(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 October 2008 (23.10.2008)

PCT

(10) International Publication Number
WO 2008/127428 A2

(51) International Patent Classification:
G06F 21/24 (2006.01)

(21) International Application Number:
PCT/US2007/085018

(22) International Filing Date:
16 November 2007 (16.11.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/859,875 17 November 2006 (17.11.2006) US

(71) Applicant *(for all designated States except US)*: **THE REGENTS OF THE UNIVERSITY OF CALIFORNIA** [US/US]; 1111 Franklin Street, 5th Floor, Oakland, CA 94607-5200 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **GROTH, Jens** [DK/DK]; Ildervej 15A, DK-8270 Hoejbjerg (DK). **SAHAI, Amit** [US/US]; P.O. Box 241631, Los Angeles, CA 90024 (US).

(74) Agent: **DELANEY, Karoline, A.**; Knobbe Martens Olson & Bear, LLP, 2040 Main Street, Fourteenth Floor, Irvine, CA 92614 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL,

[Continued on next page]

(54) **Title:** EFFICIENT NON-INTERACTIVE PROOF SYSTEMS FOR BILINEAR GROUPS

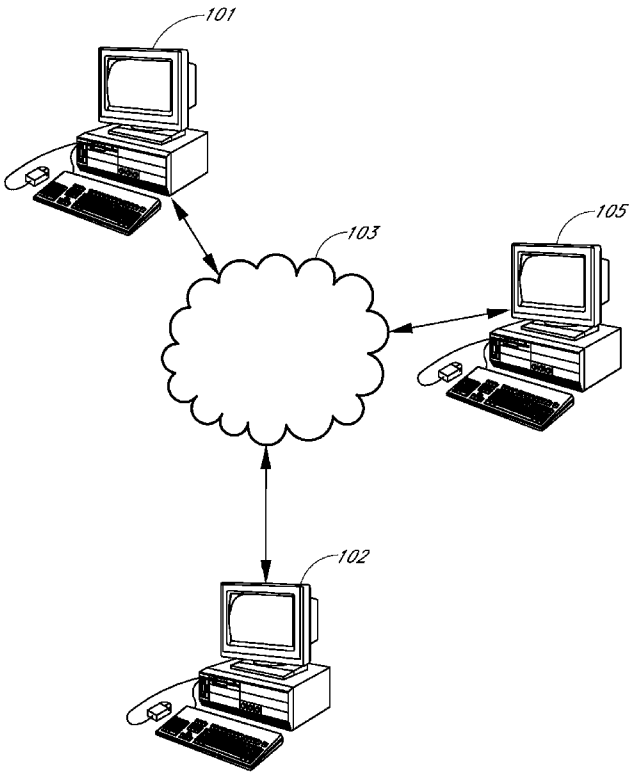


FIG. 1

(57) **Abstract:** An apparatus and method for constructing efficient non-interactive zero-knowledge proofs and non-interactive witness-indistinguishable proofs that work directly for groups with a bilinear map is described. 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 disclosure provides n on -interactive witness-indistinguishable proofs and non-interactive zero-knowledge proofs that can be used in connection with these protocols.

WO 2008/127428 A2

PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

UCLARF.009VPC / UCLA 2007-252-1

EFFICIENT NON-INTERACTIVE PROOF SYSTEMS FOR BILINEAR GROUPS

Reference to Related Applications

[0001] The present application claims priority from U.S. Provisional Application No. 60/859,875, filed November 17, 2006, titled "METHOD AND APPARATUS FOR EFFICIENT VERIFICATION OF ENCRYPTED DATA," the entire contents of which is hereby incorporated by reference

Government Interest Statement

[0002] This invention was made with Government support of Grant No. CNS0456717 awarded by the NSF. The Government has certain rights in this invention.

Background

Field of the Invention

[0003] The technical field generally relates to cryptographic systems and specifically relates to non-interactive zero-knowledge proofs.

Description of the Related Art

[0004] Non-interactive zero-knowledge (NIZK) proofs allow a prover to create a proof of membership of an NP language. The proof can be used to convince another that a statement in question belongs to the language, but the zero-knowledge property ensures that the proof will reveal nothing but the truth (or falsity) of the statement. NIZK proofs are fundamental cryptographic primitives used in many constructions, including CCA2-secure cryptosystems, digital signatures, and various cryptographic protocols. Blum, Feldman, and Micali, in Non-interactive zero-knowledge and its applications in the proceedings of STOC '88, pp. 103-112, 1988, introduced the notion of NIZK in the common random string model and showed how to construct computational NIZK proof systems for proving a single statement about any NP language. The fist computational NIZK proof system for multiple theorems was constructed by Blum, De Santis, Micali, and Persiano in Noninteractive zero-knowledge in SIAM Journal of Computation, 20(6), pp.1084-1118, 1991. Both papers based their NIZK systems on certain number-theoretic assumptions (specifically, the hardness of deciding quadratic residues modulo a composite number). Feige, Lapidot, and Shamir in Multiple non-interactive zero

knowledge proofs under general assumptions in SIAM Journal of Computing, 29(1), pp. 1-28, 1999, showed how to construct computational NIZK proofs based on a trapdoor permutation. Much research has been devoted to the construction of efficient NIZK proofs, but until now the only known method to do so has been the hidden random bits method wherein the prover has a string of random bits, which are secret to the verifier. By revealing a subset of these bits, and keeping the rest secret, the prover can convince the verifier of the truth of the statement in question.

[0005] Unfortunately, these prior 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, they have therefore not had any impact in practice.

[0006] 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 so $c = ab$". An NP-reduction of even very simple statements like these requires large circuits containing thousands of gates and the corresponding NIZK proofs become very large.

Summary

[0007] These and other problems are solved by a system for efficient non-interactive proof for bilinear groups. In one embodiment, commitment schemes are homomorphic and equipped with a bilinear map. The variables in the equations to be proved are replace with commitments to those variables. Since the commitment schemes are hiding, the equations will no longer be valid. However, we can extract out the additional terms introduced by the randomness of the commitments: An additional term is introduced by substituting the commitments. Because the additional term is a value which makes the equation true, giving it away preserves witness indistinguishability. If there are many terms, that means that these terms are not unique, and we can randomize these terms so that the equation is still true, but so that we effectively reduce to the case of there being a single term being given away with a unique value.

[0008] In one embodiment, the proof system is used for fair key exchange.

[009] In one embodiment, the proof system is used in a mix-net.

[0010] In one embodiment, the proof system is used for verifiable encryption.

Brief Description of the Figures

[0011] Figure 1 shows a first computer and a second computer provided to a computer network for exchanging encrypted data.

[0012] Figure 2 is a flow diagram of key generation in a system for verifiable encryption.

[0013] Figure 3 is a flow diagram of encryption in the system of Figure 2.

[0014] Figure 4 is a flow diagram of generation of a verification proof of membership in the system of Figure 2.

[0015] Figure 5 is a flow diagram of decryption in the system of Figure 2.

[0016] Figure 6 shows a mix-net system wherein a plurality of senders and a plurality of mix-net servers are provided to a network.

[0017] Figure 7 is a flow diagram of key generation in the system of Figure 6.

[0018] Figure 8 is a flow diagram of encryption in the system of Figure 6.

[0019] Figure 9 is a flow diagram of re-randomization in the system of Figure 6.

[0020] Figure 10 is a flow diagram of an NIZK proof of membership in the system of Figure 6.

[0021] Figure 11 is a flow diagram showing decryption in the system of Figure 6.

Description

1 Introduction

[0022] In the following disclosure, for notational convenience we will follow the tradition of mathematics and use additive notation. (Note: 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 the present setting, however, it is more convenient to use multiplicative notation to refer to the action of the bilinear map 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 $(\mathrm{n}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$ where

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

- $e : G_1 \times G_2$ is a non-degenerate bilinear map so $e(\mathcal{P}_1, \mathcal{P}_2)$ generates G_T and for all $a, b \in \mathbb{Z}_n$ 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.

[0022] In this disclosure, we develop a general set of highly efficient techniques for proving statements involving bilinear groups. First, we formulate the constructions in terms of modules over commutative rings with an associated bilinear map. This framework captures bilinear groups with cryptographic significance – for both supersingular and ordinary elliptic curves, for groups of both prime and composite order. Second, we consider 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 exponents to be "unknowns" in the statements to be proven.

[0023] With the level of generality herein, for example it would 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 the framework herein using standard techniques in arithmetization.

[0024] 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 transform these into zero-knowledge proof systems. We also provide a detailed examination of the efficiency of the constructions herein in various settings (depending on what type of bilinear group is used).

[0025] The security of constructions arising from the framework herein can be based on any of a variety of computational assumptions about bilinear groups (3 of which we discuss in detail here). Thus, the techniques herein do not rely on any one assumption in particular.

[0026] Note that while 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 so $c = ab$" 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. In the disclosure, we will therefore construct NIWI and NIZK proofs for group-dependent languages.

[0027] 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 [DBS04] 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 [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 disclosures (currently over 200) on the application of bilinear groups in cryptography, see [Bar06].

[0028] We consider equations over variables from G_1, G_2 and $\mathbb{Z}_n$ as shown in Table 1. We construct efficient 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 1.2 for precise definitions.

[0029] 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 $\mathbb{Z}_n$. 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 for $\mathcal{P}_1, \mathcal{Q}_1, \ldots, \mathcal{P}_n, \mathcal{Q}_n$ such that $t_T = \prod_{i=1}^{n} e(\mathcal{P}_i, Q_i)$.

[0030] Example Embodiment 1: Subgroup decision. The present disclosure includes a general description of the proof techniques as well as three example embodiments that illustrate the use of these techniques. The first example embodiment is based on the composite order groups introduced by Boneh, Goh and Nissim [BGN05]. Here we generate a composite order bilinear group $(n, G, G_T, e, \mathcal{P})$ where $n = pq$. We can write $G = G_p \times G_q$, where G_p, G_q are the subgroups of order p and q respectively. Boneh, Goh and Nissim introduce the subgroup decision assumption, which says that it is hard to distinguish a random element from

Variables: $\mathcal{X}_1, \ldots, \mathcal{X}_m \in G_1$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in G_2$, $x_1, \ldots, x_{m'}, y_1, \ldots, y_{n'} \in \mathbb{Z}_{\mathbf{n}}$. Footnote[a].

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:

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

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

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 $\mathbb{Z}_{\mathbf{n}}$:

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

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

[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 disclosure, but stress that the discrete logarithm problem will typically be hard in these groups.

Table 1: Equations over groups with bilinear map.

G from a random element from $G_{\mathbf{q}}$. In this disclosure, we will demonstrate that assuming the hardness of the subgroup decision problem there exists a witness-indistinguishable proof for satisfiability of a set of equations from Table 1 in the subgroup $G_{\mathbf{p}}$ and the order $\mathbf{p}$ subgroup

of G_T.

[0031] Example Embodiment 2: The symmetric external Diffie-Hellman (SXDH) problem. Let $(\mathbf{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)$ be a prime order bilinear group. The external Diffie-Hellman (XDH) assumption is that the decisional Diffie-Hellman (DDH) problem is hard in one of the groups G_1 or G_2 [Sco02, BBS04, BGdMM05, GR04, Ver04]. The Symmetric XDH assumption is that the DDH problem is hard in both G_1 and G_2. We will construct a witness-indistinguishable proof for satisfiability of a set of equations of the form given in Figure 1 under the SXDH assumption.

[0032] Example Embodiment 3: The decisional linear assumption (DLIN) problem. The DLIN for a prime order bilinear group $(\mathbf{p}, G, G_T, e, \mathcal{P})$ 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 $\alpha, \beta, r, s \in \mathbb{Z}_{\mathbf{p}}$ it is hard to tell whether $t = r + s$ or t is random. Assuming the hardness of the DLIN problem, we show a witness-indistinguishable proof for satisfiability of the equations from Table 1.

[0033] The example embodiments illustrate some of 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. These three security assumptions have been used in the cryptographic literature to build useful protocols.

[0034] For these three example embodiments, the techniques presented here yield very 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 2. We refer to Section 6 for more detailed tables.

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

Early work on NIZK proofs demonstrated that NP-languages have non-interactive proofs, however, 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. The systems and methods disclosed herein are significantly more general, and vastly more efficient.

[0035] We achieve generality, at least in part, by viewing the groups G_1, G_2, G_T as modules over the ring $\mathbb{Z}_n$. The ring $\mathbb{Z}_n$ itself can also be viewed as a $\mathbb{Z}_n$-module. We therefore look at the more general question of satisfiability of quadratic equations over $\mathbb{Z}_n$-modules A_1, A_2, A_T with a bilinear map, see Section 2 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.

[0036] 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. These modules will typically be larger modules, which give us space 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, A_T to the modules B_1, B_2, B_T. The commitment schemes are homomorphic with respect to the module operations but also with respect to the bilinear map.

[0037] It is instructive to begin with an intuition-based explanation before showing the more detailed explanation in Section 6 and related sections. Because the commitment schemes herein are homomorphic and we equip them with a bilinear map, we can take the equation that we are trying to prove, and replace the variables in the equations with commitments to those variables. Since 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. However, if there is an "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. If there are many terms, that means that these terms are not unique, and we can randomize these terms so that the equation is still true, but so that we effectively reduce to the case of there being a single term being given away with a unique value.

1.1 Applications

[0038] In one embodiment, we construct ring-signatures of sub-linear size using the NIWI

proofs in the first example embodiment, which is based on the subgroup decision problem. Groth and Lu [GL07] have used the NIWI and NIZK proofs from example embodiment 3 to construct a NIZK proof for the correctness of a shuffle. Groth [Gro07] has used the NIWI and NIZK proofs from example embodiment 3 to construct a fully anonymous group signature scheme. By attaching NIZK proofs to semantically secure public-key encryption we get an efficient non-interactive verifiable cryptosystem. This can be used for optimistic fair exchange, where two parties use a trusted but lazy third party to guarantee fairness.

1.2 Non-interactive Witness-Indistinguishable Proofs

[0039] 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 includes statements in R. For a relation that ignores gk this is of course the standard definition of an NP-language. We will, however, be more interested in the case where gk describes a bilinear group.

[0040] A non-interactive proof system for a relation R with setup incudes four probabilistic polynomial time algorithms: a setup algorithm $\mathcal{G}$, a CRS generation algorithm K, a prover P and a verifier V. The setup algorithm outputs a setup (gk, sk). In the present disclosure, 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.

[0041] WIth respect to perfect completeness, for adversaries $\mathcal{A}$ we have

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

[0042] With respect to perfect soundness, for adversaries $\mathcal{A}$ we have

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

[0043] In the standard definition of soundness defined above, the adversary is successful if creating a valid proof for $x \notin L$. We will generalize this notion to what we will call co-soundness, where the adversary is successful if creating a valid proof for $x \in L_{\text{co}}$ for some language L_{co}, which may depend on gk and σ. Standard soundness is a special case of co-soundness with L_{co} being the complement of L.

[0044] With respect to perfect L_{co}-soundness, for adversaries $\mathcal{A}$ we have

$$\Pr\left[\begin{array}{l}(gk, sk) \leftarrow \mathcal{G}(1^k);\\ \sigma \leftarrow K(gk, sk);\\ (x, \pi) \leftarrow \mathcal{A}(gk, \sigma) : V(gk, \sigma, x, \pi) = 0 \text{ if } x \in L_{\text{co}}\end{array}\right] = 1.$$

[0045] In this disclosure, we will use a strong definition of witness indistinguishability. We introduce a reference string simulator S that generates a simulated CRS. We require that the adversary cannot distinguish a real CRS from a simulated CRS. We also require that on a simulated CRS it is *perfectly* indistinguishable which witness the prover used.

[0046] In other words, for 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

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

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

[0047] Composable zero-knowledge is a strengthening of the usual notion of non-interactive zero-knowledge. First, we require that an adversary cannot distinguish a real CRS

from a simulated CRS. Second, we require that the adversary, even when it gets access to the secret simulation key τ, cannot distinguish real proofs on a simulated CRS from simulated proofs.

[0048] In other words, there exists a polynomial time simulator (S_1, S_2) so for 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

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

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

2 Modules with Bilinear Maps

[0049] 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

$$\forall r, s \in \mathcal{R}\, \forall x, y \in A \; : \; (r+s)x = rx + sx \wedge r(x+y) = rx + ry \wedge r(sx) = (rs)x \wedge 1x = x.$$

[0050] A cyclic group G of order n can in a natural way be viewed as a $\mathbb{Z}_n$-module. We will observe that the equations in Table 1 can be viewed as equations over $\mathbb{Z}_n$-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 $f : A_1 \times A_2 \rightarrow A_T$. 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

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

[0051] 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

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

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 $x_1, \ldots, x_m, y_1, \ldots, y_n$ that $\vec{x} \cdot \Gamma \vec{y} = \Gamma^{\top} \vec{x} \cdot \vec{y}$.

[0052] Now return to the equations in Table 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 we can rewrite the pairing product equation as $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{X}} \cdot \Gamma \vec{\mathcal{Y}}) = t_T$. (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.)

- 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 we can rewrite the 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 we can 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 we can rewrite the quadratic equation in $\mathbb{Z}_{\mathbf{n}}$ as $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma \vec{y} = t$.

We now 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.

3 Commitment from Modules

[0053] In the present NIWI 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.

[0054] 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 $p : B \to A$. It will also contain elements $u_1, \ldots, u_n \in B$. To commit to $x \in A$ we pick $r_1, \ldots, r_n \leftarrow \mathcal{R}$ at random and compute the commitment

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

In one embodiment, the Commitment scheme has two types of commitment keys, hiding keys and binding keys. The main assumption that we will be making throughout this disclosure is that the distribution of hiding keys and the distribution of binding keys are computationally indistinguishable. Witness-indistinguishability of the present NIWI proofs and later the zero-knowledge property of the present ZK proofs use this property.

- Hiding key: A hiding key contains $(B, \iota, p, u_1, \ldots, u_n)$ such that $\iota(G) \subseteq \langle u_1, \ldots, u_n \rangle$. The commitment $c := \iota(x) + \sum_{i=1}^{n} r_i u_i$ is therefore perfectly hiding when $r_1, \ldots r_n$ are chosen at random from $\mathcal{R}$.

- Binding key: A binding key contains $(B, \iota, p, u_1, \ldots, u_n)$ such that $\forall i : p(u_i) = 0$ and $\iota \circ p$ is non-trivial. The commitment $c := \iota(x) + \sum_{i=1}^{n} r_i u_i$ therefore contains the non-trivial information $p(c) = p(\iota(x))$ about x. In particular, if $\iota \circ p$ is the identity map on A, then the commitment is perfectly binding. (The map p is not efficiently computable. However, one can imagine scenarios where a secret key will make p efficiently computable and $\iota \circ p$ is the identity map. In this case the commitment scheme is a cryptosystem with p being the decryption operation.)

[0055] Since we will often be committing to many elements at a time let us define some convenient notation. Given elements $x_1, \ldots, x_m$ we will write $\vec{c} := \iota(\vec{x}) + R\vec{u}$ with $R \in \mathrm{Mat}_{m \times n}(\mathcal{R})$ for making commitments $c_1, \ldots, c_m$ computed as $c_i := \iota(x_i) + \sum_{j=1}^{n} r_{ij} u_j$.

3.1 Example Embodiments

[0056] The treatment of commitments using the language of modules generalizes several previous works dealing with commitments over bilinear groups.

Example Embodiment 1: Subgroup decision.

[0057] In this setting, we have a composite order group G of order $\mathbf{n} := \mathbf{pq}$. The group can in a natural way be viewed as a $\mathbb{Z}_{\mathbf{n}}$-module; using the notation above we define $A = G$ and $B = G$. The commitment key will contain an element $\mathcal{U}$. We can choose it so $\mathcal{U}$ generates G or so $\mathcal{U}$ has order q. The subgroup decision assumption tells us that the two types of commitment keys are computationally indistinguishable.

[0058] Let $\iota : G \to G$ be the identity map. Define $\lambda \in \mathbb{Z}_{\mathbf{n}}$ so $\lambda = 1 \bmod \mathbf{p}$ and $\lambda = 0 \bmod \mathbf{q}$. The map $p : G \to G$ is $p(\mathcal{X}) := \lambda\mathcal{X}$; in other words, p maps elements onto the order $\mathbf{p}$ subgroup of G. If $\mathcal{U}$ generates G, then $\mathcal{C} := \iota(\mathcal{X}) + r\mathcal{U}$ is perfectly hiding. On the other hand, if $\mathcal{U}$ has order q, then $\lambda C = \lambda\mathcal{X}$ defines $\mathcal{X}$ uniquely in $G_{\mathbf{p}}$.

[0059] We can also commit to exponents. The modules are $A' = \mathbb{Z}_{\mathbf{n}}$ and $B = G$. Let $\iota' : \mathbb{Z}_{\mathbf{n}} \to G$ be given by $\iota'(x) = x\mathcal{P}$ and $p' : G \to \mathbb{Z}_{\mathbf{n}}$ be given by $p'(x\mathcal{P}) = \lambda x$. When $\mathcal{U}$ generates G, the commitment scheme $\mathcal{C} := x\mathcal{P} + r\mathcal{U}$ is perfectly hiding. On the other hand, if $\mathcal{U}$ has order q, then the commitment determines $p'(\mathcal{C}) = \lambda x \in \mathbb{Z}_{\mathbf{n}}$.

Example embodiment 2: SXDH.

[0060] Consider a cyclic group $A := G$ of prime order $\mathbf{p}$. By entry-wise addition we get an abelian group $B := G^2$, which is a module over $\mathbb{Z}_{\mathbf{p}}$. The commitment key will contain an element $u_1 = (\mathcal{P}, \mathcal{Q})$, where $\mathcal{Q} = \alpha\mathcal{P}$ for a randomly chosen $\alpha \in \mathbb{Z}_{\mathbf{p}}^*$. It will also contain an element $u_2 = (\mathcal{U}, \mathcal{V})$ which can be chosen in one of two ways: $u_2 := tu_1$ or $u_2 := tu_1 - (\mathcal{O}, \mathcal{P})$ for a randomly chosen $t \in \mathbb{Z}_{\mathbf{p}}^*$. The former will give a perfectly binding commitment key, whereas the latter will give a perfectly hiding commitment key. The DDH assumption tells us that the two types of commitment keys are computationally indistinguishable.

[0061] Let us now describe how to commit to an element $\mathcal{X} \in G$. We define $\iota(\mathcal{X}) := (\mathcal{O}, \mathcal{X})$. Using randomness $r_1, r_2 \in \mathbb{Z}_{\mathbf{p}}$ we get a commitment of the form $c := \iota(\mathcal{X}) + r_1u_1 + r_2u_2$. If $u_2 = tu_1$ we have $c = ((r + st)\mathcal{P}, (r + st)\mathcal{Q})$ which is an ElGamal encryption of $\mathcal{P}$. We define $p : (\mathcal{C}_1, \mathcal{C}_2) \mapsto \mathcal{C}_2 - \alpha\mathcal{C}_1$ and see that the commitment is perfectly binding since $\iota \circ p$ is the identity map on G and $p(u_1) = p(u_2) = \mathcal{O}$. If u_1 and u_2 are linearly independent we have that u_1, u_2 is a basis for $B = G^2$ and therefore $\iota(G) \subseteq \langle u_1, u_2 \rangle$. When u_1 and u_2 are linearly independent we therefore have a perfectly hiding commitment.

[0062] To commit to an exponent $x \in A' := \mathbb{Z}_{\mathbf{p}}$, we use the following approach. We

define $u = u_1 + (\mathcal{O}, \mathcal{P})$ and $\iota'(x) := xu$ and $p'(c_1\mathcal{P}, c_2\mathcal{P}) := c_2 - \alpha c_1$. To commit to x using randomness $r \in \mathbb{Z}_p$ we compute $c := \iota'(x) + r\mathcal{U}_1$. On a hiding key we have $u = tu_1$ so $u \in \langle u_1 \rangle$, which implies $\iota'(\mathbb{Z}_p) \subseteq \langle u_1 \rangle$. A hiding key therefore gives us a perfectly hiding commitment scheme. On a binding key we have $c = ((r + xt)\mathcal{P}, (r + xt)\mathcal{Q} + x\mathcal{P})$, which is an ElGamal encryption of $x\mathcal{P}$. We have that $\iota' \circ p'$ is the identity map and $p'(u_1) = 0$ so the commitment scheme is perfectly binding.

Example Embodiment 3: DLIN.

[0063] In a DLIN group let $\mathcal{U} := \alpha\mathcal{P}, \mathcal{V} := \beta\mathcal{P}$ be given for random $\alpha, \beta \in \mathbb{Z}_p^*$. The DLIN assumption states that it is hard to tell whether three elements $r\mathcal{U}, s\mathcal{V}, t\mathcal{P}$ have the property that $t = r + s$. We will use the $\mathbb{Z}_p$-modules $A = G$ and $B = G^3$ formed by entry-wise addition. The commitment key will contain three elements $u_1, u_2, u_3 \in B$. We use $u_1 := (\mathcal{U}, \mathcal{O}, \mathcal{P}), u_2 := (\mathcal{O}, \mathcal{V}, \mathcal{P})$ and u_3 can be chosen as either $u_3 := ru_1 + su_2$ or $u_3 := ru_1 + su_2 - (\mathcal{O}, \mathcal{O}, \mathcal{P})$, which will give respectively a binding key and a hiding key. The DLIN assumption implies that the two types of commitment keys are computationally indistinguishable.

[0064] We will now describe how to commit to $\mathcal{X} \in G$. The map ι is defined by $\iota(\mathcal{X}) := (\mathcal{O}, \mathcal{O}, \mathcal{X})$. A commitment is formed by choosing $r_1, r_2, r_3 \in \mathbb{Z}_p$ and computing $c := \iota(\mathcal{X}) + \sum_{i=1}^{3} r_i u_i$. 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. On a binding key we have $c = ((r_1 + rr_3)\mathcal{U}, (r_2 + sr_3)\mathcal{V}, (r_1 + r_2 + (r + s)r_3)\mathcal{P} + \mathcal{X})$, which is a BBS encryption [BBS04] of $\mathcal{X}$. Defining the decryption function $p(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) := \mathcal{C}_3 - \frac{1}{\alpha}\mathcal{C}_1 - \frac{1}{\beta}\mathcal{C}_2$ we see that $p(u_1) = p(u_2) = p(u_3) = \mathcal{O}$ and $\iota \circ p$ is the identity map so the commitment is perfectly binding. (This commitment scheme coincides with the scheme of [Wat06]. We note that the different, and less efficient, commitment scheme of [Gro06] can be similarly described in the language of modules, as well.)

[0065] To commit to a message $x \in A' := \mathbb{Z}_p$ we first define $u := u_3 + (\mathcal{O}, \mathcal{O}, \mathcal{P})$ and $\iota'(x) := xu$. We commit to x using randomness r_1, r_2 by setting $c := xu + r_1u_1 + r_2u_2$. On a hiding key, we have that $u = ru_1 + su_2$ so $\iota'(\mathbb{Z}_p) \subseteq \langle u_1, u_2 \rangle$ and the commitment scheme is perfectly hiding. On a binding key, the commitment is $c = ((r_1 + rx)\mathcal{U}, (r_2 + sx)\mathcal{V}, (r_1 + r_2 + x(r + s))\mathcal{P} + x\mathcal{P})$. This corresponds to a BBS encryption of $x\mathcal{P}$. We define $p'(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3) := \mathcal{C}_3 - \frac{1}{\alpha}\mathcal{C}_1 - \frac{1}{\beta}\mathcal{C}_2)$. We have $p'(u_1) = p'(u_2) = 0$ and $\iota' \circ p'$ is the identity on $\mathbb{Z}_p$, so the commitment scheme is perfectly binding.

4 Setup

[0066] In the present NIWI proofs the common reference string contains commitment keys to commit to elements in respectively A_1 and A_2. These commitment keys specify $B_1, \iota_1, p_1, u_1, \ldots, u_{m'}$ and $B_2, \iota_2, p_2, v_1, \ldots, v_{n'}$. In addition, the common reference string will also specify a third $\mathcal{R}$-module B_T together with $\mathcal{R}$-linear maps $\iota_T : A_T \to B_T$ and $p_T : B_T \to A_T$. There will be a bilinear map $F : B_1 \times B_2 \to B_T$ as well. We require that the maps are commutative. We refer to Table 3 for an overview of the modules and the maps. For

$$\begin{array}{ccccc} A_1 & \times & A_2 & \stackrel{f}{\to} & A_T \\ \iota_1 \downarrow\uparrow p_1 & & \iota_2 \downarrow\uparrow p_2 & & \iota_T \downarrow\uparrow p_T \\ B_1 & \times & B_2 & \stackrel{F}{\to} & B_T \end{array}$$

$$\forall x \in A_1 \, \forall y \in A_2 : \; F(\iota_1(x), \iota_2(y)) = \iota_T(f(x,y))$$
$$\forall x \in B_1 \, \forall y \in B_2 : \; f(p_1(x), p_2(x)) = p_T(F(x,y))$$

Table 3: Modules and maps between them.

notational convenience, let us define for $\vec{x} \in B_1^n, \vec{y} \in B_2^n$ that

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

The final part of the common reference string is a set of matrices $H_1, \ldots, H_\eta \in \mathrm{Mat}_{m' \times n'}(\mathcal{R})$ that satisfy $\vec{u} \bullet H_i \vec{v} = 0$.

[0067] Two types of settings are of primary interest, soundness settings and witness-indistinguishability settings.

- Soundness setting: In the soundness setting, we require that the commitment keys are binding so we have $p_1(\vec{u}) = \vec{0}$ and $p_2(\vec{v}) = \vec{0}$ and the maps $\iota_1 \circ p_1$ and $\iota_2 \circ p_2$ are non-trivial.

- Witness-indistinguishability setting: In the witness-indistinguishability setting we have hiding commitment keys, so $\iota_1(G_1) \subseteq \langle u_1, \ldots u_{m'} \rangle$ and $\iota_2(G_2) \subseteq \langle v_1, \ldots, v_{n'} \rangle$. We also require that $H_1, \ldots, H_\eta$ generate the R-module of matrices H so $\vec{u} \bullet H \vec{v} = 0$.

As we will see in the next section, these matrices play a role as randomizers in the witness-indistinguishability proof.

The (only) computational assumption this disclosure is based on is that the two settings can be set up in a computationally indistinguishable way. The example embodiments show that there are many ways to get such computationally indistinguishable soundness and witness-indistinguishability setups.

4.1 Example Embodiments

Example Embodiment 1: Subgroup Decision.

[0068] The common reference string specifies $(\mathbf{p}, G, G_T, e, \mathcal{P}, \mathcal{U})$, which is sufficient to describe the entire setup given in this section. We use $B = B_1 = B_2 = G$ and $B_T = G_T$ and the bilinear map $F(\mathcal{X}, \mathcal{Y}) := e(\mathcal{X}, \mathcal{Y})$. In the witness-indistinguishability setup we use a hiding key $\mathcal{U}$ that generates G and consequently $e(\mathcal{U}, \mathcal{U})$ generates G_T. The only solution to $e(\mathcal{U}, H\mathcal{U}) = 1$ is therefore the trivial $H = 0$, so we do not need to include any H_i in the common reference string.

[0069] There are three scenarios to look at: pairing product equations, multi-scalar multiplication and quadratic equations in $\mathbb{Z}_{\mathbf{n}}$. In the pairing product equation scenario, we have $A_1 = A_2 = G$ and $A_T = G_T$ and a bilinear map $f := e$. We define the map $\iota_T : A_T \to B_T$ to be the identity map, whereas $p_T(z) := z^{\lambda}$. Observe, since $\lambda = 1 \bmod \mathbf{p}, \lambda = 0 \bmod \mathbf{q}$ that $\lambda^2 = \lambda \bmod \mathbf{n}$ so we have the commutative property $e(p_1(\mathcal{X}), p_2(\mathcal{Y})) = e(\lambda\mathcal{X}, \lambda\mathcal{Y}) = p_T(e(\mathcal{X}, \mathcal{Y}))$ and the other commutative property is trivial.

[0070] In the multi-scalar multiplication scenario, we have $A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = G, A_T = G$. The bilinear map f is the scalar multiplication function $f(x, \mathcal{Y}) := x\mathcal{Y}$. We define $\hat{\iota}_T(\mathcal{Z}) := e(\mathcal{P}, \mathcal{Z})$ and $\hat{p}_T(e(\mathcal{P}, \mathcal{Z})) = \lambda\mathcal{Z}$. This gives us the required commutative properties $e(\iota'(x), \iota(\mathcal{Y})) = e(x\mathcal{P}, \mathcal{Y}) = e(\mathcal{P}, x\mathcal{Y}) = \hat{\iota}_T(x\mathcal{Y})$ and $\hat{p}_T(e(x\mathcal{P}, \mathcal{Y})) = \lambda x\mathcal{Y} = (\lambda x)(\lambda\mathcal{Y}) = p'(x\mathcal{P})p(Y)$.

[0071] In the quadratic equation in $\mathbb{Z}_{\mathbf{n}}$, we have $A_1 = A_2 = A_T = \mathbb{Z}_{\mathbf{n}}$. The bilinear map f is the multiplication function $f(x, y) := xy \bmod \mathbf{n}$. We define $\iota'_T(z) := e(\mathcal{P}, \mathcal{P})^z$ and $p'_T(e(\mathcal{P}, \mathcal{P})^z) := \lambda z$. We have the commutative properties $e(\iota'(x), \iota'(y)) = e(x\mathcal{P}, y\mathcal{P}) = e(\mathcal{P}, \mathcal{P})^{xy} = \iota'_T(xy)$ and $p'_T(e(x\mathcal{P}, y\mathcal{P})) = \lambda xy = (\lambda x)(\lambda y) = p'(x\mathcal{P})p'(y\mathcal{P})$.

Example Embodiment 2: SXDH.

[0072] The common reference string specifies $(\mathbf{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2, u_1, u_2, v_1, v_2)$, where (u_1, u_2) is a commitment key for the group G_1 and (v_1, v_2) is a commitment key for G_2

as described in Section 3.1. We have $B_1 = G_1^2$, $B_2 = G_2^2$ and define $B_T := G_T^4$ with respectively entry-wise addition and entry-wise multiplication. The map F is defined as follows:

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

[0073] In the pairing product equation scenario, we have $A_1 = G_1, A_2 = G_2, A_T = G_T$ and $f(x, y) := e(x, y)$. The commitment keys are u_1, u_2 and v_1, v_2 for committing to respectively elements in G_1 and G_2. In the witness-indistinguishability scenario, the commitment keys are hiding, which means they are chosen so u_1 and u_2 are linearly independent and v_1 and v_2 are 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 linearly independent in this scenario. This implies that $\vec{u} \bullet H\vec{v}$ only has the trivial solution where H is the 2×2 matrix with 0-entries. As for the maps ι_T, p_T we define

$$\iota_T : z \mapsto \begin{pmatrix} 1 & 1 \\ 1 & z \end{pmatrix} \qquad , \qquad p_T(\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix}) \mapsto z_{22} z_{12}^{-\alpha_1} (z_{21} z_{11}^{-\alpha_1})^{-\alpha_2}.$$

The map p_T corresponds to first ElGamal decrypting down the columns using α_1 where $u_1 = (\mathcal{P}_1, \alpha_1 \mathcal{P}_1)$ and then ElGamal decrypting the resulting row by using α_2 where $v_1 = (\mathcal{P}_2, \alpha_2 \mathcal{P}_2)$. We note that $\iota_T \circ p_T$ is the identity map. One can check that the maps satisfy the commutative properties in Table 3.

[0074] We will now look at the case of multi-scalar multiplication in G_2. The case of multi-scalar multiplication in G_1 is treated similarly. We have $A_1 = \mathbb{Z}_\mathbf{p}, A_2 = G_2, A_T = G_2$ and the bilinear map is $f(x, \mathcal{Y}) = x\mathcal{Y}$. We will use ι', u_1 for commitments to scalars in $\mathbb{Z}_\mathbf{p}$ and ι, v_1, v_2 for commitments to elements in G_2. We define $\hat{\iota}_T(\mathcal{Z}) = \iota_T(e(\mathcal{P}, \mathcal{Z}))$. Let $e^{-1}(e(\mathcal{P}, \mathcal{Z})) := \mathcal{Z}$ and define $\hat{p}_T(z) := e^{-1}(p_T(z))$. We note that $\hat{\iota}_T \circ \hat{p}_T$ is the identity map on G_2. We see that in the witness-indistinguishability setting, where v_1, v_2 are linearly independent, the equation $u_1 \bullet H\vec{v} = 0$ only has the trivial solution where H is the 1×2 matrix containing 0-entries.

[0075] Finally, we have the case of quadratic equations in $\mathbb{Z}_\mathbf{p}$. We have $A_1 = A_2 = A_T = \mathbb{Z}_\mathbf{p}$ and the bilinear map $f(x, y) := xy \bmod \mathbf{p}$. We use u, u_1 for commitments in G_1^2 and v, v_1 for commitments in G_2^2. We define $\iota'_T(z) := \iota_T(e(\mathcal{P}, \mathcal{P})^z)$ and $p'_T(z) := \log_\mathcal{P}(\hat{p}_T(z))$. The maps satisfy the commutative properties from Table 3 and we have $\iota'_T \circ p'_T$ is the identity map on $\mathbb{Z}_\mathbf{p}$. Since $F(u_1, Hv_1)$ has no non-trivial solution we do not need to specify a set of generators $H_1, \ldots, H_\eta$.

Example embodiment 3: DLIN.

[0076] The common reference string specifies $(\mathbf{p}, G, G_T, e, \mathcal{P}, u_1, u_2, u_3)$, where (u_1, u_2, u_3) is a commitment key for the group G, and u_1, u_2 is used for committing to exponents. We have $B = G^3$.

We will use the module $B_T = G_T^9$ defining the addition of two elements to correspond to entry-wise multiplication of the 9 group elements. We will use two different bilinear maps $F, \widetilde{F}$. The map $\widetilde{F}$ is defined as follows:

$$\widetilde{F} : G^3 \times G^3 \to G_T^9 \qquad \left(\begin{pmatrix} \mathcal{X}_1 \\ \mathcal{X}_2 \\ \mathcal{X}_3 \end{pmatrix}, \begin{pmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \\ \mathcal{Y}_3 \end{pmatrix}\right) \mapsto \begin{pmatrix} e(\mathcal{X}_1, \mathcal{Y}_1) & e(\mathcal{X}_1, \mathcal{Y}_2) & e(\mathcal{X}_1, \mathcal{Y}_3) \\ e(\mathcal{X}_2, \mathcal{Y}_1) & e(\mathcal{X}_2, \mathcal{Y}_2) & e(\mathcal{X}_2, \mathcal{Y}_3) \\ e(\mathcal{X}_3, \mathcal{Y}_1) & e(\mathcal{X}_3, \mathcal{Y}_2) & e(\mathcal{X}_3, \mathcal{Y}_3) \end{pmatrix}.$$

[0077] The symmetric map F is defined by $F(x, y) := \frac{1}{2}\widetilde{F}(x, y) + \frac{1}{2}\widetilde{F}(y, x)$.

[0078] In the pairing product equation scenario, we have $A_1 = G_1, A_2 = G_2, A_T = G_T$ and $f(x, y) := e(x, y)$. The commitment key is u_1, u_2, u_3. In the witness-indistinguishability scenario, the commitment key is hiding, which means that it is chosen so $u_1, u_2 u_3$ are linearly independent and hence span of $B = G^3$. Some computation shows that the nine elements $\widetilde{F}(u_i, u_j)$ are linearly independent in the witness-indistinguishability setting. This implies that $\vec{u} \bullet H\vec{u}$ only has the trivial solution where H is the 3×3 matrix with 0-entries.

[0079] 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 so $\vec{u} \bullet H\vec{u} = 0$.

[0080] As for the maps ι_T, p_T we define

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

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

The map p_T corresponds to first BBS decrypting down the columns using the decryption key α, β and then after that BBS decrypting along the row. We note that $\iota_T \circ p_T$ is the identity map. One can check that the maps satisfy the commutative properties with both $\widetilde{F}$ and F in Table 3.

[0081] We will now look at the case of multi-scalar multiplication in G. We have $A_1 = \mathbb{Z}_{\mathbf{p}}, A_2 = G, A_T = G$ and the bilinear map is $f(x, \mathcal{Y}) = x\mathcal{Y}$. We will use ι', u_1, u_2 for commitments to scalars in $\mathbb{Z}_{\mathbf{p}}$ and ι, u_1, u_2, u_3 for commitments to elements in G. We define $\hat{\iota}_T(\mathcal{Z}) = \iota_T(e(\mathcal{P}, \mathcal{Z}))$. Let $e^{-1}(e(\mathcal{P}, \mathcal{Z})) := \mathcal{Z}$ and define $\hat{p}_T(z) := e^{-1}(p_T(z))$. We note that $\hat{\iota}_T \circ \hat{p}_T$ is the identity map on G. We see that $(u_1, u_2) \tilde{\bullet} H\vec{u} = 0$ only has the trivial solution where H is the 2×3 matrix containing 0-entries. We also have

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

generates the matrices H so $(u_1, u_2) \bullet H\vec{u} = 0$.

[0082] Finally, we have the case of quadratic equations in $\mathbb{Z}_{\mathbf{p}}$. We have $A_1 = A_2 = A_T = \mathbb{Z}_{\mathbf{p}}$ and the bilinear map $f(x, y) := xy \bmod \mathbf{p}$. We use u_1, u_2 for commitments to the exponents. We define $\iota'_T(z) := \iota_T(e(\mathcal{P}, \mathcal{P})^z$ and $p'_T(z) := \log_{\mathcal{P}}(\hat{p}_T(z))$. The maps satisfy the commutative properties from Table 3 and we have $\iota'_T \circ p'_T$ is the identity map on $\mathbb{Z}_{\mathbf{p}}$. Again we have for $\widetilde{F}$ only trivial matrices H, whereas for F we have the non-trivial basis

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

.

5 Proving that Committed Values Satisfy a Quadratic Equation

[0083] Recall that a quadratic equation looks like the following:

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

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

[0084] We will first consider the case of a single quadratic equation of the above form. The first step in the present NIWI proof is to commit to the variables $\vec{x}, \vec{y}$. The commitments are of the form

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

(Note that for various other embodiments, we will use these same commitments.)

[0085] Before giving the proof let us give some intuition. In the previous sections, we have set up the commitments so 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, also implicitly used in the $\bullet$ operation). Given that we have done this, a natural idea is to take the quadratic equation we are trying to prove, and "plug in" the commitments 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, make use of the bilinearity of $\bullet$, and rearrange terms (the details can be found in the proof of Theorem 1 below) we get

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

By the commutativity properties of the maps, the first group of three terms are equal to $\iota_T(t)$, if the equation is true. Looking at the remaining terms, note that the verifier knows $\vec{u}$ and $\vec{v}$. 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 that they match either $\vec{u}$ or $\vec{v}$ to get (again see the proof of Theorem 1 for details)

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

Now, for sake of explanation only, and not limitation, let us make some simplifying assumptions: Assume that we are working in a symmetric case where $A_1 = A_2$, and $B_1 = B_2$, and therefore $\vec{u} = \vec{v}$ and, so, the above equation can be simplified further to get:

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

Assume further, $\iota_1 \circ p_1, \iota_2 \circ p_2$ and $\iota_T \circ p_T$ are the identity maps on A_1, A_2 and A_T.

[0086] Now, suppose the prover gives to the verifier as his proof $\vec{\pi} = \left(R^\top \iota_2(\vec{b}) + R^\top \Gamma \iota_2(\vec{y}) + S^\top \iota_1(\vec{a}) + S^\top \Gamma^\top \iota_1(\vec{x})\right)$. The verifier would then check that the following *verification equation* holds:

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

[0087] It is easy to see that this proof would be convincing in the soundness setting, because 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.

[0088] 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 $\vec{\pi}$ has any information about $\vec{x}$ and $\vec{y}$ except for the information that can be inferred from the quadratic equation itself. So, consider two cases:

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

2. The simple case above might seem too good to be true, but see what it means if it isn't true. If two values $\vec{\pi}$ and $\vec{\pi}'$ both satisfy the verification equation, then just 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 $\vec{\pi}$ or $\vec{\pi}'$ exist, and furthermore we know that the matrices $H_1, \ldots, H_\eta$ generate H such that $\vec{u} \bullet H\vec{u} = 0$. Therefore, choose $r_1, \ldots, r_\eta$ at random, and consider the distribution $\vec{\pi}'' = \vec{\pi} + \sum_{i=1}^{\eta} r_i H_i \vec{u}$. We thus obtain the same distribution on $\vec{\pi}''$ regardless of what $\vec{\pi}$ we started from, and such that $\vec{\pi}''$ always satisfies the verification equation.

[0089] 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 the equation, we would also have $\vec{\psi}$ 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 $\vec{\pi}, \vec{\psi}$ that satisfy the verification equation. If we pick $T \leftarrow \mathrm{Mat}_{n' \times m'}(\mathcal{R})$ at random we have that $\vec{\psi} + T\vec{u}$ completely randomizes $\vec{\psi}$. The part we add in $\vec{\psi}$ can be "subtracted" from $\vec{\pi}$ by observing that

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

This leads to a unique distribution of proofs for the general non-symmetric case as well.

[0090] Having now explained the intuition behind the following proof system, we proceed to a formal description and proof of security properties.

Proof: Pick $T \leftarrow \mathrm{Mat}_{n' \times m'}(\mathcal{R}), r_1, \ldots, r_\eta \leftarrow \mathcal{R}$ at random. Compute

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

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

Verification: Return 1 if and only if

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

[0091] Perfect completeness of the NIWI proof will follow from the following theorem no matter whether we are in the soundness setting or the witness-indistinguishability setting.

Theorem 1

[0092] *Given $\vec{x}, \vec{y}, R, S$ satisfying*

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

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

[0093] *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

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

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

□

Theorem 2

[0094] *In the soundness setting, where we have* $p_1(\vec{u}) = \vec{0}, p_2(\vec{v}) = \vec{0}$ *a valid proof implies* $p_1(\iota_1(\vec{a})) \cdot p_2(\vec{d}) + p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) + p_1(\vec{c}) \cdot \Gamma p_2(\vec{d}) = p_T(\iota_T(t))$.

[0095] *Proof.* An acceptable proof $\vec{\pi}, \vec{\psi}$ satisfies $\iota(a) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\psi} \bullet \vec{v}$. The commutative property of the linear and bilinear maps gives us

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

□

[0096] Observe as a particularly interesting case that when $\iota_1 \circ p_1, \iota_2 \circ p_2, \iota_T \circ p_T$ are the identity maps on A_1, A_2 and A_T respectively, then this means $\vec{x} := p_1(\vec{c})$ and $\vec{y} := p_2(\vec{d})$ give us a satisfying solution to the equation $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{y} + \vec{x} \cdot \Gamma\vec{y} = t$. In this case, the theorem says that the proof is perfectly sound in the soundness setting. It is still possible though that interesting co-soundness properties emerge also in the case where these maps are not the identity-maps on A_1, A_2 and A_T.

Theorem 3

[0097] *In the witness-indistinguishable setting where* $\iota_1(G_1) \subseteq \langle u_1, \ldots, u_{m'}\rangle$, $\iota_2(G_2) \subseteq \langle v_1, \ldots, v_{n'}\rangle$ *and* $H_1, \ldots H_\eta$ *generate the matrices* H *so* $\vec{u} \bullet H\vec{v} = 0$, *the satisfying witnesses* $\vec{x}, \vec{y}, R, S$ *yield proofs* $\vec{\pi} \in \langle v_1, \ldots, v_{n'}\rangle^{m'}$ *and* $\vec{\psi} \in \langle u_1, \ldots, u_{m'}\rangle^{n'}$ *that are uni-*

formly distributed conditioned on the verification equation $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{u} \bullet \vec{\pi} + \vec{\psi} \bullet \vec{v}$.

[0098] *Proof.* Since $\iota_1(G_1) \subseteq \langle u_1, \ldots, u_{m'} \rangle$ and $\iota_2(G_2) \subseteq \langle v_1, \ldots, v_{n'} \rangle$ there exists A, B, X, Y so $\iota_1(\vec{a}) = A\vec{u}$, $\iota_1(\vec{x}) = X\vec{u}$ and $\iota_2(\vec{b}) = B\vec{v}$, $\iota_2(\vec{y}) = Y\vec{v}$. We have $\vec{c} = \vec{0} + (X + R)\vec{u}$ and $\vec{d} = \vec{0} + (Y + S)\vec{v}$. The proof is $\vec{\pi}, \vec{\psi}$ given by

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

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

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

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

[099] By perfect completeness the satisfying witnesses yield proofs where $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} - \iota_T(t) - \vec{\psi} \bullet \vec{v} = \vec{u} \bullet \vec{\pi} = \vec{u} \bullet \Pi\vec{v}$. Conditioned on the random variable Ψ we therefore have that any two possible solutions $\vec{\pi}_1, \vec{\pi}_2$ satisfy $\vec{u} \bullet (\Pi_1 - \Pi_2)\vec{v} = 0$. Since $H_1, \ldots, H_\eta$ generate the matrices H so $\vec{u} \bullet H\vec{v} = 0$ we can write this as $\Pi_1 = \Pi_2 + \sum_{i=1}^{\eta} r_i H_i$. 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 $r_1, \ldots, r_\eta$. We therefore get a uniform distribution over the $\vec{\pi}$ that satisfy the equation conditioned on $\vec{\psi}$. Since $\vec{\psi}$ is uniformly chosen, we conclude that for any witness we get a uniform distribution over $\vec{\psi}, \vec{\pi}$ conditioned on them constituting an acceptable proof. □

5.1 Linear Equations

[0100] 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 $\vec{\psi} := \vec{0}$ and $\vec{\pi} := R^\top \iota_2(\vec{b}) + \sum_{i=1}^{\eta} r_i H_i \vec{v}$. Theorem 1 still applies with $T = 0$. Theorem 2 gives us $p_1(\vec{c}) \cdot p_2(\iota_2(\vec{b})) = p_T(\iota_T(t))$, which will give us soundness. Finally, we have the following theorem.

Theorem 4

[0101] *In the witness-indistinguishable setting where* $\iota_1(G_1) \subseteq \langle u_1, \ldots, u_{m'} \rangle$, $\iota_2(G_2) \subseteq \langle v_1, \ldots, v_{n'} \rangle$ *and* $H_1, \ldots, H_\eta$ *generate the matrices* H *so* $\vec{u} \bullet H\vec{v} = 0$, *the satisfying witnesses* $\vec{x}, \vec{y}, R, S$ *yield the uniform distribution of the proof* $\vec{\pi} \in \langle v_1, \ldots, v_{n'} \rangle^{m'}$ *conditioned on the verification equation* $\vec{c} \bullet \iota_2(\vec{b}) = \iota_T(t) + \vec{u} \bullet \vec{\pi}$ *being satisfied.*

[0102] *Proof.* As in the proof of Theorem 3 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 the matrices H so $\vec{u} \bullet H\vec{v} = 0$ we have that Π has a uniform distribution over the matrices Π satisfying the verification equation. □

5.2 The Symmetric Case

[0103] An interesting special case is when $B := B_1 = B_2$, $m' \leq n'$ with $u_1 = v_1, \ldots, u_{m'} = v_{m'}$ and for the $x, y \in B$ 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 $\vec{\psi}$ with zeroes in the end to extend the length to n', call this vector $\vec{\psi}'$, and revealing the proof $\vec{\phi} = \vec{\pi} + \vec{\psi}'$. In the verification, we check that

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

Theorem 1 and Theorem 3 still hold in this setting. With respect to soundness we have the following theorem.

Theorem 5

[0104] *In the soundness setting, where we have* $p_2(\vec{v}) = \vec{0}$ *a valid proof implies*

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

[0105] *Proof.* An acceptable proof $\vec{\phi}$ satisfies $\iota_1(\vec{a}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{b}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t) + \vec{\phi} \bullet \vec{v}$. The commutative property of the linear and bilinear maps gives us

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

□

[0106] We can simplify the computation of the proof in the symmetric case. We have

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

and extend ψ to ψ' by padding it with $m' - n'$ 0's. Another way to accomplish this padding is by padding T with $m' - n'$ 0-rows and S with $m' - n'$ 0-columns and H_i with $m' - n'$ 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 $\vec{u} \bullet (T' - (T')^{\top})\vec{u} = 0$, so if we have a set $H_1', \ldots, H_{\eta'}'$ that generates the matrices H' so $\vec{u} \bullet H'\vec{u} = 0$, then we can rewrite the proof as

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

6 NIWI Proof for Satisfiability of a Set of Quadratic Equations

[0107] We will now give the full composable NIWI proof for satisfiability of a set of quadratic equations in a module with a bilinear map. The proof will have L_{co}-soundness, where

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

Observe that L_{co}-soundness and soundness are the same notions in the common case where $\iota_1 \circ p_1, \iota_2 \circ p_2$ and $\iota_t \circ p_T$ are the identity maps on respectively A_1, A_2 and A_T.

[0108] The cryptographic assumption we make is that the common reference string is created by one of two algorithms 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.

[0109] Setup: $(gk, sk) := ((\mathcal{R}, A_1, A_2, A_T, f), sk) \leftarrow \mathcal{G}(1^k)$.

[0110] Soundness string: $\sigma := (B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}) \leftarrow K(gk, sk)$.

[0111] Witness-indistinguishability string: $\sigma := (B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}) \leftarrow S(gk, sk)$.

[0112] Proof: The input includes 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 $R \leftarrow \mathrm{Mat}_{m \times m'}(\mathcal{R})$ and $S \leftarrow \mathrm{Mat}_{n \times n'}(\mathcal{R})$ and commit to 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 5. In other words, pick $T_i \leftarrow \mathrm{Mat}_{n' \times m'}(\mathcal{R})$ and $r_{i1}, \ldots, r_{i\eta} \leftarrow \mathcal{R}$ compute

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

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

[0113] Verification: The input is $gk, \sigma, \{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and the proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\psi}_i)\})$. For each equation check

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

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

Theorem 6

[0114] *The protocol given above is a NIWI proof for satisfiability of a set of quadratic equations with perfect completeness, perfect L_{co}-soundness and composable witness-indistinguishability.*

[0115] *Proof.* Perfect completeness follows from Theorem 1.

[0116] Consider a proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\psi}_i)\})$ on a soundness string. Define $\vec{x} := p_1(\vec{c}), \vec{y} := p_2(\vec{d})$. It follows from Theorem 2 that for each equation we have

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

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

[0117] In the present disclosure, a computational assumption is that soundness strings and witness-indistinguishability strings are computationally indistinguishable (or at least computationally similar). 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 3 says that in either equation each of two possible witnesses yield the same distribution on the proof for that equation. A straightforward hybrid argument then shows that we have perfect witness-indistinguishability. □

[0118] 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 it is straightforward to compute the witness $\vec{x} = p_1(\vec{c})$ and $\vec{y} = p_2(\vec{d})$, so the proof is a perfect proof of knowledge.

[0119] Proof size. The size of the common reference string is m' elements in B_1 and n' elements in B_2 in addition to the description of the modules and the maps. The size of the proof is $m + Nn'$ elements in B_1 and $n + Nm'$ elements in B_2.

[0120] Typically, m' and 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.

6.1 NIWI Proofs for Bilinear Groups

[0121] 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 2, 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:** $A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = G_2, A_T = G_T, f(x, \mathcal{Y}) = x\mathcal{Y}$.
- **Quadratic equations in $\mathbb{Z}_{\mathbf{n}}$:** $A_1 = \mathbb{Z}_{\mathbf{n}}, A_2 = \mathbb{Z}_{\mathbf{n}}, A_T = \mathbb{Z}_{\mathbf{n}}, f(x, y) = xy \bmod \mathbf{n}$.

[0122] The common reference string will specify commitment schemes to respectively scalars and group elements. We first commit to 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 equations, i.e., for instance we only commit to a scalar x once and we use the same commitment in the proof whether the equation 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 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.

Example Embodiment 1: Subgroup decision.

[0123] Setup: $(gk, sk) := ((\mathbf{n}, G, G_T, e, \mathcal{P}), (\mathbf{p}, \mathbf{q})) \leftarrow \mathcal{G}(1^k)$, where $\mathbf{n} = \mathbf{pq}$.

[0124] Soundness string: On input (gk, sk) return $\sigma := \mathcal{U}$ where $U := rp\mathcal{P}$ for random $r \in \mathbb{Z}_{\mathbf{n}}^*$.

[0125] Witness-indistinguishability string: On input (gk, sk) return $\sigma := \mathcal{U}$ where $U := r\mathcal{P}$ for random $r \in \mathbb{Z}_{\mathbf{n}}^*$.

[0126] Proof: On input $(\mathbf{n}, G, G_T, e, \mathcal{P}, \mathcal{U})$, a set of equations and a witness $\vec{x}, \vec{\mathcal{Y}}$ do:

1. Commit to each exponent $x_1, \ldots, x_m$ and each element $\mathcal{Y}_1, \ldots, \mathcal{Y}_n$ as respectively $\mathcal{C}_i := x_i\mathcal{P} + r_i\mathcal{U}$ and $\mathcal{D}_i := \mathcal{Y}_i + s_i\mathcal{U}$ for randomly chosen $\vec{r}, \vec{s}$.

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 5.2. Writing it out and doing calculations, we get

$$\phi := \vec{s}^{\top}\vec{\mathcal{A}} + \vec{s}^{\top}(\Gamma + \Gamma^{\top})\vec{\mathcal{Y}} + \vec{s}^{\top}\Gamma\vec{s}\mathcal{U} = \sum_{i=1}^{n} s_i\mathcal{A}_i + \sum_{i=1}^{n}\sum_{j=1}^{n}(\gamma_{ij} + \gamma_{ji})s_i\mathcal{Y}_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}s_is_j\mathcal{U}.$$

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

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

[0127] Verification: On input $(\mathrm{n}, G, G_T, e, \mathcal{P}, \mathcal{U})$, 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 $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ check that $\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{D}_i) \cdot \prod_{i=1}^{n}\prod_{j=1}^{n} e(D_i, D_j)^{\gamma_{ij}} = t_T e(\mathcal{U}, \phi)$.
2. For each multi-scalar multiplication $\vec{a}\cdot\vec{\mathcal{Y}}+\vec{x}\cdot\vec{\mathcal{B}}+\vec{x}\cdot\Gamma\vec{\mathcal{Y}} = \mathcal{T}$ 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 $\mathbb{Z}_{\mathrm{n}}$ check that $\prod_{i=1}^{m} e(\mathcal{C}_i, b_i\mathcal{P}) \cdot \prod_{i=1}^{m}\prod_{j=1}^{m} e(\mathcal{C}_i, \mathcal{C}_j)^{\gamma_{ij}} = e(\mathcal{P}, \mathcal{P})^t e(\mathcal{U}, \phi)$.

[0128] Define L_{co} to be the sets of quadratic equations over $\mathbb{Z}_{\mathrm{n}}$ that are unsatisfiable in the order p subgroups of $\mathbb{Z}_{\mathrm{n}}$, G and G_T.

Theorem 7

[0129] *The NIWI proof given above has perfect completeness, perfect L_{co}-soundness and composable witness-indistinguishability.*

[0130] *Proof.* Perfect completeness follows from Theorem 1. Perfect L_{co}-soundness follows from Theorem 2 since the $\iota \circ p$ maps go to the order p subgroups of $\mathbb{Z}_{\mathrm{n}}$, G and G_T. The subgroup decision problem gives us that we cannot distinguish whether $\mathcal{U}$ has order q or order n so 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 Theorem 3. □

[0131] The size of the 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 is the number of equations.

Example Embodiment 2: SXDH.

[0132] Setup: $gk := (\mathrm{p}, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2) \leftarrow \mathcal{G}(1^k)$.

[0133] Soundness string: On input gk return $\sigma := (u_1, u_2, v_1, v_2)$ from the soundness setup described in Section 4. This gives us $u_2 = t_1 u_1$ and $v_2 = t_2 v_2$ for random $t_1, t_2 \leftarrow \mathbb{Z}_{\mathrm{p}}$ so the elements are linearly dependent.

[0134] [Witness-indistinguishability string:] On input gk return $\sigma := (u_1, u_2, v_1, v_2)$ from the witness-indistinguishability setup described in Section 4. This gives us $u_2 = t_1 u_1 - (\mathcal{O}, \mathcal{P}_1)$ and $v_2 = t_2 v_1 - (\mathcal{O}, \mathcal{P}_2)$ for random $t_1, t_2 \leftarrow \mathbb{Z}_{\mathrm{p}}$.

[0135] 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 group elements $\vec{\mathcal{X}}$ as $\vec{c} := \iota_1(\vec{\mathcal{X}}) + R\vec{u}$ for $R \leftarrow \mathrm{Mat}_{m\times 2}(\mathbb{Z}_\mathbf{p})$ and group elements $\vec{\mathcal{Y}}$ as $d := \iota_2(\vec{\mathcal{Y}}) + S\vec{v}$ for $S \leftarrow \mathrm{Mat}_{m\times 2}(\mathbb{Z}_\mathbf{p})$. Commit to exponents $\vec{x}$ as $\vec{c}' := \iota_1'(x) + \vec{r}u_1$ and exponents y as $d' := \iota_2'(y) + \vec{s}v_1$ for $\vec{r} \leftarrow \mathbb{Z}_\mathbf{p}^{m'}, \vec{s} \leftarrow \mathbb{Z}_\mathbf{p}^{n'}$.

2. For each pairing product equation $(\vec{\mathcal{A}}\cdot\vec{\mathcal{Y}})(\vec{\mathcal{X}}\cdot\vec{\mathcal{B}})(\vec{\mathcal{Y}}\cdot\Gamma\vec{\mathcal{Y}}) = t_T$ make a proof as described in section 5. Writing it out we have for $T \leftarrow \mathrm{Mat}_{2\times 2}(\mathbb{Z}_\mathbf{p})$ the following proof.

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

   For each linear equation $\vec{\mathcal{A}}\cdot\vec{\mathcal{Y}} = t_T$ we use $\vec{\psi} := S^\top\iota_1(\vec{\mathcal{A}})$.

   For each linear equation $\vec{\mathcal{X}}\cdot\vec{\mathcal{B}} = t_T$ we use $\vec{\pi} := R^\top\iota_2(\vec{\mathcal{B}})$.

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 $T \leftarrow \mathrm{Mat}_{1\times 2}(\mathbb{Z}_\mathbf{p})$

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

   For each linear equation $\vec{\mathcal{A}}\cdot\vec{y} = \mathcal{T}_1$ the proof is $\psi := \vec{s}^\top\iota_1(\vec{\mathcal{A}})$.

   For each linear equation $\vec{\mathcal{X}}\cdot\vec{b} = \mathcal{T}_1$ the proof is $\vec{\pi} := R^\top\iota_2'(\vec{b})$.

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 $T \leftarrow \mathrm{Mat}_{2\times 1}(\mathbb{Z}_\mathbf{p})$

$$\begin{aligned}\pi &:= \vec{r}^\top\iota_2(\vec{\mathcal{B}}) + \vec{r}^\top\Gamma\iota_2(\vec{\mathcal{Y}}) + (\vec{r}^\top\Gamma S - T^\top)\vec{v}\\ \vec{\psi} &:= S^\top\iota_1'(\vec{a}) + S^\top\Gamma^\top\iota_1'(\vec{x}) + Tu_1\end{aligned}$$

   For each linear equation $\vec{a}\cdot\vec{\mathcal{Y}} = \mathcal{T}_2$ the proof is $\vec{\pi} := S^\top\iota_1'(\vec{a})$.

   For each linear equation $\vec{x}\cdot\vec{\mathcal{B}} = \mathcal{T}_2$ the proof is $\pi := \vec{r}^\top\iota_2(\vec{\mathcal{B}})$.

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

$$\begin{aligned}\pi &:= \vec{r}^\top\iota_2'(\vec{b}) + \vec{r}^\top\Gamma\iota_2'(\vec{y}) + (\vec{r}^\top\Gamma\vec{s} - T)v_1\\ \varphi &:= \vec{s}^\top\iota_1'(\vec{a}) + \vec{s}^\top\Gamma^\top\iota_1'(\vec{x}) + Tu_1\end{aligned}$$

For each linear equation $\vec{a} \cdot \vec{y} = t$ we use $\psi := \vec{s}^{\top} \iota_1'(\vec{a})$.

For each linear equation $\vec{x} \cdot \vec{b} = t$ we use $\pi := \vec{r}^{\top} \iota_2'(\vec{b})$.

[0136] Verification: On input (gk, σ), a set of equations and a proof $\vec{c}, \vec{d}, \vec{c}', \vec{d}', \{\vec{\pi}_i, \vec{\psi}_i\}_{i=1}^{N}$ do:

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

$$\iota_1(\vec{\mathcal{A}}) \bullet \vec{d} + \vec{c} \bullet \iota_2(\vec{\mathcal{B}}) + \vec{c} \bullet \Gamma\vec{d} = \iota_T(t_T) + \vec{u} \bullet \vec{\pi} + \vec{\psi} \bullet \vec{v}.$$

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 check that

$$\iota_1(\vec{\mathcal{A}}) \bullet \vec{d}' + \vec{c} \bullet \iota_2'(\vec{b}) + \vec{c} \bullet \Gamma\vec{d}' = \tilde{\iota_T}(\mathcal{T}_1) + \vec{u} \bullet \vec{\pi} + F(\psi, v_1).$$

3. For each multi-scalar multiplication $\vec{a} \cdot \vec{\mathcal{Y}} + \vec{x} \cdot \vec{\mathcal{B}} + \vec{x} \cdot \Gamma\vec{\mathcal{Y}} = \mathcal{T}_2$ in G_2 check that

$$\iota_1'(\vec{a}) \bullet \vec{d} + \vec{c}' \bullet \iota_2(\vec{\mathcal{B}}) + \vec{c}' \bullet \Gamma\vec{d} = \hat{\iota_T}(\mathcal{T}_2) + F(u_1, \pi) + \vec{\psi} \bullet \vec{v}.$$

4. For each quadratic equation $\vec{a} \cdot \vec{y} + \vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{y} = t$ in $\mathbb{Z}_{\mathbf{p}}$ check that

$$\iota_1'(\vec{a}) \bullet \vec{d}' + \vec{c}' \bullet \iota_2'(\vec{b}) + \vec{c}' \bullet \Gamma\vec{d}' = \iota_T'(t) + F(u_1, \pi) + F(\psi, v_1).$$

Theorem 8

[0137] *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.*

[0138] Perfect completeness follows from Theorem 1. Perfect soundness follows from Theorem 2 since the $\iota \circ p$ 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 Theorem 3. □

[0139] The modules we work in are $B_1 = G_1^2$ and $B_2 = G_2^2$, so each element in a module includes two group elements from respectively G_1 and G_2. Table 4 list the cost of the different types of equations.

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

Table 4: Cost of each variable and equation measured in elements from G_1 and G_2.

Example Embodiment 3: DLIN.

[0140] Setup: $gk := (\mathbf{p}, G, G_T, e, \mathcal{P}) \leftarrow \mathcal{G}(1^k)$.

[0141] Soundness string: On input gk return $\sigma := (u_1, u_2, u_3)$ from the soundness setup described in Section 4. This gives us $u_3 = t_1 u_1 + t_2 u_2$ for random $t_1, t_2 \leftarrow \mathbb{Z}_{\mathbf{p}}$ so the elements are linearly dependent.

[0142] Witness-indistinguishability string: On input gk return $\sigma := (u_1, u_2, u_3)$ from the witness-indistinguishability setup described in Section 4. This gives us $u_1 = (\alpha\mathcal{P}, \mathcal{O}, \mathcal{P}), u_2 = (\mathcal{O}, \beta\mathcal{P}, \mathcal{P}), u_3 = (\mathcal{O} - \mathcal{P}) + t_1 u_1 + t_2 u_2)$ for random $\alpha, \beta \leftarrow \mathbb{Z}_{\mathbf{p}}^*$ and $t_1, t_2 \leftarrow \mathbb{Z}_{\mathbf{p}}$. Define for notational convenience $\vec{v} := (u_1, u_2)$.

[0143] Proof: On input gk, σ, a set of equations and a witness $\vec{x}, \vec{\mathcal{Y}}$ do:

1. Commit to exponents $\vec{x}$ as $\vec{c} := \iota'(\vec{x}) + R\vec{v}$ for $R \leftarrow \mathrm{Mat}_{m\times 2}(\mathbb{Z}_{\mathbf{p}})$. Commit to group elements $\vec{\mathcal{Y}}$ as $\vec{d} := \iota(\vec{\mathcal{Y}}) + S\vec{u}$ for $S \leftarrow \mathrm{Mat}_{n\times 3}(\mathbb{Z}_{\mathbf{p}})$.

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 5 using the symmetric map F.

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

[0144] For each linear equation $\vec{\mathcal{Y}} \cdot \vec{\mathcal{B}} = t_T$ we use the asymmetric map $\widetilde{F}$ to get the proof

$$\vec{\phi} := S^{\top} \iota(\vec{\mathcal{B}}).$$

We remark that the reason we use the asymmetric $\widetilde{F}$ is that there are no matrices non-trivial H so $\vec{u} \bullet H\vec{u} = 0$, which simplifies the proof. Observe that $\vec{\phi} = \iota(S^{\top}\vec{\mathcal{B}}) = S^{\top}\iota(\vec{\mathcal{B}})$ and vice versa $p(\vec{\phi}) = S^{\top}\vec{\mathcal{B}}$ is easily computable in this special setting, since $\iota(\mathcal{B}_i) = (\mathcal{O}, \mathcal{O}, \mathcal{B}_i)$. We can therefore just reveal the proof $\phi' := p(\vec{\phi}) = S^{\top}\vec{\mathcal{B}}$, which is 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 $\widetilde{F}$. The proof is for random $r_1 \leftarrow \mathbb{Z}_{\mathrm{p}}$

$$\vec{\phi} := R^{\top}\iota(\vec{\mathcal{B}}) + R^{\top}\Gamma\iota(\vec{\mathcal{Y}}) + (S')^{\top}\iota'(\vec{a}) + (S')^{\top}\Gamma^{\top}\iota'(\vec{x}) + R^{\top}\Gamma S'\vec{u} + r_1 H_1 \vec{u}.$$

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

$$\vec{\phi} := S^{\top}\iota'(\vec{b}).$$

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

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

$$\phi := R^{\top}\iota(\vec{\mathcal{B}}).$$

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

4. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{x} = t$ in $\mathbb{Z}_{\mathrm{p}}$ we use the symmetric map F. There is one matrix H_1 that generates the H so $\vec{v} \bullet H\vec{v}$. The proof is for random $r_1 \leftarrow \mathbb{Z}_{\mathrm{p}}$

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

For each linear equation $\vec{x} \cdot \vec{b} = t$ we use the asymmetric map $\widetilde{F}$ to get the proof

$\vec{\phi} := R^{\top}\iota'(\vec{b})$. It suffices to reveal just $R^{\top}\vec{b}$, from which the verifier can compute $\vec{\phi} = \iota'(R^{\top}\vec{b})$.

[0145] Verification: 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 $(\vec{A} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{Y}} \cdot \Gamma\vec{\mathcal{Y}}) = t_T$ check that

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

For each linear equation $\vec{\mathcal{Y}} \cdot \vec{\mathcal{B}} = t_T$ check

$$\vec{d} \,\tilde{\bullet}\, \iota(\vec{\mathcal{B}}) = \iota_T(t_T) + \vec{u} \,\tilde{\bullet}\, \vec{\phi}.$$

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

$$\vec{d} \,\tilde{\bullet}\, \iota'(\vec{b}) = \hat{\iota_T}(\mathcal{T}) + \vec{u} \,\tilde{\bullet}\, \vec{\phi}.$$

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

$$\vec{c} \,\tilde{\bullet}\, \iota(\vec{\mathcal{B}}) = \hat{\iota_T}(\mathcal{T}) + \vec{v} \,\tilde{\bullet}\, \vec{\phi}.$$

3. For each quadratic equation $\vec{x} \cdot \vec{b} + \vec{x} \cdot \Gamma\vec{x} = t$ in $\mathbb{Z}_p$ 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 $\vec{x} \cdot \vec{b} = t$ check

$$\vec{c} \,\tilde{\bullet}\, \iota'(\vec{b}) = \iota'_T(t) + \vec{v} \,\tilde{\bullet}\, \vec{\phi}.$$

Theorem 9

[0146] *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.*

[0147] Perfect completeness follows from Theorem 1. Perfect soundness follows from Theorem 2 since the $\iota \circ p$ 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 Theorem 5. □

[0148] The module we work in is $B = G^3$, so each element in the module includes 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 a shorter proof. Table 5 list the cost of the different types of equations.

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: $\vec{\mathcal{Y}} \cdot \vec{\mathcal{B}} = t_T$	3	0
Multi-scalar multiplication equations	9	0
- Linear equation: $\vec{\mathcal{Y}} \cdot \vec{b} = \mathcal{T}$	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 5: Cost of each variable and equation measured in elements from G.

7 Zero-Knowledge

[0149] We will show that in many cases it is possible to make zero-knowledge proofs for satisfiability of quadratic equations. One strategy is to use the NIWI proofs directly, however, such proofs may not be zero-knowledge because the zero-knowledge simulator may not be able to compute any witness for satisfiability of the equations. However, we can often modify the set of quadratic equations into an equivalent set of quadratic equations where a witness can be found.

[0150] We consider first the case where $A_1 = \mathcal{R}, A_2 = A_T, f(r, y) = ry$ and where S outputs an extra piece of information τ that makes it possible to trapdoor open the commit-

ments in B_1. More precisely, τ permits the computation of $\vec{s} \in \mathcal{R}^{m'}$ so $\iota_1(1) = \iota_1(0) + \vec{s}^{\top}\vec{u}$. We remark that this is a common case; in bilinear groups both multi-scalar multiplication equations in G_1, G_2 and quadratic equations in $\mathbb{Z}_n$ have this structure.

[0151] Define $c = \iota_1(1)$ to be a commitment to $\phi = 1$. Let us rewrite the equations in the statement as

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

We have introduced a new variable ϕ and if we choose the variables in these modified equations to be 0 then we have a satisfying witness. In the simulation, we give the simulator trapdoor information that permits it to open c to 0 and we can now use the NIWI proof from Section 6.

[0152] **Setup:** $(gk, sk) := ((\mathcal{R}, A_1, A_2, A_T, f), sk) \leftarrow \mathcal{G}(1^k)$.

[0153] **Soundness string:** $\sigma := (B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}) \leftarrow K(gk, sk)$.

[0154] **Proof:** This protocol is exactly the same as in the NIWI proof. The input includes 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 $R \leftarrow \mathrm{Mat}_{m\times m'}(\mathcal{R})$ and $S \leftarrow \mathrm{Mat}_{n\times n'}(\mathcal{R})$ and commit to 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 5. In other words, pick $T_i \leftarrow \mathrm{Mat}_{n'\times m'}(\mathcal{R})$ and $r_{i1}, \ldots, r_{i\eta} \leftarrow \mathcal{R}$ and compute

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

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

[0155] **Verification:** The input is gk, σ, $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and the proof $(\vec{c}, \vec{d}, \{(\vec{\pi}_i, \vec{\psi}_i)\})$. For each equation check

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

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

[0156] **Simulation string:** $(\sigma, \tau) := ((B_1, B_2, B_T, F, \iota_1, p_1, \iota_2, p_2, \iota_T, p_T, \vec{u}, \vec{v}), \vec{s}) \leftarrow S_1(gk, sk)$, where $\iota_1(1) = \iota_1(0) + \sum_{i=1}^{m'} s_i u_i$.

[0157] **Simulated proof:** The input includes gk, σ, a list of quadratic equations $\{(\vec{a}_i, \vec{b}_i, \Gamma_i, t_i)\}_{i=1}^N$ and a satisfying witness $\vec{x}, \vec{y}$.

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

Pick at random $R \leftarrow \mathrm{Mat}_{m \times m'}(\mathcal{R})$ and $S \leftarrow \mathrm{Mat}_{n \times n'}(\mathcal{R})$ and commit to the the variables as $\vec{c} := \vec{0} + R\vec{u}$ and $\vec{d} := \vec{0} + S\vec{v}$. We have $c := \iota_1(1) = \iota_1(0) + \sum_{i=1}^{m'} s_i u_i$.

For each modified equation $(\vec{a}_i, \vec{b}_i, -t_i, \Gamma_i, 0)$ make a proof as described in Section 5. Return the simulated proof $\{(\vec{c}, \vec{d}, \vec{\pi}_i, \vec{\psi}_i)\}_{i=1}^{N}$.

Theorem 10

[0158] *The protocol described above is a composable NIZK proof for satisfiability of pairing product equations with perfect completeness, perfect L_{co}-soundness and composable zero-knowledge.*

[0159] *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 $c = \iota_1(0) + \sum_{i=1}^{m'} s_i u_i$. It therefore knows a witness $\vec{0}, \vec{0}, \phi = 0$ for satisfiability of the modified equations. It therefore outputs a proof $\{(\vec{c}, \vec{d}, \vec{\pi}_i, \vec{\psi}_i)\}_{i=1}^{N}$ such that for i we have

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

The commutative properties 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 proof satisfies the equation the verifier checks. Perfect completeness on a simulation string now follows from the perfect completeness of the NIWI proof as well.

[0160] Perfect L_{co}-soundness follows from the perfect L_{co}-soundness of the NIWI proof.

[0161] 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 $\vec{x}, \vec{y}, \phi = 1$ or $\vec{0}, \vec{0}, \phi = 0$. Theorem 3 now tells us that the proofs $\vec{\pi}_i, \vec{\psi}_i$ made with either type of opening of $\vec{c}, \vec{d}, c$ are uniformly distributed over the possible choices of $\{(\vec{\psi}_i, \vec{\pi}_i)\}_{i=1}^{N}$ 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. □

7.1 NIZK Proofs for Bilinear Groups

[0162] Let us return to the four types of quadratic equations given in Table 1. If we set up the common reference string such that we can trapdoor open respectively $\iota_1'(1)$ and $\iota_2'(1)$ to 0

then multi-scalar multiplication equations and quadratic equations in $\mathbb{Z}_n$ are of the form for which we can give zero-knowledge proofs (at no additional cost).

[0163] 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 satisfiability witness. We do observe though that in the special case, where $t_T = 1$ the choice of $\vec{\mathcal{X}} = \vec{\mathcal{O}}, \vec{\mathcal{Y}} = \vec{\mathcal{O}}$ is a satisfactory witness. Since we also use $\vec{\mathcal{X}} = \vec{\mathcal{O}}, \vec{\mathcal{Y}} = \vec{\mathcal{O}}$ in the other zero-knowledge proofs, the simulator can use this witness and give a NIWI proof. In the special case where $t_T = 1$ we can therefore make NIZK proofs for satisfiability of the set of pairing product equations.

[0164] Next, let us look at the 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 $\mathcal{P}_i, \mathcal{Q}_i$. In this case, we can add linear equations $\mathcal{Z}_i = \mathcal{P}_i$ to the set of multi-scalar multiplication equations in G_1. We already know that such equations have zero-knowledge proofs. We can now rewrite the pairing product equation as $(\vec{\mathcal{A}} \cdot \vec{\mathcal{Y}})(\vec{\mathcal{X}} \cdot \vec{\mathcal{B}})(\vec{\mathcal{Z}} \cdot \vec{\mathcal{Q}})(\vec{\mathcal{X}} \cdot \Gamma\vec{\mathcal{Y}}) = 1$. This is a pairing product equation of the type where we can make a zero-knowledge proof. We can therefore also make zero-knowledge proofs for a set of quadratic equations over a bilinear group if 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 $\mathcal{P}_i, \mathcal{Q}_i$.

[0165] The case of pairing product equations points to a couple of differences between witness-indistinguishable proofs and zero-knowledge proofs using the techniques herein. 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.

8 Application Embodiments

8.1 Fair Key Exchange

[0166] Suppose two parties want to exchange a pair of keys. However, neither party wants to give away his key without having assurance that the other party will give him a key in return. In the standalone setting, there is no fair protocol to implement this objective, since either party may abort the protocol after learning his output. If we introduce a trusted party, the problem is of course easily solved, the parties can hand their keys to the trusted party that then gives each party the desired key.

[0167] In one embodiment, each party encrypts the key under the public key of a trusted

party. Now both parties exchange their keys. If either party aborts, the other party can call on the trusted party to get his key, however, if both parties act honestly there will be no need to call upon the trusted party. This way, we reduce the burden on the honest party that is only invoked in case of protocol breaches.

[0168] Verifiable encryption can be used to solve this problem. In addition to encrypting the key, we also make a proof that we have encrypted a proper key. Of course this proof should not reveal the nature of the key we are encrypting.

[0169] Given witness indistinguishable proofs, it is straightforward to construct NIZK proofs. We can therefore use an NIZK proof to prove that we have encrypted a proper key. This NIZK proof suffices for our purpose, since it guarantees the correctness of the encryption, yet reveals nothing else. The present techniques makes these NIZK proofs efficient enough to be practical, when we set up the cryptosystem in groups with bilinear maps. We therefore get a satisfactory solution to the fair key exchange problem.

8.2 Verifiable Encryption

In one embodiment, the NIZK proof system is used to provide verifiable encryption. Figure 1 shows a first computer 101, a second computer 102 provided to a computer network 103 for exchanging encrypted data, and, optionally, a third party computer 105. One of ordinary skill in the art will recognize that one or more of the computers 101, 102, and 105 can be combined to provide the functionality shown in Figure 1. A message is encrypted in the computer 101 and sent to the second computer 103 where it can be decrypted and displayed or the second computer 103 can use techniques as describe herein to use an NIZK proof of membership for one or more aspects of the encrypted message without decrypting (or even being able to decrypt) the message. In one embodiment, a third computer (not shown) is provided as a third party that uses a proof of membership algorithm to verify one or more aspects of the encrypted message without decrypting the message. This allows two parties to exchange information while using a third party to verify one or more aspects of the message without revealing the contents of the message to the third party. Thus the NIZK proof allows two parties to communicate through a third party (e.g., an escrow party) who verifies aspects of the message and/or escrows the messages. This permits encryption of a message $\mathcal{X}$ and construction of an NIZK proof that $\mathcal{X}$ satisfies a certain equation. For the purpose of this example I have chosen the equation $e(\mathcal{X}, \mathcal{Q} + m\mathcal{P}) = e(\mathcal{P}, \mathcal{P})$. This equation has practical value, such an $\mathcal{X}$ is a Boneh-Boyen signature on m (with public verification key $\mathcal{Q}$). So it is

a verifiable encryption of a signature on m. Those skilled in the art can extend the verifiable encryption scheme to encrypt multiple messages and prove that they simultaneously satisfy multiple equations. Figure 2 is a flow diagram of key generation in a system for verifiable encryption. Figure 3 is a flow diagram of encryption in the system of Figure 2. Figure 4 is a flow diagram of generation of a verification proof of membership in the system of Figure 2. Figure 5 is a flow diagram of decryption in the system of Figure 2.

8.3 Verifiable encryption based on the DLIN embodiment

[0170] Verifiable encryption includes key generation, encryption, and verification.

- Key generation: Generate bilinear group $gk = (p, G, G_T, e, \mathcal{P}) \leftarrow \mathcal{G}(1^k)$. Pick a soundness reference string for the NIWI proof $\sigma = (u_1, u_2, u_3)$ $(\in G^{3\times 3})$. Pick at random $a, b \leftarrow \mathbb{Z}_p^*$ and set $\mathcal{A} = a\mathcal{P}$ and $\mathcal{B} = b\mathcal{P}$.

  The public key is $pk = (gk, \sigma, \mathcal{A}, \mathcal{B})$. The secret decryption key is $sk = (a, b)$.

- Encryption: To encrypt a message $\mathcal{X}$ pick at random $r, s \leftarrow \mathbb{Z}_p$ and let the ciphertext be $c = (\mathcal{U}, \mathcal{V}, \mathcal{W}) = (r\mathcal{A}, s\mathcal{B}, \mathcal{X} + (r+s)\mathcal{P})$.

- Verification proof: To prove that the ciphertext $c = (\mathcal{U}, \mathcal{V}, \mathcal{W})$ contains $\mathcal{X}$ satisfying $e(\mathcal{X}, \mathcal{Q} + m\mathcal{P}) = e(\mathcal{P}, \mathcal{P})$ we need to prove

$$\exists r, s, \mathcal{X} : \mathcal{U} = r\mathcal{A} \ \wedge \ \mathcal{V} = s\mathcal{B} \ \wedge \ \mathcal{W} = \mathcal{X} + (r+s)\mathcal{P} \ \wedge \ e(\mathcal{X}, \mathcal{Q} + m\mathcal{P}) = e(\mathcal{P}, \mathcal{P}).$$

Since we need an NIZK proof, we start by rewriting the equations as described in section 7:

$$\exists \phi, r, s, \mathcal{X}, \mathcal{U}', \mathcal{V}', \mathcal{W}', \mathcal{Q}' :$$

$$\phi = 1 \bmod p \wedge \mathcal{U}' = \phi\mathcal{U} \wedge \mathcal{V}' = \phi\mathcal{V} \wedge \mathcal{W}' = \phi W + (1-\phi)\mathcal{P} \wedge \mathcal{Q}' = \phi(\mathcal{Q} + m\mathcal{P}) + (1-\phi)\mathcal{P}$$

$$\wedge \mathcal{U}' = r\mathcal{A} \ \wedge \ \mathcal{V}' = s\mathcal{B} \ \wedge \ \mathcal{W}' = \mathcal{X} + (r+s)\mathcal{P} \ \wedge \ e(\mathcal{X}, \mathcal{Q}') = e(\mathcal{P}, \mathcal{P}).$$

Observe, $\phi = 1$ and $r, s, \mathcal{X}$ chosen as in the encryption phase gives a satisfying witness for the statement (provided we have indeed encrypted a Boneh-Boyen signature $\mathcal{X}$).

[0171] As in Section 7, by choosing a simulation reference string, we can typically open ϕ to both 0 and 1. This means we can choose $r = 0, s = 0, \mathcal{X} = \mathcal{P}, \mathcal{U}' = \mathcal{O}, \mathcal{V}' = \mathcal{O}, \mathcal{W}' =$

$\mathcal{P}, \mathcal{Q}' = \mathcal{P}$ such that all equations are satisfied. This in turn means we can make a ZK simulation of the proof, without knowing how the ciphertext c was generated.)

[0172] We now give an NIWI proof $\vec{\pi}$ as described in Section 6 for all the equations above being simultaneously satisfiable.

- Verifying ciphertext and proof: Write out the equations as described above and verify the NIWI proof $\vec{\pi}$.
- Decryption: To decrypt $c = (\mathcal{U}, \mathcal{V}, \mathcal{W})$ using the secret decryption key (a, b) compute $\mathcal{X} = \mathcal{W} - a^{-1}\mathcal{U} - b^{-1}\mathcal{V}$.

8.4 Mix-nets

[0173] Figure 6 shows a mix-net system wherein a plurality of senders 601 and a plurality of mix-net servers 602 are provided to a network 604. A mix-net takes a set of messages from one or more senders 601 as input and publishes them in random order (e.g., to one or more receivers 603. At the recievers 603, the message can be decrypted and displayed. The sender of each message is thus hidden among all the other senders, so it provides some degree of anonymity. Mix-nets are for instance used in internet-voting protocols, anonymous broadcast protocols, etc. The goal for the parties is to publish a message without revealing the sender. One place where this is useful is in internet-voting protocols, where voters anonymously publish their votes.

[0174] A standard way of constructing mix-nets is to use a homomorphic cryptosystem, since such ciphertexts can be rerandomized. The senders encrypt their intended message and send them to the mix-net. The mix-servers one by one take the encrypted messages, permute them and rerandomize them. After they have all rerandomized and permuted the ciphertexts, they use threshold decryption to get out the ciphertext. Provided just one server is honest, the ciphertexts get permuted completely and thus loose their link to the sender. This is what gives us anonymity. It is of course important that the decryption keys are shared between the servers, such that no single server can decrypt the incoming or intermediate ciphertexts.

[0175] The construction described here works well as long as the servers are honest but curious. However, it is easy to imagine a setting where a server might wish to replace messages with other messages, for instance votes for a particular candidate. To guard against this, it has been suggested to provide a proof of correctness of the shuffle, i.e., the permutation

and rerandomization of the ciphertexts. Such a proof would guarantee that no messages are replaced. However, it is of course important that this proof keeps the permutation secret.

[0176] Research in this area has resulted in a number of interactive proofs for correctness of a shuffle that hide the permutation. To minimize server interaction, it is desirable to reduce the round complexity and several 3-move schemes have been suggested.

[0177] Some use a permutation network based approach for proving the correctness of a shuffle. The idea is to write the permutation as $n \log n$ potential transpositions. For each transposition, we may choose to transpose the ciphertexts or choose not to transpose them, thus giving us the potential of selecting all of the $n!$ possible permutations. By publishing the intermediate ciphertexts in this network and by making a proof for each potential transposition that we have transposed them or not, it is straightforward to build an interactive zero-knowledge proof for the correctness of the shuffle.

[0178] The witness-indistinguishable proofs in the present disclosure will give us the first non-interactive shuffle proof. We first describe our setup. We need a homomorphic cryptosystem, for instance the one based on the DLIN problem, i.e., we encrypt $m \in G$ as $(f^r, h^s, g^{r+s}m)$. This cryptosystem is obviously, semantically secure and homomorphic and it is easy to set up a threshold decryption structure for it. The mix-servers will in addition also publish (u, v, w). We will now encrypt as $(f^r u^t, h^s v^t, g^{r+s} w^t m)$. If $u = f^x, v = h^y, w = g^{x+y}$ this is fine, we just get a slightly more complicated way of encrypting m. However, if we set it up with $u = f^x, v = h^y, w = g^z$ for $z \neq x+y$, then we have a perfectly hiding commitment scheme instead.

[0179] We can now make Abe's shuffle proof non-interactive as follows. We compute all the intermediate ciphertexts in the network. For each potential transposition, we now make a WI proof that either we transposed the ciphertext or we kept them in place, i.e., in either case we did not introduce new messages into the shuffle. In the perfect binding case, i.e., when $u = f^x, v = h^y, w = g^{x+y}$ we can set up the proof with perfect soundness. This means we have a non-interactive proof that the shuffle is correct. On the other hand, we may compare with using $u = f^x, v = h^y, w = g^z$ in which case we have perfectly binding commitments. In this case, there are many possible witnesses and we can set up our proofs so they are perfectly witness indistinguishable. We can therefore argue that the permutation is computationally hidden because the cryptosystem setup that we use in the mix-net is computationally indistinguishable from the perfect hiding setup, where we do not reveal the permutation.

[0180] The mix-net is run by a set of mix-servers $M_1, \ldots, M_N$. Each sender encrypts his message (for privacy) and sends it to the mix-net. We will now describe what the mix-servers do with the ciphertexts. The first mix-server M_1 permutes and re-randomizes the ciphertexts. It also provides an NIZK proof for having permuted and re-randomized correctly (otherwise it would be able to replace some ciphertexts and thus alter the messages). The second mix-server M_2 permutes and re-randomizes the output from M_2. It also provides an NIZK proof for having done this correctly. The mix-servers continue like this until all of them have permuted and re-randomized the ciphertexts. If at least one of the mix-servers is honest the messages have now been permuted and re-randomized so it is impossible to trace them back to the senders. The mix-servers now cooperate to decrypt.

[0181] A mix-net showing permutations of the messages in rows 1,...N. (The messages are encrypted, so outsiders do not actually see these permutations).

Input : 1 2 3 4 5 6

M1 out: 6 3 4 2 1 5

M2 out: 2 4 5 1 6 3

[0182] Each mix-server permutes and re-randomizes all the ciphertexts that the previous mix-server outputs. It must prove that this has been done correctly. This can be done by creating a permutation network of $\log N$ layers. In each layer, we have $N/2$ pairs of ciphertexts, which can either pass on to the next layer after re-randomization or be swapped and re-randomized. (Any permutation of N elements can be built from $N \log N$ swaps/not swaps)

A permutation of N elements built from swaps/non-swaps of pairs of messages. E.g.

Layer 1: 1 2 3 4 5 6

Layer 2: 2 1 4 3 5 6 swapping/non-swapping neighbours

Layer 3: 3 6 4 2 5 1 swapping/non-swapping 3 spaces apart

Layer 4: 6 3 4 2 1 5 swapping/non-swapping neighbours

The key operation is therefore an NIZK proof of having swapped or not swapped two ciphertexts.

[0183] Figure 7 is a flow diagram of key generation in the system of Figure 6. Figure 8 is a flow diagram of encryption in the system of Figure 6. Figure 9 is a flow diagram of re-randomization in the system of Figure 6. Figure 10 is a flow diagram of an NIZK proof of membership in the system of Figure 6.

8.5 Encryption with swap/non-swap NIZK proofs based on SXDH embodiment

[0184] Encryption with swap/non-swap NIZK proofs includes key generation, encryption, etc.

Key generation: Generate a group $gk = ((n, G_1, G_2, G_T, e, \mathcal{P}_1, \mathcal{P}_2)) \leftarrow \mathcal{G}(1^k)$. Generate a soundness reference string σ as described in Section 6. Generate an encryption key by selecting at random $a \leftarrow \mathbb{Z}_p$ and setting $\mathcal{A} = a\mathcal{P}_1$. The public key is $(gk, \sigma, \mathcal{A})$ and the secret decryption key is a.

Encryption: To encrypt a message $\mathcal{X} \in G_1$ pick at random $r \leftarrow \mathbb{Z}_p$ and let the ciphertext be $(\mathcal{U}, \mathcal{V}) = (r\mathcal{P}_1, \mathcal{X} + r\mathcal{A})$.

Re-randomization: To re-randomize a cipertext $(\mathcal{U}, \mathcal{V})$ pick at random $s \leftarrow \mathbb{Z}_p$ and set $(\mathcal{U}', \mathcal{V}') = (\mathcal{U} + s\mathcal{P}_1, \mathcal{V} + s\mathcal{A})$.

NIZK swap proof: Given input ciphertexts $(\mathcal{U}_1, \mathcal{V}_1)$ and $(\mathcal{U}_2, \mathcal{V}_2)$ and output ciphertext $(\mathcal{U}_1', V_1'), (U_2', V_2')$ we want to make an NIZK proof for them being swapped or not swapped

$$\exists r, s :$$

$$\mathcal{U}_1' = \mathcal{U}_1 + r\mathcal{P}_1 \ \wedge \ \mathcal{V}_1' = \mathcal{V}_1 + r\mathcal{A} \ \wedge \ U_2' = \mathcal{U}_2 + r\mathcal{P}_1 \ \wedge \ V_2' = \mathcal{V}_2 + r\mathcal{A}$$

$$\text{OR } \mathcal{U}_1' = \mathcal{U}_2 + s\mathcal{P}_1 \ \wedge \ V_1' = \mathcal{V}_2 + s\mathcal{A} \ \wedge \ \mathcal{U}_2' = \mathcal{U}_1 + r\mathcal{P}_1 \wedge \mathcal{V}_2' = \mathcal{V}_2 + r\mathcal{A}.$$

As in Section 7 we do this by rewriting the equations as

$$\exists \phi, r, s : \phi(\phi - 1) = 0$$

$$\phi(\mathcal{U}_1' - \mathcal{U}_1 - r\mathcal{P}_1) = \mathcal{O} \ \wedge \ \phi\tau(\mathcal{V}_1' - \mathcal{V}_1 - r\mathcal{A}) = \mathcal{O}$$
$$\wedge \ \phi(U_2' - \mathcal{U}_2 - r\mathcal{P}_1) = \mathcal{O} \ \wedge \ \phi(V_2' - \mathcal{V}_2 - r\mathcal{A}) = 1$$

$$\text{and } (1 - \phi)(\mathcal{U}_1' - \mathcal{U}_2 - s\mathcal{P}_1) = \mathcal{O} \ \wedge \ (1 - \phi)(V_1' - \mathcal{V}_2 - s\mathcal{A}) = \mathcal{O}$$
$$\wedge \ (1 - \phi)(\mathcal{U}_2' - \mathcal{U}_1 - r\mathcal{P}_1) = \mathcal{O} \wedge (1 - \phi)(\mathcal{V}_2' - \mathcal{V}_2 - r\mathcal{A}) = \mathcal{O}.$$

We then give an NIWI proof $(\vec{\pi}, \vec{\psi})$ as in Section 6 for these equations being simultaneously satisfiable.

Verifying swap proof: Verify the NIWI proof $(\vec{\pi}, \vec{\psi})$ for the equations above.

Decryption: To decrypt $(\mathcal{U}, \mathcal{V})$ compute $\mathcal{X} = \mathcal{V} - a\mathcal{U}$.

8.6 Blind Signatures

[0185] In blind signatures, there is a signing server and a set of users. The users should be able to obtain signatures on messages of their choice from the signing server. At the same time, the signing server should not learn, which message it is signing. Blind signatures has application in e-cash and anonymous credentials.

[0186] It is straightforward to construct a blind signature scheme using the present witness-indistinguishable techniques. The server will have a verification key for a signature scheme as well as a public key for a commitment scheme. There will be two types of keys for the commitment scheme, one being such that a secret decryption key can be used to extract messages. The other type of public key will give a perfectly hiding commitment. The blind signature protocol now works as follows. The user commits to his message and send it to the signing server. The signing server signs this message. The user can now take his message and create a WI proof for having a commitment to a pair of a commitment to the message and a signature on this message. Since the commitment is perfectly hiding and the WI proof perfectly witness-indistinguishable, there is no way to link the message and the original input to the server.

8.7 Ring Signatures

[0187] In ring signatures, we have a bunch of public verification keys for various users. We want to make a signature such that we know one of the users have signed, yet we do not want to reveal which user signed the message. This could for instance be useful in whistleblower-cases for instance, enabling employees of a company to anonymously identify themselves as being from the particular company and testify to malpractice, yet remain anonymous. The central idea in this protocol is that the signer makes a witness-indistinguishable proof of knowledge that he knows the signature on the message under one of the keys, yet does not reveal which of the verification keys the signature correspond to.

[0188] The above disclosure shows the construction of efficient non-interactive cryptographic proofs for use in bilinear groups. These proofs can be instantiated with many different types of bilinear groups and the security of the proofs can be based on many different types of intractability assumptions, of which we have given various example embodiments and applications. One of ordinary skill in the art will recognize that other embodiments will be apparent from the disclosure. For example, the embodiments shown are based on the modules on bilinear groups. One of ordinary skill in the art will recognize that these techniques do not require the modules to be cyclic as is the case for bilinear groups. Other types of modules with a bilinear map exist, which are not constructed from bilinear groups.

[0189] While the present disclosure has been described in connection with various embodiments , it is understood that similar aspects may be used or modifications and additions may be made to the described aspects of the disclosed embodiments for performing the same function of the present disclosure without deviating therefrom. Therefore, the present disclosure should not be limited to any single aspect, but rather construed in breadth and scope in accordance with the appended claims.

References

[Bar06] Paulo Barreto. The pairing-based crypto lounge, 2006. Available at `http://paginas.terra.com.br/informatica/paulobarreto/pblounge`

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *proceedings of CRYPTO '04, LNCS series, volume 3152*, pages 41–55, 2004.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *proceedings of EUROCRYPT '04, LNCS series, volume 3027*, pages 506–522, 2004.

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

[BGdMM05] Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose. Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive, Report 2005/417, 2005. Available at `http://eprint.iacr.org/2005/417`.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *proceedings of TCC '05, LNCS series, volume 3378*, pages 325–341, 2005.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *proceedings of EUROCRYPT '06, LNCS series, volume 4004*, pages 573–592, 2006.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In *proceedings of EUROCRYPT '06, LNCS series, volume 4004*, pages 427–444, 2006.

[DBS04] Ratna Dutta, Rana Barua, and Palash Sarkar. Pairing-based cryptographic protocols : A survey. Cryptology ePrint Archive, Report 2004/064, 2004. `http://eprint.iacr.org/`.

[GL07] Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In *proceedings of ASIACRYPT '07, LNCS series, volume 4833*, pages 51–67, 2007.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS '06*, pages 89–98, 2006.

[GR04] Steven D. Galbraith and Victor Rotger. Easy decision Diffie-Hellman groups. *London Mathematical Society Journal of Computation and Mathematics*, 7:201–218, 2004.

[Gro06] Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In *proceedings of ASIACRYPT '06, LNCS series*, 2006. Full paper available at `http://www.brics.dk/~jg/NIZKGroupSignFull.pdf`.

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. In *proceedings of ASIACRYPT '06, LNCS series*, 2007. Full paper available at `http://www.brics.dk/~jg/NIZKGroupSignFull.pdf`.

[Sco02] Mike Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number. Cryptology ePrint Archive, Report 2002/164, 2002. Available at `http://eprint.iacr.org/2002/164`.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *proceedings of EUROCRYPT '05, LNCS series, volume 3494*, pages 457–473, 2005.

[Ver04] Eric R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *proceedings of EUROCRYPT '05, LNCS series, volume 3494*, pages 114–127, 2005.

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

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 $A_1 \times A_2 \to A_T$ and $F : B_1 \times B_2 \to B_T$.

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

Properties that follows from bilinearity:

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

Commutative diagram of maps in setup.

$$\begin{array}{ccccc} A_1 & \times & A_2 & \xrightarrow{f} & A_T \\ \iota_1 \downarrow\uparrow p_1 & & \iota_2 \downarrow\uparrow p_2 & & \iota_T \downarrow\uparrow p_T \\ B_1 & \times & B_2 & \xrightarrow{F} & B_T \end{array}$$

Commutative properties:

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

Equations.

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

(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: $\vec{u} \in B_1^{m'}, \vec{v} \in B_2^{n'}$.

Commitments:

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

NIWI proofs.

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

Randomness in proofs: $T \leftarrow \mathrm{Mat}_{m' \times n'}(\mathcal{R})$, $r_1, \ldots, r_\eta \leftarrow \mathcal{R}$.

Proofs:

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

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

WHAT IS CLAIMED IS:

1. A method for verification of encrypted data, comprising:

encrypting data using an encryption algorithm related to groups with a bilinear map to produce encrypted data; and

using a witness-independent algorithm to verify said data.

2. The method of Claim 1, further comprising sending said encrypted data as part of a key exchange.

3. The method of Claim 1, further comprising sending said encrypted data as part of a non-interactive shuffle.

4. An apparatus for verification of encrypted data, comprising:

a computer memory provided to a computer processor; and

a program loaded into said computer memory, said program configured to verify encrypted data using a witness-independent algorithm and according to selected groups with a bilinear map, wherein said witness-independent algorithm uses commitments of variables from said bilinear map to verify said encrypted data.

5. A method for generating a proof of membership, the method comprising:

receiving a common reference string comprising a group order, a description of a first group having the group order, a description of a second group having the group order, a description of a bilinear map from the first group to the second group, a first generator of the first group, and a second generator of a proper nontrivial subgroup of the first group;

receiving a message from a first computing entity;

identifying a ciphertext encrypting the message;

determining a proof value comprising a triple of values from the first group, said triple of values generated using a unit from the group of integers modulo the group order, the first generator, the second generator, the message, and the secret integer value; and

communicating the proof value to a second computing entity.

6. The method as recited in Claim 5, wherein identifying a ciphertext comprises receiving the ciphertext from the first computing entity.

7. The method as recited in Claim 5, wherein identifying a ciphertext comprises computing the ciphertext using at least the first generator, the message, the second generator, and a secret integer value.

8. A proof system comprising:

a common reference string computed from a group order, a description of a first group having the group order, a description of a second group having the group order, a description of a bilinear map from the first group to the second group, a first generator of the first group, wherein said common reference string is computed from commitments of variables mapped by said bilinear map;

a message, said message having been generated by a first computing entity;

a ciphertext representing an encryption of the message, said ciphertext having been generated using elements of the common reference string and a secret integer value;

a proof value comprising a plurality of values from the first group, said plurality of values generated using a unit from the group of integers modulo the group order, the first generator, the second generator, the message, and the secret integer value; and

a communications module for communicating the proof value to a second computing entity.

9. The system of claim 8 wherein the plurality is a triple.

10. The system of Claim 8, further comprising: a verifier configured to receive the common reference key, the ciphertext, and the proof value and to verify that relationships hold when the bilinear map is applied to selected elements of the common reference key, the ciphertext, and the proof value.

11. A method of verifying a proof, the method comprising:

receiving a common reference key comprising a group order, a description of a first group having the group order, a description of a second group having the group order, a description of a bilinear map from the first group to the second group, a first generator of the first group, and a second generator of a proper subgroup of the first group;

receiving a ciphertext encrypting a message;

receiving a proof value, said proof value comprising a triple of values from the first group;

using the bilinear map, the first generator, the second generator, the ciphertext, and the proof value to determine whether the ciphertext encrypts a value from a set of values; and

generating a signal representative of the determination.

12. A method for generating a proof, the method comprising:

receiving a common reference string computed at least in part from, a description of a first group, said first group comprising a DLIN group, a description of a second group having the group order, a description of a bilinear map from the first group to the second group, a plurality of generators of the first group, and a first plurality of values from the first group, wherein the first plurality of values from the first group have a relationship to the plurality of generators;

receiving a message from a first computing entity;

identifying a ciphertext encrypting the message and comprising a second plurality of values from the first group, the values of the second plurality of values from the first group determined at least in part by a relationship of the plurality generators;

determining a proof value comprising a matrix of values from the first group, said matrix of values computed at least in part from commitments to variables in said relationship and from a satisfying witness to said relationship; and

communicating the proof value to a second computing entity.

13. The method as recited in Claim 12, wherein identifying a ciphertext comprises receiving the ciphertext from the first computing entity.

14. The method as recited in Claim 12, wherein identifying a ciphertext comprises computing the ciphertext.

15. A system for generating a proof, the system comprising:

a common reference string comprising a prime group order p, a description of a first group as an SXDH group, a description of a second group, a description of a bilinear map from the first group to the second group, a plurality of generators of the first group, and a first plurality of values from the first group;

a message received from a first computing entity;

a ciphertext encrypting the message and satisfying at least one first equation;

a proof value computed from proof equations, wherein coefficients for said proof equations are computed at least in part form commitments to said first equation; and

a communications module for communicating the proof value to a second computing entity.

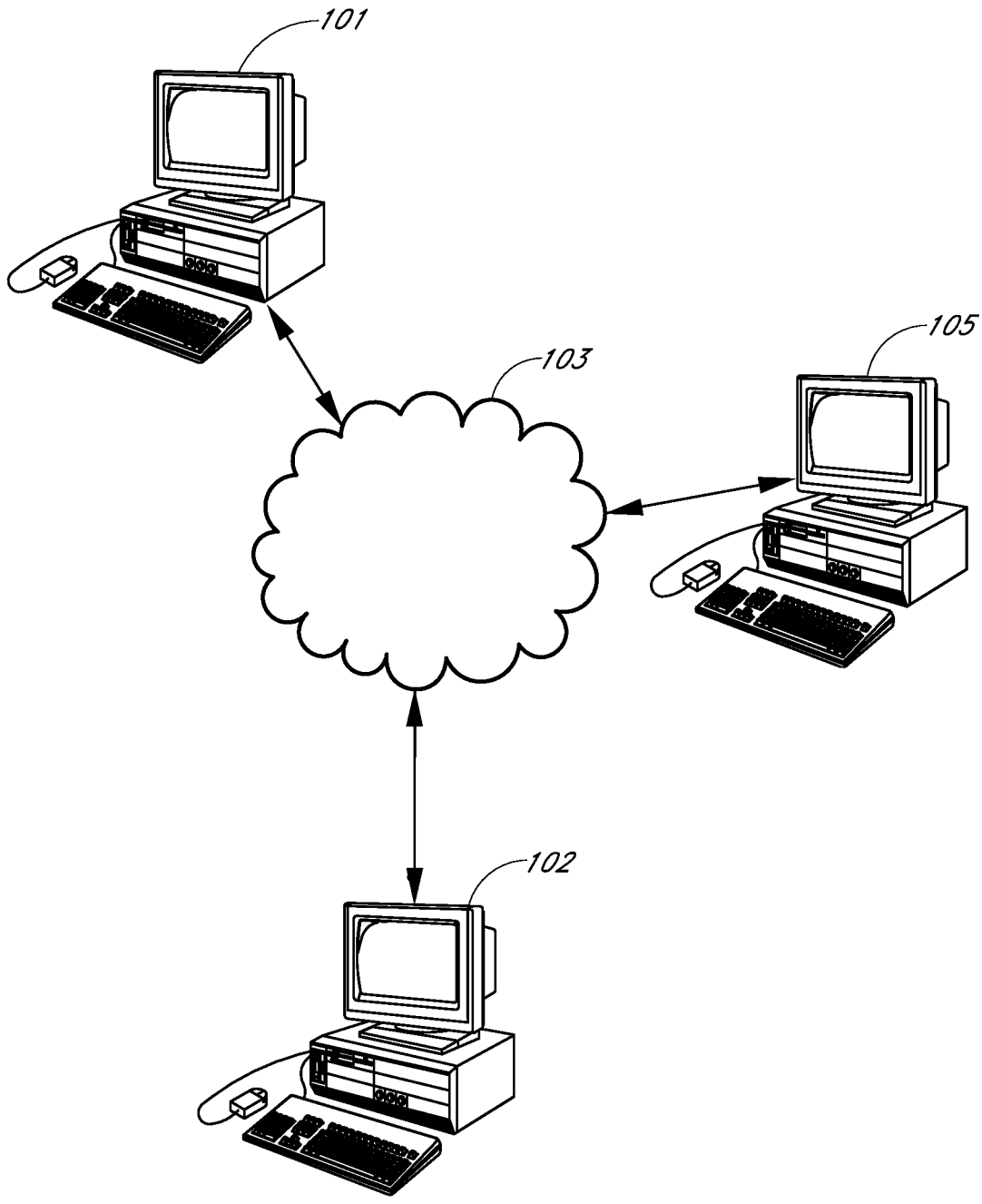


FIG. 1

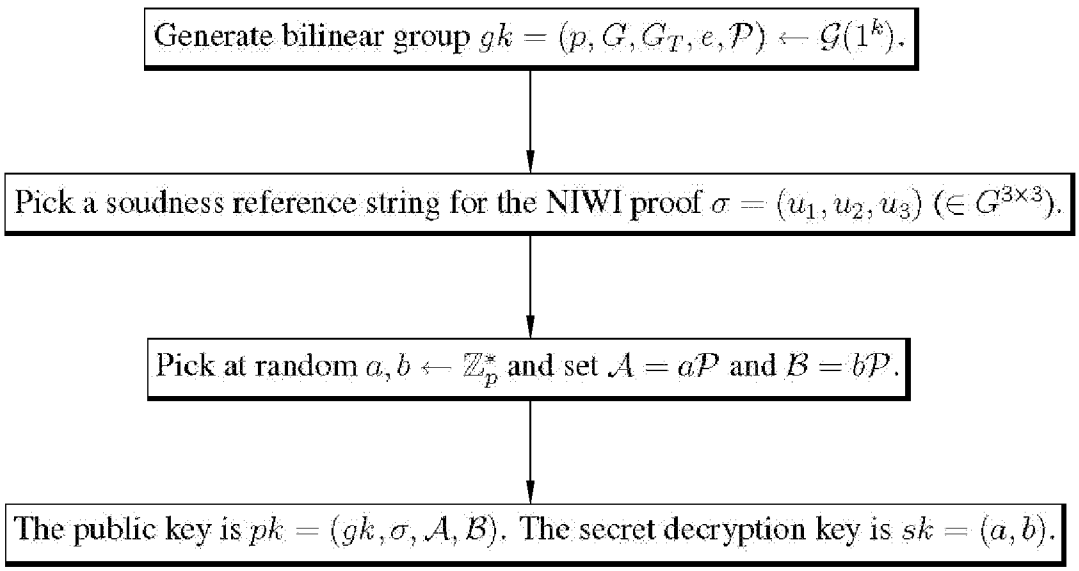

Generate bilinear group $gk = (p, G, G_T, e, \mathcal{P}) \leftarrow \mathcal{G}(1^k)$.
Pick a soudness reference string for the NIWI proof $\sigma = (u_1, u_2, u_3)$ $(\in G^{3\times 3})$.
Pick at random $a, b \leftarrow \mathbb{Z}_p^*$ and set $\mathcal{A} = a\mathcal{P}$ and $\mathcal{B} = b\mathcal{P}$.
The public key is $pk = (gk, \sigma, \mathcal{A}, \mathcal{B})$. The secret decryption key is $sk = (a, b)$.

FIG. 2

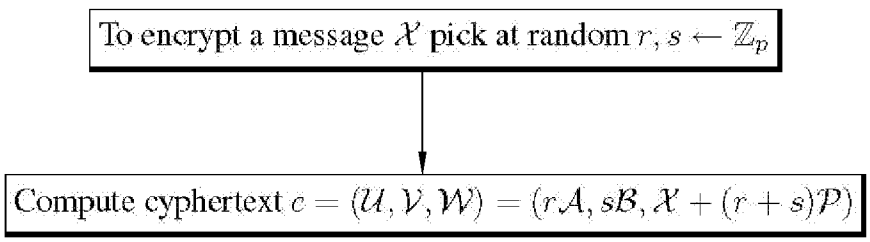

To encrypt a message $\mathcal{X}$ pick at random $r, s \leftarrow \mathbb{Z}_p$
Compute cyphertext $c = (\mathcal{U}, \mathcal{V}, \mathcal{W}) = (r\mathcal{A}, s\mathcal{B}, \mathcal{X} + (r + s)\mathcal{P})$

FIG. 3

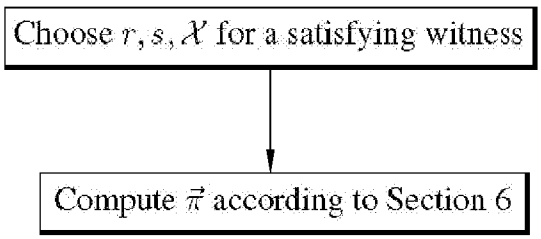

Choose $r, s, \mathcal{X}$ for a satisfying witness
Compute $\vec{\pi}$ according to Section 6

FIG. 4

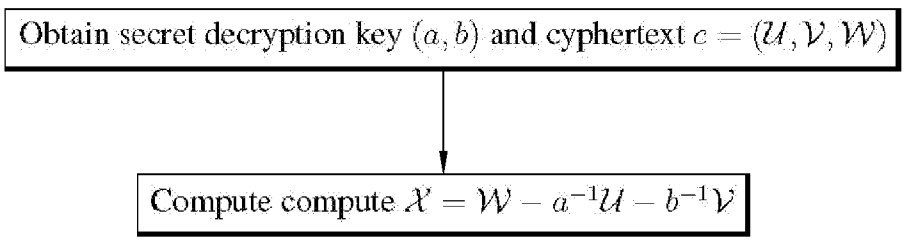

Obtain secret decryption key (a, b) and cyphertext $c = (\mathcal{U}, \mathcal{V}, \mathcal{W})$
Compute compute $\mathcal{X} = \mathcal{W} - a^{-1}\mathcal{U} - b^{-1}\mathcal{V}$

FIG. 5

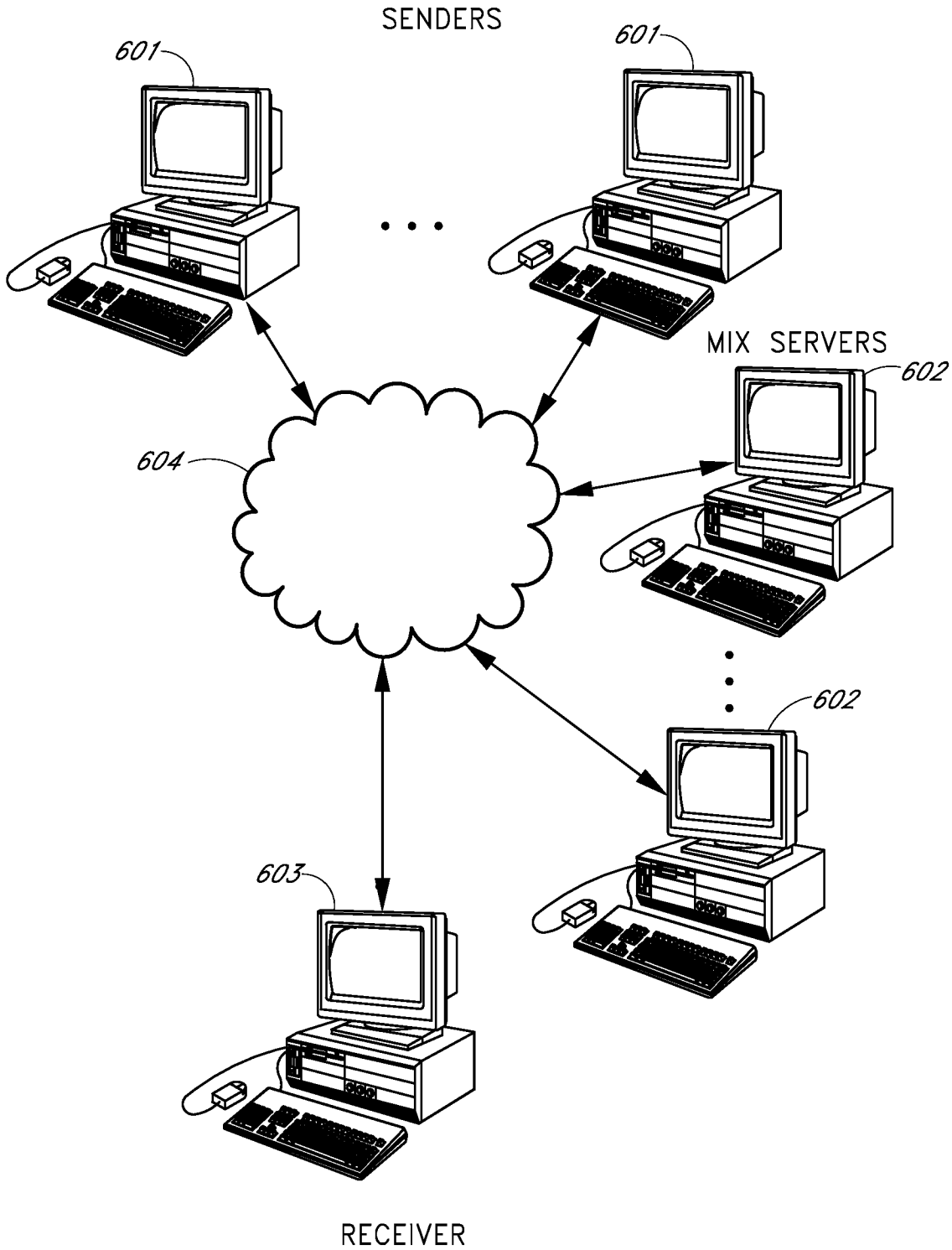


FIG. 6

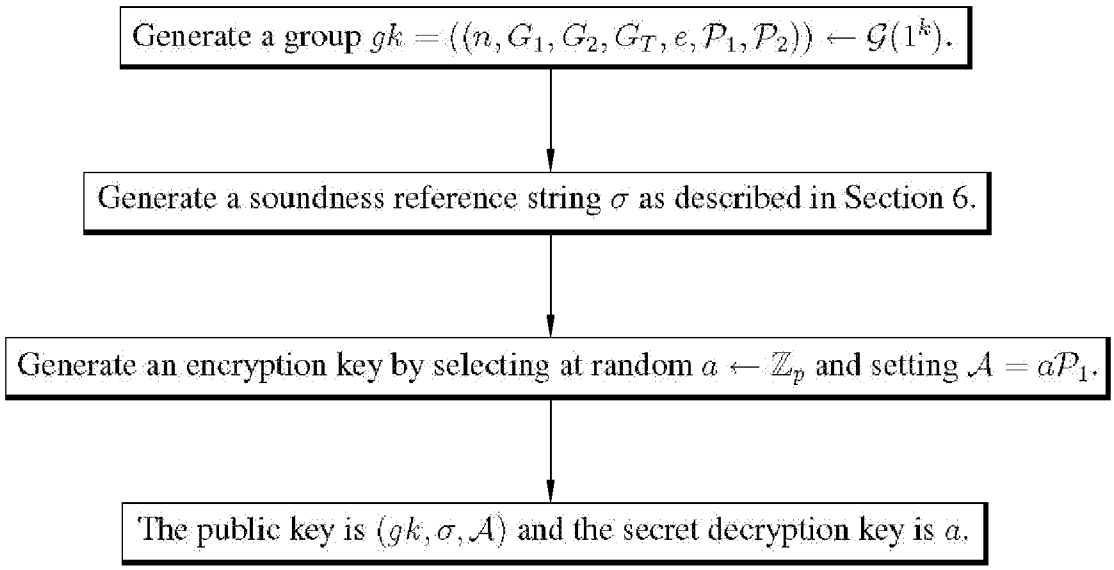


FIG. 7

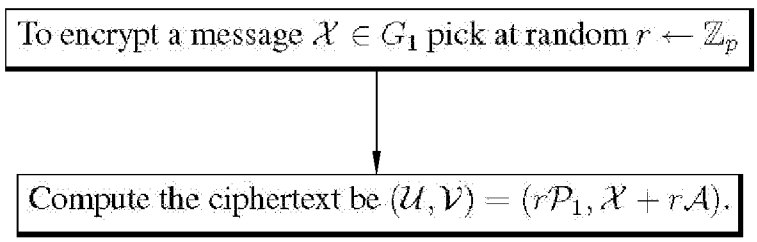


FIG. 8

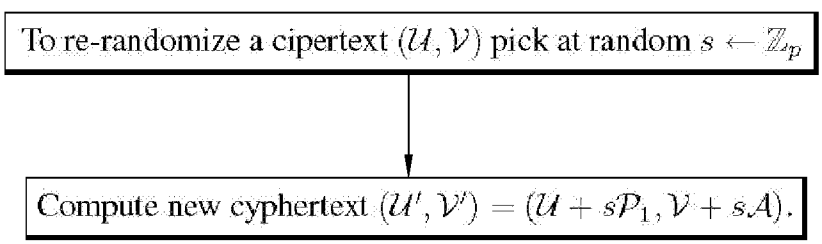
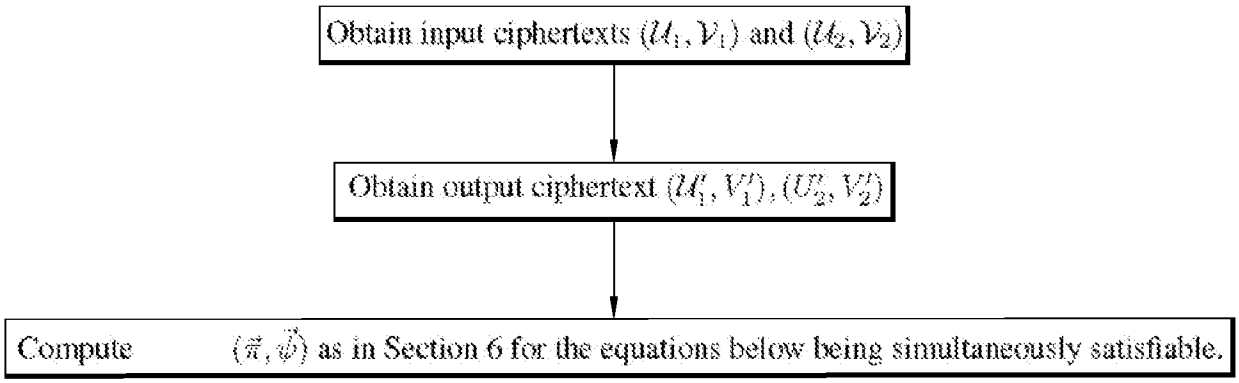


FIG. 9



$$\phi(\mathcal{U}'_1 - \mathcal{U}_1 - r\mathcal{P}_1) = \mathcal{O} \ \wedge \ \phi\tau(\mathcal{V}'_1 - \mathcal{V}_1 - r\mathcal{A}) = \mathcal{O}$$
$$\wedge \phi(U'_2 - \mathcal{U}_2 - r\mathcal{P}_1) = \mathcal{O} \ \wedge \ \phi(V'_2 - \mathcal{V}_2 - r\mathcal{A}) = 1$$

$$\text{and } (1-\phi)(\mathcal{U}'_1 - \mathcal{U}_2 - s\mathcal{P}_1) = \mathcal{O} \ \wedge \ (1-\phi)(V'_1 - \mathcal{V}_2 - s\mathcal{A}) = \mathcal{O}$$
$$\wedge (1-\phi)(\mathcal{U}'_2 - \mathcal{U}_1 - r\mathcal{P}_1) = \mathcal{O} \wedge (1-\phi)(\mathcal{V}'_2 - \mathcal{V}_2 - r\mathcal{A}) = \mathcal{O}.$$

FIG. 10

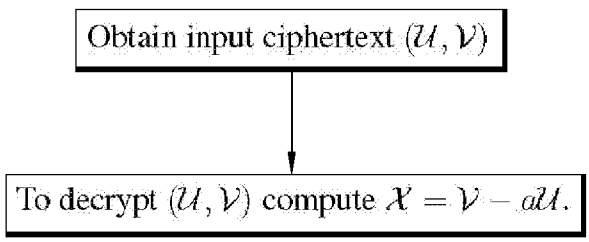


FIG. 11