledge also follows easily: the simulator generates the same CRS as above but retains D and b as trapdoors. Now, given a language candidate $\vec{\boldsymbol{l}}$, the proof is simply $\vec{\mathbf{p}} := \vec{\boldsymbol{l}} \cdot \begin{bmatrix} \mathsf{D} \\ b^{-1} \cdot \mathrm{I}^{s\times s} \end{bmatrix}$. If $\vec{\boldsymbol{l}}$ is in the language, i.e., it is $\vec{\mathrm{x}} \cdot \mathbf{\mathsf{A}}$ for some $\vec{\mathrm{x}}$, then the distribution of the simulated proof is identical to the real world proof.

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

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

Now we show that if an efficient adversary can produce a "proof" $\vec{\mathbf{p}}$ for which the above pairing test holds and yet the candidate $\vec{\boldsymbol{l}}$ is not in $L_{\mathbf{\mathsf{A}}}$, then it implies an efficient adversary that can break DDH in group $\mathbb{G}_2$. So consider a DDH game, where a challenger either provides a real DDH-tuple $\langle \mathbf{g}_2, b \cdot \mathbf{g}_2, r \cdot \mathbf{g}_2, \chi = br \cdot \mathbf{g}_2 \rangle$ or a fake DDH tuple $\langle \mathbf{g}_2, b \cdot \mathbf{g}_2, r \cdot \mathbf{g}_2, \chi = br' \cdot \mathbf{g}_2 \rangle$. Let us partition the $\mathbb{Z}_q$ matrix A as $\left[\mathsf{A}_0^{t\times t} \middle| \mathsf{A}_1^{t\times s} \right]$ and the candidate vector $\vec{\boldsymbol{l}}$ as $\left[\vec{\boldsymbol{l}}_0^{\,1\times t} \middle| \vec{\boldsymbol{l}}_1^{\,1\times s} \right]$. Note that, since A_0 has rank t, the elements of $\vec{\boldsymbol{l}}_0$ are 'free' elements and $\vec{\boldsymbol{l}}_0$ can be extended to a unique n element vector $\vec{\boldsymbol{l}}\,'$, which is a member of $L_{\mathbf{\mathsf{A}}}$. This member vector $\vec{\boldsymbol{l}}\,'$ can be computed as $\vec{\boldsymbol{l}}\,' := \left[\vec{\boldsymbol{l}}_0 \;\middle|\; -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W} \right]$, nothing $\mathsf{W} = -\mathsf{A}_0^{-1}\mathsf{A}_1$. The proof of $\vec{\boldsymbol{l}}\,'$ is computed as $\vec{\mathbf{p}}\,' := \vec{\boldsymbol{l}}_0 \cdot \mathsf{D}'$. Since both $(\vec{\boldsymbol{l}}, \vec{\mathbf{p}})$ and $(\vec{\boldsymbol{l}}\,', \vec{\mathbf{p}}\,')$ pass the verification equation, we obtain: $\vec{\boldsymbol{l}}_1' - \vec{\boldsymbol{l}}_1 = b(\vec{\mathbf{p}}\,' - \vec{\mathbf{p}})$, where $\vec{\boldsymbol{l}}_1' = -\vec{\boldsymbol{l}}_0 \cdot \mathsf{W}$. In particular there exists $i \in [1, s]$, such that, $\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i} = b(\mathbf{p}_i' - \mathbf{p}_i) \neq \mathbf{0}_1$. This gives us a straightforward test for the DDH challenge: $e(\boldsymbol{l}_{1i}' - \boldsymbol{l}_{1i}, r \cdot \mathbf{g}_2) \stackrel{?}{=} e(\mathbf{p}_i' - \mathbf{p}_i, \chi)$. This leads to a proof of soundness of the QA-NIZK.

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

Definition 1. *Consider a generation algorithm $\mathcal{G}$ taking the security parameter as input, that outputs a tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathbf{g}_1, \mathbf{g}_2)$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q with generators $\mathbf{g}_1, \mathbf{g}_2$ and $e(\mathbf{g}_1, \mathbf{g}_2)$ respectively and which allow an efficiently computable $\mathbb{Z}_q$-bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The assumption asserts that the following problem is hard: Given $\mathbf{f}, \mathbf{f}^b \xleftarrow{\$} \mathbb{G}_2$, output $\mathbf{h}, \mathbf{h}' \in \mathbb{G}_1$, such that $\mathbf{h}' = \mathbf{h}^b \neq \mathbf{0}_1$.*

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

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

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

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

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

4 Extensions

In this section we consider some useful extensions of the concepts and constructions of QA-NIZK systems. We show how the previous system can be extended to include tags. The tags are elements of $\mathbb{Z}_q$, are included as part of the proof and are used as part of the defining equations of the language. We define a notion called split-CRS QA-NIZK system, where the prover and verifier use distinct parts of a CRS and we construct a split-CRS system for affine systems.

Tags. While our system works for any number of components in the tuple (except the first t) being dependent on any number of tags, to simplify the presentation we will focus on only one dependent element and only one tag. Also for simplicity, we will assume that this element is an affine function of the tag (the function being defined by parameters). We can handle arbitrary polynomial functions of the tags as well, but we will focus on affine functions here as most applications seem to need just affine functions. Then, the languages we handle can be characterized as

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

where $\mathbf{A}^{t\times(n-1)}$, $\vec{\mathbf{a}}_1^{1\times t}$ and $\vec{\mathbf{a}}_2^{1\times t}$ are parameters of the language. A distribution is still called robust (as in Section 3) if with overwhelming probability the first t columns of $\mathbf{A}$ are full-ranked. Write $\mathbf{A}$ as $[\mathbf{A}_l^{t\times t} \mid \mathbf{A}_r^{t\times(n-1-t)}]$, where without loss of generality, $\mathbf{A}_l$ is non-singular. While the first $n-1-t$ components in excess of the unknowns, corresponding to $\mathbf{A}_r$, can be verified just as in Section 3, for the last component we proceed as follows.

Algorithm K_1**.** The CRS is generated as:

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

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

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

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

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

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

$$e\left(\left[\vec{\boldsymbol{l}}' \mid \vec{\mathbf{p}}\right], \mathbf{CRS}_{v,0} + \textsc{tag}\cdot\mathbf{CRS}_{v,1}\right) \stackrel{?}{=} \mathbf{0}_T$$

The size of the proof is 1 element in the group $\mathbb{G}_1$. The proof of completeness, soundness and zero-knowledge for this quasi-adaptive system is similar to proof in Section 3 and a proof sketch can be found in [15].

Split-CRS QA-NIZK Proofs. We note that the QA-NIZK described in Section 3 has an interesting split-CRS property. In a **split-CRS QA-NIZK** for a distribution of relations, the CRS generator K_1 generates two CRS-es ψ_p and ψ_v, such that the prover P *only* needs ψ_p, and the verifier V *only* needs ψ_v. In addition, the CRS ψ_v is *independent* of the particular relation R_ρ. In other words the CRS generator K_1 can be split into two PPTs K_{11} and K_{12}, such that K_{11} generates ψ_v using just λ, and K_{12} generates ψ_p using ρ and a state output by K_{11}. The key generation simulator S_1 is also split similarly. The formal definition is given in [15].

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

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

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

The parameter language $\mathcal{L}_{\text{par}}$ just specifies $\mathbf{A}$ and $\vec{\mathbf{a}}$. A distribution over $\mathcal{L}_{\text{par}}$ is called robust if with overwhelming probability the left most $t\times t$ sub-matrix of $\mathbf{A}$

is non-singular (full-ranked). If $\vec{\mathbf{a}}$ is given as part of the verifier CRS, then a QA-NIZK for distributions over this class follows directly from the construction in Section 3. However, that would make the QA-NIZK non split-CRS. We now show that the techniques of Section 3 can be extended to give a split-CRS QA-NIZK for (robust and witness-samplable) distributions over affine spaces.

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

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

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

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

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

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

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

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

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

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

$$e\left(\left[\vec{\boldsymbol{l}} \mid \vec{\mathbf{p}}\right], \mathbf{CRS}_v\right) \overset{?}{=} \vec{\mathbf{f}}$$

We provide a proof sketch in [15]. The split-CRS QA-NIZK for affine spaces also naturally extends to include tags as described before in this section.

5 Applications

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

Adaptive UC Commitments in the Erasure Model. The SXDH-based commitment scheme from [11] requires the following quasi-adaptive NIZK proof (see [15] for details)

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

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 [11] stated a scheme using Groth-Sahai NIZK proofs requiring $(5+16)$ elements. More details can be found in [15].

One-time (Relatively) Simulation-Sound NIZK for DDH and Others. In [14] 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 [18,22], the NIZK proof can be made one-time simulation sound using a projective hash proof [7] and proving in addition that the hash proof is correct. For the DLIN language, this one-time simulation sound proof (in Groth-Sahai system) required 15 group elements, whereas the quasi-adaptive proof in this paper leads to a proof of size only 5 group elements.

Signatures. We will now show a generic construction of existentially unforgeable signature scheme (against adaptive adversaries) from labeled CCA2-encryption schemes and split-CRS QA-NIZK proof system (as defined in Section 4) for a related language distribution. This construction is a generalization of a signature scheme from [5] which used (fully) adaptive NIZK proofs and *required* constructions based on groups in which the CDH assumption holds.

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

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

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

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

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

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

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

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

Theorem 2. *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 [15]. 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 [5]), 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 [14]). Thus, we just need a (split-CRS) QA-NIZK for the tag-based *affine* system (it is affine because of the additive constant $\mathbf{u}$). There is one variable r, and three equations (four if we consider the original CCA-2 encryption) Thus, we just need $(3-1)*1\,(=2)$ proof elements, leading to a total signature size of 5 elements (i.e. $R, S, \mathbf{u}+H$, and the two proof elements) under the SXDH assumption.

Dual-System Fully Secure IBE. It is well-known that Identity Based Encryption (IBE) implies signature schemes (due to Naor), but the question arises whether the above signature scheme using Cramer-Shoup CCA2-encryption and the related QA-NIZK can be converted into an IBE scheme. To achieve this, we take a hint from Naor's IBE to Signature Scheme conversion, and let the signatures (on identities) be private keys of the various identities. The verification of the QA-NIZK from Section 3 works by checking $e\left(\left[\vec{l} \mid \vec{\mathbf{p}}\right], \mathbf{CRS}_v\right) \overset{?}{=} \mathbf{0}_T^{1\times s}$ (or more precisely, $e\left(\left[\vec{l} \mid \vec{\mathbf{p}}\right], \mathbf{CRS}_v\right) \overset{?}{=} \vec{\mathbf{f}}$ for the affine language). However, there are two issues: (1) $\mathbf{CRS}_v$ 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 $\mathbf{CRS}_v$ by a random value s, and then taking a linear combination of the two equations using a public random tag. The right hand side $s \cdot \vec{\mathbf{f}}$ can then serve as secret one-time pad for encryption. Rather than being a provable generic construction, this is more a hint to get to a really short IBE. We give the construction in Appendix A and a complete proof in [15]. It shows an IBE scheme under the SXDH assumption where the ciphertext has only four group

($\mathbb{G}_1$) elements plus a $\mathbb{Z}_q$-tag, which is the shortest IBE known under standard static assumptions[7].

Publicly-Verifiable CCA2 Fully-Secure IBE. We can also extend our IBE scheme above to be publicly-verifiable CCA2-secure [21,1]. Public verifiability is an informal but practical notion: most CCA2-secure schemes have a test of well-formedness of ciphertext, and on passing the test a CPA-secure scheme style decryption suffices. However, if this test can be performed publicly, i.e. without access to the secret key, then we call the scheme publicly-verifiable. While there is a well known reduction from hierarchical IBE to make an IBE scheme CCA2-secure [4], that reduction does not make the scheme publicly-verifiable CCA2 in a useful manner. In the IBE setting, publicly-verifiable *also* requires that it be verifiable if the ciphertext is *valid for the claimed identity.* This can have interesting applications where the network can act as a filter. We show that our scheme above can be extended to be publicly-verifiable CCA2-fully-secure IBE with *only* two additional group elements in the ciphertext (and two additional group elements in the keys). We give the construction in Appendix B and a complete proof in [15]. The IBE scheme above has four group elements (and a tag), where one group element serves as one-time pad for encrypting the plaintext. The remaining three group elements form a linear subspace with one variable as witness and three integer tags corresponding to: (a) the identity, (b) the tag needed in the IBE scheme, and (c) a 1-1 (or universal one-way) hash of some of the elements. We show that if these three group elements can be QA-NIZK proven to be consistent, and given the unique proof property of our QA-NIZKs, then the above IBE scheme can be made CCA2-secure - the dual-system already has implicit simulation-soundness as explained in the signature scheme above, and we show that this QA-NIZK need not be simulation-sound. Since, there are three components, and one variable (see the appendix for details), the QA-NIZK requires only two group elements under SXDH.

References

1. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)
3. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: STOC, pp. 103–112 (1988)
4. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. 36(5), 1301–1328 (2007)

[7] [6] have recently and independently obtained a short IBE under SXDH, but our IBE ciphertexts are even shorter. See Table 2 in the Introduction for detailed comparison.

5. Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
6. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (2013)
7. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
8. Damgård, I.: On Σ protocols, http://www.daimi.au.dk/~ivan/Sigma.pdf
9. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for diffie-hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (2013)
10. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
11. Fischlin, M., Libert, B., Manulis, M.: Non-interactive and re-usable universally composable string commitments with adaptive security. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 468–485. Springer, Heidelberg (2011)
12. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
13. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
14. Jutla, C., Roy, A.: Relatively-sound NIZKs and password-based key-exchange. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 485–503. Springer, Heidelberg (2012)
15. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. Cryptology ePrint Archive, Report 2013/109 (2013), http://eprint.iacr.org/
16. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)
17. Lewko, A.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (2012)
18. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC Annual ACM Symposium on Theory of Computing. ACM Press (May 1990)
19. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
20. Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 214–231. Springer, Heidelberg (2009)
21. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)

22. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS Annual Symposium on Foundations of Computer Science, pp. 543–553. IEEE Computer Society Press (October 1999)
23. Shacham, H.: 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/`
24. Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)

A Dual System IBE under SXDH Assumption

For ease of reading, we switch to multiplicative group notation in the following. **Setup**: The authority uses a group generation algorithm for which the SXDH assumption holds to generate a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with $\mathbf{g}_1$ and $\mathbf{g}_2$ as generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Assume that $\mathbb{G}_1$ and $\mathbb{G}_2$ are of order q, and let e be a bilinear pairing on $\mathbb{G}_1 \times \mathbb{G}_2$. Then it picks c at random from $\mathbb{Z}_q$, and sets $\mathbf{f} = \mathbf{g}_2^c$. It further picks $\varDelta_1$, $\varDelta_2$, $\varDelta_3$, $\varDelta_4$, b, d, e, u from $\mathbb{Z}_q$, and publishes the following public key **PK**:
$\mathbf{g}_1$, $\mathbf{g}_1^b$, $\mathbf{v}_1 = \mathbf{g}_1^{-\varDelta_1 \cdot b + d}$, $\mathbf{v}_2 = \mathbf{g}_1^{-\varDelta_2 \cdot b + e}$, $\mathbf{v}_3 = \mathbf{g}_1^{-\varDelta_3 \cdot b + c}$, and $\mathbf{k} = e(\mathbf{g}_1, \mathbf{g}_2)^{-\varDelta_4 \cdot b + u}$. The authority retains the following master secret key **MSK**: $\mathbf{g}_2$, $\mathbf{f} = (\mathbf{g}_2^c)$, and $\varDelta_1$, $\varDelta_2$, $\varDelta_3$, $\varDelta_4$, d, e, u.

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

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

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

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

$$R = \mathbf{g}_2^r, S = \mathbf{g}_2^{r \cdot c}, T = \mathbf{g}_2^{u + r \cdot (d + i \cdot e)},\ W_1 = \mathbf{g}_2^{-\varDelta_4 - r \cdot (\varDelta_1 + i \cdot \varDelta_2)}, W_2 = \mathbf{g}_2^{-r \cdot \varDelta_3}$$

as the secret key K_i for identity i.

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

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

and output C_0/κ.

Theorem 3. *Under the SXDH Assumption, the above scheme is a fully-secure IBE scheme.*

B Publicly Verifiable CCA2-IBE under SXDH Assumption

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

Consider the language:

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

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

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

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

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

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

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

as the secret key K_i for identity i.

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

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

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

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

Theorem 4. *Under the SXDH Assumption, the above scheme is a CCA2 fully-secure IBE scheme.*