Efficient Non-interactive Proof Systems for Bilinear Groups *

Jens Groth
University College London
j.groth@ucl.ac.uk†

Amit Sahai
University of California Los Angeles
sahai@cs.ucla.edu‡

April 7, 2016

Abstract

Non-interactive zero-knowledge proofs and non-interactive witness-indistinguishable proofs have played a significant role in the theory of cryptography. However, lack of efficiency has prevented them from being used in practice. One of the roots of this inefficiency is that non-interactive zero-knowledge proofs have been constructed for general NP-complete languages such as Circuit Satisfiability, causing an expensive blowup in the size of the statement when reducing it to a circuit. The contribution of this paper is a general methodology for constructing very simple and efficient non-interactive zero-knowledge proofs and non-interactive witness-indistinguishable proofs that work directly for a wide class of languages that are relevant in practice (namely, ones involving the satisfiability of equations over bilinear groups), without needing a reduction to Circuit Satisfiability.

Groups with bilinear maps have enjoyed tremendous success in the field of cryptography in recent years and have been used to construct a plethora of protocols. This paper provides non-interactive witness-indistinguishable proofs and non-interactive zero-knowledge proofs that can be used in connection with these protocols. Our goal is to spread the use of non-interactive cryptographic proofs from mainly theoretical purposes to the large class of practical cryptographic protocols based on bilinear groups.

Keywords: Non-interactive witness-indistinguishability, non-interactive zero-knowledge, common reference string, bilinear groups.

1 Introduction

Non-interactive zero-knowledge proofs and non-interactive witness-indistinguishable proofs have played a significant role in the theory of cryptography. However, lack of efficiency has prevented

*Work presented and part of work done while participating in Securing Cyberspace: Applications and Foundations of Cryptography and Computer Security, Institute of Pure and Applied Mathematics, UCLA, 2006. An extended abstract was presented at Advances in Cryptology – EUROCRYPT 2008, LNCS 4965, pages 415-432. The full version was published in SIAM Journal on Computing 41(5), pages 1193-1232, 2012.

†Supported by EPSRC grants EP/G013829/1 and EP/J009520/1. Part of work done while at UCLA supported by NSF grant 0456717.

‡Research supported in part from a DARPA/ONR PROCEED award, NSF grants 1136174, 1118096, 1065276, 0916574, 0830803, 0627781, 0456717, 0716389, and 0205594, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, an Okawa Foundation Research Grant, a subgrant from SRI as part of the Army Cyber-TA program, and an Alfred P. Sloan Foundation Research Fellowship. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0389. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

them from being used in practice. Our goal is to construct efficient and practical non-interactive zero-knowledge (NIZK) proofs and non-interactive witness-indistinguishable (NIWI) proofs.

Blum, Feldman and Micali [BFM88] introduced NIZK proofs. Their paper and subsequent work, e.g., [FLS99, Dam92, KP98, DDP02], demonstrate that NIZK proofs exist for all of NP. Unfortunately, these NIZK proofs are all very inefficient. While leading to interesting theoretical results, such as the construction of public-key encryption secure against chosen ciphertext attack by Dolev, Dwork and Naor [DDN00], they have not been used in practice.

Since we want to construct NIZK proofs that can be used in practice, it is worthwhile to identify the roots of the inefficiency in the above-mentioned NIZK proofs. One drawback is that they were designed with a general NP-complete language in mind, e.g., Circuit Satisfiability. In practice, we want to prove statements such as "the ciphertext c encrypts a signature on the message m" or "the three commitments c_a, c_b, c_c contain messages a, b, c such that $c = ab$". An NP-reduction of even very simple statements like these gives us big circuits containing thousands of gates and the corresponding NIZK proofs become very large.

Although we want to avoid an expensive NP-reduction, it is still desirable to have a general way to express statements that arise in practice instead of having to construct non-interactive proofs on an ad hoc basis. A useful observation in this context is that many public-key cryptography protocols are based on finite abelian groups. If we can capture statements that express relations between group elements, then we can express statements that come up in practice such as "the commitments c_a, c_b, c_c contain messages such that $c = ab$" or "the plaintext of c is a signature on m", as long as those commitment, encryption, and signature schemes work over the same finite group. We will therefore construct NIWI and NIZK proofs for *group-dependent* languages.

The next issue to address is where to find suitable group-dependent languages. We will look at statements related to groups with a bilinear map, which have become widely used in the design of cryptographic protocols. Not only have bilinear groups been used to give new constructions of such cryptographic staples as public-key encryption, digital signatures, and key agreement (see [Pat05] and the references therein), but bilinear groups have enabled the first constructions achieving goals that had never been attained before. The most notable of these is the Identity-Based Encryption scheme of Boneh and Franklin [BF03] (see also [BB11, BB04, Wat05]), and there are many others, such as Attribute-Based Encryption [SW05, GPSW06], Searchable Public-Key Encryption [BCOP04, BSW06, BW06], and One-time Double-Homomorphic Encryption [BGN05]. For an incomplete list of papers (currently over 200) on the application of bilinear groups in cryptography, see [Bar06].

1.1 Our contribution

For completeness, let us recap the definition of a bilinear group. *Please note that for notational convenience we will follow the tradition of mathematics and use additive notation*[1] *for the binary operations in G_1 and G_2.* We have a probabilistic polynomial time algorithm $\mathcal{G}$ that takes a security parameter as input and outputs $(\mathbf{n}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$. In some cases, $G_1 = G_2$ and $\mathcal{P}_1 = \mathcal{P}_2$, in which case we write $(\mathbf{n}, G, G_T, e, \mathcal{P})$.

- G_1, G_2, G_T are descriptions of cyclic groups of order $\mathbf{n}$.
- The elements $\mathcal{P}_1, \mathcal{P}_2$ generate G_1 and G_2 respectively.

[1] We remark that in the cryptographic literature it is more common to use multiplicative notation for these groups, since the "discrete log problem" is believed to be hard in these groups, which is also important to us. In our setting, however, it will be much more convenient to use multiplicative notation to refer to the action of the bilinear map.

- $e : G_1 \times G_2$ is a non-degenerate bilinear map such that $e(\mathcal{P}_1, \mathcal{P}_2)$ generates G_T and for all $a, b \in \mathbb{Z}_{\mathbf{n}}$ we have $e(a\mathcal{P}_1, b\mathcal{P}_2) = e(\mathcal{P}_1, \mathcal{P}_2)^{ab}$.
- We can efficiently compute group operations, compute the bilinear map and decide membership.

In this work, we develop a general set of highly efficient techniques for proving statements involving bilinear groups. The generality of our work extends in two directions. First, we formulate our constructions in terms of modules over commutative rings with an associated bilinear map. This framework captures all known bilinear groups with cryptographic significance – for both supersingular and ordinary elliptic curves, for groups of both prime and composite order. Second, we consider all mathematical operations that can take place in the context of a bilinear group - addition in G_1 and G_2, scalar point-multiplication, addition or multiplication of scalars, and use of the bilinear map. We also allow both group elements and scalars to be "unknowns" in the statements to be proven.

Since we cover all operations over the bilinear group, we can prove any statement formulated in terms of the operations associated with the bilinear group. With our level of generality, it would for example be easy to write down a short statement, using the operations above, that encodes "c is an encryption of the value committed to in d under the product of the two keys committed to in a and b" where the encryptions and commitments being referred to are existing cryptographic constructions based on bilinear groups. Logical operations like AND and OR are also easy to encode into our framework using standard techniques in arithmetization. The ability to encode logical operations implies we can use our proof system for the NP-complete language Circuit Satisfiability but the main novelty and advantage is the natural way we can directly handle statements over bilinear groups without using NP-reductions.

The proof systems we build are *non-interactive*. This allows them to be used in contexts where interaction is undesirable or impossible. We first build highly efficient witness-indistinguishable proof systems, which are of independent interest. We then show how to, under certain conditions, transform these into zero-knowledge proof systems. We also provide a detailed examination of the efficiency of our constructions in various settings (depending on what type of bilinear group and cryptographic assumption is used).

The security of constructions arising from our framework can be based on *any* of a variety of computational assumptions about bilinear groups (three of which we discuss in detail here).

Informal statement of our results. We consider equations over variables from G_1, G_2 and $\mathbb{Z}_{\mathbf{n}}$ as described in Figure 1. We construct efficient non-interactive witness-indistinguishable proofs for the simultaneous satisfiability of a set of such equations. The witness-indistinguishable proofs have perfect completeness and there are two computationally indistinguishable types of common reference strings giving respectively perfect soundness and perfect witness indistinguishability. We refer to Section 2 for precise definitions.

We also consider the question of non-interactive zero-knowledge. We show that we can give zero-knowledge proofs for multi-scalar multiplication in G_1 or G_2 and for quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. We can also give zero-knowledge proofs for pairing product equations with $t_T = 1$. When $t_T \neq 1$ we can still give zero-knowledge proofs if we can find $\mathcal{P}_1, \mathcal{Q}_1, \ldots, \mathcal{P}_n, \mathcal{Q}_n$ such that $t_T = \prod_{i=1}^{n} e(\mathcal{P}_i, Q_i)$.

In the first part of the article, we give a general description of our techniques. In Section 8, Section 9 and Section 10 we then offer three concrete instantiations that illustrate the use of our techniques. They are based on respectively the subgroup decision assumption [BGN05], the assumption that the decision Diffie-Hellman problem is hard in both G_1 and G_2 (SXDH), and the decisional linear assumption (DLIN) [BBS04]. We note that there are many other possible

Variables: [a]

$$\mathcal{X}_1, \ldots, \mathcal{X}_m \in G_1 \ , \quad \mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G_2 \ , \quad x_1, \ldots, x_{m'}, y_1, \ldots, y_{n'} \in \mathbb{Z}_{\mathbf{n}}.$$

Pairing product equation:

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{Y}_i) \cdot \prod_{i=1}^{m} e(\mathcal{X}_i, \mathcal{B}_i) \cdot \prod_{i=1}^{m}\prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{ij}} \ = \ t_T,$$

for constants $\mathcal{A}_i \in G_1, \mathcal{B}_i \in G_2, t_T \in G_T, \gamma_{ij} \in \mathbb{Z}_{\mathbf{n}}$.

Multi-scalar multiplication equation in G_1: [b]

$$\sum_{i=1}^{n'} y_i \mathcal{A}_i + \sum_{i=1}^{m} b_i \mathcal{X}_i + \sum_{i=1}^{m}\sum_{j=1}^{n'} \gamma_{ij} y_j \mathcal{X}_i \ = \ \mathcal{T}_1,$$

for constants $\mathcal{A}_i, \mathcal{T}_1 \in G_1$ and $b_i, \gamma_{ij} \in \mathbb{Z}_{\mathbf{n}}$.

Multi-scalar multiplication equation in G_2:

$$\sum_{i=1}^{n} a_i \mathcal{Y}_i + \sum_{i=1}^{m'} x_i \mathcal{B}_i + \sum_{i=1}^{m'}\sum_{j=1}^{n} \gamma_{ij} x_i \mathcal{Y}_j \ = \ \mathcal{T}_2$$

for constants $\mathcal{B}_i, \mathcal{T}_2 \in G_2$ and $a_i, \gamma_{ij} \in \mathbb{Z}_{\mathbf{n}}$.

Quadratic equation in $\mathbb{Z}_{\mathbf{n}}$:

$$\sum_{i=1}^{n'} a_i y_i + \sum_{i=1}^{m'} x_i b_i + \sum_{i=1}^{m'}\sum_{j=1}^{n'} \gamma_{ij} x_i y_j \ \equiv \ t \bmod \mathbf{n}$$

for constants $a_i, b_i, \gamma_{ij}, t \in \mathbb{Z}_{\mathbf{n}}$.

[a] We list variables in $\mathbb{Z}_{\mathbf{n}}$ in two separate groups because we will treat them differently in the NIWI proofs. If we wish to deal with only one group of variables in $\mathbb{Z}_{\mathbf{n}}$ we can add equations in $\mathbb{Z}_{\mathbf{n}}$ of the form $x_1 = y_1, x_2 = y_2$, etc.

[b] With multiplicative notation, these equations would be multi-exponentiation equations. We use additive notation for G_1 and G_2, since this will be notationally convenient in the paper, but again stress that the discrete logarithm problem will typically be hard in these groups.

Figure 1: Equations over groups with bilinear map.

instantiations. The instantiations illustrate the variety of ways bilinear groups can be constructed. We can choose prime order groups or composite order groups, we can have $G_1 = G_2$ and $G_1 \neq G_2$, and we can make various cryptographic assumptions. All three security assumptions have been used in the cryptographic literature to build interesting protocols.

For all three instantiations, the techniques presented here yield efficient witness-indistinguishable proofs. In particular, the cost in proof size of each extra equation is constant and independent of the number of variables in the equation. The size of the proofs can be computed by adding the cost, measured in group elements from G_1 or G_2, of each variable and each equation listed in Figure 1. We refer to Sections 8, Section 9 and Section 10 for more detailed tables. The tables should be read with care because the size of a group element depends on the type of bilinear group [GPS08]. We expect the SXDH-based instantiation to yield the smallest proofs when taking the sizes of group elements into account.

	Subgroup decision	SXDH	DLIN
Variable in G_1 or G_2	1	2	3
Variable in $\mathbb{Z}_{\mathbf{n}}$ or $\mathbb{Z}_{\mathbf{p}}$	1	2	3
Paring product equation	1	8	9
Multi-scalar multiplication in G_1 or G_2	1	6	9
Quadratic equation in $\mathbb{Z}_{\mathbf{n}}$ or $\mathbb{Z}_{\mathbf{p}}$	1	4	6

Table 1: Number of group elements each variable or equation adds to the size of a NIWI proof.

1.2 Related work

As we mentioned before, early work on NIZK proofs demonstrated that all NP-languages have non-interactive proofs, but did not yield efficient proofs. One cause for these proofs being inefficient in practice was the need for an expensive NP-reduction to, e.g., Circuit Satisfiability. Another cause of inefficiency was the reliance on the so-called hidden bits model, which even for small circuits is inefficient.

Groth, Ostrovsky, and Sahai [GOS06b, GOS06a] investigated NIZK proofs for Circuit Satisfiability using bilinear groups. This addressed the second cause of inefficiency since their techniques give efficient proofs for Circuit Satisfiability, but to use their proofs one must still make an NP-reduction to Circuit Satisfiability. We stress that while [GOS06b, GOS06a] used bilinear groups, their application was to build proof systems for Circuit Satisfiability. Here, we devise entirely new techniques to deal with general statements about equations in bilinear groups, *without* having to reduce to an NP-complete language.

Addressing the issue of avoiding an expensive NP-reduction, we have works by Boyen and Waters [BW06, BW07] that suggest efficient NIWI proofs for statements related to group signatures. These proofs are based on bilinear groups of composite order and rely on the subgroup decision assumption.

Groth [Gro06] was the first to suggest a general group-dependent language and NIZK proofs for statements in this language. He investigated satisfiability of pairing product equations and only allowed group elements to be variables. He looked at the special case of prime order groups G, G_T with a bilinear map $e : G \times G \to G_T$ and, based on the decisional linear assumption [BBS04], constructed NIZK proofs for such pairing product equations. However, even for very small statements, the very different and much more complicated techniques of Groth yield proofs consisting of thousands of group elements (whereas ours would be in the tens). Our techniques are much easier to understand, significantly more general, and vastly more efficient.

We summarize our comparison with other works on NIZK proofs in Table 2.

We note that there have been many earlier works (starting with [GMR89]) dealing with efficient *interactive* zero-knowledge protocols for a number of algebraic relations. Here, we focus on *non-interactive* proofs. We also note that even for interactive zero-knowledge proofs, no set of techniques was known for dealing with general algebraic assertions arising in bilinear groups, as we do here.

	Inefficient	Efficient
Circuit Satisfiability	Example: Kilian and Petrank [KP98]	Groth, Ostrovsky and Sahai [GOS06b, GOS06a]
Group-dependent language	Groth [Gro06] (restricted case)	This work

Table 2: Classification of NIZK proofs according to usefulness.

1.3 New techniques

[GOS06b, GOS06a, Gro06] start by constructing non-interactive proofs for simple statements and then combine many of them to get more powerful proofs. The main building block in [GOS06b], for instance, is a proof that a given commitment contains either 0 or 1, which has little expressive power on its own. Our approach is the opposite: we directly construct proofs for very expressive languages; as such, our techniques are very different from previous work.

The way we achieve our generality is by viewing the groups G_1, G_2, G_T as modules over the ring $\mathbb{Z}_{\mathbf{n}}$. The ring $\mathbb{Z}_{\mathbf{n}}$ itself can also be viewed as a $\mathbb{Z}_{\mathbf{n}}$-module. We therefore look at the more general question of satisfiability of quadratic equations over $\mathbb{Z}_{\mathbf{n}}$-modules A_1, A_2, A_T with a bilinear map, see Section 3 for details. Since many bilinear groups with various cryptographic assumptions and various mathematical properties can be viewed as modules we are not bound to any particular bilinear group or any particular assumption.

Given modules A_1, A_2, A_T with a bilinear map, we construct new modules B_1, B_2, B_T, also equipped with a bilinear map, and we map the elements in A_1, A_2, A_T into B_1, B_2, B_T. The latter modules will typically be larger thereby giving us room to hide the elements of A_1, A_2, A_T. More precisely, we devise commitment schemes that map variables from A_1, A_2 to the modules B_1, B_2. The commitment schemes are homomorphic both with respect to the module operations and also with respect to the bilinear map.

Our techniques for constructing witness-indistinguishable proofs are fairly involved mathematically, but we will try to present some high level intuition here. (We give more detailed intuition later in Section 6, where we present our main proof system). The main idea is the following: because our commitment schemes are homomorphic *and* we equip them with a bilinear map, we can take the equation that we are trying to prove, and just replace the variables in the equation with commitments to those variables. Of course, because the commitment schemes are hiding, the equations will no longer be valid. Intuitively, however, we can extract out the additional terms introduced by the randomness of the commitments: if we give away these terms in the proof, then this would be a *convincing* proof of the equation's validity (again, because of the homomorphic properties). But, giving away these terms might destroy witness indistinguishability. Suppose, however, that there is only one "additional term" introduced by substituting the commitments. Then, because it would be the unique value which makes the equation true, giving it away would preserve witness indistinguishability! In general, we are not so lucky. But if there are many terms, the nice algebraic environment allows us to randomize the terms such that their distribution is uniform over all possible terms satisfying the equation. We now get witness indistinguishability because all possible witnesses after randomization yield the same uniform distribution of terms satisfying the equation.

1.4 Applications

Independently of our work, Boyen and Waters [BW07] have constructed non-interactive proofs that they use for group signatures (see also their earlier paper [BW06]). These proofs can be seen as examples of the NIWI proofs in instantiation 1 based on the subgroup decision problem.

Subsequent to the announcement of our work, several papers have built upon it: Chandran, Groth and Sahai [CGS07] have constructed ring-signatures of sub-linear size using the NIWI proofs in the first instantiation, which is based on the subgroup decision problem. Groth and Lu [GL07] have used the NIWI and NIZK proofs from the third instantiation to construct a NIZK proof for the correctness of a shuffle. Groth [Gro07] has used the NIWI and NIZK proofs from the third instantiation to construct a fully anonymous group signature scheme. Belenkiy, Chase, Kohlweiss and Lysyanskaya [BCKL08] have used the second and third instantiations to construct non-interactive anonymous credentials. Green and Hohenberger [GH08] have used the third instantiation in a universally composable adaptive oblivious transfer protocol. Also, by attaching NIZK proofs to semantically secure public-key encryption in any instantiation, we get an efficient non-interactive verifiable cryptosystem. Boneh [Bon06] has suggested using this for optimistic fair exchange [Mic03], where two parties use a trusted but lazy third party to guarantee fairness.

1.5 Roadmap

The main result is the NIWI proof that can be found in Section 7. Sections 3, 4, 5 and 6 explain the structure of the NIWI proof, which goes through modules, commitments, a description of the common reference string, and an explanation of how the NIWI proof works. For a concrete illustration of the steps, we refer the reader to the instantiation in Section 8. Other instantiations are given in Section 9 and Section 10. In many cases, our NIWI proofs can also be used as NIZK proofs, which we discuss in Section 11.

2 Non-interactive witness-indistinguishable proofs

Notation. We write $y = A(x; r)$ when the algorithm A, on input x and randomness r, outputs y. We write $y \leftarrow A(x)$ for the process of picking randomness r uniformly at random and setting $y = A(x; r)$. More generally, we write $y \leftarrow S$ for sampling y from the set S according to some probability distribution on S, using the uniform distribution as the default when nothing else is specified.

We write $a \leftarrow A; b \leftarrow B(a); \ldots$ for running the experiment where a is chosen from A, then b is chosen from B, which may depend on a, etc. This yields a probability distribution over the outputs and we write $\Pr\left[a \leftarrow A; b \leftarrow B(a); \ldots : C(a, b, \ldots)\right]$ for the probability of the condition $C(a, b, \ldots)$ being satisfied after running the experiment.

The security of our schemes is governed by a security parameter k, which can be used to scale up the security. Given two functions $f, g : \mathbb{N} \rightarrow [0, 1]$ we write $f(k) \approx g(k)$ when $|f(k) - g(k)| = O(k^{-c})$ for every constant c. We say that f is *negligible* when $f(k) \approx 0$ and that it is *overwhelming* when $f(k) \approx 1$. We say that two families of probability distributions $\{S_1(k)\}_{k\in\mathbb{N}}, \{S_2(k)\}_{k\in\mathbb{N}}$ are perfectly indistinguishable when they are the same for all sufficiently large $k \in \mathbb{N}$, and we say they are computationally indistinguishable if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have

$$\Pr\left[y \leftarrow S_1(k) : \mathcal{A}(1^k, y) = 1\right] \approx \Pr\left[y \leftarrow S_2(k) : \mathcal{A}(1^k, y) = 1\right].$$

Group dependent languages. Let R be an efficiently computable ternary relation. For triplets $(gk, x, w) \in R$ we call gk the setup, x the statement and w the witness. Given some gk we let L be the language consisting of statements x that have a witness w such that $(gk, x, w) \in R$. For a relation that ignores gk this is of course the standard definition of an NP-language. We will be more interested in the case where gk describes a bilinear group, though.

Non-interactive proofs. A non-interactive proof system for a relation R with setup consists of four probabilistic polynomial time algorithms: a setup algorithm $\mathcal{G}$, a common reference string (CRS) generation algorithm K, a prover P and a verifier V. The setup algorithm outputs a setup (gk, sk). In our paper, gk will be a description of a bilinear group. The setup algorithm may output some related information sk, for instance the factorization of the group order. A cleaner case, however, is when sk is just the empty string, meaning the protocol is built on top of the group without knowledge of any trapdoors. The CRS generation algorithm takes (gk, sk) as input and produces a common reference string σ. The prover takes as input (gk, σ, x, w) and produces a proof π. The verifier takes as input (gk, σ, x, π) and outputs 1 if the proof is acceptable and 0 if rejecting the proof. We call $(\mathcal{G}, K, P, V)$ a non-interactive proof system for R with setup $\mathcal{G}$ if it has the completeness and soundness properties described below.

Perfect completeness. A non-interactive proof is complete if an honest prover can convince an honest verifier whenever the statement belongs to the language and the prover holds a witness testifying to this fact.

Definition 1 (Perfect completeness) *We say $(\mathcal{G}, K, P, V)$ is perfectly complete if for all adversaries $\mathcal{A}$ we have*[2]

$$\Pr\Big[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow K(gk, sk); (x, w) \leftarrow \mathcal{A}(gk, \sigma); \pi \leftarrow P(gk, \sigma, x, w) :$$
$$V(gk, \sigma, x, \pi) = 1 \text{ if } (gk, x, w) \in R\Big] = 1.$$

Perfect soundness. A non-interactive proof is sound if it is impossible to prove a false statement.

Definition 2 (Perfect soundness) *We say $(\mathcal{G}, K, P, V)$ is perfectly sound if for all adversaries $\mathcal{A}$ we have*

$$\Pr\Big[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow K(gk, sk); (x, \pi) \leftarrow \mathcal{A}(gk, \sigma) : V(gk, \sigma, x, \pi) = 0 \text{ if } x \notin L\Big] = 1.$$

Perfect culpable soundness. In the standard definition of soundness given above, the adversary tries to create a valid proof for $x \in \bar{L}$. Groth, Ostrovsky and Sahai [GOS06b, Gro06] generalized the notion of soundness to disallowing false proofs of statements $x \in L_{\mathrm{guilt}}$, where L_{guilt} is a language that may depend on gk and σ. They call this notion *culpable* soundness.[3] Standard soundness is a special case with $L_{\mathrm{guilt}} = \bar{L}$, but the notion can be used to capture other interesting cases as well. The instantiation in Section 8 uses groups of composite order $\mathbf{n} = \mathbf{pq}$ and offers an example where culpable soundness captures the inability of the adversary to produce convincing proofs for statements that are false in the order $\mathbf{p}$ subgroups of G and G_T (here $L_{\mathrm{guilt}} \subseteq \bar{L}$ is the language of statements that are false in the order $\mathbf{p}$ subgroups).

Definition 3 (Perfect culpable soundness) *We say $(\mathcal{G}, K, P, V)$ has perfect L_{guilt}-soundness if for all adversaries $\mathcal{A}$ we have*

$$\Pr\Big[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow K(gk, sk); (x, \pi) \leftarrow \mathcal{A}(gk, \sigma) : V(gk, \sigma, x, \pi) = 0 \text{ if } x \in L_{\mathrm{guilt}}\Big] = 1.$$

[2] Since the probability is exactly 1, the definition quantifies over all gk in the support of $\mathcal{G}$ and all $(gk, x, w) \in R$.

[3] In an earlier version of their paper, Groth, Ostrovsky and Sahai [GOS06b] used the term co-soundness instead of culpable soundness.

Composable witness indistinguishability. A statement may have many possible witnesses. A non-interactive proof is witness indistinguishable if the proof does not reveal which of those witnesses the prover has used. The standard definition of witness-indstinguishability requires that proofs using different witnesses for the same statement are computationally indistinguishable. We will obtain a stronger definition of witness indistinguishability called composable witness indistinguishability. In this definition there is a reference string simulator S that generates a simulated CRS and we require that the adversary cannot distinguish a real CRS from a simulated CRS. We also require that on a simulated CRS there is no information whatsoever to distinguish the different witnesses that might have been used to construct the proof. The advantage of this definition is that different types of proofs using the same type of real/simulated CRS can share the same CRS, which facilitates easier security proofs. We will use this composability property in the instantiations in Sections 8, 9 and 10.

Definition 4 (Composable witness indistinguishability) *We say $(\mathcal{G}, K, P, V)$ is composable witness indistinguishable, if there is a probabilistic polynomial time simulator S, such that for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\begin{aligned} & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow K(gk, sk) : \mathcal{A}(gk, \sigma) = 1\right] \\ \approx\ & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow S(gk, sk) : \mathcal{A}(gk, \sigma) = 1\right], \end{aligned}$$

and for all adversaries $\mathcal{A}$ we have

$$\begin{aligned} & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow S(gk, sk); (x, w_0, w_1) \leftarrow \mathcal{A}(gk, \sigma); \pi \leftarrow P(gk, \sigma, x, w_0) : \mathcal{A}(\pi) = 1\right] \\ =\ & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow S(gk, sk); (x, w_0, w_1) \leftarrow \mathcal{A}(gk, \sigma); \pi \leftarrow P(gk, \sigma, x, w_1) : \mathcal{A}(\pi) = 1\right], \end{aligned}$$

where we require $(gk, x, w_0), (gk, x, w_1) \in R$.

Composable zero-knowledge. A zero-knowledge proof, is a proof that shows the statement is true, but does not reveal anything else. Traditionally, this is defined by having a simulator (S_1, S_2) that can simulate the CRS and the proof, resepctively. The first part of the simulator outputs a simulated CRS and a simulation trapdoor τ, and the second part of the simulator uses the simulation trapdoor to simulate proofs for statements without knowing the corresponding witnesses. The standard definition of (multi-theorem) zero-knowledge then says that real proofs on a real CRS should be computationally indistinguishable from simulated proofs on a simulated CRS.

We will obtain a strong notion of zero-knowledge, called composable zero-knowledge [Gro06]. Composable zero-knowledge implies standard zero-knowledge [Gro06] and has the advantage that it is simpler to work with, since it separates the computational indistinguishability into two separate parts addressing the CRS and the proofs, respectively. In composable zero-knowledge, the real CRS and the simulated CRS are computationally indistinguishable. Moreover, the adversary, *even when it gets access to the secret simulation key τ*, cannot distinguish real proofs from simulated proofs on a simulated CRS.

Definition 5 (Composable zero-knowledge) *We say $(\mathcal{G}, K, P, V)$ is composable zero-knowledge if there exists a probabilistic polynomial time simulator (S_1, S_2) such that for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\begin{aligned} & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); \sigma \leftarrow K(gk, sk) : \mathcal{A}(gk, \sigma) = 1\right] \\ \approx\ & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); (\sigma, \tau) \leftarrow S_1(gk, sk) : \mathcal{A}(gk, \sigma) = 1\right], \end{aligned}$$

and for all interactive adversaries $\mathcal{A}$ we have

$$\begin{aligned} & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); (\sigma, \tau) \leftarrow S_1(gk, sk); (x, w) \leftarrow \mathcal{A}(gk, \sigma, \tau); \pi \leftarrow P(gk, \sigma, x, w) : \mathcal{A}(\pi) = 1\right] \\ = \; & \Pr\left[(gk, sk) \leftarrow \mathcal{G}(1^k); (\sigma, \tau) \leftarrow S_1(gk, sk); (x, w) \leftarrow \mathcal{A}(gk, \sigma, \tau); \pi \leftarrow S_2(gk, \sigma, \tau, x) : \mathcal{A}(\pi) = 1\right], \end{aligned}$$

where $\mathcal{A}$ outputs (x, w) so $(gk, x, w) \in R$.

3 Modules with bilinear maps

Let $(\mathcal{R}, +, \cdot, 0, 1)$ be a finite commutative ring. Recall that an $\mathcal{R}$-module A is an abelian group $(A, +, 0)$ where the ring acts on the group such that for all $r, s \in \mathcal{R}$ and all $x, y \in A$

$$(r+s)x = rx + sx \qquad r(x+y) = rx + ry \qquad r(sx) = (rs)x \qquad 1x = x.$$

A cyclic group G of order $\mathbf{n}$ can in a natural way be viewed as a $\mathbb{Z}_{\mathbf{n}}$-module. We observe that all the equations in Figure 1 can be viewed as equations over $\mathbb{Z}_{\mathbf{n}}$-modules with a bilinear map. To generalize completely, let $\mathcal{R}$ be a finite commutative ring and let A_1, A_2, A_T be finite $\mathcal{R}$-modules with a bilinear map $f : A_1 \times A_2 \to A_T$. We will consider quadratic equations over variables $x_1, \ldots, x_m \in A_1, y_1, \ldots, y_n \in A_2$ of the form

$$\sum_{j=1}^{n} f(a_j, y_j) + \sum_{i=1}^{m} f(x_i, b_i) + \sum_{i=1}^{m}\sum_{j=1}^{n} \gamma_{ij} f(x_i, y_j) = t.$$

In order to simplify notation, let us for $x_1, \ldots, x_n \in A_1, y_1, \ldots, y_n \in A_2$ define

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} f(x_i, y_i).$$

The equations can now be written as

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t,$$

where $\vec{a} \in A_1^n, \vec{b} \in A_2^m, \Gamma \in \mathrm{Mat}_{m \times n}(\mathcal{R})$. We note for future use that due to the bilinear properties of f, we have for any matrix $\Gamma \in \mathrm{Mat}_{m \times n}(\mathcal{R})$ and for any $\vec{x} \in A_1^m, \vec{y} \in A_2^n$ that $\vec{x} \cdot \Gamma \vec{y} = \Gamma^\top \vec{x} \cdot \vec{y}$.

Let us now return to the equations in Figure 1 and see how they can be recast as quadratic equations over $\mathbb{Z}_{\mathbf{n}}$-modules with a bilinear map.

Pairing product equations: Define $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = G_1, A_2 = G_2, A_T = G_T, f(x, y) = e(x, y)$ and rewrite[4] the pairing product equation as $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{X}} \cdot \Gamma \vec{\mathcal{Y}}) = t_T$.

Multi-scalar multiplication in G_1: Define $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = G_1, A_2 = \mathbb{Z}_{\mathbf{n}}, A_T = G_1, f(\mathcal{X}, y) = y\mathcal{X}$ and rewrite the multi-scalar multiplication equation as $\vec{\mathcal{A}} \cdot \vec{y} + \vec{\mathcal{X}} \cdot \vec{b} + \vec{\mathcal{X}} \cdot \Gamma \vec{y} = \mathcal{T}_1$.

Multi-scalar multiplication in G_2: Define $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = G_2, A_T = G_2, f(x, \mathcal{Y}) = x\mathcal{Y}$ and rewrite the multi-scalar multiplication equation as $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma \vec{\mathcal{Y}} = \mathcal{T}_2$.

Quadratic equation in $\mathbb{Z}_{\mathbf{n}}$: Define $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = \mathbb{Z}_{\mathbf{n}}, A_T = \mathbb{Z}_{\mathbf{n}}, f(x, y) = xy \bmod \mathbf{n}$ and rewrite the quadratic equation in $\mathbb{Z}_{\mathbf{n}}$ as $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} \equiv t \bmod \mathbf{n}$.

We will therefore first focus on the more general problem of constructing non-interactive composable witness-indistinguishable proofs for satisfiability of quadratic equations over $\mathcal{R}$-modules A_1, A_2, A_T (using additive notation for all modules) with a bilinear map $f : A_1 \times A_2 \to A_T$.

[4] We use multiplicative notation here, because, usually G_T is written multiplicatively in the literature. When we work with the abstract modules, however, we will use additive notation.

4 Commitment from modules

In our NIWI and NIZK proofs we will commit to the variables $x_1, \ldots, x_m \in A_1, y_1, \ldots, y_n \in A_2$. We do this by mapping them into other $\mathcal{R}$-modules B_1, B_2 and making the commitments in those modules.

Let us for now just consider how to commit to elements from one $\mathcal{R}$-module A. The public key for the commitment scheme will describe another $\mathcal{R}$-module B and $\mathcal{R}$-linear maps $\iota : A \to B$ and $p : B \to A$. Operations in the module and computation of the map ι will be efficiently computable but p is hard to compute.[5] The public key will also contain elements $u_1, \ldots, u_{\hat{m}} \in B$. To commit to $x \in A$ we pick $r_1, \ldots, r_{\hat{m}} \leftarrow \mathcal{R}$ at random and compute the commitment

$$c := \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i.$$

Our commitment scheme will have two types of commitment keys.

Binding key: A binding key defines $(B, \iota, p, u_1, \ldots, u_{\hat{m}})$ where $\forall i : p(u_i) = 0$ and $p \circ \iota$ is non-trivial. The commitment $c := \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$ therefore contains the non-trivial information $p(c) = p(\iota(x))$ about x. In particular, if $p \circ \iota$ is the identity map on A, then the commitment is perfectly binding to x.

Hiding key: A hiding key defines $(B, \iota, p, u_1, \ldots, u_{\hat{m}})$ where $\iota(A) \subseteq \langle u_1, \ldots, u_{\hat{m}} \rangle$. The commitment $c := \iota(x) + \sum_{i=1}^{\hat{m}} r_i u_i$ therefore perfectly hides the element x when $r_1, \ldots, r_{\hat{m}}$ are chosen at random from $\mathcal{R}$.

Computational indistinguishability: For security we need binding keys and hiding keys to be computationally indistinguishable. Witness-indistinguishability of our NIWI proofs and later the zero-knowledge property of our NIZK proofs will rely on this.

The treatment of commitments using the language of modules generalizes several previous works dealing with commitments over bilinear groups, including [BGN05, GOS06b, GOS06a, Gro06, Wat06].

Since we will often be committing to many elements at a time let us define some convenient notation. Given elements $x_1, \ldots, x_m \in A$ we will write $\vec{c} := \iota(\vec{x}) + R\vec{u}$ with $R \in \mathrm{Mat}_{m \times \hat{m}}(\mathcal{R})$ for making commitments $c_1, \ldots, c_m$ computed as $c_i := \iota(x_i) + \sum_{j=1}^{\hat{m}} r_{ij} u_j$.

5 Setup

In our NIWI and NIZK proofs the setup and the common reference string are

$$\begin{aligned} &gk \text{ defining } (\mathcal{R}, A_1, A_2, A_T, f), \\ &\sigma \text{ together with } gk \text{ defining } (B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}, H_1, \ldots, H_\eta). \end{aligned}$$

Part of the common reference string specifies $B_1, \iota_1, p_1, u_1, \ldots, u_{\hat{m}}$ and $B_2, \iota_2, p_2, v_1, \ldots, v_{\hat{n}}$ that are commitment keys for A_1 and A_2. We note that many of these components may be given implicitly instead of being described explicitly in the common reference string.

Another part of the common reference string specifies a third $\mathcal{R}$-module B_T together with $\mathcal{R}$-linear maps $\iota_T : A_T \to B_T$ and $p_T : B_T \to A_T$ and a bilinear map $F : B_1 \times B_2 \to B_T$. We require

$$
\begin{tikzcd}
A_1 \arrow[d, shift right, "\iota_1"'] & \times & A_2 \arrow[r, "f"] \arrow[d, shift right, "\iota_2"'] & A_T \arrow[d, shift right, "\iota_T"'] \\
B_1 \arrow[u, shift right, "p_1"'] & \times & B_2 \arrow[r, "F"] \arrow[u, shift right, "p_2"'] & B_T \arrow[u, shift right, "p_T"']
\end{tikzcd}
$$

$$
\begin{aligned}
\forall x \in A_1\ \forall y \in A_2 :&\ \ F(\iota_1(x), \iota_2(y)) = \iota_T(f(x, y)) \\
\forall x \in B_1\ \forall y \in B_2 :&\ \ f(p_1(x), p_2(y)) = p_T(F(x, y))
\end{aligned}
$$

Figure 2: Modules and maps between them.

that the maps are commutative as described in Figure 2 below, and with the exception of p_1, p_2 and p_T, that they are efficiently computable.
For notational convenience, we define for $\vec{x} \in B_1^n, \vec{y} \in B_2^n$ that

$$\vec{x} \bullet \vec{y} = \sum_{i=1}^{n} F(x_i, y_i).$$

Due to the bilinear properties of F we have for all vectors and matrices with appropriate dimensions

$$\vec{x} \bullet \Gamma\vec{y} = \Gamma^\top \vec{x} \bullet \vec{y}.$$

The final part of the common reference string is a set of matrices $H_1, \ldots, H_\eta \in \mathrm{Mat}_{\hat{m}\times\hat{n}}(\mathcal{R})$ that all satisfy $\vec{u} \bullet H_i\vec{v} = 0$. The exact number of matrices $H_1, \ldots, H_\eta$ that is needed, depends on the concrete setting. In many cases, we need no matrices at all and we have $\eta = 0$, but there are also cases where they are needed as we shall see in the instantiation in Section 10.

There will be two different settings of interest to us.

Soundness setting: In the soundness setting, we have binding commitment keys. This means $p_1(\vec{u}) = \vec{0}$ and $p_2(\vec{v}) = \vec{0}$, and the maps $p_1 \circ \iota_1$ and $p_2 \circ \iota_2$ are non-trivial. We will also want $p_T \circ \iota_T$ to be non-trivial.

Witness-indistinguishability setting: In the witness-indistinguishability setting we have hiding commitment keys, such that $\iota_1(A_1) \subseteq \langle u_1, \ldots, u_{\hat{m}}\rangle$ and $\iota_2(A_2) \subseteq \langle v_1, \ldots, v_{\hat{n}}\rangle$. We also require that $H_1, \ldots, H_\eta$ generate the R-module of all matrices $H \in \mathrm{Mat}_{\hat{m}\times\hat{n}}(\mathcal{R})$ such that $\vec{u} \bullet H\vec{v} = 0$. As we will see in the next section, these matrices play a role in the randomization of the NIWI proofs.

Computational indistinguishability: The (only) computational assumption this paper is based on is that the two settings can be set up in a computationally indistinguishable way. The instantiations in Sections 8, 9 and 10 show that there are many ways to get such computationally indistinguishable soundness and witness-indistinguishability setups.

[5] There are scenarios where a secret key will make p efficiently computable and $p \circ \iota$ is the identity map. In this case the commitment scheme is a public key encryption scheme with p being the decryption operation.

6 Proving that committed values satisfy a quadratic equation

Recall that in our setting, a quadratic equation looks like

$$\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t, \tag{1}$$

with constants $\vec{a} \in A_1^n, \vec{b} \in A_2^m, \Gamma \in \mathrm{Mat}_{m\times n}(\mathcal{R}), t \in A_T$. We will first consider the case of a single quadratic equation of the above form. The first step in our NIWI proof will be to commit to all the variables $\vec{x}, \vec{y}$. The commitments are of the form

$$\vec{c} = \iota_1(\vec{x}) + R\vec{u} \quad , \quad \vec{d} = \iota_2(\vec{y}) + S\vec{v}, \tag{2}$$

with $R \in \mathrm{Mat}_{m\times \hat{m}}(\mathcal{R}), S \in \mathrm{Mat}_{n\times \hat{n}}(\mathcal{R})$. The prover's task is to convince the verifier that the commitments contain $\vec{x} \in A_1^m, \vec{y} \in A_2^n$ that satisfy the quadratic equation. (Note that for all equations we will use these same commitments.)

Intuition. Before giving the construction let us give some intuition. In the previous sections, we have carefully set up our commitments such that the commitments themselves also "behave" like the values being committed to: they also belong to modules (the B modules) equipped with a bilinear map (the map F, implicitly used in the $\bullet$ operation). Given that we have done this, a natural idea is to take the quadratic equation (1), and "plug in" the commitments (2) in place of the variables; let us evaluate

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d}.$$

After some computations, where we expand the commitments (2), make use of the bilinearity of $\bullet$, and rearrange terms (the details can be found in the proof of Theorem 6) we get

$$\begin{aligned}&\Big(\iota_1(\vec{a}) \bullet \iota_2(\vec{y}) + \iota_1(\vec{x}) \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma\iota_2(\vec{y})\Big)\\ &+\iota_1(\vec{a}) \bullet S\vec{v} + R\vec{u} \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma S\vec{v} + R\vec{u} \bullet \Gamma\iota_2(\vec{y}) + R\vec{u} \bullet \Gamma S\vec{v}.\end{aligned}$$

By the commutative properties of the maps, the first group of three terms is equal to $\iota_T(t)$, if (1) holds. Looking at the remaining terms, note that $\vec{u}$ and $\vec{v}$ are part of the common reference string and therefore known to the verifier. Using the fact that bilinearity implies that for any $\vec{x}, \vec{y}$ we have $\vec{x} \bullet \Gamma\vec{y} = \Gamma^\top \vec{x} \bullet \vec{y}$, we can sort the remaining terms so they match either $\vec{u}$ or $\vec{v}$ to get (again see the proof of Theorem 6 for details)

$$\iota_T(t) + \vec{u} \bullet \Big(R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v}\Big) + \Big(S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x})\Big) \bullet \vec{v}. \tag{3}$$

Now, for the sake of intuition, let us make some simplifying assumptions. Let us assume that we are working in a symmetric case where $A_1 = A_2$, $B_1 = B_2$, $\vec{u} = \vec{v}$, and F is symmetric, and so, the above equation can be simplified further to get

$$\iota_T(t) + \vec{u} \bullet \Big(R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{u} + S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x})\Big).$$

Now, suppose the prover gives to the verifier as his proof $\vec{\pi} = \Big(R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{u} + S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x})\Big)$. The verifier would then check that the following *verification equation* holds:

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi}.$$

Suppose further $p_1 \circ \iota_1, p_2 \circ \iota_2, p_T \circ \iota_T$ are the identity maps on A_1, A_2, A_T. It is easy to see that the proof is convincing in the soundness setting, because in that setting we have that $p_1(\vec{u}) = \vec{0}$. Then the verifier would know (but not be able to compute) that by applying the maps p_1, p_2, p_T we get

$$\vec{a} \bullet p_2(\vec{d}) + p_1(\vec{c}) \bullet \vec{b} + p_1(\vec{c}) \bullet \Gamma p_2(\vec{d}) = t + p_1(\vec{u}) \bullet p_2(\vec{\pi}) = t.$$

This gives us soundness, since $\vec{x} := p_1(\vec{c})$ and $\vec{y} := p_2(\vec{d})$ satisfy the equations.

The remaining problem is to get witness-indistinguishability. Recall that in the witness-indistinguishability setting, the commitments are perfectly hiding. Therefore, in the verification equation, nothing except for $\vec{\pi}$ holds any information about $\vec{x}$ and $\vec{y}$ (except for the information that can be inferred from the quadratic equation itself). So, let's consider two cases:

1. Suppose that $\vec{\pi}$ is the unique value such that the verification equation is valid. In this case, we trivially have witness indistinguishability, since the uniqueness means that any witness would lead to the same value for $\vec{\pi}$.

2. The simple case above might seem too good to be true, but let us see what it means if it is not true. If two proofs $\vec{\pi}$ and $\vec{\pi}'$ both satisfy the verification equation, then subtracting the equations shows that $\vec{u} \bullet (\vec{\pi} - \vec{\pi}') = 0$. On the other hand, recall that in the witness indistinguishability setting, the $\vec{u}$ vectors generate the entire space where $\vec{\pi}$ and $\vec{\pi}'$ live, and furthermore we know that the matrices $H_1, \dots, H_\eta$ generate all H such that $\vec{u} \bullet H\vec{u} = 0$. Therefore, let us choose $r_1, \dots, r_\eta$ at random, and consider the distribution $\vec{\pi}'' = \vec{\pi} + \sum_{i=1}^{\eta} r_i H_i \vec{u}$. We obtain the same distribution on $\vec{\pi}''$ that satisfies the verification equation regardless of whether we started from $\vec{\pi}$ or $\vec{\pi}'$ or any other proof.

Thus, for the symmetric case we obtain a witness indistinguishable proof system. For the general non-symmetric case, instead of having just $\vec{\pi}$ for the $\vec{u}$ part of (3), we would also have a proof $\vec{\theta}$ for the $\vec{v}$ part. In this case, we would also have to make sure that this split does not reveal any information about the witness. What we will do is to randomize the proofs such that they get a uniform distribution on all $\vec{\pi}, \vec{\theta}$ that satisfy the verification equation. If we pick $T \leftarrow \mathrm{Mat}_{\hat{n} \times \hat{m}}(\mathcal{R})$ at random we have that $\vec{\theta} + T\vec{u}$ completely randomizes $\vec{\theta}$. The part we add in $\vec{\theta}$ can be "subtracted" from $\vec{\pi}$ by observing that

$$\iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v} = \iota_T(t) + \vec{u} \bullet \left(\vec{\pi} - T^\top \vec{v}\right) + \left(\vec{\theta} + T\vec{u}\right) \bullet \vec{v}.$$

By randomizing $\vec{\pi}$ this leads to a uniform distribution of proofs for the general non-symmetric case as well.

6.1 The general case

Having explained the intuition behind the proof system, we proceed to a formal description of how the prover handles a single equation and the security properties the procedure has.

Prover: Pick $T \leftarrow \mathrm{Mat}_{\hat{n} \times \hat{m}}(\mathcal{R}), r_1, \dots, r_\eta \leftarrow \mathcal{R}$ at random. Compute

$$\begin{aligned} \vec{\pi} &:= R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v} - T^\top \vec{v} + \sum_{i=1}^{\eta} r_i H_i \vec{v} \\ \vec{\theta} &:= S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x}) + T\vec{u} \end{aligned}$$

and return the proof $(\vec{\theta}, \vec{\pi})$.

Verifier: Return 1 if and only if

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}.$$

Perfect completeness of our NIWI proof will follow from the following theorem regardless of whether we are in the soundness setting or the witness-indistinguishability setting.

Theorem 6 *Given $\vec{x} \in A_1^m, \vec{y} \in A_2^n, R \in \mathrm{Mat}_{m\times\hat{m}}(\mathcal{R}), S \in \mathrm{Mat}_{n\times\hat{n}}(\mathcal{R})$ satisfying*

$$\vec{c} = \iota_1(\vec{x}) + R\vec{u} \qquad \vec{d} = \iota_2(\vec{y}) + S\vec{v} \qquad \vec{a}\cdot\vec{y} + \vec{x}\cdot\vec{b} + \vec{x}\cdot\Gamma\vec{y} = t,$$

we have for all choices of $T, r_1, \ldots, r_\eta$ that the proofs $\vec{\pi}, \vec{\theta}$ constructed as above will be accepted.

Proof. The commutative property of the linear and bilinear maps gives us $\iota_1(\vec{a}) \bullet \iota_2(\vec{y}) + \iota_1(\vec{x}) \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma\iota_2(\vec{y}) = \iota_T(t)$. For any choice of $T, r_1, \ldots, r_\eta$ we have

$$\begin{aligned}
& \iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} \\
= \; & \iota_1(\vec{a}) \bullet \Big(\iota_2(\vec{y}) + S\vec{v}\Big) + \Big(\iota_1(\vec{x}) + R\vec{u}\Big) \bullet \iota_2(\vec{b}) + \Big(\iota_1(\vec{x}) + R\vec{u}\Big) \bullet \Gamma\Big(\iota_2(\vec{y}) + S\vec{v}\Big) \\
= \; & \iota_1(\vec{a}) \bullet \iota_2(\vec{y}) + \iota_1(\vec{x}) \bullet \iota_2(\vec{b}) + \iota_1(\vec{x}) \bullet \Gamma\iota_2(\vec{y}) \\
& \qquad + R\vec{u} \bullet \iota_2(\vec{b}) + R\vec{u} \bullet \Gamma\iota_2(\vec{y}) + R\vec{u} \bullet \Gamma S\vec{v} + \iota_1(\vec{a}) \bullet S\vec{v} + \iota_1(\vec{x}) \bullet \Gamma S\vec{v} \\
= \; & \iota_T(t) + \vec{u} \bullet \Big(R^\top\iota_2(\vec{b}) + R^\top\Gamma\iota_2(\vec{y}) + R^\top\Gamma S\vec{v}\Big) + \Big(S^\top\iota_1(\vec{a}) + S^\top\Gamma^\top\iota_1(\vec{x})\Big) \bullet \vec{v} \\
= \; & \iota_T(t) + \vec{u} \bullet \Big(R^\top\iota_2(\vec{b}) + R^\top\Gamma\iota_2(\vec{y}) + R^\top\Gamma S\vec{v}\Big) + \sum_{i=1}^{\eta} r_i(\vec{u} \bullet H_i\vec{v}) - \vec{u} \bullet T^\top\vec{v} \\
& \qquad + T\vec{u} \bullet \vec{v} + \Big(S^\top\iota_1(\vec{a}) + S^\top\Gamma^\top\iota_1(\vec{x})\Big) \bullet \vec{v} \\
= \; & \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v} \qquad \square
\end{aligned}$$

Theorem 7 *In the soundness setting, where we have $p_1(\vec{u}) = \vec{0}$ and $p_2(\vec{v}) = \vec{0}$, a valid proof implies*

$$p_1(\iota_1(\vec{a})) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) + p_1(\vec{c}) \cdot \Gamma p_2(\vec{d}) = p_T(\iota_T(t)).$$

Proof. An acceptable proof $\vec{\pi}, \vec{\theta}$ satisfies $\iota(a) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}$. The commutative property of the linear and bilinear maps gives us

$$\begin{aligned}
& p_1(\iota_1(\vec{a})) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) + p_1(\vec{c}) \cdot \Gamma p_2(\vec{d}) \\
= \; & p_T(\iota_T(t)) + p_1(\vec{u}) \cdot p_2(\vec{\pi}) + p_1(\vec{\theta}) \cdot p_2(\vec{v}) = p_T(\iota_T(t)) \qquad \square
\end{aligned}$$

Observe as a particularly interesting case that when $p_1 \circ \iota_1, p_2 \circ \iota_2, p_T \circ \iota_T$ are the identity maps on A_1, A_2 and A_T, respectively, this means $\vec{x} := p_1(\vec{c})$ and $\vec{y} := p_2(\vec{d})$ give us a satisfying solution to the equation $\vec{a}\cdot\vec{y} + \vec{x}\cdot\vec{b} + \vec{x}\cdot\Gamma\vec{y} = t$. In this case, the theorem says that the proof is perfectly sound in the soundness setting. In the case where they are not the identity maps it is still possible to have a form of culpable soundness, see the instantiation in Section 8 for an example based on composite order bilinear groups.

Theorem 8 *In the witness-indistinguishable setting where $\iota_1(A_1) \subseteq \langle u_1, \ldots, u_{\hat{m}}\rangle$, $\iota_2(A_2) \subseteq \langle v_1, \ldots, v_{\hat{n}}\rangle$ and $H_1, \ldots, H_\eta$ generate all matrices H such that $\vec{u} \bullet H\vec{v} = 0$, all satisfying witnesses $\vec{x}, \vec{y}, R, S$ yield proofs $\vec{\pi} \in \langle v_1, \ldots, v_{\hat{n}}\rangle^{\hat{m}}$ and $\vec{\theta} \in \langle u_1, \ldots, u_{\hat{m}}\rangle^{\hat{n}}$ that are uniformly distributed conditioned on the verification equation $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}$.*

Proof. Since $\iota_1(A_1) \subseteq \langle u_1, \ldots, u_{\hat{m}} \rangle$ and $\iota_2(A_2) \subseteq \langle v_1, \ldots, v_{\hat{n}} \rangle$ there exists A, B, X, Y such that $\iota_1(\vec{a}) = A\vec{u}$, $\iota_1(\vec{x}) = X\vec{u}$ and $\iota_2(\vec{b}) = B\vec{v}, \iota_2(\vec{y}) = Y\vec{v}$. We have $\vec{c} = (X+R)\vec{u}$ and $\vec{d} = (Y+S)\vec{v}$. The proof is $(\vec{\pi}, \vec{\theta})$ given by

$$\vec{\theta} = S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x}) + T\vec{u} = \Big(S^\top A + S^\top \Gamma^\top X + T\Big)\vec{u}$$

$$\vec{\pi} = R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v})) - T^\top \vec{v} + \sum_{i=1}^{\eta} r_i H_i \vec{v}$$

$$= \Big(R^\top B + R^\top \Gamma Y + R^\top \Gamma S - T^\top\Big)\vec{v} + \Big(\sum_{i=1}^{\eta} r_i H_i\Big)\vec{v}.$$

We choose T at random, so we can think of $\vec{\theta}$ being a uniformly random variable given by $\vec{\theta} = \Theta\vec{v}$ for a randomly chosen matrix Θ. We can think of $\vec{\pi}$ as being written $\vec{\pi} = \Pi\vec{v}$, where Π is a random variable that depends on Θ.

By perfect completeness all satisfying witnesses yield proofs where $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} - \iota_T(t) - \vec{\theta} \bullet \vec{v} = \vec{u} \bullet \vec{\pi} = \vec{u} \bullet \Pi\vec{v}$. Conditioned on the random variable Θ we therefore have that any two possible solutions $\vec{\pi}, \vec{\pi}'$ satisfy $\vec{u} \bullet (\Pi - \Pi')\vec{v} = 0$. Since $H_1, \ldots, H_\eta$ generate all matrices H such that $\vec{u} \bullet H\vec{v} = 0$ we can write this as $\Pi = \Pi' + \sum_{i=1}^{\eta} r_i' H_i$. In constructing $\vec{\pi}$ we form it as $\Big(R^\top B + R^\top \Gamma Y + R^\top \Gamma S - T^\top\Big)\vec{v} + \Big(\sum_{i=1}^{\eta} r_i H_i\Big)\vec{v}$ for randomly chosen $r_1, \ldots, r_\eta \in \mathcal{R}$. We therefore get a uniform distribution over all $\vec{\pi}$ that satisfy the equation conditioned on $\vec{\theta}$. Since $\vec{\theta}$ is uniformly chosen, we conclude that for any witness we get a uniform distribution over $(\vec{\theta}, \vec{\pi})$ conditioned on it being an acceptable proof. □

6.2 Linear equations

As a special case, we will consider the proof system when $\vec{a} = 0$ and $\Gamma = 0$. In this case the equation is simply

$$\vec{x} \cdot \vec{b} = t.$$

The scheme can be simplified in this case by choosing $T = 0$ in the proof, which gives $\vec{\theta} := \vec{0}$ and $\vec{\pi} := R^\top \iota_2(\vec{b}) + \sum_{i=1}^{\eta} r_i H_i \vec{v}$. Theorem 6 still applies with $T = 0$, which will give us completeness. Theorem 7 says $p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) = p_T(\iota_T(t))$, which will give us soundness. Finally, we have the following theorem.

Theorem 9 *In the witness-indistinguishable setting where $\iota_1(A_1) \subseteq \langle u_1, \ldots, u_{\hat{m}} \rangle$, $\iota_2(A_2) \subseteq \langle v_1, \ldots, v_{\hat{n}} \rangle$ and $H_1, \ldots, H_\eta$ generate all matrices H such that $\vec{u} \bullet H\vec{v} = 0$, all satisfying witnesses $\vec{x}, \vec{y}, R, S$ yield the uniform distribution of the proof $\vec{\pi} \in \langle v_1, \ldots, v_{\hat{n}} \rangle^{\hat{m}}$ conditioned on the verification equation $\vec{c} \bullet \iota_2(\vec{b}) = \iota_T(t) + \vec{u} \bullet \vec{\pi}$ being satisfied.*

Proof. As in the proof of Theorem 8 we can write $\vec{\pi} = \Pi\vec{v}$. Any witness gives a proof that satisfies

$$\vec{c} \bullet \iota_1(\vec{b}) - \iota_T(t) = \vec{u} \bullet \vec{\pi} = \vec{u} \bullet \Pi\vec{v}.$$

Since $H_1, \ldots, H_\eta$ generate all matrices H such that $\vec{u} \bullet H\vec{v} = 0$ we have that Π has a uniform distribution over all matrices Π satisfying the verification equation. □

6.3 The symmetric case

An interesting special case is when $B := B_1 = B_2$, $\hat{m} \geq \hat{n}$ with $u_1 = v_1, \ldots, u_{\hat{n}} = v_{\hat{n}}$, and F is symmetric, i.e., for all $x, y \in B$ we have $F(x, y) = F(y, x)$. We call this the symmetric case. In the symmetric case, we can simplify the scheme by just padding $\vec{\theta}$ with zeroes in the end to extend the length to $\hat{m}$, call this vector $\vec{\theta}'$, and reveal the proof $\vec{\phi} = \vec{\pi} + \vec{\theta}'$. In the verification, we check that

$$\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\phi}.$$

Theorem 6 gives us completeness, and Theorem 8 implies that all witnesses yield the same distrubutions on the proofs $\vec{\pi}, \vec{\theta}$ and therefore the same distributions on the proofs $\vec{\phi}$. With respect to soundness we have the following theorem.

Theorem 10 *In the soundness setting, where we have $p_1(\vec{u}) = \vec{0}$ a valid proof implies*

$$p_1(\iota_1(a)) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) + p_1(\vec{c}) \cdot \Gamma p_2(\vec{d}) = p_T(\iota_T(t)).$$

Proof. An acceptable proof $\vec{\phi}$ satisfies $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\phi}$. The commutative property of the linear and bilinear maps gives us

$$p_1(\iota_1(\vec{a})) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) + p_1(\vec{c}) \cdot \Gamma p_2(\vec{d}) = p_T(\iota_T(t)) + p_1(\vec{u}) \cdot p_2(\vec{\phi}) = p_T(\iota_T(t)). \qquad \square$$

We can simplify the computation of the proof in the symmetric case. We have

$$\begin{aligned} \vec{\pi} &:= R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v} - T^\top \vec{v} + \sum_{i=1}^{\eta} r_i H_i \vec{v} \\ \vec{\theta} &:= S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x}) + T \vec{u}, \end{aligned}$$

and extend θ to θ' by padding it with $\hat{m} - \hat{n}$ 0's. Another way to accomplish this padding is by padding T with $\hat{m} - \hat{n}$ 0-rows and S with $\hat{m} - \hat{n}$ 0-columns and each H_i with $\hat{m} - \hat{n}$ 0-columns. We then have

$$\vec{\phi} := R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S' \vec{u} - (T')^\top \vec{u} + \sum_{i=1}^{\eta} r_i H_i' \vec{u} + (S')^\top \iota_1(\vec{a}) + (S')^\top \Gamma^\top \iota_1(\vec{x}) + T' \vec{u}.$$

Since the map is symmetric we have $\vec{u} \bullet (T' - (T')^\top) \vec{u} = 0$. If we have a set $H_1', \ldots, H_{\eta'}'$ that generates all matrices H' such that $\vec{u} \bullet H' \vec{u} = 0$, then we have $T' - (T')^\top$ is in the span of them. This means the following simpler proof is also witness-indistinguishable

$$\vec{\phi} := R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + (S')^\top \iota_1(\vec{a}) + (S')^\top \Gamma^\top \iota_1(\vec{x}) + R^\top \Gamma S' \vec{u} + \sum_{i=1}^{\eta'} r_i H_i' \vec{u}.$$

7 NIWI proof for satisfiability of a set of quadratic equations

We will now give the full composable NIWI proof for satisfiability of a set of quadratic equations in a module with a bilinear map, i.e., the language

$$L = \left\{ \{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N \;\middle|\; \exists \vec{x}, \vec{y}\; \forall i : \vec{a}_i \cdot \vec{y} + \vec{x} \cdot \vec{b}_i + \vec{x} \cdot \Gamma_i \vec{y} = t_i \right\}.$$

The proof will have L_{guilt}-soundness for

$$L_{\text{guilt}} = \left\{ \{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N \;\middle|\; \forall \vec{x}, \vec{y}\; \exists i : p_1(\iota_1(\vec{a}_i)) \cdot \vec{y} + \vec{x} \cdot p_2(\iota_2(\vec{b}_i)) + \vec{x} \cdot \Gamma_i \vec{y} \neq p_T(\iota_T(t_i)) \right\}.$$

Observe as an important special case that if $p_1 \circ \iota_1, p_2 \circ \iota_2, p_T \circ \iota_T$ are the identity maps on A_1, A_2 and A_T, then $L_{\text{guilt}} = \bar{L}$ making soundness and L_{guilt}-soundness the same notion.

The cryptographic assumption we make is that the common reference string is created by one of two algorithm K or S and that their outputs are computationally indistinguishable. The first algorithm outputs a common reference string that specifies a soundness setting, whereas the second algorithm outputs a common reference string that specifies a witness-indistinguishability setting.

Setup: $(gk, sk) = ((\mathcal{R}, A_1, A_2, A_T, f), sk) \leftarrow \mathcal{G}(1^k)$.

CRS generators: The common reference string defines $(B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}, H_1, \ldots, H_\eta)$. It can be generated as a soundness string $\sigma \leftarrow K(gk, sk)$ or as a witness-indistinguishability string $\sigma \leftarrow S(gk, sk)$.

Prover: The input consists of gk, σ, a list of quadratic equations $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and a satisfying witness $\vec{x} \in A_1^m, \vec{y} \in A_2^n$.

Pick at random $R \leftarrow \text{Mat}_{m\times\hat{m}}(\mathcal{R})$ and $S \leftarrow \text{Mat}_{n\times\hat{n}}(\mathcal{R})$ and commit to all the variables as $\vec{c} := \vec{x} + R\vec{u}$ and $\vec{d} := \vec{y} + S\vec{v}$.

For each equation $(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)$ make a proof as described in Section 6. In other words, pick $T_i \leftarrow \text{Mat}_{\hat{n}\times\hat{m}}(\mathcal{R})$ and $r_{i1}, \ldots, r_{i\eta} \leftarrow \mathcal{R}$ and compute

$$\begin{aligned} \vec{\pi}_i &:= R^\top \iota_2(\vec{b}_i) + R^\top \Gamma_i \iota_2(\vec{y}) + R^\top \Gamma_i S \vec{v} - T_i^\top \vec{v} + \sum_{j=1}^{\eta} r_{ij} H_j \vec{v} \\ \vec{\theta}_i &:= S^\top \iota_1(\vec{a}_i) + S^\top \Gamma_i^\top \iota_1(\vec{x}) + T_i \vec{u}. \end{aligned}$$

Output the proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N)$.

Verifier: The input is gk, σ, $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and the proof is $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N)$.

For each equation check

$$\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}_i) + \vec{c} \bullet \Gamma_i \vec{d} = \iota_T(t_i) + \vec{u} \bullet \vec{\pi}_i + \vec{\theta}_i \bullet \vec{v}.$$

Output 1 if all the checks pass, else output 0.

Theorem 11 *The proof system $(\mathcal{G}, K, P, V)$ given above is a NIWI proof for satisfiability of a set of quadratic equations with perfect completeness, perfect L_{guilt}-soundness and composable witness-indistinguishability.*

Proof. Perfect completeness follows from Theorem 6.

Consider a proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\theta}_i)\})$ on a soundness string. Define $\vec{x} := p_1(\vec{c}), \vec{y} := p_2(\vec{d})$. It follows from Theorem 7 that for each equation we have

$$\begin{aligned} & p_1(\iota_1(\vec{a}_i)) \cdot \vec{y} + \vec{x} \cdot p_2(\iota_2(\vec{b}_i)) + \vec{x} \cdot \Gamma_i \vec{y} \\ = \; & p_1(\iota_1(\vec{a}_i)) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b}_i)) + p_1(\vec{c}) \cdot \Gamma_i p_2(\vec{d}) = p_T(\iota_T(t_i)). \end{aligned}$$

This means we have perfect L_{guilt}-soundness.

We have assumed that soundness strings and witness-indistinguishability strings are computationally indistinguishable. Consider now a witness-indistinguishability string σ. The commitments are perfectly hiding, so they do not reveal the witness $\vec{x}, \vec{y}$ that the prover uses in the commitments $\vec{c}, \vec{d}$. Theorem 8 says that in each equation either of two possible witnesses yields the same distribution on the proof for that equation. A straightforward hybrid argument then shows that we have perfect witness-indistinguishability. □

Proof of knowledge. We observe that if K outputs an additional secret piece of information ξ that makes it possible to efficiently compute p_1 and p_2, then ξ makes it possible to extract the witness $\vec{x} = p_1(\vec{c})$ and $\vec{y} = p_2(\vec{d})$.

Proof size. The size of the common reference string is $\hat{m}$ elements in B_1 and $\hat{n}$ elements in B_2 in addition to the description of the modules, the maps and $H_1, \dots, H_\eta$. The size of the proof is $m + N\hat{n}$ elements in B_1 and $n + N\hat{m}$ elements in B_2.

Typically, $\hat{m}$ and $\hat{n}$ will be small, giving us a proof size that is $O(m + n + N)$ elements in B_1 and B_2. The proof size may thus be smaller than the description of the statement, which can be of size up to Nn elements in A_1, Nm elements in A_2, Nmn elements in $\mathcal{R}$ and N elements in A_T.

7.1 NIWI proofs for bilinear groups

We will now outline the strategy for making NIWI proofs for satisfiability of a set of quadratic equations over bilinear groups. As we described in Section 3, there are four different types of equations corresponding to the following four combinations of $\mathbb{Z}_{\mathbf{n}}$-modules:

Pairing product equations: $A_1 = G_1, A_2 = G_2, A_T = G_T, f(\mathcal{X}, \mathcal{Y}) = e(\mathcal{X}, \mathcal{Y})$.

Multi-scalar multiplication in G_1: $A_1 = G_1, A_2 = \mathbb{Z}_{\mathbf{n}}, A_T = G_1, f(\mathcal{X}, y) = y\mathcal{X}$.

Multi-scalar multiplication in G_2: $A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = G_2, A_T = G_2, f(x, \mathcal{Y}) = x\mathcal{Y}$.

Quadratic equations in $\mathbb{Z}_{\mathbf{n}}$: $A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = \mathbb{Z}_{\mathbf{n}}, A_T = \mathbb{Z}_{\mathbf{n}}, f(x, y) = xy \bmod \mathbf{n}$.

The common reference string will specify commitment schemes to respectively scalars and group elements. We first commit to all the variables and then make the NIWI proofs that correspond to the types of equations that we are looking at. It is important that we use the same commitment schemes and commitments for all equations, i.e., for instance we only commit to a scalar x once and we use the same commitment in the proof whether x is involved in is a multi-scalar multiplication in G_2 or a quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. The use of the same commitment in all the equations is necessary to ensure a consistent choice of x throughout the proof. As a consequence of this we use the same module B_1' to commit to x in both multi-scalar multiplication in G_2 and quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. We therefore end up with at most four different modules B_1, B_1', B_2, B_2' to commit to respectively $\mathcal{X}, x, \mathcal{Y}, y$ variables.

8 Instantiation based on the subgroup decision assumption

Setup. The first instantiation is based on the composite order groups introduced by Boneh, Goh and Nissim [BGN05]. The setup algorithm $\mathcal{G}_{\mathrm{BGN}}$ outputs (gk, sk) where $gk = (\mathbf{n}, G, G_T, e, \mathcal{P})$ describes a bilinear group of composite order $\mathbf{n}$ and $sk = (\mathbf{p}, \mathbf{q})$ consists of two primes such that $\mathbf{n} = \mathbf{pq}$. Boneh, Goh and Nissim also introduced the subgroup decision assumption, which says that it is hard to distinguish a random element of order $\mathbf{q}$ from a random element of order $\mathbf{n}$.

Definition 12 (Subgroup decision assumption) *We say the* subgroup decision assumption *holds for $\mathcal{G}_{\mathrm{BGN}}$ if for all non-uniform polynomial time $\mathcal{A}$:*

$$\begin{aligned} & \Pr[(gk, sk) \leftarrow \mathcal{G}_{\mathrm{BGN}}(1^k); \alpha \leftarrow \mathbb{Z}_{\mathbf{n}}^*; \mathcal{U} := \alpha\mathbf{p}\mathcal{P} : \mathcal{A}(gk, \mathcal{U}) = 1] \\ \approx\ & \Pr[(gk, sk) \leftarrow \mathcal{G}_{\mathrm{BGN}}(1^k); \alpha \leftarrow \mathbb{Z}_{\mathbf{n}}^*; \mathcal{U} := \alpha\mathcal{P} : \mathcal{A}(gk, \mathcal{U}) = 1]. \end{aligned}$$

Statements. Based on the subgroup decision assumption we will construct NIWI proofs for the language consisting of pairing product equations, multi-scalar multiplication equations and quadratic equations as described in Figure 1. A statement consists of N_{P} pairing product equations of the form $\prod_i e(\mathcal{A}_i, \mathcal{Y}_i) \cdot \prod_{i,j} e(\mathcal{Y}_i, \mathcal{Y}_j)^{\gamma_{ij}} = t_T$, N_{M} multi-scalar multiplication equations of the form $\sum_i a_i \mathcal{Y}_i + \sum_i x_i \mathcal{B}_i + \sum_{i,j} \gamma_{ij} x_i \mathcal{Y}_j = \mathcal{T}$ and N_{Q} quadratic equations of the form $\sum_i a_i x_i + \sum_{i,j} \gamma_{ij} x_i x_j \equiv t \bmod \mathbf{n}$, and a claim that there are $x_1, \ldots, x_m \in \mathbb{Z}_{\mathbf{n}}$ and $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G$ that satisfy all equations.

Formally, given a setup $gk = (\mathbf{n}, G, G_T, e, \mathcal{P})$ we define the language:

$$
\begin{aligned}
L \;=\; & \Big\{ \Big(\{(\vec{\mathcal{A}}_i, \Gamma_i^{\mathrm{P}}, t_{Ti})\}_{i=1}^{N_{\mathrm{P}}}, \{(\vec{a}_i, \vec{\mathcal{B}}_i, \Gamma_i^{\mathrm{M}}, \mathcal{T}_i)\}_{i=1}^{N_{\mathrm{M}}}, \{(\vec{b}_i, \Gamma_i^{\mathrm{Q}}, t_i)\}_{i=1}^{N_{\mathrm{Q}}} \Big) \;\Big|\; \exists m, n \in \mathbb{N} \exists \vec{x} \in \mathbb{Z}_{\mathbf{n}}^m \exists \vec{\mathcal{Y}} \in G^n : \\
& \forall i \in [N_{\mathrm{P}}] : \vec{\mathcal{A}}_i \in G^n \;\wedge\; \Gamma_i^{\mathrm{P}} \in \mathrm{Mat}_{n \times n}(\mathbb{Z}_{\mathbf{n}}) \;\wedge t_{Ti} \in G_T \;\wedge\; (\vec{\mathcal{A}}_i \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma_i^{\mathrm{P}} \vec{\mathcal{Y}}) = t_{Ti} \\
\wedge \quad & \forall i \in [N_{\mathrm{M}}] : \vec{a}_i \in \mathbb{Z}_{\mathbf{n}}^m \wedge \vec{\mathcal{B}}_i \in G^n \wedge \Gamma_i^{\mathrm{M}} \in \mathrm{Mat}_{m \times n}(\mathbb{Z}_{\mathbf{n}}) \wedge \mathcal{T}_i \in G \wedge \vec{a}_i \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}}_i + \vec{x} \cdot \Gamma_i^{\mathrm{M}} \vec{\mathcal{Y}} = \mathcal{T}_i \\
\wedge \quad & \forall i \in [N_{\mathrm{Q}}] : \vec{b}_i \in \mathbb{Z}_{\mathbf{n}}^m \;\wedge\; \Gamma_i^{\mathrm{Q}} \in \mathrm{Mat}_{m \times m}(\mathbb{Z}_{\mathbf{n}}) \;\wedge\; t_i \in \mathbb{Z}_{\mathbf{n}} \wedge\; \vec{x} \cdot \vec{b}_i + \vec{x} \cdot \Gamma_i^{\mathrm{P}} \vec{x} \equiv t_i \bmod \mathbf{n} \Big\}.
\end{aligned}
$$

Soundness will hold in the order $\mathbf{p}$ subgroups of G, G_T and $\mathbb{Z}_{\mathbf{n}}$. More precisely, define $\lambda \in \mathbb{Z}_{\mathbf{n}}$ as an integer satisfying $\lambda \equiv 1 \bmod \mathbf{p}$ and $\lambda \equiv 0 \bmod \mathbf{q}$. Then $\lambda \mathbb{Z}_{\mathbf{n}}$ is the order $\mathbf{p}$ subgroup of $\mathbb{Z}_{\mathbf{n}}$ and λG is the order $\mathbf{p}$ subgroup of G. We will get L_{guilt}-soundness for

$$
\begin{aligned}
L_{\mathrm{guilt}} \;=\; & \Big\{ \Big(\{(\vec{\mathcal{A}}_i, \Gamma_i^{\mathrm{P}}, t_{Ti})\}_{i=1}^{N_{\mathrm{P}}}, \{(\vec{a}_i, \vec{\mathcal{B}}_i, \Gamma_i^{\mathrm{M}}, \mathcal{T}_i)\}_{i=1}^{N^{\mathrm{M}}}, \{(\vec{b}_i, \Gamma_i^{\mathrm{Q}}, t_i)\}_{i=1}^{N_{\mathrm{P}}} \Big) \;\Big|\; \forall m, n \in \mathbb{N} \forall \vec{x} \in (\lambda \mathbb{Z}_{\mathbf{n}})^m \forall \vec{\mathcal{Y}} \in (\lambda G)^n : \\
& \exists i \in [N_{\mathrm{P}}] : \vec{\mathcal{A}}_i \notin G^n \;\vee\; \Gamma_i^{\mathrm{P}} \notin \mathrm{Mat}_{n \times n}(\mathbb{Z}_{\mathbf{n}}) \;\vee t_{Ti} \notin G_T \;\vee\; (\vec{\mathcal{A}}_i \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma_i^{\mathrm{P}} \vec{\mathcal{Y}}) \neq t_{Ti}^{\lambda} \\
\vee \quad & \exists i \in [N_{\mathrm{M}}] : \vec{a}_i \notin \mathbb{Z}_{\mathbf{n}}^m \vee \vec{\mathcal{B}}_i \notin G^n \vee \Gamma_i^{\mathrm{M}} \notin \mathrm{Mat}_{m \times n}(\mathbb{Z}_{\mathbf{n}}) \vee \mathcal{T}_i \notin G \vee \vec{a}_i \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}}_i + \vec{x} \cdot \Gamma_i^{\mathrm{M}} \vec{\mathcal{Y}} \neq \lambda \mathcal{T}_i \\
\vee \quad & \exists i \in [N_{\mathrm{Q}}] : \vec{b}_i \notin \mathbb{Z}_{\mathbf{n}}^m \;\vee\; \Gamma_i^{\mathrm{Q}} \notin \mathrm{Mat}_{m \times m}(\mathbb{Z}_{\mathbf{n}}) \;\vee\; t_i \notin \mathbb{Z}_{\mathbf{n}} \vee\; \vec{x} \cdot \vec{b}_i + \vec{x} \cdot \Gamma_i^{\mathrm{Q}} \vec{x} \not\equiv t_i \bmod \mathbf{p} \Big\}.
\end{aligned}
$$

Multi-scalar multiplication equations. We will build our full NIWI proof from a combination of NIWI proofs for pairing-product equations, multi-scalar multiplication equations and quadratic equations. First consider the case where we only have multi-scalar multiplication equations. Define L^{M} ($L^{\mathrm{M}}_{\mathrm{guilt}}$) to be L (L_{guilt}) restricted to $N_{\mathrm{P}} = N_{\mathrm{Q}} = 0$ such that it only has N_{M} multi-scalar multiplication equations.

We can use our framework to get NIWI proofs for L^{M}. The multi-scalar multiplication case corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = G, A_T = G, f(x, \mathcal{Y}) = x\mathcal{Y}$ and equations of the form $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma \vec{\mathcal{Y}} = \mathcal{T}$ over variables $\vec{x} \in A_1^m$ and $\vec{\mathcal{Y}} \in A_2^n$.

The setup $gk = (\mathbf{n}, G, G_T, e, \mathcal{P})$ implicitly defines A_1, A_2, A_T, f. It also implicitly defines $B_1 = B_2 = B_T = G$ and $F(\mathcal{X}, \mathcal{Y}) = e(\mathcal{X}, \mathcal{Y})$ and the linear maps[6]

$$
\begin{array}{lll}
\iota_1(x) = x\mathcal{P} & \iota_2(\mathcal{Y}) = \mathcal{Y} & \iota_T(\mathcal{T}) = e(\mathcal{P}, \mathcal{T}) \\
p_1(x\mathcal{P}) = \lambda x \bmod \mathbf{n} & p_2(\mathcal{Y}) = \lambda \mathcal{Y} & p_T(e(\mathcal{P}, \mathcal{T})) = \lambda \mathcal{T}.
\end{array}
$$

Since $\lambda^2 \equiv \lambda \bmod \mathbf{n}$ the maps commute as described in Figure 2. That is, we have

$$
\begin{tikzcd}
(x, \mathcal{Y}) \arrow[r, mapsto, "f"] \arrow[d, "{(\iota_1, \iota_2)}"] & x\mathcal{Y} \arrow[d, "\iota_T"] \\
(x\mathcal{P}, \mathcal{Y}) \arrow[r, mapsto, "F"] & e(\mathcal{P}, x\mathcal{Y})
\end{tikzcd}
$$

[6]To uniquely define the maps let the setup include a bit indicating whether $\mathbf{p}$ is the large or the small prime factor of $\mathbf{n}$.

and we have

$$
\begin{tikzcd}
(\lambda x, \lambda \mathcal{Y}) \arrow[r, mapsto, "f"] & \lambda x \mathcal{Y} \\
(x\mathcal{P}, \mathcal{Y}) \arrow[u, mapsto, "{(p_1, p_2)}"] \arrow[r, mapsto, "F"] & e(\mathcal{P}, x\mathcal{Y}) \arrow[u, mapsto, "p_T"']
\end{tikzcd}
$$

The common reference string σ consists of an element $\mathcal{U} \in G$. In the soundness setting it is generated as $\mathcal{U} = \alpha \mathbf{p} \mathcal{P}$ and in the witness-indistinguishability setting it is generated as $\mathcal{U} = \alpha \mathcal{P}$, where $\alpha \leftarrow \mathbb{Z}_{\mathbf{n}}^*$. The subgroup decision assumption implies that soundness strings and witness-indistinguishability strings are computationally indistinguishable.

We will be using $\mathcal{U}$ as a commitment key in both B_1 and in B_2. In order to commit to $x \in A_1 = \mathbb{Z}_{\mathbf{n}}$ we pick $r \in \mathbb{Z}_{\mathbf{n}}$ and compute the commitment $\mathcal{C} := \iota_1(x) + r\mathcal{U} = x\mathcal{P} + r\mathcal{U} \in B_1 = G$. In order to commit to $\mathcal{Y} \in A_2 = G$ we pick $s \leftarrow \mathbb{Z}_{\mathbf{n}}$ and compute the commitment $\mathcal{D} := \iota_2(\mathcal{Y}) + s\mathcal{U} = \mathcal{Y} + s\mathcal{U} \in B_2 = G$.

On a soundness string, $\mathcal{U}$ describes a binding key for both commitment schemes. We have $p_1(\mathcal{U}) \equiv p_1(\alpha \mathbf{p} \mathcal{P}) \equiv \lambda \alpha \mathcal{P} \equiv 0 \bmod \mathbf{n}$ and $p_2(\mathcal{U}) = \lambda \alpha \mathbf{p} \mathcal{U} = \mathcal{O}$. Furthermore, the maps $p_1 \circ \iota_1(x) = p_1(x\mathcal{P}) = \lambda x \bmod \mathbf{n}$ and $p_2 \circ \iota_2(\mathcal{Y}) = p_2(\mathcal{Y}) = \lambda \mathcal{Y}$ and $p_T \circ \iota_T(\mathcal{T}) = p_T(e(\mathcal{P}, \mathcal{T})) = \lambda \mathcal{T}$ are all non-trivial. A commitment $\mathcal{C} \in B_1$ defines the committed value uniquely in $\lambda \mathbb{Z}_{\mathbf{n}}$, and a commitment $\mathcal{D} \in B_2$ defines the committed value uniquely in λG.

On a witness-indistinguishability string, $\mathcal{U}$ describes a hiding key for both commitment schemes. Since $\mathcal{U}$ is a generator for $B_1 = B_2 = G$ we have $\iota_1(A_1) = \iota_1(\mathbb{Z}_{\mathbf{n}}) = G = \langle \mathcal{U} \rangle$ and $\iota_2(A_2) = \iota_2(G) = G = \langle \mathcal{U} \rangle$. This implies that the commitment schemes are perfectly hiding. The only solution $H \in \mathrm{Mat}_{1 \times 1}(\mathbb{Z}_{\mathbf{n}})$ to $\mathcal{U} \bullet H\mathcal{U} = 1$, i.e., $e(\mathcal{U}, H\mathcal{U}) = 1$ is $H = 0$. We do therefore not need to include any $H_1, \ldots, H_\eta$ in the common reference string.

Theorem 11 now gives us a NIWI proof for the simultaneous satisfiability of a set of multi-scalar multiplication equations with perfect completeness, perfect $L_{\text{guilt}}^{\text{M}}$-soundness and composable witness-indistinguishability.

Pairing product equations. Now consider the case where we only have pairing product equations. Define L^{P} ($L_{\text{guilt}}^{\text{P}}$) to be L (L_{guilt}) restricted to $N_{\text{M}} = N_{\text{Q}} = 0$ such that it only has N_{P} pairing product equations. Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = A_2 = G, A_T = G_T, f(x, y) = e(x, y)$, and equations of the form $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma \vec{\mathcal{Y}}) = t_T$ over variables $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G$. The setup also defines modules $B_1 = B_2 = G$ and $B_T = G_T$ and the bilinear map $F(\mathcal{X}, \mathcal{Y}) = e(\mathcal{X}, \mathcal{Y})$. We use the maps $\iota_2(\mathcal{Y}) = \mathcal{Y}$ and $p_2(\mathcal{Y}) = \lambda \mathcal{Y}$ described in the multi-scalar multiplication case above together with $\iota_T(z_T) = z_T$ and $p_T(z_T) = z_T^\lambda$ to get the commutative diagram

$$
\begin{tikzcd}
A_1 = G \arrow[d, shift right, "\iota_2"'] & \times & A_2 = G \arrow[d, shift right, "\iota_2"'] \arrow[r, "{f(\mathcal{X}, \mathcal{Y}) = e(\mathcal{X}, \mathcal{Y})}"'] & A_T = G_T \arrow[d, shift right, "\iota_T"'] \\
B_1 = G \arrow[u, shift right, "p_2"'] & \times & B_2 = G \arrow[u, shift right, "p_2"'] \arrow[r, "{F(\mathcal{X}, \mathcal{Y}) = e(\mathcal{X}, \mathcal{Y})}"] & B_T = G_T \arrow[u, shift right, "p_T"']
\end{tikzcd}
$$

Using the same type of common reference string as in the multi-scalar multiplication case described above, we get a NIWI proof for the simultaneous satisfiability of pairing product equations with perfect completeness, perfect $L^{\mathrm{P}}_{\mathrm{guilt}}$-soundness and composable witness-indistinguishability.

Quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. Finally, consider the case where we only have quadratic equations. Define L^{Q} ($L^{\mathrm{Q}}_{\mathrm{guilt}}$) to be L (L_{guilt}) restricted to $N_{\mathrm{P}} = N_{\mathrm{M}} = 0$ such that it only has N_{Q} quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{n}}, A_1 = A_2 = A_T = \mathbb{Z}_{\mathbf{n}}, f(x,y) = xy \bmod \mathbf{n}$, and equations of the form $\vec{x}\cdot\vec{b}+\vec{x}\cdot\Gamma\vec{x} \equiv t \bmod \mathbf{n}$ over variables $x_1,\ldots,x_m \in \mathbb{Z}_{\mathbf{n}}$. The setup also defines modules $B_1 = B_2 = G$ and $B_T = G_T$ and the bilinear map $F(x\mathcal{P}, y\mathcal{P}) = e(x\mathcal{P}, y\mathcal{P})$. We use the maps $\iota_1(x) = x\mathcal{P}$ and $p_1(x\mathcal{P}) = \lambda x$ described in the multi-scalar multiplication case above together with $\iota_T(t) = e(\mathcal{P}, t\mathcal{P})$ and $p_T(e(\mathcal{P}, t\mathcal{P})) = \lambda t \bmod \mathbf{n}$ to get the commutative diagram

$$\begin{array}{ccccc}
A_1 = \mathbb{Z}_{\mathbf{n}} & \times & A_2 = \mathbb{Z}_{\mathbf{n}} & \xrightarrow[f(x,y) = xy \bmod \mathbf{n}]{} & A_T = \mathbb{Z}_{\mathbf{n}} \\
\iota_1 \downarrow\uparrow p_1 & & \iota_1 \downarrow\uparrow p_1 & & \iota_T \downarrow\uparrow p_T \\
B_1 = G & \times & B_2 = G & \xrightarrow{F(\mathcal{X},\mathcal{Y}) = e(\mathcal{X},\mathcal{Y})} & B_T = G_T
\end{array}$$

Using the same type of common reference string as in the multi-scalar multiplication case described above, we get a NIWI proof for the simultaneous satisfiability of quadratic equations with perfect completeness, perfect $L^{\mathrm{Q}}_{\mathrm{guilt}}$-soundness and composable witness-indistinguishability.

The general case. In the three special cases described above, we used the same type of common reference string $\sigma = \mathcal{U}$. To get a NIWI proof for the simultaneous satisfiability of equations we will combine them by using the same $\mathcal{U}$ for all three types of equations. The same commitments to scalars $x_i \in \mathbb{Z}_{\mathbf{n}}$ are used in both multi-scalar multiplication equations and in quadratic equations in $\mathbb{Z}_{\mathbf{n}}$ and the same commitments to variables $\mathcal{Y}_j \in G$ are used in both pairing product equations and in multi-scalar multiplication equations to enforce consistency accross different types of equations. The full NIWI proof for L is as follows:

Setup: $(gk, sk) := ((\mathbf{n}, G, G_T, e, \mathcal{P}), (\mathbf{p}, \mathbf{q})) \leftarrow \mathcal{G}(1^k)$, where $\mathbf{n} = \mathbf{pq}$.

Soundness string: On input (gk, sk) return $\sigma := \mathcal{U}$ where $\mathcal{U} := r\mathbf{p}\mathcal{P}$ for random $r \in \mathbb{Z}^*_{\mathbf{n}}$.

Witness-indistinguishability string: On input (gk, sk) return $\sigma := \mathcal{U}$ where $\mathcal{U} := r\mathcal{P}$ for random $r \in \mathbb{Z}^*_{\mathbf{n}}$.

Prover: On input $(\mathbf{n}, G, G_T, e, \mathcal{P}, \mathcal{U})$, a set of $N = N_{\mathrm{P}} + N_{\mathrm{M}} + N_{\mathrm{Q}}$ equations, and a witness $\vec{x}, \vec{\mathcal{Y}}$ do:

1. Commit to the scalars $x_1,\ldots,x_m \in \mathbb{Z}_{\mathbf{n}}$ and the group elements $\mathcal{Y}_1,\ldots,\mathcal{Y}_n \in G$ as
$$\mathcal{C}_i := x_i\mathcal{P} + r_i\mathcal{U} \qquad\qquad \mathcal{D}_i := \mathcal{Y}_i + s_i\mathcal{U}$$
for randomly chosen $\vec{r} \in \mathbb{Z}^m_{\mathbf{n}}, \vec{s} \in \mathbb{Z}^n_{\mathbf{n}}$.
2. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ make a proof as described in Section 6.3
$$\begin{aligned}\phi \;&:=\; \vec{s}^\top\vec{\mathcal{A}} + \vec{s}^\top(\Gamma + \Gamma^\top)\vec{\mathcal{Y}} + \vec{s}^\top\Gamma\vec{s}\mathcal{U} \\ &=\; \sum_{i=1}^n s_i\mathcal{A}_i + \sum_{i=1}^n\sum_{j=1}^n(\gamma_{ij} + \gamma_{ji})s_i\mathcal{Y}_j + \sum_{i=1}^n\sum_{j=1}^n \gamma_{ij}s_is_j\mathcal{U}.\end{aligned}$$

3. For each multi-scalar multiplication equation $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}$ the proof is

$$\begin{aligned} \phi : &= \vec{r}^\top \vec{\mathcal{B}} + \vec{r}^\top \Gamma\vec{\mathcal{Y}} + \vec{r}^\top \Gamma\vec{s}\mathcal{U} + \vec{s}^\top \vec{a}\mathcal{P} + \vec{s}^\top \Gamma\vec{x}\mathcal{P} \\ &= \sum_{i=1}^m r_i\mathcal{B}_i + \sum_{i=1}^m\sum_{j=1}^n r_i\gamma_{ij}\mathcal{Y}_j + \sum_{i=1}^m\sum_{j=1}^n \gamma_{ij}r_is_j\mathcal{U} + \sum_{i=1}^n s_i(a_i + \sum_{j=1}^m \gamma_{ij}x_j)\mathcal{P}. \end{aligned}$$

4. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{x} = t$ in $\mathbb{Z}_\mathbf{n}$ we have

$$\begin{aligned} \phi &:= \vec{r}^\top \vec{b}\mathcal{P} + \vec{r}^\top(\Gamma + \Gamma^\top)\vec{x}\mathcal{P} + \vec{r}^\top\Gamma\vec{r}\mathcal{U} \\ &= (\sum_{i=1}^m r_ib_i + \sum_{i=1}^m\sum_{j=1}^m(\gamma_{ij} + \gamma_{ji})r_ix_j)\mathcal{P} + \sum_{i=1}^m\sum_{j=1}^m \gamma_{ij}r_ir_j\mathcal{U}. \end{aligned}$$

Verifier: On input $(\mathbf{n}, G, G_T, e, \mathcal{P}, \mathcal{U})$, a set of equations and a proof $\vec{\mathcal{C}}, \vec{\mathcal{D}}, \{\phi_i\}_{i=1}^N$ do:

1. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ with proof ϕ check that

$$\prod_{i=1}^n e(\mathcal{A}_i, \mathcal{D}_i) \cdot \prod_{i=1}^n\prod_{j=1}^n e(\mathcal{D}_i, \mathcal{D}_j)^{\gamma_{ij}} = t_T e(\mathcal{U}, \phi).$$

2. For each multi-scalar multiplication $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}$ with proof ϕ check that

$$\prod_{i=1}^n e(a_i\mathcal{P}, \mathcal{D}_i) \cdot \prod_{i=1}^m e(\mathcal{C}_i, \mathcal{B}_i) \cdot \prod_{i=1}^m\prod_{j=1}^n e(\mathcal{C}_i, \mathcal{D}_j)^{\gamma_{ij}} = e(\mathcal{P}, \mathcal{T})e(\mathcal{U}, \phi).$$

3. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{x} = t$ in $\mathbb{Z}_\mathbf{n}$ with proof ϕ check that

$$\prod_{i=1}^m e(\mathcal{C}_i, b_i\mathcal{P}) \cdot \prod_{i=1}^m\prod_{j=1}^m e(\mathcal{C}_i, \mathcal{C}_j)^{\gamma_{ij}} = e(\mathcal{P}, \mathcal{P})^t e(\mathcal{U}, \phi).$$

Theorem 13 *The NIWI proof for L given above has perfect completeness, perfect L_{guilt}-soundness and composable witness-indistinguishability.*

Proof. Perfect completeness follows from the perfect completeness of each of the three types of proofs. Perfect L_{guilt}-soundness follows from Theorem 10 since we use the same commitments and maps p_1, p_2 accross different types of equations thus making the order $\mathbf{p}$ solutions $\vec{x} = p_1(\vec{\mathcal{C}}), \vec{\mathcal{Y}} = p_2(\vec{\mathcal{D}})$ consistent with each other for all three types of equations. The subgroup decision assumption implies that soundness and witness-indistinguishability common reference strings are indistinguishable. On a witness-indistinguishability string the commitments are perfectly hiding and we get perfect witness-indistinguishability from Theorem 8. □

Size. The size of the NIWI proof is $m + n + N$ group elements in G, where m is the number of variables in $\vec{x}$, n is the number of variables in $\vec{\mathcal{Y}}$ and $N = N_\text{P} + N_\text{M} + N_\text{Q}$ is the total number of equations.

9 Instantiation based on the SXDH assumption

Setup. The setup algorithm $\mathcal{G}_{\text{SXDH}}$ returns a prime order bilinear group $gk = (\mathbf{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$. We will assume the decision Diffie-Hellman problem is hard in both groups, i.e., the Symmetric External Diffie-Hellman (SXDH) assumption.

Definition 14 (SXDH assumption) *We say the SXDH assumption holds for $\mathcal{G}_{\text{SXDH}}$ if for all non-uniform polynomial time $\mathcal{A}$ and all $b \in \{1,2\}$ we have*

$$\begin{aligned} & \Pr[gk \leftarrow \mathcal{G}_{\text{SXDH}}(1^k); \alpha, t \leftarrow \mathbb{Z}_{\mathbf{p}}^* : \mathcal{A}(gk, \alpha\mathcal{P}_b, t\mathcal{P}_b, \alpha t\mathcal{P}_b) = 1] \\ \approx\ & \Pr[gk \leftarrow \mathcal{G}_{\text{SXDH}}(1^k); \alpha, t, r \leftarrow \mathbb{Z}_{\mathbf{p}}^* : \mathcal{A}(gk, \alpha\mathcal{P}_b, t\mathcal{P}_b, r\mathcal{P}_b) = 1] \end{aligned}$$

Statements. The setup $gk = (\mathbf{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$ defines the ring $\mathbb{Z}_{\mathbf{p}}$ and modules $\mathbb{Z}_{\mathbf{p}}, G_1, G_2, G_T$ and bilinear maps corresponding to multiplication in $\mathbb{Z}_{\mathbf{p}}$, scalar multiplication in G_1 and G_2, and the pairing $e : G_1 \times G_2 \to G_T$.

With this setup we can define pairing product equations, multi-scalar multiplication equations and quadratic equations as follows:

Pairing product equations: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{p}}, A_1 = G_1, A_2 = G_2, A_T = G_T, f(x,y) = e(x,y)$, and equations of the form $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{X}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$.

Multi-scalar multiplication in G_1: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{p}}, A_1 = G_1, A_2 = \mathbb{Z}_{\mathbf{p}}, A_T = G_1, f(\mathcal{X}, y) = y\mathcal{X}$, and equations of the form $\vec{\mathcal{A}} \cdot \vec{y} + \vec{\mathcal{X}} \cdot \vec{b} + \vec{\mathcal{X}} \cdot \Gamma\vec{y} = \mathcal{T}_1$.

Multi-scalar multiplication in G_2: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{p}}, A_1 = \mathbb{Z}_{\mathbf{p}}, A_2 = G_2, A_T = G_2, f(x, \mathcal{Y}) = x\mathcal{Y}$, and equations of the form $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}_2$.

Quadratic equation in $\mathbb{Z}_{\mathbf{p}}$: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_{\mathbf{p}}, A_1 = \mathbb{Z}_{\mathbf{p}}, A_2 = \mathbb{Z}_{\mathbf{p}}, A_T = \mathbb{Z}_{\mathbf{p}}, f(x,y) = xy \bmod \mathbf{p}$, and equations of the form $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = t$.

We consider statements that consist of sets of pairing product equations, multi-scalar multiplications in G_1 and G_2, and quadratic equations as described above. The equations are over variables $x_1, \ldots, x_{m'}, y_1, \ldots, y_{n'} \in \mathbb{Z}_{\mathbf{p}}$ and $\mathcal{X}_1, \ldots, \mathcal{X}_m \in G_1$ and $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G_2$. We let L be the language of statements where there exists a solution $\vec{x}, \vec{y}, \vec{\mathcal{X}}, \vec{\mathcal{Y}}$ that simultaneously satisfies all equations of all types.

Commitments. Consider a group G of prime order $\mathbf{p}$. With entry-wise addition we get the $\mathbb{Z}_{\mathbf{p}}$-module $B := G^2$. We will use a commitment key of the form

$$u_1 = (\mathcal{P}, \mathcal{Q}) := (\mathcal{P}, \alpha\mathcal{P}) \qquad\qquad u_2 = (\mathcal{U}, \mathcal{V}),$$

where $\alpha \leftarrow \mathbb{Z}_{\mathbf{p}}^*$ is chosen at random. We can choose $u_2 = (\mathcal{U}, \mathcal{V})$ in two different ways: $u_2 := tu_1$ or $u_2 := tu_1 - (\mathcal{O}, \mathcal{P})$ for a random $t \in \mathbb{Z}_{\mathbf{p}}^*$. The former choice of u_2 gives a perfectly binding commitment key, whereas the latter choice of u_2 gives a perfectly hiding commitment key. The two types of commitment keys are computationally indistinguishable under the DDH assumption.

Let us now describe how to commit to an element $\mathcal{X} \in G_1$ using randomness $r_1, r_2 \in \mathbb{Z}_{\mathbf{p}}$:

$$\iota(\mathcal{Z}) := (\mathcal{O}, \mathcal{Z}) \qquad\qquad p(\mathcal{Z}_1, \mathcal{Z}_2) := \mathcal{Z}_2 - \alpha\mathcal{Z}_1 \qquad\qquad c := \iota(\mathcal{X}) + r_1u_1 + r_2u_2.$$

On a binding key where $u_2 = tu_1$ we have that $p \circ \iota$ is the identity map on G and $p(u_1) = p(u_2) = \mathcal{O}$. The commitment $c = ((r_1 + r_2t)\mathcal{P}, (r_1 + r_2t)\mathcal{Q} + \mathcal{X})$ corresponds to an ElGamal encryption of $\mathcal{X}$.

On a hiding key on the other hand, u_1 and u_2 are linearly independent. This means u_1, u_2 form a basis for $B = G^2$ and $\iota(G) \subseteq \langle u_1, u_2 \rangle$ giving a perfectly hiding commitment.

Commitment to a scalar $x \in \mathbb{Z}_\mathbf{p}$ using randomness $r \in \mathbb{Z}_\mathbf{p}$ works as follows:

$$u := u_2 + (\mathcal{O}, \mathcal{P}) \qquad \iota'(z) := zu \qquad p'(z_1\mathcal{P}, z_2\mathcal{P}) := z_2 - \alpha z_1 \qquad c := \iota'(x) + ru_1.$$

On a binding key $p' \circ \iota'$ is the identity map and $p'(u_1) = 0$, so the commitment scheme is perfectly binding, and in fact the commitment $c = ((r + xt)\mathcal{P}, (r + xt)\mathcal{Q} + x\mathcal{P})$ is an ElGamal encryption of $x\mathcal{P}$. On a hiding key we have $u = tu_1$ so $u \in \langle u_1 \rangle$, which implies $\iota'(\mathbb{Z}_\mathbf{p}) \subseteq \langle u_1 \rangle$. A hiding key therefore gives us a perfectly hiding commitment scheme.

Common reference string. The common reference string is of the form (u_1, u_2, v_1, v_2), where (u_1, u_2) is a commitment key for the group G_1 implicitly defining maps $\iota_1, p_1, \iota_1', p_1'$ as described above, and (v_1, v_2) is a commitment key for G_2 implicitly defining maps $\iota_2, p_2, \iota_2', p_2'$ as described above.

We will always use $B_1 = G_1^2, B_2 = G_2^2$ and we define $B_T := G_T^4$ with addition being entry-wise multiplication. The map F is defined as follows:

$$F : G_1^2 \times G_2^2 \to G_T^4 \qquad \left(\begin{pmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \end{pmatrix}, \begin{pmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \end{pmatrix} \right) \mapsto \begin{pmatrix} e(\mathcal{X}_1, \mathcal{Y}_1) & e(\mathcal{X}_1, \mathcal{Y}_2) \\ e(\mathcal{X}_2, \mathcal{Y}_1) & e(\mathcal{X}_2, \mathcal{Y}_2) \end{pmatrix}.$$

On a witness-indistinguishability string, we have hiding commitment keys u_1, u_2 and v_1, v_2 where each pair of vectors is linearly independent. The four elements $F(u_1, v_1), F(u_1, v_2), F(u_2, v_1), F(u_2, v_2)$ are also linearly independent in the witness-indistinguishability scenario. This implies that $\vec{u} \bullet H\vec{v} = 0$ only has the trivial solution where H is the 2×2 matrix with 0-entries. Therefore, the common reference string does not need to include any matrices $H_1, \ldots, H_\eta$ for the pairing product equations. The same holds true for the other types of equations, we do not need any matrices $H_1, \ldots, H_\eta$ in the common reference string.

Pairing product equations. Consider first the restricted language $L^{\mathrm{P}} \subset L$, where the statements only have pairing product equations. The common reference string describes $\mathcal{R} = \mathbb{Z}_\mathbf{p}, A_1 = G_1, A_2 = G_2, A_T = G_T$, and $B_1 = G_1^2, B_2 = G_2^2, B_T = G_T^4$, and commitment keys u_1, u_2, v_1, v_2, and the following commuting linear and bilinear maps:

$$\begin{array}{ccc}
(\mathcal{X}, \mathcal{Y}) & \xmapsto{\quad f \quad} & e(\mathcal{X}, \mathcal{Y}) \\
\Big\downarrow {\scriptstyle (\iota_1, \iota_2)} & & \Big\downarrow {\scriptstyle \iota_T} \\
\left(\begin{pmatrix} \mathcal{O} \\ \mathcal{X} \end{pmatrix}, \begin{pmatrix} \mathcal{O} \\ \mathcal{Y} \end{pmatrix} \right) & \xmapsto{\quad F \quad} & \begin{pmatrix} 1 & 1 \\ 1 & e(\mathcal{X}, \mathcal{Y}) \end{pmatrix}
\end{array}$$

For the following maps we recall $u_1 = (\mathcal{P}_1, \alpha_1\mathcal{P}_1)$ and $v_1 = (\mathcal{P}_2, \alpha_2\mathcal{P}_2)$:

$$\begin{array}{ccc}
(\mathcal{X}_2-\alpha_1\mathcal{X}_1,\mathcal{Y}_2-\alpha_2\mathcal{Y}_1) & \stackrel{f}{\longmapsto} & e(\mathcal{X}_2-\alpha_1\mathcal{X}_1,\mathcal{Y}_2-\alpha_2\mathcal{Y}_1) \\
\uparrow (p_1,p_2) & & \uparrow p_T \\
\left(\begin{pmatrix}\mathcal{X}_1\\ \mathcal{X}_2\end{pmatrix},\begin{pmatrix}\mathcal{Y}_1\\ \mathcal{Y}_2\end{pmatrix}\right) & \stackrel{F}{\longmapsto} & \begin{pmatrix} e(\mathcal{X}_1,\mathcal{Y}_1) & e(\mathcal{X}_1,\mathcal{Y}_2)\\ e(\mathcal{X}_2,\mathcal{Y}_1) & e(\mathcal{X}_2,\mathcal{Y}_2)\end{pmatrix}
\end{array}$$

This gives us the setup from Section 5 and we can use the NIWI proofs described in Section 6 on the pairing product equations.

Multi-scalar multiplication in G_1 or G_2. For multi-scalar multiplications in G_1, we will need maps $\widetilde{\iota}_T : G_1 \to G_T^4$ and $\widetilde{p}_T : G_T^4 \to G_1$. For multi-scalar multiplications in G_2 we will need maps $\hat{\iota}_T : G_2 \to G_T^4$ and $\hat{p}_T : G_T^4 \to G_2$. The two cases are symmetric, so we will just focus on multi-scalar multiplication in G_2 here.

We define

$$\hat{\iota}_T(\mathcal{Z}) := F(\iota_1'(1),\iota_2(\mathcal{Z})) = F(u,(\mathcal{O},\mathcal{Z})) \qquad\qquad \hat{p}_T = e^{-1}(p_T(z)),$$

where $e^{-1}(e(\mathcal{P}_1,\mathcal{Z})) := \mathcal{Z}$. In the soundness setting $\hat{p}_T \circ \hat{\iota}_T$ is the identity map on G_2.

We have $F(\iota_1'(x),\iota_2(\mathcal{Y})) = F(\iota_1'(1),\iota_2(x\mathcal{Y})) = \hat{\iota}_T(x\mathcal{Y})$ by the linearity and bilinearity of the maps, and $p_1'(x_1\mathcal{P}_1,x_2\mathcal{P}_1)p_2(\mathcal{Y}_1,\mathcal{Y}_2) = (x_2-\alpha_1x_1)(\mathcal{Y}_2-\alpha_2\mathcal{Y}_1) = x_2\mathcal{Y}_2-\alpha_1x_1\mathcal{Y}_2-\alpha_2(x_2\mathcal{Y}_1-\alpha_1x_1\mathcal{Y}_1) = \hat{p}_T(F((x_1\mathcal{P}_1,x_2\mathcal{P}_2),(\mathcal{Y}_1,\mathcal{Y}_2)))$. This gives us the commutative diagram of linear and bilinear maps:

$$\begin{array}{ccccc}
A_1=\mathbb{Z}_\mathbf{p} & \times & A_2=G_2 & \xrightarrow[f(x,\mathcal{Y})=x\mathcal{Y}]{} & A_T=G_T \\
\iota_1' \downarrow\uparrow p_1' & & \iota_2 \downarrow\uparrow p_2 & & \hat{\iota}_T \downarrow\uparrow \hat{p}_T \\
B_1=G_1^2 & \times & B_2=G_2^2 & \xrightarrow{F} & B_T=G_T^4
\end{array}$$

Using this setup, we can apply the NIWI proof from Section 6 to multi-scalar multiplication equations in G_2. The case of multi-scalar multiplication in G_1 is treated similarly.

Quadratic equations. For quadratic equations in $\mathbb{Z}_\mathbf{p}$ we define the maps $\iota_T' : \mathbb{Z}_\mathbf{p} \to G_T^4$ and $p_T' : G_T^4 \to \mathbb{Z}_\mathbf{p}$ as follows

$$\iota_T'(t) := F(\iota_1'(1),\iota_2'(t)) = F(u,tv) \qquad\qquad p_T'(z) := \log_{e(\mathcal{P}_1,\mathcal{P}_2)}(p_T(z)).$$

In the soundness setting $p_T' \circ \iota_T'$ is the identity map on $\mathbb{Z}_\mathbf{p}$. To see that the maps satisfy the two commutative properties from Figure 2, observe $F(\iota_1'(x),\iota_2'(y)) = F(\iota_1'(1),\iota_2(xy)) = \iota'(xy)$ by the linearity and bilinearity of the maps, and

$$\begin{aligned}
& p_1'(x_1\mathcal{P}_1,x_2\mathcal{P}_1)p_2'(y_1\mathcal{P}_2,y_2\mathcal{P}_2) \\
= \;& (x_2-\alpha_1x_1)(y_2-\alpha_2y_1) = x_2y_2-\alpha_1x_1y_2-\alpha_2(x_2y_1-\alpha_1x_1y_1) = p_T'(F((x_1\mathcal{P}_1,x_2\mathcal{P}_2),(y_1\mathcal{P}_2,y_2\mathcal{P}_2))).
\end{aligned}$$

This gives us the following setup:

$$\begin{tikzcd}
A_1 = \mathbb{Z}_{\mathbf{p}} \arrow[d, shift right, "\iota_1'"'] & \times & A_2 = \mathbb{Z}_{\mathbf{p}} \arrow[r, "f(x,y) = xy \bmod \mathbf{p}"'] \arrow[d, shift right, "\iota_2'"'] & A_T = \mathbb{Z}_{\mathbf{p}} \arrow[d, shift right, "\iota_T'"'] \\
B_1 = G_1^2 \arrow[u, shift right, "p_1'"'] & \times & B_2 = G_2^2 \arrow[r, "F"] \arrow[u, shift right, "p_2'"'] & B_T = G_T^4 \arrow[u, shift right, "p_T'"']
\end{tikzcd}$$

NIWI proof. We now give the full NIWI proof for L.

Setup: $gk := (\mathbf{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2) \leftarrow \mathcal{G}_{\mathrm{SXDH}}(1^k)$.

Soundness string: On input gk return $\sigma := (u_1, u_2, v_1, v_2)$ where $u_2 = t_1 u_1$ and $v_2 = t_2 v_1$ for random $t_1, t_2 \leftarrow \mathbb{Z}_{\mathbf{p}}$.

Witness-indistinguishability string: On input gk return $\sigma := (u_1, u_2, v_1, v_2)$ where $u_2 = t_1 u_1 - (\mathcal{O}, \mathcal{P}_1)$ and $v_2 = t_2 v_1 - (\mathcal{O}, \mathcal{P}_2)$ for random $t_1, t_2 \leftarrow \mathbb{Z}_{\mathbf{p}}$.

NIWI proof: On input gk, σ, a set of equations and a witness $\vec{\mathcal{X}}, \vec{\mathcal{Y}}, \vec{x}, \vec{y}$ do:

1. Commit to the group elements $\vec{\mathcal{X}} \in G_1^m$ and the scalars $\vec{x} \in \mathbb{Z}_{\mathbf{p}}^{m'}$ as

$$\vec{c} := \iota_1(\vec{\mathcal{X}}) + R\vec{u} \qquad \vec{c}' := \iota_1'(x) + \vec{r}u_1 \quad \text{ where } \; R \leftarrow \mathrm{Mat}_{m\times 2}(\mathbb{Z}_{\mathbf{p}}), \vec{r} \leftarrow \mathbb{Z}_{\mathbf{p}}^{m'}.$$

   Commit to the group elements $\vec{\mathcal{Y}} \in G_2^n$ and the scalars $\vec{y} \in \mathbb{Z}_{\mathbf{p}}^{n'}$ as

$$\vec{d} := \iota_2(\vec{\mathcal{Y}}) + S\vec{v} \qquad \vec{d}' := \iota_2'(y) + \vec{s}v_1 \quad \text{ where } \; S \leftarrow \mathrm{Mat}_{n\times 2}(\mathbb{Z}_{\mathbf{p}}), \vec{s} \leftarrow \mathbb{Z}_{\mathbf{p}}^{n'}.$$

2. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{X}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ make a proof as described in Section 6. Writing it out, we have for $T \leftarrow \mathrm{Mat}_{2\times 2}(\mathbb{Z}_{\mathbf{p}})$ the following proof:

$$\begin{aligned}
\vec{\pi} &:= R^\top \iota_2(\vec{\mathcal{B}}) + R^\top \Gamma \iota_2(\vec{\mathcal{Y}}) + (R^\top \Gamma S - T^\top)\vec{v} \\
\vec{\theta} &:= S^\top \iota_1(\vec{\mathcal{A}}) + S^\top \Gamma^\top \iota_1(\vec{\mathcal{X}}) + T\vec{u}
\end{aligned}$$

   Following Section 6.2, for each linear equation $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = t_T$ we use $\vec{\pi} := \vec{0}$ and $\vec{\theta} := S^\top \iota_1(\vec{\mathcal{A}})$. There is a bijective correspondence between $S^\top \vec{\mathcal{A}} = p_1(\vec{\theta})$ and $\vec{\theta} = \iota_1(S^\top \vec{\mathcal{A}})$. The proof can therefore be communicated by sending $S^\top \vec{\mathcal{A}}$, which consists of two group elements in G_1.
   For each linear equation $\vec{\mathcal{X}} \cdot \vec{\mathcal{B}} = t_T$ we use $\vec{\pi} := R^\top \iota_2(\vec{\mathcal{B}})$ and $\vec{\theta} := \vec{0}$. As above, the proof can be communicated by sending the two group elements $R^\top \vec{\mathcal{B}}$ in G_2.
3. For each multi-scalar multiplication equation $\vec{\mathcal{A}} \cdot \vec{y} + \vec{\mathcal{X}} \cdot \vec{b} + \vec{\mathcal{X}} \cdot \Gamma\vec{y} = \mathcal{T}_1$ in G_1 the proof is for random $T \leftarrow \mathrm{Mat}_{1\times 2}(\mathbb{Z}_{\mathbf{p}})$

$$\begin{aligned}
\vec{\pi} &:= R^\top \iota_2'(\vec{b}) + R^\top \Gamma \iota_2'(\vec{y}) + (R^\top \Gamma \vec{s} - T^\top)v_1 \\
\theta &:= \vec{s}^\top \iota_1(\vec{\mathcal{A}}) + \vec{s}^\top \Gamma^\top \iota_1(\vec{\mathcal{X}}) + T\vec{u}
\end{aligned}$$

   For each linear equation $\vec{\mathcal{A}} \cdot \vec{y} = \mathcal{T}_1$ the proof is $\vec{\pi} := \vec{0}$ and $\theta := \vec{s}^\top \iota_1(\vec{\mathcal{A}})$. There is a bijective correspondence between $\vec{s}^\top \vec{\mathcal{A}} = p_1(\vec{\theta})$ and $\theta = \iota_1(\vec{s}^\top \vec{\mathcal{A}})$. The proof can therefore be communicated by sending $\vec{s}^\top \vec{\mathcal{A}}$, which consists of one group element in G_1.
   For each linear equation $\vec{\mathcal{X}} \cdot \vec{b} = \mathcal{T}_1$ the proof is $\vec{\pi} := R^\top \iota_2'(\vec{b})$ and $\theta := 0$. As above, the proof can be communicated by sending the two field elements $R^\top \vec{b}$.

4. For each multi-scalar multiplication equation $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma \vec{\mathcal{Y}} = \mathcal{T}_2$ in G_2 the proof is for random $T \leftarrow \mathrm{Mat}_{2\times 1}(\mathbb{Z}_{\mathbf{p}})$

$$\begin{aligned}\pi &:= \vec{r}^\top \iota_2(\vec{\mathcal{B}}) + \vec{r}^\top \Gamma \iota_2(\vec{\mathcal{Y}}) + (\vec{r}^\top \Gamma S - T^\top)\vec{v} \\ \vec{\theta} &:= S^\top \iota_1'(\vec{a}) + S^\top \Gamma^\top \iota_1'(\vec{x}) + T u_1\end{aligned}$$

For each linear equation $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{T}_2$ the proof is $\pi := 0$ and $\vec{\theta} := S^\top \iota_1'(\vec{a})$. There is a bijective correspondence between $S^\top \vec{a} = p_1'(\vec{\theta})$ and $\vec{\theta} = \iota_1'(S^\top \vec{a})$. The proof can therefore be communicated by sending $S^\top \vec{a}$, which consists of two field elements.

For each linear equation $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}_2$ the proof is $\pi := \vec{r}^\top \iota_2(\vec{\mathcal{B}})$ and $\vec{\theta} := 0$. As above, the proof can be communicated by sending the single group element $\vec{r}^\top \vec{\mathcal{B}}$.

5. For each quadratic equation $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$ in $\mathbb{Z}_{\mathbf{p}}$ the proof is for random $T \leftarrow \mathbb{Z}_{\mathbf{p}}$

$$\begin{aligned}\pi &:= \vec{r}^\top \iota_2'(\vec{b}) + \vec{r}^\top \Gamma \iota_2'(\vec{y}) + (\vec{r}^\top \Gamma \vec{s} - T)v_1 \\ \theta &:= \vec{s}^\top \iota_1'(\vec{a}) + \vec{s}^\top \Gamma^\top \iota_1'(\vec{x}) + T u_1\end{aligned}$$

For each linear equation $\vec{a} \cdot \vec{y} = t$ we use $\pi := 0$ and $\theta := \vec{s}^\top \iota_1'(\vec{a})$. There is a bijective correspondence between $\vec{s}^\top \vec{a} = p_1'(\theta)$ and $\theta = \iota_1'(\vec{s}^\top \vec{a})$. The proof can therefore be communicated by sending $\vec{s}^\top \vec{a}$, which consists of one field element.

For each linear equation $\vec{x} \cdot \vec{b} = t$ we use $\pi := \vec{r}^\top \iota_2'(\vec{b})$. As above, the proof can be communicated by sending the single field element $\vec{r}^\top \vec{b}$.

Verifier: On input (gk, σ), a set of equations and a proof $\vec{c}, \vec{d}, \vec{c}', \vec{d}', \{\vec{\pi}_i, \vec{\theta}_i\}_{i=1}^N$ do:

1. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{X}} \cdot \Gamma \vec{\mathcal{Y}}) = t_T$ with proof $(\vec{\pi}, \vec{\theta})$ check that
$$\iota_1(\vec{\mathcal{A}}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{\mathcal{B}}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t_T) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}.$$

2. For each multi-scalar equation $\vec{\mathcal{A}} \cdot \vec{y} + \vec{\mathcal{X}} \cdot \vec{b} + \vec{\mathcal{X}} \cdot \Gamma \vec{y} = \mathcal{T}_1$ in G_1 with proof $(\vec{\pi}, \theta)$ check that
$$\iota_1(\vec{\mathcal{A}}) \bullet \vec{d}' + \vec{c} \bullet \iota_2'(\vec{b}) + \vec{c} \bullet \Gamma \vec{d}' = \tilde{\iota}_T(\mathcal{T}_1) + \vec{u} \bullet \vec{\pi} + F(\theta, v_1).$$

3. For each multi-scalar equation $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma \vec{\mathcal{Y}} = \mathcal{T}_2$ in G_2 with proof $(\pi, \vec{\theta})$ check that
$$\iota_1'(\vec{a}) \bullet \vec{d} + \vec{c}' \bullet \iota_2(\vec{\mathcal{B}}) + \vec{c}' \bullet \Gamma \vec{d} = \hat{\iota}_T(\mathcal{T}_2) + F(u_1, \pi) + \vec{\theta} \bullet \vec{v}.$$

4. For each quadratic equation $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$ in $\mathbb{Z}_{\mathbf{p}}$ with proof (π, θ) check that
$$\iota_1'(\vec{a}) \bullet \vec{d}' + \vec{c}' \bullet \iota_2'(\vec{b}) + \vec{c}' \bullet \Gamma \vec{d}' = \iota_T'(t) + F(u_1, \pi) + F(\theta, v_1).$$

Theorem 15 *The protocol is a NIWI proof with perfect completeness, perfect soundness and composable witness-indistinguishability for satisfiability of a set of equations over a bilinear group where the SXDH problem is hard.*

Proof. Perfect completeness follows from Theorem 6. Perfect soundness follows from Theorem 7 since the $p \circ \iota$ maps are identity maps on $\mathbb{Z}_{\mathbf{p}}, G_1, G_2$ and G_T. The SXDH assumption gives us that the two types of common reference strings are computationally indistinguishable. On a witness-indistinguishability string, the commitments are perfectly hiding and we get perfect witness-indistinguishability from Theorems 8 and 9. □

Assumption: SXDH	G_1	G_2	$\mathbb{Z}_\mathbf{p}$
Variables $x \in \mathbb{Z}_\mathbf{p}, \mathcal{X} \in G_1$	2	0	0
Variables $y \in \mathbb{Z}_\mathbf{p}, \mathcal{Y} \in G_2$	0	2	0
Pairing product equations	4	4	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = t_T$	2	0	0
- Linear equation: $\vec{\mathcal{X}} \cdot \vec{\mathcal{B}} = t_T$	0	2	0
Multi-scalar multiplication equations in G_1	2	4	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{y} = \mathcal{T}_1$	1	0	0
- Linear equation: $\vec{\mathcal{X}} \cdot \vec{b} = \mathcal{T}_1$	0	0	2
Multi-scalar multiplication equations in G_2	4	2	0
- Linear equation: $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{T}_2$	0	0	2
- Linear equation: $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}_2$	0	1	0
Quadratic equations in $\mathbb{Z}_\mathbf{p}$	2	2	0
- Linear equation: $\vec{a} \cdot \vec{y} = t$	0	0	1
- Linear equation: $\vec{x} \cdot \vec{b} = t$	0	0	1

Table 3: Cost of each variable and equation measured in elements from G_1, G_2 and $\mathbb{Z}_\mathbf{p}$.

Size. The modules we work in are $B_1 = G_1^2$ and $B_2 = G_2^2$, so each element in a module consists of two group elements from respectively G_1 and G_2. Table 3 lists the cost of all the different types of equations.

10 Instantiation based on the DLIN assumption

Setup. Let $\mathcal{G}_{\mathrm{DLIN}}$ be a generator of a bilinear group $(\mathbf{p}, G, G_T, e, \mathcal{P})$. The decisional linear assumption (DLIN) introduced by Boneh, Boyen and Shacham [BBS04] states that given $(\alpha\mathcal{P}, \beta\mathcal{P}, r\alpha\mathcal{P}, s\beta\mathcal{P}, t\mathcal{P})$ for random α, β, r, s it is hard to tell whether $t = r + s$ or t is random.

Definition 16 (DLIN assumption) *The decisional linear assumption holds for $\mathcal{G}_{\mathrm{DLIN}}$ if for all non-uniform polynomial time $\mathcal{A}$ we have*

$$\begin{aligned} & \Pr[gk \leftarrow \mathcal{G}_{\mathrm{DLIN}}(1^k); \alpha, \beta, r, s \leftarrow \mathbb{Z}_\mathbf{p}^* : \mathcal{A}(gk, \alpha\mathcal{P}, \beta\mathcal{P}, r\alpha\mathcal{P}, s\beta\mathcal{P}, (r+s)\mathcal{P}) = 1] \\ \approx \ & \Pr[gk \leftarrow \mathcal{G}_{\mathrm{DLIN}}(1^k); \alpha, \beta, r, s, t \leftarrow \mathbb{Z}_\mathbf{p}^* : \mathcal{A}(gk, \alpha\mathcal{P}, \beta\mathcal{P}, r\alpha\mathcal{P}, s\beta\mathcal{P}, t\mathcal{P}) = 1]. \end{aligned}$$

Statements. The setup $gk = (\mathbf{p}, G, G_T, e, \mathcal{P})$ describes three $\mathbb{Z}_\mathbf{p}$-modules $\mathbb{Z}_\mathbf{p}, G$ and G_T. A statement will consist of a set of equations, which can include quadratic equations in $\mathbb{Z}_\mathbf{p}$, multi-scalar multiplication equations in G and pairing product equations. The equations are over variables $x_1, \ldots, x_m \in \mathbb{Z}_\mathbf{p}$ and $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G$.

Pairing product equations: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_\mathbf{p}, A_1 = G, A_2 = G, A_T = G_T, f(x, y) = e(x, y)$, and equations of the form $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$.

Multi-scalar multiplication in G: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_\mathbf{p}, A_1 = \mathbb{Z}_\mathbf{n}, A_2 = G, A_T = G, f(x, \mathcal{Y}) = x\mathcal{Y}$, and equations of the form $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}$.

Quadratic equations: Using our framework, this corresponds to $\mathcal{R} = \mathbb{Z}_\mathbf{p}, A_1 = \mathbb{Z}_\mathbf{p}, A_2 = \mathbb{Z}_\mathbf{p}, A_T = \mathbb{Z}_\mathbf{p}, f(x, y) = xy \bmod \mathbf{p}$, and equations of the form $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{x} = t$.

We will construct NIWI proofs for the language L that consists of statements with pairing product equation, multi-scalar multiplication equations and quadratic equations for which there is a choice of $\vec{x}, \vec{\mathcal{Y}}$ satisfying all equations simultaneously.

Commitments. We will now describe how to commit to elements in $\mathbb{Z}_{\mathbf{p}}$ and G. The commitments will belong to the $\mathbb{Z}_{\mathbf{p}}$-module $B = G^3$ formed by entry-wise addition. The commitment key is of the form

$$u_1 := (\mathcal{U}, \mathcal{O}, \mathcal{P}) = (\alpha\mathcal{P}, \mathcal{O}, \mathcal{P}) \qquad u_2 := (\mathcal{O}, \mathcal{V}, \mathcal{P}) = (\mathcal{O}, \beta\mathcal{P}, \mathcal{P}) \qquad u_3 = (\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3),$$

where $\alpha, \beta \leftarrow \mathbb{Z}_{\mathbf{p}}^*$. The vector u_3 can be chosen as either $u_3 := ru_1 + su_2$ or $u_3 := ru_1 + su_2 - (\mathcal{O}, \mathcal{O}, \mathcal{P})$ giving a binding key or a hiding key, respectively. The DLIN assumption says it is hard to tell whether three elements $r\mathcal{U}, s\mathcal{V}, t\mathcal{P}$ have the property that $t = r + s$, which implies that the two types of commitment keys are computationally indistinguishable.

For committing to $\mathcal{Y} \in G$ using randomness $(s_1, s_2, s_3) \leftarrow \mathbb{Z}_{\mathbf{p}}^3$ we define

$$\iota(\mathcal{Z}) := (\mathcal{O}, \mathcal{O}, \mathcal{Z}) \qquad p(\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3) := \mathcal{Z}_3 - \frac{1}{\alpha}\mathcal{Z}_1 - \frac{1}{\beta}\mathcal{Z}_2 \quad \text{giving us } c := \iota(\mathcal{Y}) + \sum_{i=1}^{3} s_i u_i.$$

On a binding key we have $p \circ \iota$ is the identity map and $p(u_1) = p(u_2) = p(u_3) = \mathcal{O}$ so the commitment is perfectly binding, and in fact $c = ((s_1 + rs_3)\mathcal{U}, (s_2 + ss_3)\mathcal{V}, (s_1 + s_2 + (r + s)s_3)\mathcal{P} + \mathcal{Y})$ is a linear encryption [BBS04] of $\mathcal{Y}$ with p being the decryption algorithm. On a hiding key u_1, u_2, u_3 are linearly independent so they form a basis for $B = G^3$ and therefore $\iota(G) \subseteq \langle u_1, u_2, u_3\rangle$ so the commitment scheme is perfectly hiding. The commitment scheme described here coincides with the scheme of Waters [Wat06]. We note that the different, and less efficient, commitment scheme of Groth [Gro06] can be similarly described in our language of modules.

To commit to a scalar $x \in \mathbb{Z}_{\mathbf{p}}$ we define $u := u_3 + (\mathcal{O}, \mathcal{O}, \mathcal{P})$ and using randomness $r_1, r_2 \in \mathbb{Z}_{\mathbf{p}}$ let

$$\iota'(z) := zu \qquad p'(z_1\mathcal{P}, z_2\mathcal{P}, z_3\mathcal{P}) := z_3 - \frac{1}{\alpha}z_1 - \frac{1}{\beta}z_2 \quad \text{giving us } c := xu + r_1u_1 + r_2u_2.$$

On a binding key, $p' \circ \iota'$ is the identity map on $\mathbb{Z}_{\mathbf{p}}$ and $p'(u_1) = p'(u_2) = 0$ so the commitment $c = ((r_1 + rx)\mathcal{U}, (r_2 + sx)\mathcal{V}, (r_1 + r_2 + x(r + s))\mathcal{P} + x\mathcal{P})$ is perfectly binding. On a hiding key, we have that $u = ru_1 + su_2$ so $\iota'(\mathbb{Z}_{\mathbf{p}}) \subseteq \langle u_1, u_2\rangle$ and the commitment scheme is perfectly hiding.

Common reference string. The common reference string is of the form (u_1, u_2, u_3), which implicitly defines maps ι, p, ι', p' and commitment schemes in $B = G^3$ as described above.

We use the module $B_T := G_T^9$ with addition corresponding to entry-wise multiplication. We use two different bilinear maps $F, \widetilde{F}$. The map $\widetilde{F} : G^3 \times G^3 \to G_T^9$ is defined as follows:

$$\widetilde{F} : (\begin{pmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \end{pmatrix}, \begin{pmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \\ \mathcal{Y}_3 \end{pmatrix}) \mapsto \begin{pmatrix} e(\mathcal{X}_1, \mathcal{Y}_1) & e(\mathcal{X}_1, \mathcal{Y}_2) & e(\mathcal{X}_1, \mathcal{Y}_3) \\ e(\mathcal{X}_2, \mathcal{Y}_1) & e(\mathcal{X}_2, \mathcal{Y}_2) & e(\mathcal{X}_2, \mathcal{Y}_3) \\ e(\mathcal{X}_3, \mathcal{Y}_1) & e(\mathcal{X}_3, \mathcal{Y}_2) & e(\mathcal{X}_3, \mathcal{Y}_3) \end{pmatrix}.$$

The symmetric map F is defined by

$$F(x, y) := \frac{1}{2}\widetilde{F}(x, y) + \frac{1}{2}\widetilde{F}(y, x).$$

We use the notation $\tilde{\bullet}$ and $\bullet$ when using $\tilde{F}$ and F, respectively, as the underlying bilinear maps

Pairing product equations. For pairing product equations we define

$$\iota_T(z) := \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & z \end{pmatrix}$$

$$p_T(\begin{pmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{pmatrix}) := z_{33} z_{13}^{-\frac{1}{\alpha}} z_{23}^{-\frac{1}{\beta}} \left(z_{31} z_{11}^{-\frac{1}{\alpha}} z_{21}^{-\frac{1}{\beta}} \right)^{-\frac{1}{\alpha}} \left(z_{32} z_{12}^{-\frac{1}{\alpha}} z_{22}^{-\frac{1}{\beta}} \right)^{-\frac{1}{\beta}}.$$

The map p_T corresponds to first decrypting down the columns using the decryption key α, β for the linear encryption scheme [BBS04] and then decrypting along the resulting row. We note that $p_T \circ \iota_T$ is the identity map. Both $\widetilde{F}$ and F satisfy the two commutative properties in Figure 2.

Some computation shows that the nine elements $\widetilde{F}(u_i, u_j)$ are linearly independent in the witness-indistinguishability setting. This implies that $\vec{u} \mathbin{\widetilde{\bullet}} H\vec{u}$ only has the trivial solution where H is the 3×3 matrix with 0-entries. On the other hand, the map F has non-trivial solutions to $\vec{u} \bullet H\vec{u}$ corresponding to the identities $F(u_i, u_j) = F(u_j, u_i)$. Some computation shows that the matrices

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad H_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \qquad H_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

form a basis for the matrices H such that $\vec{u} \bullet H\vec{u} = 0$. Since these matrices are fixed, we do not need to define them explicitly in the common reference string.

Multi-scalar multiplication equations. We will now look at the case of multi-scalar multiplication in G. We define

$$\widetilde{\iota}_T(\mathcal{Z}) := \widetilde{F}(\iota'(1), \iota(\mathcal{Z})) = \widetilde{F}(u, (\mathcal{O}, \mathcal{O}, \mathcal{Z})) \qquad \hat{\iota}_T(\mathcal{Z}) := F(\iota'(1), \iota(\mathcal{Z})) = F(u, (\mathcal{O}, \mathcal{O}, \mathcal{Z}))$$

$$\widetilde{p}_T(z) = \hat{p}_T(z) := e^{-1}(p_T(z)) \qquad \text{where} \qquad e^{-1}(e(\mathcal{P}, \mathcal{Z})) := \mathcal{Z}.$$

In the soundness setting $\widetilde{p}_T \circ \widetilde{\iota}_T$ and $\hat{p}_T \circ \hat{\iota}_T$ are the identity maps on G. $\widetilde{F}$ satisfies the two commutative properties, since by the linear and bilinear properties give $\widetilde{F}(\iota'(x), \iota(\mathcal{Y})) = \widetilde{F}(\iota'(1), \iota(x\mathcal{Y})) = \widetilde{\iota}_T(x\mathcal{Y})$ and $p'(x_1\mathcal{P}, x_2\mathcal{P}, x_3\mathcal{P})p(\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3) = (x_3 - \frac{1}{\alpha}x_1 - \frac{1}{\beta}x_2)(\mathcal{Y}_3 - \frac{1}{\alpha}\mathcal{Y}_1 - \frac{1}{\beta}\mathcal{Y}_2) = \widetilde{p}_T(\widetilde{F}((x_1\mathcal{P}, x_2\mathcal{P}, x_3\mathcal{P}), (\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3))$. F also satisfies the two commutative properties, since the bilinearity gives us $F(\iota'(x), \iota(\mathcal{Y})) = F(\iota'(1), \iota(x\mathcal{Y})) = \hat{\iota}_T(x\mathcal{Y})$ and $p'(x)p(y) = \frac{1}{2}p'(x)p(y) + \frac{1}{2}p'(y)p(x) = \frac{1}{2}\widetilde{p}_T(\widetilde{F}(x, y)) + \frac{1}{2}\widetilde{p}_T(\widetilde{F}(y, x)) = \hat{p}_T(F(x, y))$.

Quadratic equations. Finally, we have the case of quadratic equations in $\mathbb{Z}_{\mathbf{p}}$. We define

$$\widetilde{\iota}'_T(z) := \widetilde{F}(\iota'(1), \iota'(z)) \qquad \iota'_T(z) := F(\iota'(1), \iota'(z)) \qquad p'_T(z) := \log_{e(\mathcal{P}, \mathcal{P})}(p_T(z)).$$

On a soundness string $p'_T \circ \widetilde{\iota}'_T$ and $p'_T \circ \iota'_T$ are the identity maps on $\mathbb{Z}_{\mathbf{p}}$.

$\widetilde{F}$ satisfies the commutative properties from Figure 2, since by the linear and bilinear properties $\widetilde{F}(\iota'(x), \iota'(y)) = \widetilde{F}(\iota'(1), \iota'(xy)) = \widetilde{\iota}_T(xy)$ and $p'(x_1\mathcal{P}, x_2\mathcal{P}, x_3\mathcal{P})p'(y_1\mathcal{P}, y_2\mathcal{P}, y_3\mathcal{P}) = (x_3 - \frac{1}{\alpha}x_1 - \frac{1}{\beta}x_2)(y_3 - \frac{1}{\alpha}y_1 - \frac{1}{\beta}y_2) = p_T(\widetilde{F}((x_1\mathcal{P}, x_2\mathcal{P}, x_3\mathcal{P}), (y_1\mathcal{P}, y_2\mathcal{P}, y_3\mathcal{P}))$. F also satisfies the two commutative properties, since the bilinearity gives us $F(\iota'(x), \iota'(y)) = F(\iota'(1), \iota'(xy)) = \iota'_T(xy)$ and $p'(x)p'(y) = \frac{1}{2}p'(x)p'(y) + \frac{1}{2}p'(y)p'(x) = \frac{1}{2}p'_T(\widetilde{F}(x, y)) + \frac{1}{2}p'_T(\widetilde{F}(y, x)) = p'_T(F(x, y))$.

NIWI proof. Having described the modules, maps and matrices that are implicitly given by the common reference string above, we are now ready to give the full NIWI proof.

Setup: $gk := (\mathbf{p}, G, G_T, e, \mathcal{P}) \leftarrow \mathcal{G}_{\text{DLIN}}(1^k)$.

Soundness string: On input gk return $\sigma := (u_1, u_2, u_3)$, where $u_1 = (\alpha\mathcal{P}, \mathcal{O}, \mathcal{P}), u_2 = (\mathcal{O}, \beta\mathcal{P}, \mathcal{P}), u_3 = ru_1 + su_2$ for random $\alpha, \beta \leftarrow \mathbb{Z}_{\mathbf{p}}^*$ and $r, s \leftarrow \mathbb{Z}_{\mathbf{p}}$.

Witness-indistinguishability string: On input gk return $\sigma := (u_1, u_2, u_3)$, where $u_1 = (\alpha\mathcal{P}, \mathcal{O}, \mathcal{P}), u_2 = (\mathcal{O}, \beta\mathcal{P}, \mathcal{P}), u_3 = ru_1 + su_2 - (\mathcal{O}, \mathcal{O}, \mathcal{P})$ for random $\alpha, \beta \leftarrow \mathbb{Z}_{\mathbf{p}}^*$ and $r, s \leftarrow \mathbb{Z}_{\mathbf{p}}$.

Prover: For notational convenience let $\vec{v} = (u_1, u_2)$. On input gk, σ, a set of equations and a witness $\vec{x}, \vec{\mathcal{Y}}$ do:

1. Commit to the scalars $\vec{x} \in \mathbb{Z}_{\mathbf{p}}^m$ and the group elements $\vec{\mathcal{Y}} \in G^n$ as

$$\vec{c} := \iota'(\vec{x}) + R\vec{v} \qquad\qquad \vec{d} := \iota(\vec{\mathcal{Y}}) + S\vec{u}$$

   for randomly chosen $R \leftarrow \text{Mat}_{m\times 2}(\mathbb{Z}_{\mathbf{p}}), S \leftarrow \text{Mat}_{n\times 3}(\mathbb{Z}_{\mathbf{p}})$.

2. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ make a proof as described in Section 6.3 using the symmetric map F and random $r_1, r_2, r_3 \leftarrow \mathbb{Z}_{\mathbf{p}}$.

$$\vec{\phi} := S^\top \iota(\vec{\mathcal{A}}) + S^\top(\Gamma + \Gamma^\top)\iota(\vec{\mathcal{Y}}) + S^\top \Gamma S\vec{u} + \sum_{i=1}^{3} r_i H_i \vec{u}.$$

   For each linear equation $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = t_T$ we use the asymmetric map $\widetilde{F}$ following Section 6.2 to get the proof

$$\vec{\pi} = \vec{0} \qquad\qquad \vec{\theta} := S^\top \iota(\vec{\mathcal{A}}).$$

   The reason we use the asymmetric $\widetilde{F}$ for the linear equation is that there are no non-trivial matrices H such that $\vec{u} \,\widetilde{\bullet}\, H\vec{u} = 0$, which simplifies the proof. Observe that $\vec{\theta} = \iota(S^\top \vec{\mathcal{A}}) = S^\top \iota(\vec{\mathcal{A}})$ and, conversely, $p(\vec{\theta}) = S^\top \vec{\mathcal{A}}$ is easily computable in this special setting, since $\iota(\mathcal{A}) = (\mathcal{O}, \mathcal{O}, \mathcal{A})$. We can therefore just reveal the proof $\vec{\phi} := p(\vec{\theta}) = S^\top \vec{\mathcal{A}}$, which consists of only three group elements.

3. For each multi-scalar multiplication equation $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}_2$ we use the symmetric map F and as in Section 6.3 let R' be R with an appended 0-row. The proof is for random $r_1, r_2, r_3 \leftarrow \mathbb{Z}_{\mathbf{p}}$:

$$\vec{\phi} := (R')^\top \iota(\vec{\mathcal{B}}) + (R')^\top \Gamma \iota(\vec{\mathcal{Y}}) + S^\top \iota'(\vec{a}) + S^\top \Gamma^\top \iota'(\vec{x}) + (R')^\top \Gamma S\vec{u} + \sum_{i=1}^{3} r_i H_i \vec{u}.$$

   For each linear equation $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{T}$ we use the asymmetric map $\widetilde{F}$ to get the proof

$$\vec{\pi} = \vec{0} \qquad\qquad \vec{\theta} := S^\top \iota'(\vec{a}).$$

   It suffices to reveal the value $\vec{\phi} = S^\top \vec{a}$. Since $\vec{\theta}$ determines $\vec{\phi}$ uniquely, this does not compromise the perfect witness-indistinguishability we have on witness-indistinguishability strings. The verifier can compute $\vec{\theta} = \iota'(\vec{\phi})$. The proof now consists of only 3 elements in $\mathbb{Z}_{\mathbf{p}}$.

For each linear equation $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}$ we use $\widetilde{F}$ to get the proof

$$\vec{\pi} := R^\top \iota(\vec{\mathcal{B}}) \qquad\qquad \vec{\theta} = \vec{0}.$$

We can use $\vec{\phi} = R^\top \vec{\mathcal{B}}$ as the proof, since it allows the verifier to compute $\vec{\pi} = \iota(\vec{\phi})$. The proof therefore consists of only 2 group elements.

4. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{x} = t$ in $\mathbb{Z}_{\mathbf{p}}$ we use the symmetric map F. The matrix $H_1' = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ generates all H such that $\vec{v} \bullet H\vec{v}$. The proof is for random $r_1 \leftarrow \mathbb{Z}_{\mathbf{p}}$:

$$\vec{\phi} := R^\top \iota'(\vec{b}) + R^\top(\Gamma + \Gamma^\top)\iota'(x) + R^\top \Gamma R \vec{v} + r_1 H_1 \vec{v}.$$

For each linear equation $\vec{x} \cdot \vec{b} = t$ we use the asymmetric map $\widetilde{F}$ to get the proof $\vec{\pi} := R^\top \iota'(\vec{b})$. It suffices to reveal just $\vec{\phi} = R^\top \vec{b}$, from which the verifier can compute $\vec{\pi} = \iota'(\vec{\phi})$.

Verifier: On input (gk, σ), a set of equations and a proof $\vec{c}, \vec{d}, \{\vec{\phi}_i\}_{i=1}^N$ do:

1. For each pairing product equation $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma \vec{\mathcal{Y}}) = t_T$ with proof $\vec{\phi}$ check that

$$\iota(\vec{\mathcal{A}}) \bullet \vec{d} + \vec{d} \bullet \Gamma \vec{d} = \iota_T(t_T) + \vec{u} \bullet \vec{\phi}.$$

For each linear equation $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = t_T$ with proof $\vec{\phi}$ check

$$\iota(\vec{\mathcal{A}}) \mathbin{\widetilde{\bullet}} \vec{d} = \iota_T(t_T) + \iota(\vec{\phi}) \mathbin{\widetilde{\bullet}} \vec{u}\ .$$

2. For each multi-scalar multiplication $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma \vec{\mathcal{Y}} = \mathcal{T}$ with proof $\vec{\phi}$ check that

$$\iota'(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota(\vec{\mathcal{B}}) + \vec{c} \bullet \Gamma \vec{d} = \hat{\iota}_T(\mathcal{T}) + \vec{u} \bullet \vec{\phi}.$$

For each linear equation $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{T}$ with proof $\vec{\phi}$ check

$$\iota'(\vec{a}) \mathbin{\widetilde{\bullet}} \vec{d} = \hat{\iota}_T(\mathcal{T}) + \iota'(\vec{\phi}) \mathbin{\widetilde{\bullet}} \vec{u}.$$

For each linear equation $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}$ with proof $\vec{\phi}$ check

$$\vec{c} \mathbin{\widetilde{\bullet}} \iota(\vec{\mathcal{B}}) = \hat{\iota}_T(\mathcal{T}) + \vec{v} \mathbin{\widetilde{\bullet}} \iota(\vec{\phi}).$$

3. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{x} = t$ in $\mathbb{Z}_{\mathbf{p}}$ with proof $\vec{\phi}$ check that

$$\vec{c} \bullet \iota'(\vec{b}) + \vec{c} \bullet \Gamma \vec{c} = \iota_T'(t) + \vec{v} \bullet \vec{\phi}.$$

For each linear equation $\vec{x} \cdot \vec{b} = t$ with proof $\vec{\phi}$ check

$$\vec{c} \mathbin{\widetilde{\bullet}} \iota'(\vec{b}) = \iota_T'(t) + \vec{v} \mathbin{\widetilde{\bullet}} \iota'(\vec{\phi}).$$

Theorem 17 *The protocol is a NIWI proof with perfect completeness, perfect soundness and composable witness-indistinguishability for satisfiability of a set of equations over a bilinear group where the DLIN problem is hard.*

Proof. Perfect completeness follows from Theorem 6. Perfect soundness follows from Theorems 7 and 10 since the $p \circ \iota$ maps are identity maps on $\mathbb{Z}_{\mathbf{p}}, G$ and G_T. The DLIN assumption gives us that the two types of common reference strings are computationally indistinguishable. On a witness-indistinguishability string, the commitments are perfectly hiding and we get perfect witness-indistinguishability from Theorems 8 and 9. □

Assumption: DLIN	G	$\mathbb{Z}_\mathbf{p}$
Variables $x \in \mathbb{Z}_\mathbf{p}, \mathcal{Y} \in G$	3	0
Pairing product equations	9	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = t_T$	3	0
Multi-scalar multiplication equations	9	0
- Linear equation: $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{T}$	0	3
- Linear equation: $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}$	2	0
Quadratic equations in $\mathbb{Z}_\mathbf{p}$	6	0
- Linear equation: $\vec{x} \cdot \vec{b} = t$	0	2

Table 4: Cost of each variable and equation measured in elements from $\mathbb{Z}_\mathbf{p}$ and G.

Size. The module we work in is $B = G^3$, so each element in the module consists of three group elements from G. In some of the linear equations, we can compute $p(\vec{\phi})$ efficiently and we have $\iota(p(\vec{\phi})) = \vec{\phi}$, which gives us shorter proofs. Figure 4 list the cost of all the different types of equations.

11 Zero-knowledge

We will now show that in many cases it is possible to make zero-knowledge proofs for satisfiability of quadratic equations. An obvious strategy is to use our NIWI proofs directly, however, one could imagine such proofs might not be zero-knowledge because the zero-knowledge simulator might not be able to compute any witness for satisfiability of the equations. It turns out that the strategy is better than it seems at first sight though; we will often be able to modify the set of quadratic equations into an equivalent set of quadratic equations where a witness can be found and which has the same distribution of proofs.

We will consider the case where $A_1 = \mathcal{R}, A_2 = A_T, f(r, y) = ry$. We remark that it is quite common to have $\mathcal{A}_1 = \mathcal{R}$; in bilinear groups both multi-scalar multiplication equations in G_1, G_2 and quadratic equations in $\mathbb{Z}_\mathbf{n}$ have this structure.

The first stage of the simulator S_1 will output a witness-indistinguishability string and a simulation trapdoor τ that makes it possible to trapdoor open the commitments in B_1. More precisely, $\tau = \vec{s} \in \mathcal{R}^{\hat{m}}$ such that $\iota_1(1) = \iota_1(0) + \vec{s}^\top \vec{u}$. Define $c := \iota_1(1)$, which is a commitment to $\delta = 1$ with trivial randomness. The idea in the simulation is that we can rewrite the statement as

$$\vec{a}_i \cdot \vec{y} + f(-\delta, t_i) + \vec{x} \cdot \vec{b}_i + \vec{x} \cdot \Gamma \vec{y} = 0.$$

We have introduced a new variable δ and by choosing all variables to be 0 gives a satisfying witness. In the simulation, the simulator S_2 will use the trapdoor information τ to open c to 0 and it can now use the NIWI proof from Section 7.

We are now ready to give the NIZK proof for the language L consisting of statements with quadratic equations that are simultaneously satisfiable defined in Section 6. These are statements consisting of one or more equations of the form $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$ such that there is some choice of $\vec{x}, \vec{y}$ that satisfy all equations. The witness for membership of L is $w = (\vec{x}, \vec{y})$. The NIZK proof will have perfect L_{guilt}-soundness as defined in Section 6. When $L_{\text{guilt}} = \bar{L}$ this corresponds to standard perfect soundness.

Setup: $(gk, sk) = ((\mathcal{R}, A_1, A_2, A_T, f), sk) \leftarrow \mathcal{G}(1^k)$, where $A_1 = \mathcal{R}$ and $A_2 = A_T$.

Soundness string: $\sigma = (B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}, H_1, \ldots, H_\eta) \leftarrow K(gk, sk)$.

Prover: This protocol is exactly the same as in the NIWI proof, we do not even need to rewrite the equations. The input consists of gk, σ, a list of quadratic equations $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and a satisfying witness $\vec{x}, \vec{y}$.

Pick at random $R \leftarrow \mathrm{Mat}_{m\times\hat{m}}(\mathcal{R})$ and $S \leftarrow \mathrm{Mat}_{n\times\hat{n}}(\mathcal{R})$ and commit to all the variables as $\vec{c} := \iota_1(\vec{x}) + R\vec{u}$ and $\vec{d} := \iota_2(\vec{y}) + S\vec{v}$.

For each equation $(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)$ make a proof as described in Section 6. In other words, pick $T_i \leftarrow \mathrm{Mat}_{\hat{n}\times\hat{m}}(\mathcal{R})$ and $r_{i1}, \ldots, r_{i\eta} \leftarrow \mathcal{R}$ and compute

$$\begin{aligned} \vec{\pi}_i &:= R^\top \iota_2(\vec{b}_i) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S\vec{v} - T_i^\top \vec{v} + \sum_{j=1}^{\eta} r_{ij} H_j \vec{v} \\ \vec{\theta}_i &:= S^\top \iota_1(\vec{a}_i) + S^\top \Gamma^\top \iota_1(\vec{x}) + T_i \vec{u}. \end{aligned}$$

Output the proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N)$.

Verifier: The input is gk, σ, $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and the proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\theta}_i)\})$.

For each equation check

$$\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}_i) + \vec{c} \bullet \Gamma_i \vec{d} = \iota_T(t_i) + \vec{u} \bullet \vec{\pi}_i + \vec{\theta}_i \bullet \vec{v}.$$

Output 1 if all the checks pass, else output 0.

Simulation string: $(\sigma, \tau) = ((B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}, H_1, \ldots, H_\eta), \vec{s}) \leftarrow S_1(gk, sk)$, where $\iota_1(1) = \iota_1(0) + \vec{s}^\top \vec{u}$.

Proof simulator: The input consists of gk, σ and a list of quadratic equations $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and the simulation trapdoor $\tau = \vec{s}$.

Rewrite each equation as $\vec{a}_i \cdot \vec{y} + \vec{x} \cdot \vec{b}_i + f(\delta, -t_i) + \vec{x} \cdot \Gamma_i \vec{y} = 0$. Define $\vec{x} := \vec{0}, \vec{y} := \vec{0}$ and $\delta = 0$ to get a witness that satisfies all the modified equations.

Pick at random $R \leftarrow \mathrm{Mat}_{m\times\hat{m}}(\mathcal{R})$ and $S \leftarrow \mathrm{Mat}_{n\times\hat{n}}(\mathcal{R})$ and commit to all the variables as $\vec{c} := \vec{0} + R\vec{u}$ and $\vec{d} := \vec{0} + S\vec{v}$. We also use $c := \iota_1(1) = \iota_1(0) + \vec{s}^\top \vec{u}$ and append it to $\vec{c}$.

For each modified equation $(\vec{a}_i, \vec{b}_i, -t_i, \Gamma_i, 0)$ make a proof as described in Section 6. Return the simulated proof $\{(\vec{c}, \vec{d}, \vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N$.

Theorem 18 *The protocol described above is a composable NIZK proof for satisfiability of quadratic equations with perfect completeness, perfect L_{guilt}-soundness and composable zero-knowledge.*

Proof. Perfect completeness on a soundness string follows from the perfect completeness of the NIWI proof: The simulator knows an opening of $c := \iota_1(1)$ to $c = \iota_1(0) + \sum_{i=1}^{\hat{m}} s_i u_i$. It therefore knows a witness $\vec{0}, \vec{0}, \delta = 0$ for satisfiability of all the modified equations. It therefore outputs a proof $\{(\vec{c}, \vec{d}, \vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N$ such that for all i we have

$$\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}_i) + F(\iota_1(1), -\iota_2(t_i)) + \vec{c} \bullet \Gamma_i \vec{d} = \iota_T(0) + \vec{u} \bullet \vec{\pi}_i + \vec{\theta}_i \bullet \vec{v}.$$

The commutative property of the maps gives us $F(\iota_1(1), \iota_2(t_i)) = \iota_T(f(1, t_i)) = \iota_T(t_i)$, so the NIZK proofs satisfy the equations the verifier checks. Perfect completeness on a simulation string now follows from the perfect completeness of the NIWI proof as well.

Perfect L_{guilt}-soundness follows from the perfect L_{guilt}-soundness of the NIWI proof.

We will now show that on a simulation string we have perfect zero-knowledge. The commitments $\vec{c}, \vec{d}$ and $c = \iota_1(1)$ are perfectly hiding and therefore have the same distribution whether we use witness $\vec{x}, \vec{y}, \delta = 1$ or $\vec{0}, \vec{0}, \delta = 0$. Theorem 8 now tells us that the proofs $\vec{\pi}_i, \vec{\theta}_i$ made with either type of opening of $\vec{c}, \vec{d}, c$ are uniformly distributed over all possible choices of $\{(\vec{\pi}_i, \vec{\theta}_i)\}_{i=1}^N$ that satisfy the equations $\iota_1(\vec{a}_i) \bullet \vec{d} + \vec{c} \bullet \vec{b}_i + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t)$. We therefore have perfect zero-knowledge on a simulation string. □

Since the NIZK proof is exactly the same as the NIWI proof, there is no additional cost associated with getting composable zero-knowledge for full quadratic equations. If we look at linear equations, there are two cases to consider. On a linear equation of the form $\vec{x} \cdot \vec{b} = t$, the simulator can rewrite it as $\vec{x} \cdot \vec{b} + f(-\delta, t) = 0$, which is a linear equation of the same form. The shorter NIWI proofs for this type of linear equations can therefore also be perfectly simulated on a simulation string. NIWI proofs for linear equations of the form $\vec{a} \cdot \vec{y} = t$ on the other hand cannot be simulated as easily, because if the simulator rewrites the equation as $\vec{a} \cdot \vec{y} + (-\delta, t) = 0$, then it is no longer a linear equation. To get composable zero-knowledge for the latter type of linear equation, the prover can instead use the NIWI proof for the full quadratic equation.

11.1 NIZK proofs for bilinear groups

Let us now consider bilinear groups and the four types of quadratic equations given in Figure 1. If we set up the common reference string such that we can trapdoor open both $\iota_1'(1)$ and $\iota_2'(1)$ to 0 then multi-scalar multiplication equations and quadratic equations in $\mathbb{Z}_{\mathbf{n}}$ are of the form for which we can get a perfect simulation.

In the case of pairing product equations we do not know how to get zero-knowledge, since even with the trapdoors we may not be able to compute a witness. We do observe though that in the special case, where all $t_T = 1$ the choice of $\vec{\mathcal{X}} = \vec{\mathcal{O}}, \vec{\mathcal{Y}} = \vec{\mathcal{O}}$ is a satisfactory witness. Since we also use the witness $\vec{\mathcal{X}} = \vec{\mathcal{O}}, \vec{\mathcal{Y}} = \vec{\mathcal{O}}$ in the other types of equations, the simulator can use this witness in the simulation. In the special case where all $t_T = 1$ we can therefore make NIZK proofs for satisfiability of a set of quadratic equations.

In another special case where we have a pairing product equation with $t_T = \prod_{i=1}^n e(\mathcal{P}_i, \mathcal{Q}_i)$ for some known $\mathcal{P}_i, \mathcal{Q}_i$ there is another technique that can be useful to get zero-knowledge. In this case, we can add the equations $\delta \mathcal{Z}_i - \delta \mathcal{Q}_i = \mathcal{O}$ to the set of multi-scalar multiplication equations in G_2 and rewrite the pairing product equation as $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{P}} \cdot \vec{\mathcal{Z}})(\vec{\mathcal{X}} \cdot \Gamma \vec{\mathcal{Y}}) = 1$. This gives us pairing product equations of the type where we can make zero-knowledge proofs. We can therefore also make zero-knowledge proofs for a set of quadratic equations over a bilinear group if all the pairing product equations have t_T of the form $t_T = \prod_{i=1}^n e(\mathcal{P}_i, \mathcal{Q}_i)$ for some known $\mathcal{P}_i, \mathcal{Q}_i$.

The case of pairing product equations points to a couple of differences between witness-indistinguishable proofs and zero-knowledge proofs using our techniques. NIWI proofs can handle any target t_T, whereas zero-knowledge proofs can only handle special types of target t_T. Furthermore, if $t_T \neq 1$ the size of the NIWI proof for this equation is constant, whereas the NIZK proof for the same equation may be larger.

We conclude our discussion of NIZK proofs with Figure 3 and Figure 4 that give the costs for proving the satisfiability of a set of quadratic equations in the SXDH and DLIN instantiations. For the subgroup decision instantiation, NIZK proofs for sets of quadratic equations where all $t_T = 1$ are the same as those given in Figure 1.

Assumption: SXDH	G_1	G_2	$\mathbb{Z}_\mathbf{p}$
Variables $x \in \mathbb{Z}_\mathbf{p}, \mathcal{X} \in G_1$	2	0	0
Variables $y \in \mathbb{Z}_\mathbf{p}, \mathcal{Y} \in G_2$	0	2	0
Pairing product equations with $t_T = 1$	4	4	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = 1$	2	0	0
- Linear equation: $\vec{\mathcal{X}} \cdot \vec{\mathcal{B}} = 1$	0	2	0
Multi-scalar multiplication equations in G_1	2	4	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{y} = \mathcal{T}_1$	1	0	0
- Linear equation: $\vec{\mathcal{X}} \cdot \vec{b} = \mathcal{O}$	0	0	2
Multi-scalar multiplication equations in G_2	4	2	0
- Linear equation: $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{O}$	0	0	2
- Linear equation: $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}_2$	0	1	0
Quadratic equations in $\mathbb{Z}_\mathbf{p}$	2	2	0
- Linear equation: $\vec{a} \cdot \vec{y} = t$	0	0	1
- Linear equation: $\vec{x} \cdot \vec{b} = t$	0	0	1

Figure 3: Cost of each variable and equation in an NIZK proof in the SXDH instantiation.

Assumption: DLIN	G	$\mathbb{Z}_\mathbf{p}$
Variables $x \in \mathbb{Z}_\mathbf{p}, \mathcal{Y} \in G$	3	0
Pairing product equations with $t_T = 1$	9	0
- Linear equation: $\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}} = 1$	3	0
Multi-scalar multiplication equations	9	0
- Linear equation: $\vec{a} \cdot \vec{\mathcal{Y}} = \mathcal{O}$	0	3
- Linear equation: $\vec{x} \cdot \vec{\mathcal{B}} = \mathcal{T}$	2	0
Quadratic equations in $\mathbb{Z}_\mathbf{p}$	6	0
- Linear equation: $\vec{x} \cdot \vec{b} = t$	0	2

Figure 4: Cost of each variable and equation in an NIZK proof in the DLIN instantiation.

12 Conclusion and an open problem

Our main contribution in this paper is the construction of efficient non-interactive cryptographic proofs for use in bilinear groups. Our proofs can be instantiated with many different types of bilinear groups and the security of our proofs can be based on many different types of intractability assumptions. We have given three concrete examples of instantiations based on the subgroup decision assumption, the SXDH assumption, and the DLIN assumption, respectively.

Because of their interest for applications, we have focused on bilinear groups in our instantiations. However, our techniques generalize beyond bilinear groups; for instance we do not require the modules to be cyclic (as is the case for bilinear groups). It is possible that other types of modules with a bilinear map exist, which are not constructed from bilinear groups. The existence of such modules might lead to efficient NIWI and NIZK proofs based on entirely different intractability assumptions. We leave the construction of such modules with a bilinear map as an interesting open problem.

Acknowledgements

We gratefully acknowledge Brent Waters for a number of helpful ideas, comments, and conversations related to this work. In particular, our module-based approach can be seen as formalizing part of the intuition expressed by Waters that the Decisional Linear Assumption, Subgroup Decision

Assumption in composite-order groups, and SXDH can typically be exchanged for one another. (We were inspired by previously such connections made by [GOS06a, Wat06].) We thank Dan Boneh for his encouragement and for suggesting using our techniques to get fair exchange. We also thank Ghadafi, Smart, and Warinschi [GSW10] for their helpful feedback regarding earlier online versions of this paper, observing and correcting some errors in the instantiations based on the SXDH and DLIN assumptions. Likewise we thank Keita Xagawa [Xag15] for bringing an error in Section 6.3 and in the DLIN instantiation to our attention (this error was introduced in earlier revisions of our paper, and has been fixed now) and for suggesting additional randomization of the multi-scalar multiplication equation proofs as a solution.

References

[Bar06] Paulo Barreto. The pairing-based crypto lounge, 2006.

[BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459, 2004.

[BB11] Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, 2011.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, 2004.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signature and non-interactive anonymous credentials. In *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374, 2008.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

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

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, 2005.

[Bon06] Dan Boneh. Personal communication, 2006.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 573–592, 2006.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444, 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15, 2007.

[CGS07] Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434, 2007.

[Dam92] Ivan Damgård. Non-interactive circuit based proofs and non-interactive perfect zero-knowledge with preprocessing. In *EUROCRYPT*, volume 658 of *Lecture Notes in Computer Science*, pages 341–355, 1992.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DDP02] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Randomness-optimal characterization of two NP proof systems. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 179–193, 2002.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[GH08] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 179–197, 2008.

[GL07] Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 51–67, 2007.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111, 2006.

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero-knowledge for NP. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358, 2006.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS*, pages 89–98, 2006.

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, volume 4248 of *Lecture Notes in Computer Science*, pages 444–459, 2006.

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 164–180, 2007.

[GSW10] Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Groth-sahai proofs revisited. In *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 177–192, 2010.

[KP98] Joe Kilian and Erez Petrank. An efficient noninteractive zero-knowledge proof system for NP with general assumptions. *Journal of Cryptology*, 11(1):1–27, 1998.

[Mic03] Silvio Micali. Simple and fast optimistic protocols for fair electronic exchange. In *PODC*, pages 12–19, 2003.

[Pat05] Kenneth G. Paterson. Cryptography from pairings. In I.F. Blake, G. Seroussi, and N.P. Smart, editors, *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*, pages 215–251. Cambridge University Press, 2005.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, 2005.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, 2005.

[Wat06] Brent Waters. New techniques for slightly 2-homomorphic encryption, 2006. Manuscript.

[Xag15] Keita Xagawa. Groth-sahai proofs, revisited, revisited, 2015.

A Quick reference to notation

Bilinear groups

G_1, G_2, G_T: cyclic groups with bilinear map $e : G_1 \times G_2 \to G_T$.

$\mathcal{P}_1, \mathcal{P}_2$: generators of respectively G_1 and G_2.

Group order: prime order $\mathbf{p}$ or composite order $\mathbf{n}$.

Modules with bilinear map

$\mathcal{R}$: finite commutative ring $(\mathcal{R}, +, \cdot, 0, 1)$.

$A_1, A_2, A_T, B_1, B_2, B_T$: $\mathcal{R}$-modules.

f, F: bilinear maps $f : A_1 \times A_2 \to A_T$ and $F : B_1 \times B_2 \to B_T$.

$$\vec{x} \cdot \vec{y} := \sum_{i=1}^{n} f(x_i, y_i) \qquad , \qquad \vec{x} \bullet \vec{y} := \sum_{i=1}^{n} F(x_i, y_i).$$

Properties that follows from bilinearity:

$$\vec{x} \cdot M\vec{y} = M^\top \vec{x} \cdot \vec{y} \qquad , \qquad \vec{x} \bullet M\vec{y} = M^\top \vec{x} \bullet \vec{y}.$$

Commutative diagram of maps in setup

$$\begin{tikzcd}
A_1 \arrow[d, shift right, "\iota_1"'] & \times & A_2 \arrow[r, "f"] \arrow[d, shift right, "\iota_2"'] & A_T \arrow[d, shift right, "\iota_T"'] \\
B_1 \arrow[u, shift right, "p_1"'] & \times & B_2 \arrow[r, "F"] \arrow[u, shift right, "p_2"'] & B_T \arrow[u, shift right, "p_T"']
\end{tikzcd}$$

Commutative properties:

$$F(\iota_1(x), \iota_2(y)) = \iota_T(f(x,y)) \qquad , \qquad f(p_1(x), p_2(y)) = p_T(F(x,y)).$$

Equations

(Secret) variables: $\vec{x} \in A_1^m, \vec{y} \in A_2^n$.

(Public) constants: $\vec{a} \in A_1^n, \vec{b} \in A_2^m, \Gamma \in \mathrm{Mat}_{m \times n}(\mathcal{R}), t \in A_T$.

Equations: $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$.

Commitments

Commitment keys: $\vec{u} \in B_1^{\hat{m}}, \vec{v} \in B_2^{\hat{n}}$.

Commitments:

$$\vec{c} := \iota_1(\vec{x}) + R\vec{u} \in B_1^m \qquad , \qquad \vec{d} := \iota_2(\vec{y}) + S\vec{v} \in B_2^n.$$

NIWI proofs

Additional setup information: $H_1, \ldots, H_\eta$ such that $\vec{u} \bullet H_i \vec{v} = 0$.

Randomness in proofs: $T \leftarrow \mathrm{Mat}_{\hat{n} \times \hat{m}}(\mathcal{R})$, $r_1, \ldots, r_\eta \leftarrow \mathcal{R}$.

Proofs:

$$\begin{aligned}
\vec{\pi} &:= R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + R^\top \Gamma S \vec{v} - T^\top \vec{v} + \sum_{i=1}^{\eta} r_i H_i \vec{v} \\
\vec{\theta} &:= S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x}) + T \vec{u}
\end{aligned}$$

Verification: $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma \vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\theta} \bullet \vec{v}$.