Ring Referral: Efficient Publicly Verifiable Ad hoc Credential Scheme with Issuer and Strong User Anonymity for Decentralized Identity and More

The-Anh Ta
CSIRO Data61

Xiangyu Hui
University of Melbourne

Sid Chi-Kin Chau*
CSIRO Data61

***Abstract*—In this paper, we present a *ring referral scheme*, by which a user can publicly prove her knowledge of a valid signature for a private message that is signed by one of an ad hoc set of authorized issuers, without revealing the signing issuer. Ring referral is a natural extension to traditional ring signature by allowing a prover to obtain a signature from a third-party signer. Our scheme is useful for diverse applications, such as certificate-hiding decentralized identity, privacy-enhancing federated authentication, anonymous endorsement and privacy -preserving referral marketing. In contrast with prior issuer-hiding credential schemes, our ring referral scheme supports more distinguishing features, such as (1) public verifiability over an ad hoc ring, (2) strong user anonymity against collusion among the issuers and verifier to track a user, (3) transparent setup, (4) message hiding, (5) efficient multi-message logarithmic verifiability, (6) threshold scheme for requiring multiple co-signing issuers. Finally, we implemented our ring referral scheme with extensive empirical evaluation.**[1]

***Index Terms*—Decentralized Identity, Issuer Hiding, Strong User Anonymity, Logarithmic Verifiability, Ring Signatures**

1. Introduction

In a traditional ring signature scheme [2], a prover (also a signer) convinces a public verifier by a proof-of-knowledge of a valid signature for a message that is signed by a secret key corresponding to one of an ad hoc set of authorized public keys (a.k.a. a ring). This paper presents a ring referral scheme, a natural extension to a ring signature, by allowing a prover to obtain a signature from a third-party signer, without possessing the secret key. In a *ring referral scheme*, a user (not necessarily a signer) convinces a public verifier by a proof-of-knowledge of a valid signature for a message that is signed by one of an ad hoc set of authorized issuers (each of which is associated with a public key), without revealing which the signer or the associated public key is.

A ring referral scheme is useful for a variety of applications, where the anonymity of the signer (a.k.a. *issuer anonymity*) is an important consideration. We provide several useful applications of a ring referral scheme as follows:

1) ***Certificate-hiding Decentralized Identity***: In normal decentralized identity (e.g., OpenID [3]), a user (i.e., credential holder) presents a proof of credentials along with a certificate issued by an authorized issuer (as depicted in Fig. 1a). For example, if requested for a proof-of-adulthood (age 18+), a user presents a certificate of a digital driving license issued by the authority, along with a proof showing the date-of-birth in the certified driving license within a valid range, without disclosing any personal credentials (e.g., name, date-of-birth, address). There are multiple authorized issuers of credentials, e.g., each state in the US issues driving licenses. Revealing the identity of the issuer may leak sensitive personal information, e.g., revealing the state of a driving license may leak a user's location. For stronger privacy protection, a user needs to hide the certificate by showing only a proof of a valid certificate from one of the authorized issuers, along with a proof of credentials that are certified by the hidden certificate (as depicted in Fig. 1b).
2) ***Privacy-enhancing Federated Authentication***: To support a single sign-on and eliminate repetitive authentication in distributed systems, federated authentication enables a website to grant a user's access, if the user possesses an account from a set of authorized service providers (e.g., Google, Facebook, Microsoft). OAuth [4] is a popular protocol for federated authentication. An authorized service provider first signs an access token linked to the destined website, after authenticating the user. The user then utilizes the signed access token, verifiable by the public key of the authorized service provider, to gain access to the destined website. To enhance privacy, the user may not want to disclose the identity of the authorized service provider to prevent the

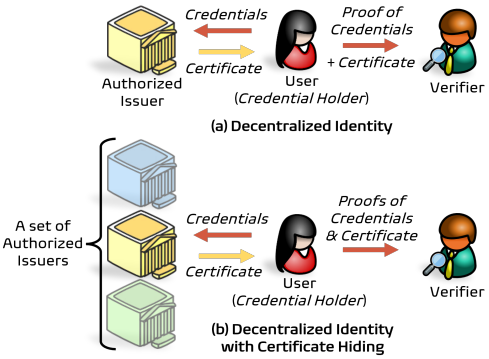


Figure 1: Certificate-hiding decentralized identity.

*Corresponding author: Sid Chi-Kin Chau (sid.chau@acm.org)

1. This is an extended version of the paper in IEEE S&P '25 [1].

destined website from tracking her activities.

3) ***Privacy-preserving Referral Marketing***: A business operator (e.g., bookstore) would offer discounts to certain shoppers with referrals from a list of recognized sources (e.g., students enrolling in a recognized institute). But the shoppers do not need to disclose the sources of referrals (e.g., the name of the enrolled institute), by presenting only a proof of eligibility for discounts.
4) ***Anonymous Endorsement***: The notion of Decentralized Autonomous Organizations (e.g., DAOs) is to allow governance by a community without centralized leadership. It is desirable that the members of a community can put forward a motion, if there are sufficient endorsements from the community. *Anonymous endorsement* allows a motion to be triggered by the endorsements from a quorum, without revealing the identities of the endorsers.

In general applications, there is a list of authorized issuers (i.e., signers) who can sign a message for a user. The user then presents a proof-of-knowledge to a verifier that the message has been signed by an authorized issuer, without revealing the identity of the issuer in the list.

Although a similar idea of *issuer anonymity* was studied recently in the literature, such as issuer-hiding credential schemes [5], [6], [7] and multi-issuer credential scheme [8], this paper offers more distinguishing features with stronger privacy protection, higher efficiency and more applicability. Our ring referral scheme supports the following features:

- ***Public Verifiability over Ad hoc Ring***: The issuer-hiding credential schemes in [5], [6], [7] rely on private verifiability over a verifier-defined static ring, whereas our ring referral scheme supports *public verifiability* over an *ad hoc ring*. Namely, any public member can verify a ring referral without the interaction with a user, and anyone (a user or third-party) can construct or update a ring without the need for approval by a verifier. This especially enables decentralized applications on permissionless blockchain platforms for publicly verifiable certificate-hiding decentralized identity and anonymous endorsement of DAOs.
- ***Issuer Oblivion***: The issuers should be oblivious of what applications of their signatures will be used for. Issuer oblivion not only prevents the issuers from interference, but also facilitates the setup without coordinating the issuers and verifier. The issuer-hiding credential schemes in [5], [6], [7] relies on issuers' special signatures via a verifier-defined master key, whereas our ring referral scheme uses standard BBS signature, without letting the issuers know if their signatures are used in a ring referral.
- ***Transparent Setup***: The schemes in [6], [7] require a trusted third-party for the setup of structured parameters, which may incur security loopholes or additional overhead. [8] requires a trusted setup for the accumulator in its credential scheme, but not for signature verification. Our scheme supports a transparent setup to eliminate any trusted third-party in a decentralized setting.
- ***Message Hiding***: To support comprehensive privacy, we consider hiding the signed message from the verifier in a ring referral. The schemes in [5], [6], [7], however, require the revelation of the signed message to the verifier.
- ***Strong User Anonymity***: While hiding the issuer can protect user anonymity, it is insufficient if there is a collusion among the issuers and verifier to track a user. For instance, when there is a data breach with the issuers, such that all previously signed messages and signatures are exposed, it may be possible to link a ring referral with an issuer. We introduce the notion of *strong user anonymity*, which unlinks a signed message and its signature from a ring referral. None of the extant schemes can support strong user anonymity, as [5], [6], [7] are not message-hiding and [8] needs to reveal partial signature to the verifier.
- ***Multi-message Logarithmic Verifiability***: The schemes in [5], [7], [8] require linear-sized public keys or proofs for proving multi-message signatures, while [6] does not support multi-message signatures. We apply a recursive compression technique (Dory [9]) to attain an efficient logarithmically verifiable multi-message signature scheme, which may be of independent interest, and improve the verification efficiency of our multi-message scheme.
- ***Threshold Scheme***: In anonymous endorsement, a motion needs to be co-signed by at least k distinct endorsers. Thus, we also provide a threshold version of ring referral scheme that requires at least k distinct co-signing issuers in a single ring referral. None of the extant schemes [5], [6], [7], [8] can support threshold requirements.

TABLE 1: Feature comparison of issuer-hiding/multi-issuer credential schemes and our ring referral scheme RR.bbs. ♯[8] requires no trusted setup for signature verification.

Scheme	Base Signature	Issuer Anonymity	Public Verifiability	Ad hoc Ring	Issuer Oblivion	Transparent Setup	Message Hiding	Strong User Anonymity	MultiMsg LogVerify	Threshold
BEK+21 [5]	Groth [10]	✓	✗	✗	✗	✓	✗	✗	✗	✗
BFGP22 [6]	PS [11]	✓	✗	✗	✗	✗	✗	✗	✗	✗
ST23 [7]	PS [11]	✓	✗	✗	✗	✗	✗	✗	✗	✗
ECA21 [8]	Groth [10]	✓	✓	✓	✓	✓♯	✓	✗	✗	✗
RR.bbs	BBS [12]	✓	✓	✓	✓	✓	✓	✓	✓	✓

To sum up, we compare the features of the issuer-hiding/multi-issuer credential schemes [5], [6], [7], [8] and our scheme in Table 1. Our scheme supports all the aforementioned features.

Contributions: We make the following contributions:

1) We formalize the notion of a ring referral scheme and the associated security properties, such as unforgeability, issuer anonymity and strong user anonymity.
2) We present a ring referral scheme RR.bbs, based on BBS signature, supporting all the features in Table 1.
3) We design a new succinct and logarithmically verifiable multi-message BBS signature scheme, which enables succinct and efficiently verifiable multi-message RR.bbs.
4) We extend RR.bbs to support threshold requirements.
5) We implemented our schemes and conducted extensive empirical evaluation of RR.bbs.

Organization: Sec. 2 surveys the related work. Sec. 3 provides a technical overview. Sec. 4 presents the preliminaries and Sec. 5 presents a formal model of ring referral scheme.

Sec. 6 presents a novel succinct multi-message BBS signature scheme. Secs. 7, 8 and Appendix C present the single-message, multi-message and threshold RR.bbs schemes, respectively. Sec. 9 empirically evaluates our scheme. More technical definitions and proofs are deferred to Appendix.

2. Related Work

► **Anonymous Signatures:** Group signatures [13] and ring signatures [2] are two main types of anonymous signatures to let a user sign a message on behalf of a group without revealing her identity with respect to the group. Our ring referral scheme is closely related to ring signatures by allowing an ad hoc ring. But a ring referral also distinguishes between an issuer (i.e. signer) and a user (i.e. not a signer) and ensures anonymity for both issuer and user. There are other privacy extensions to signature schemes, such as multi-user blind signatures [14] and zero-knowledge credentials [15]. But these schemes do not hide the signer with a ring.
► **Compressed Anonymous Signatures:** Constructions of anonymous signatures often involve zero-knowledge proofs of knowledge for group membership [16], [17]. One can use compressed argument systems to design succinct ring signatures. Our ring referral scheme draws on Dory [9] – a pairing-based recursive argument system with a logarithmically verifiable (and logarithmic-sized) proof in transparent setup. DualDory [18] and LLRing [19] are ring signature schemes using Dory to achieve succinct signatures and efficient verifiability. Omniring [20] is a ring signature based on Bulletproofs [21] with a logarithmic-sized (but linearly verifiable) proof. Our ring referral scheme is inspired by the signature compression techniques in Omniring and LLRing.
► **Issuer-hiding Credentials:** In issuer-hiding credential schemes [5], [6], [7], a verifier specifies a fixed set of authorized issuers by a master key, and a user reveals to the verifier a signed message with an anonymous signature by an authorized issuer. Among the features in Table 1, our ring referral scheme supports issuer anonymity, but allows an ad hoc and publicly verifiable set of issuers.
► **Multi-issuer Credentials:** In multi-issuer credential schemes [8], [22], [23], a credential is issued from a set of issuers without a root authority, which support issuer anonymity[2]. One-out-of-many proof [17] is used in [8] to produce a logarithmic-sized (but linearly verifiable) proof with respect to the ring. Unlike our scheme, these schemes do not guarantee strong user anonymity against collusion among the issuers and verifier. Our scheme based on Dory yields a logarithmically verifiable proof with respect to both the ring and message size, and is extensible to support threshold requirements. There are other distributed credential schemes, e.g., [24], [25] are primarily designed to support fault-tolerance of issuers, rather than issuer anonymity.

3. Technical Overview

► **Basic Notations:** We first define some basic notations, before providing a technical overview of our results. Denote a cyclic group of prime order p by $\mathbb{G}$, and a ring of integers modulo p by $\mathbb{Z}_p$. $\mathbf{1}_{\mathbb{G}}$ denotes the identity element in $\mathbb{G}$. Let $\mathbb{Z}_p^* \triangleq \mathbb{Z}_p \backslash \{0\}$. We denote a vector in bold font with an arrow symbol and its coordinates in normal font. For example, $\vec{\mathbf{a}} \triangleq (a_1, ..., a_n) \in \mathbb{Z}_p^n$ denotes a scalar vector and $\vec{\mathbf{G}} \triangleq (G_1, ..., G_n) \in \mathbb{G}^n$ denotes a vector of generators from a finite group. Define the following basic vector operations:

- $\vec{\mathbf{a}} \circ \vec{\mathbf{b}} \triangleq (a_1 \cdot b_1, ..., a_n \cdot b_n) \in \mathbb{Z}_p^n$,
- $\vec{\mathbf{G}} \circ \vec{\mathbf{H}} \triangleq (G_1 \cdot H_1, ..., G_n \cdot H_n) \in \mathbb{G}^n$,
- $\vec{\mathbf{G}}^{\circ \vec{\mathbf{a}}} \triangleq (G_1^{a_1}, ..., G_n^{a_n}) \in \mathbb{G}^n$,
- $\vec{\mathbf{G}}^{\vec{\mathbf{a}}} \triangleq \prod_{i \in [n]} G_i^{a_i} \in \mathbb{G}$.

Definition 3.1 (Bilinear Pairing)**.** *Given three cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order p, a* bilinear pairing *is a map $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$ that satisfies the following properties: (i)* Bilinearity: *for any $\Omega, \Gamma \in \mathbb{G}_1$, $\Theta, \Lambda \in \mathbb{G}_2$, $\alpha, \beta \in \mathbb{Z}_p$, we have $\mathsf{e}(\Omega^\alpha \cdot \Gamma^\beta, \Theta) = \mathsf{e}(\Omega, \Theta)^\alpha \cdot \mathsf{e}(\Gamma, \Theta)^\beta$ and $\mathsf{e}(\Omega, \Theta^\alpha \cdot \Lambda^\beta) = \mathsf{e}(\Omega, \Theta)^\alpha \cdot \mathsf{e}(\Omega, \Lambda)^\beta$; (ii)* Non-degeneracy: *if Ω, Θ are generators of $\mathbb{G}_1, \mathbb{G}_2$, then $\mathsf{e}(\Omega, \Theta)$ is a generator of $\mathbb{G}_T$; (iii)* Efficiency: *$\mathsf{e}(\cdot, \cdot)$ is efficiently computable.*

We also define the inner-product relation via bilinear pairing for given $(\vec{\mathbf{\Omega}} \in \mathbb{G}_1^n, \vec{\mathbf{\Theta}} \in \mathbb{G}_2^n, \vec{\mathbf{c}} \in \mathbb{Z}_p^n)$ as follows:

$$\mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Theta}}) \triangleq \prod_{i \in [n]} \mathsf{e}(\Omega_i, \Theta_i), \ \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Theta}})^{\vec{\mathbf{c}}} \triangleq \prod_{i \in [n]} \mathsf{e}(\Omega_i, \Theta_i)^{c_i}.$$

► **Overview:** We next provide a high-level overview of our results. For the brevity of presentation, here we omit zero knowledge, which will be rectified in the full scheme.

1) ***Compressed Message-hiding Multi-message BBS Signature:*** Our ring referral scheme relies on a message-hiding signature scheme, in which the signed message $\vec{\mathsf{m}}$ is not revealed during the verification, except a commitment $\mathtt{Cm}(\vec{\mathsf{m}})$. The ramifications of a message-hiding signature are three-fold: (1) it can be extended to a subsequent proof-of-knowledge that also hides the signature and the public key, (2) it incorporates compression of a vector message by a commitment, which enables logarithmic verifiability of a multi-message signature via Dory, (3) it enables strong user anonymity by hiding the message that is known by the issuer from the verifier. First, we adapt BBS signature scheme [12] to support message hiding, which may be of independent interest. The validity of a BBS signature (σ_0, σ_1) on message $\vec{\mathsf{m}}$ with respect to public key pk can be checked by the bilinear pairing equation:
$$\mathsf{e}(\sigma_0, \ G_2^{\sigma_1} \cdot \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathsf{e}(\vec{\mathbf{H}}, \vec{G}_2)^{\vec{\mathsf{m}}},$$
where $G_1, G_2, \vec{G}_2, \vec{\mathbf{H}}$ are generators from the setup. In our scheme, the verifier outsources the computation of $\mathsf{e}(\vec{\mathbf{H}}, \vec{G}_2)^{\vec{\mathsf{m}}}$ to the prover. To check the consistency of the same message $\vec{\mathsf{m}}$ in both $\mathsf{e}(\vec{\mathbf{H}}, \vec{G}_2)^{\vec{\mathsf{m}}}$ and $\mathtt{Cm}(\vec{\mathsf{m}})$, we use Dory, a compressed zero-knowledge argument system with logarithmic proof size and verification time.
2) ***Proof-of-Knowledge for BBS Signature:*** In addition to $\mathtt{Cm}(\vec{\mathsf{m}})$, the prover also commits the public key and signature to $\mathtt{Cm}(\mathsf{pk})$, $\mathtt{Cm}(\sigma_0)$, $\mathtt{Cm}(\sigma_1)$. We devise a way

2. Note that [8] refers to issuer anonymity by "user anonymity".

to validate a BBS signature based on these commitments only. The basic idea is to use Dory to check if these commitments relate to the same committed values. For example, Dory can check the consistency of the same values (σ_0, σ_1) in $\mathsf{e}(\sigma_0, G_2^{\sigma_1})$ and $(\mathtt{Cm}(\sigma_0), \mathtt{Cm}(\sigma_1))$, and the same values (σ_0, pk) in $\mathsf{e}(\sigma_0, \mathsf{pk})$ and $(\mathtt{Cm}(\sigma_0), \mathtt{Cm}(\mathsf{pk}))$. As a result, we can validate the knowledge of a BBS signature by checking the blinear pairing equation:

$$\mathsf{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathsf{e}(\sigma_0, \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathsf{e}(\vec{\mathbf{H}}, \vec{G}_2)^{\vec{\mathsf{m}}}.$$

3) ***Proof-of-Knowledge for a Ring:*** In our ring referral scheme, we also ensure that $\mathtt{Cm}(\mathsf{pk})$ is within an designated ring $(\mathsf{pk}_i)_{i\in n}$. We adopt the binary selector technique from LLRing [19] and Omniring [20]. Suppose $i^* \in [n]$ is the index of a secret issuer. We define a unit basis vector $\vec{\mathbf{b}}$, such that $b_{i^*} = 1$ and $b_i = 0$ for $i \neq i^*$. As a result, we can validate the knowledge of $\mathtt{Cm}(\mathsf{pk})$ in a ring by checking the bilinear pairing equation:

$$\prod\nolimits_{i\in[n]} \mathsf{e}\big(G_1, (\mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})^{b_i}\big) \stackrel{?}{=} \mathsf{e}(G_1, \mathsf{pk}) \cdot \mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}},$$

where $\mathsf{h}_i \triangleq \mathtt{Hash}[\mathsf{pk}_i]$ and the well-formedness of $\mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}}$ can be validated by a standard Schnorr proof-of-knowledge. Note that $\prod_{i\in[n]} \mathsf{e}(G_1, \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})^{b_i}$ is a commitment of $\vec{\mathbf{b}}$, and hence, the satisfiability of $\vec{\mathbf{b}}$ as a unit basis vector can also be checked by Dory.

4) ***Multi-message & Threshold Ring Referrals:*** Overall, our multi-message ring referral scheme is an integration of compressed message-hiding BBS signature, proof-of-knowledge of BBS signature and proof-of-knowledge for a ring. Our scheme can be extended to support the k-issuer threshold requirement by allowing $\vec{\mathbf{b}}$ to be a binary vector with the sum of its coordinates equating to k.

3.1. Integration with Decentralized Identity System

To demonstrate a practical application, we discuss a potential integration of our ring referral scheme with OpenID [3], which is a framework of a third-party identity provider for authentication. OpenID involves three parties: user, third-party identity provider (IDP) who certifies users' identity and credentials for authentication, and relying party (RP) website which requires users' credential certification.

In OpenID, a user first registers an account with OpenID identity providers and obtains unique identifiers. When the user visits an RP website, the user is redirected to the corresponding IDP based on its unique identifier. The IDP validates the user's information, like username, passwords, then issues an OpenID – a certificate for the user's identity and credentials. Then, the user shows the OpenID to the RP website for verification of the OpenID.

We can integrate our ring referral scheme with OpenID as follows. First, the user will send a message to its IDP asking for a signature to authenticate his identity and credentials. This step can be performed before the user interaction with an RP website. Second, when the user visits an RP website, it collects a set of IDP providers as the ring of issuers in the ring referral scheme, then generates a ring referral proof as a certificate to access the RP website. The user can generate a proof from the non-interactive ring referral scheme. Finally, the RP website will run the ring referral verification algorithm. Note that our ring referral scheme can be readily integrated with other protocols like OAuth [4] and SAML [26].

Moreover, the public verifiability over an ad hoc ring allows our ring referral scheme to be deployed on permissionless blockchain platforms for DAOs.

4. Preliminaries

We present the preliminaries for our scheme. Let λ be the security level parameter and $\mathrm{negl}(\lambda)$ be a negligible function of λ. PPT denotes "probabilistic polynomial time". "$\stackrel{\$}{\leftarrow}$" denotes a uniformly random selection from a set.

Commitment. A commitment scheme is a mapping $\mathtt{Cm} : \mathcal{M}^n \times \mathcal{R} \to \mathcal{C}$ from a (vector) multi-message[3] space $\mathcal{M}^n$ and a random mask space $\mathcal{R}$ to a commitment space $\mathcal{C}$. A commitment scheme is *homomorphic*, if for any $\vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2 \in \mathcal{M}^n, \mathsf{r}_1, \mathsf{r}_2 \in \mathcal{R}$:

$$\mathtt{Cm}(\vec{\mathsf{m}}_1; \mathsf{r}_1) \cdot \mathtt{Cm}(\vec{\mathsf{m}}_2; \mathsf{r}_2) = \mathtt{Cm}(\vec{\mathsf{m}}_1 + \vec{\mathsf{m}}_2; \mathsf{r}_1 + \mathsf{r}_2).$$

We use the well-known Pedersen commitment and AFGHO commitment, both are homomorphic commitment schemes that support usual perfect hiding and computational binding.

Definition 4.1 (Pedersen Commitment)**.** *Let $\mathcal{M} = \mathbb{Z}_p^n$, $\mathcal{R} = \mathbb{Z}_p^*$ and $\mathcal{C} = \mathbb{G}$ of order p. $\vec{\mathbf{G}} \stackrel{\$}{\leftarrow} \mathbb{G}^n, Q \stackrel{\$}{\leftarrow} \mathbb{G}$ are randomly selected generators. Define Pedersen commitment by*

$$\mathtt{Cm}(\vec{\mathsf{m}}; \mathsf{r}) \triangleq \vec{\mathbf{G}}^{\vec{\mathsf{m}}} \cdot Q^{\mathsf{r}} = \Big(\prod\nolimits_{i\in[n]} G_i^{\mathsf{m}_i}\Big) \cdot Q^{\mathsf{r}}.$$

Definition 4.2 (AFGHO Commitment [27])**.** *Let $\mathcal{M} = \mathbb{Z}_p^n$, $\mathcal{R} = \mathbb{Z}_p^*$ and $\mathcal{C} = \mathbb{G}_T$ of order p. $\vec{\mathbf{G}} \stackrel{\$}{\leftarrow} \mathbb{G}_1^n, \vec{\mathbf{\Lambda}} \stackrel{\$}{\leftarrow} \mathbb{G}_2^n, Q_1 \stackrel{\$}{\leftarrow} \mathbb{G}_1, Q_2 \stackrel{\$}{\leftarrow} \mathbb{G}_2$ are randomly selected generators. Let $\mathtt{Q} \triangleq \mathsf{e}(Q_1, Q_2)$. Define AFGHO commitment by*

$$\mathtt{Cm}(\vec{\mathsf{m}}; \mathsf{r}) \triangleq \mathsf{e}(\vec{\mathbf{G}}, \vec{\mathbf{\Lambda}})^{\vec{\mathsf{m}}} \cdot \mathtt{Q}^{\mathsf{r}} = \Big(\prod\nolimits_{i\in[n]} \mathsf{e}(G_i, \Lambda_i)^{\mathsf{m}_i}\Big) \cdot \mathtt{Q}^{\mathsf{r}}.$$

4.1. Zero-Knowledge Arguments of Knowledge

An *argument system* is consisted of three PPT algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$, where $\mathcal{G}$ is the setup algorithm for public parameters pp, $\mathcal{P}$ and $\mathcal{V}$ are the prover and verifier algorithms. Denote the communication transcript between the prover and verifier by $\mathsf{tr} \leftarrow \langle \mathcal{P}(\cdot), \mathcal{V}(\cdot) \rangle$. At the end, the transcript will produce a binary decision: $\mathtt{Accept}[\mathsf{tr}] \in \{0, 1\}$. A key class of argument systems are *zero-knowledge arguments of knowledge* (e.g., ring signature, ring referral).

Definition 4.3 (Argument of Knowledge)**.** *An argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is an* argument of knowledge *for a relation, if it satisfies perfect completeness (Def. (B.1)) and computational witness-extended emulation (CWE) (Def. (B.2)).*

3. For credential applications, messages are called "attributes". A credential can be represented by a commitment of a list of private attributes.

CWE captures the idea of *knowledge-sound* arguments. Informally, if an adversary produces an acceptable argument with some probability, there exists an emulator that produces a similar argument and a witness with the same probability.

We are interested in *Special Honest-Verifier Zero-Knowledge (SHVZK)* arguments (Def. (B.4)) that do not leak the information about the witness beyond what can be inferred from the truth of the statement.

An argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called *public-coin*, if the verifier chooses her messages uniformly at random, independent from the messages sent by the prover. Let e be the public-coin challenge. The transcript of a public-coin argument system is defined as $\mathsf{tr} = \langle \mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x; e) \rangle$.

Definition 4.4 (Fiat-Shamir Transformation)**.** *A multi-move interactive public-coin argument of knowledge can be converted to a non-interactive argument of knowledge by replacing the public-coin challenges by the output of a cryptographic hash function, which produces seemingly random output and is regarded as a replacement for a verifier.*

In this paper, we focus on *multi-move interactive public-coin protocols* for arguments of knowledge. The Fiat-Shamir transformation can be applied to convert our interactive protocols to non-interactive arguments using the random oracle model in the security proofs [28]. This is especially useful for reducing a logarithmic number of moves to a single move in a publicly verifiable scheme.

4.2. Dory: Compressed Arguments of Knowledge

New techniques have been developed to compress zero-knowledge arguments of knowledge to obtain a smaller proof size or faster verification. Recently, *Dory* [9] is a compressive protocol with logarithmic verification efficiency and proof size based on a recursion for checking the following inner-product relations in the pairing setting:

$$\begin{cases} \mathtt{D}_0 \overset{?}{=} \mathsf{e}(\vec{\boldsymbol{\Omega}}, \vec{\boldsymbol{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_0}, \\ \mathtt{D}_1 \overset{?}{=} \mathsf{e}(\vec{\boldsymbol{\Omega}}, \vec{\boldsymbol{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathtt{D}_2 \overset{?}{=} \mathsf{e}(\vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_2}, \end{cases} \tag{1}$$

where $\mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2 \in \mathbb{G}_T$ and random generators $\vec{\boldsymbol{\Gamma}} \xleftarrow{\$} \mathbb{G}_1^n$, $\vec{\boldsymbol{\Lambda}} \xleftarrow{\$} \mathbb{G}_2^n$ are the given input, and $(\vec{\boldsymbol{\Omega}} \in \mathbb{G}_1^n, \vec{\boldsymbol{\Theta}} \in \mathbb{G}_2^n$, $\mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2 \in \mathbb{Z}_p^*)$ are the private witness. We color the witness in Eqn. (1) to enhance readability.

Dory produces a proof with $6 \log n$ $\mathbb{G}_T$ elements, 1 $\mathbb{G}_1$ element and 1 $\mathbb{G}_2$ element. The verification takes 1 pairing, $9 \log n + 9$ $\mathbb{G}_T$ exponentiations, 1 $\mathbb{G}_1$ exponentiation and 1 $\mathbb{G}_2$ exponentiation. The precomputation takes $3n$ pairings, which only involves the a-priori known generators. The proving takes $3n$ pairings, $2 \log n$ $\mathbb{G}_1$ exponentiations and $2 \log n$ $\mathbb{G}_2$ exponentiations.

We denote the syntax of interactive Dory protocol by

$$\Pi_{\mathsf{do.ip}}[n, \vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Lambda}}, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2; \vec{\boldsymbol{\Omega}}, \vec{\boldsymbol{\Theta}}, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2].$$

We also denote the syntax of *non-interactive* Dory argument via Fiat-Shamir transformation by $(\mathcal{P}_{\mathsf{do.ip}}, \mathcal{V}_{\mathsf{do.ip}})$:

- ***Proving***: $\mathcal{P}_{\mathsf{do.ip}}[n, \vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Lambda}}, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2; \vec{\boldsymbol{\Omega}}, \vec{\boldsymbol{\Theta}}, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2] \mapsto \pi$. This produces a non-interactive proof π from the input $(n, \vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Lambda}}, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2)$ and witness $(\vec{\boldsymbol{\Omega}}, \vec{\boldsymbol{\Theta}}, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2)$.
- ***Verification***: $\mathcal{V}_{\mathsf{do.ip}}[n, \vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Lambda}}, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2, \pi] \mapsto \{0, 1\}$. This takes the input $(n, \vec{\boldsymbol{\Gamma}}, \vec{\boldsymbol{\Lambda}}, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2)$ and checks the validity of proof π. It returns 1 for a valid proof or 0 otherwise.

We also denote a scalar version of Dory protocol by $\Pi_{\mathsf{do.sp}}$ for checking the scalar-product relations:

$\mathtt{D}_0 \overset{?}{=} \mathsf{e}(\Omega, \Theta) \cdot \mathtt{Q}^{\mathsf{r}_0}$, $\mathtt{D}_1 \overset{?}{=} \mathsf{e}(\Omega, \Lambda) \cdot \mathtt{Q}^{\mathsf{r}_1}$, $\mathtt{D}_2 \overset{?}{=} \mathsf{e}(\Gamma, \Theta) \cdot \mathtt{Q}^{\mathsf{r}_2}$, where $\Omega, \Gamma \in \mathbb{G}_1, \Theta, \Lambda \in \mathbb{G}_2$ are 1-dimensional elements.

The details of protocols $\Pi_{\mathsf{do.ip}}$, $\Pi_{\mathsf{do.sp}}$ and argument $(\mathcal{P}_{\mathsf{do.ip}}, \mathcal{V}_{\mathsf{do.ip}})$ are described in Appendix B.1. Note that it is possible to batch multiple Dory arguments with common generators into a single argument [9] (see Appendix B.1).

In the next sections, we will use Dory to produce compressed multi-message signature and ring referral schemes.

5. Definitions & Properties of Ring Referrals

In this section, we formally define the syntaxes of signature and ring referral schemes and the security properties.

Let the feasible message space be $\mathcal{M}$. In a ring referral scheme, a user $\mathcal{U}$ obtains a base signature σ on a (vector) multi-message $\vec{\mathsf{m}} \in \mathcal{M}$ from an issuer $\mathcal{I} \in \{\mathcal{I}_i\}_{i \in [n]}$ and presents a ring referral proof π to a verifier $\mathcal{V}$. Each issuer $\mathcal{I}_i$ has a public key pk_i. We collectively denote the set of n public keys (i.e., the ring) by $\vec{\mathbf{pk}} \triangleq \{\mathsf{pk}_i\}_{i \in [n]}$.

5.1. Syntax of Base Signature Scheme

Definition 5.1 (Signature Scheme)**.** *We define a* signature scheme $\mathtt{Sig}$ *used by the issuers by the following methods:*

1) $\mathtt{Setup}[1^\lambda] \mapsto \mathsf{pp}$*: This method is given a security level parameter λ and produces a public set-up parameter* pp*. For brevity, input* pp *to other methods is implicit.*
2) $\mathtt{KeyGen}[\mathsf{pp}] \mapsto (\mathsf{pk}, \mathsf{sk})$*: Given a public parameter* pp*, each application of this method produces a public key* pk *and the corresponding secret key* sk*. Each issuer generates his own key pair and keeps his secret key private. We denote the vector public keys of issuers as* $\vec{\mathbf{pk}}$ *and the corresponding vector of secret keys* $\vec{\mathbf{sk}}$*.*
3) $\mathtt{Sign}[\mathsf{pp}, \vec{\mathsf{m}}; \mathsf{sk}] \mapsto \sigma$*: Given a public parameter* pp *and a message* $\vec{\mathsf{m}}$*, this method produces a signature σ that is signed by a secret key* sk*.*
4) $\mathtt{VfySig}[\mathsf{pp}, \mathsf{pk}, \vec{\mathsf{m}}, \sigma] \mapsto \{0, 1\}$*: This method checks if the message* $\vec{\mathsf{m}}$ *and signature σ is signed by the secret key corresponding to the public key* pk*. It returns 1 for a valid signature or 0 otherwise.*

Definition 5.2 (Compressed Message-hiding Signature Scheme)**.** *We also define a compressed* message-hiding signature scheme*, where the message* $\vec{\mathsf{m}}$ *can be compressed to a commitment* $\mathtt{Cm}_\mathsf{m}$ *during verification, by additional methods:*

1) $\mathtt{Compress}[\mathsf{pp}, \vec{\mathsf{m}}] \mapsto (\mathtt{Cm}_\mathsf{m}, \pi)$*: Given a public parameter* pp *and a message* $\vec{\mathsf{m}}$*, this method compresses* $\vec{\mathsf{m}}$ *to a commitment* $\mathtt{Cm}_\mathsf{m}$ *with an commitment proof π.*

2) $\underline{\texttt{VfyCSig}[\mathsf{pp}, \mathsf{pk}, \texttt{Cm}_\mathsf{m}, \sigma, \pi] \mapsto \{0,1\}}$*: This method checks if the compressed message in the commitment* $\texttt{Cm}_\mathsf{m}$ *and the signature* σ *is signed by the secret key corresponding to the public key* pk *via the commitment proof* π*. It returns 1 for a valid signature or 0 otherwise.*

We present a concrete instantiation of compressed message-hiding signature scheme based on BBS signature in Section 6. See the Appendix E for further details.

5.2. Syntax of Ring Referral Scheme

Definition 5.3 (Basic Ring Referral Scheme)**.** *We first define a basic* ring referral scheme RR *on a given base signature scheme* Sig *by the following methods:*

1) $\underline{\texttt{Setup}[1^\lambda] \mapsto \mathsf{pp}}$*: This method is given a security level parameter* λ *and produces a public parameter* pp*, which is also used to set up base signature scheme* Sig*.*
2) $\underline{\texttt{Prove}[\mathsf{pp}, \vec{\mathbf{pk}}, \vec{\mathsf{m}}, \sigma] \mapsto \pi}$*: This method produces a ring referral proof* π *for a signature* σ *that is signed on message* $\vec{\mathsf{m}}$ *by a secret key corresponding to one of the public keys in* $\vec{\mathbf{pk}}$ *using base signature scheme* Sig*.*
3) $\underline{\texttt{Verify}[\mathsf{pp}, \vec{\mathbf{pk}}, \vec{\mathsf{m}}, \pi] \mapsto \{0,1\}}$*: This method checks based on proof* π *if message* $\vec{\mathsf{m}}$ *has a valid signature that is signed by a secret key corresponding to one of the public keys in* $\vec{\mathbf{pk}}$ *using base signature scheme* Sig*. It returns 1 if a valid signature exists or 0 otherwise.*

Definition 5.4 (Message-Hiding Ring Referral Scheme)**.** *We also define a general ring referral scheme supporting* message hiding*, which does not require the explicit knowledge of the signed message at the verifier. Let a message sub-space* $\mathcal{M}^* \subset \mathcal{M}$ *denote a set of feasible messages of interest for the verifier. A common example is* $\mathcal{M}_\mathsf{C} = \{\vec{\mathsf{m}} \in \mathcal{M} \mid \mathsf{C} = \texttt{Cm}(\vec{\mathsf{m}}; \mathsf{r}), \mathsf{r} \in \mathcal{R}\}$ *as the set of feasible messages for a given commitment* C*. We define a* message-hiding ring referral scheme RR *on a given base signature scheme* Sig *by the following variant methods:*

1) $\underline{\texttt{Prove}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}^*, \vec{\mathsf{m}}, \sigma] \mapsto \pi}$*: This method produces a ring referral proof* π *for a signature* σ *that is signed on* $\vec{\mathsf{m}} \in \mathcal{M}^*$ *by a secret key corresponding to one of the public keys in* $\vec{\mathbf{pk}}$ *using base signature scheme* Sig*.*
2) $\underline{\texttt{Verify}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}^*, \pi] \mapsto \{0,1\}}$*: This method checks based on proof* π *if there exists a message* $\vec{\mathsf{m}} \in \mathcal{M}^*$ *having a valid signature that is signed by a secret key corresponding to one of the public keys in* $\vec{\mathbf{pk}}$ *using base signature scheme* Sig*. It returns 1 if message* $\vec{\mathsf{m}} \in \mathcal{M}^*$ *with a valid signature exists or 0 otherwise.*

Note that if we let $\mathcal{M}^* = \{\vec{\mathsf{m}}\}$ *be a singleton set, then this becomes a ring referral scheme without message hiding.*

5.3. Security Properties of Ring Referrals

We define the desirable security properties of a ring referral scheme, and prove these properties in our scheme. For clarity, we consider the feasible message space as $\mathcal{M}_\mathsf{C}$. It is possible to generalize $\mathcal{M}_\mathsf{C}$ to be a feasible message space $\mathcal{M}^*$ subject to additional constraints (e.g., the feasible messages are within a specific range for credentials).

► **Completeness.** This property captures the basic notion of correctness, if the scheme is executed faithfully.

Definition 5.5 (Completeness)**.** *A ring referral scheme* RR *satisfies* completeness*, if*

$$\Pr\left[\begin{array}{l} \texttt{Verify}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}_\mathsf{C}, \pi] \\ = 1 \end{array} \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \texttt{Setup}(1^\lambda), \\ (\vec{\mathbf{pk}}, \vec{\mathbf{sk}}) \leftarrow \texttt{Sig.Gen}[\mathsf{pp}], \\ i^* \in [n],\ \mathsf{C} \in \mathcal{C},\ \vec{\mathsf{m}} \in \mathcal{M}_\mathsf{C} \subset \mathcal{M}, \\ \sigma \leftarrow \texttt{Sig.Sign}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}_{i^*}], \\ \pi \leftarrow \texttt{Prove}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}_\mathsf{C}, \vec{\mathsf{m}}, \sigma] \end{array}\right] \geq 1 - \mathrm{negl}(\lambda).$$

The ring referral scheme satisfies *perfect completeness*, if the above probability is exactly 1.

► **Unforgeability.** This property means that without knowing a valid base signature signed by a private key in a ring $\vec{\mathbf{pk}}$ on a message $\vec{\mathsf{m}} \in \mathcal{M}_\mathsf{C}$ for some commitment value C, it is infeasible for an adversary to convince the verifier to accept $(\vec{\mathbf{pk}}, \mathcal{M}_\mathsf{C})$.

We allow an adversary's possible access to a corruption oracle $\mathcal{CO}$, a base signature oracle $\mathcal{BSO}$ and a proving oracle $\mathcal{PO}$, as defined in the following.

Definition 5.6 (Corruption Oracle $\mathcal{CO}$)**.** *Initialize the set of corrupted keys by* $\vec{\mathbf{pk}}_{\mathcal{CO}} = \varnothing$*. When* $\mathcal{CO}$ *is queried with a public key* pk*, it returns the corresponding secret key* sk*, and then adds* pk *to* $\vec{\mathbf{pk}}_{\mathcal{CO}}$*.*

Definition 5.7 (Base Signature Oracle $\mathcal{BSO}$)**.** *Initialize the set of queries by* $\mathbf{\Sigma}_{\mathcal{BSO}} = \varnothing$*. When* $\mathcal{BSO}$ *is queried with a public parameter* pp*, a public key* pk *and a message* $\vec{\mathsf{m}}$*, it returns a valid base signature* σ *on the message* $\vec{\mathsf{m}}$ *using the secret key of* pk*, then adds* $(\mathsf{pp}, \mathsf{pk}, \vec{\mathsf{m}}, \sigma)$ *to* $\mathbf{\Sigma}_{\mathcal{BSO}}$*.*

Definition 5.8 (Proving Oracle $\mathcal{PO}$)**.** *Initialize the set of queries by* $\mathbf{\Pi}_{\mathcal{PO}} = \varnothing$*. When* $\mathcal{PO}$ *is queried with a public parameter* pp*, a ring* $\vec{\mathbf{pk}}$ *and a message space* $\mathcal{M}_\mathsf{C}$*, it returns a valid ring referral proof* π *on a message* $\vec{\mathsf{m}}$ *in* $\mathcal{M}_\mathsf{C}$ *with respect to* $\vec{\mathbf{pk}}$*, then adds* $(\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}_\mathsf{C}, \pi)$ *to* $\mathbf{\Pi}_{\mathcal{PO}}$*.*

Given a ring $\vec{\mathbf{pk}}$, an adversary $\mathcal{A}$ can obtain certain secret keys of corrupted issuers in $\vec{\mathbf{pk}}$, and certain exposed signatures and ring referral proofs with respect to a subset of $\vec{\mathbf{pk}}$. Unforgeability requires that it is infeasible for $\mathcal{A}$ to convince the verifier that it has a valid signature with respect to a subset of uncorrupted keys in $\vec{\mathbf{pk}} \backslash \vec{\mathbf{pk}}_{\mathcal{CO}}$, with unexposed signatures and ring referral proofs on these uncorrupted keys and the given message space. The notion of unforgeability is captured by the following security game:

1) First, a security game instance is set up by generating a public parameter pp and a ring $\vec{\mathbf{pk}}$.
2) Adversary $\mathcal{A}$ can query the oracles, and obtains the set of corrupted keys $\vec{\mathbf{pk}}_{\mathcal{CO}}$ and the set of queries $\mathbf{\Sigma}_{\mathcal{SO}}$.
3) $\mathcal{A}$ then chooses $(\vec{\mathbf{pk}}', \mathsf{C}, \pi')$, where $\vec{\mathbf{pk}}'$ are not the corrupted keys and no base signature has been queried on $(\mathsf{pp}, \vec{\mathbf{pk}}', \vec{\mathsf{m}})$ for any $\vec{\mathsf{m}} \in \mathcal{M}_\mathsf{C}$ to the base signature oracle before, as well as $(\mathsf{pp}, \vec{\mathbf{pk}}', \mathcal{M}_\mathsf{C}, \pi')$ has not been queried to the proving oracle before.

4) $\mathcal{A}$ wins, if $\mathcal{A}$ can produce an accepted proof for the ring referral scheme on $(\vec{\mathbf{pk}}', \mathcal{M}_{\mathsf{C}})$. Unforgeability is satisfied, if the probability of $\mathcal{A}$ winning is negligible.

Definition 5.9 (Unforgeability)**.** *A ring referral scheme* RR *satisfies* unforgeability*, if for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{c} \mathtt{Verify}[\mathsf{pp}, \vec{\mathbf{pk}}', \mathcal{M}_{\mathsf{C}}, \pi']=1 \\ \wedge\ \vec{\mathbf{pk}}' \subset \vec{\mathbf{pk}} \backslash \vec{\mathbf{pk}}_{\mathcal{CO}} \\ \wedge\ (\mathsf{pp}, \mathsf{pk}, \vec{\mathsf{m}}, \cdot) \notin \mathbf{\Sigma}_{\mathcal{BSO}}, \\ \forall (\mathsf{pk}, \vec{\mathsf{m}}) \in \vec{\mathbf{pk}}' \times \mathcal{M}_{\mathsf{C}} \\ \wedge\ (\mathsf{pp}, \vec{\mathbf{pk}}', \mathcal{M}_{\mathsf{C}}, \pi') \notin \mathbf{\Pi}_{\mathcal{PO}} \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda), \\ (\vec{\mathbf{pk}}, \vec{\mathbf{sk}}) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}], \\ (\vec{\mathbf{pk}}', \mathsf{C}, \pi') \\ \leftarrow \mathcal{A}^{\mathcal{CO}, \mathcal{BSO}, \mathcal{PO}}[\mathsf{pp}, \vec{\mathbf{pk}}] \end{array}\right] \leq \mathrm{negl}(\lambda).$$

Remark: The definition of unforgeability of ring referral scheme extends the one of a ring signature scheme, but there is a subtle difference between the two. The access to a base signature oracle is insufficient to forge a ring signature, but it is sufficient in a ring referral scheme. Hence, we need to limit its access in the unforgeability of ring referral scheme.

► **Issuer Anonymity.** This property means that the verifier cannot guess the signing issuer better than random guessing. The notion of issuer anonymity is captured as follows:

1) First, a security game instance is set up by generating a public parameter pp and a ring $\vec{\mathbf{pk}}$.
2) Adversary $\mathcal{A}$ picks two distinct issuers $(\mathcal{I}_{i_1}, \mathcal{I}_{i_2})$, a commitment value C and a message $\vec{\mathsf{m}} \in \mathcal{M}_{\mathsf{C}}$.
3) $\mathcal{I}_{i_1}$ produces a base signature σ_1 on $\vec{\mathsf{m}}$, whereas $\mathcal{I}_{i_2}$ produces a base signature σ_2 on $\vec{\mathsf{m}}$.
4) Without $\mathcal{A}$'s knowledge, a random $b \xleftarrow{\$} \{1, 2\}$ is selected and a ring referral proof π is generated from $(\mathcal{M}_{\mathsf{C}}, \vec{\mathsf{m}}, \sigma_b)$.
5) $\mathcal{A}$ wins, if $\mathcal{A}$ can guess b from $(\sigma_1, \sigma_2, \pi)$. Issuer anonymity is satisfied, if the net probability of $\mathcal{A}$ winning as compared with random guessing is negligible.
6) Note that $\mathcal{A}$ can access the corruption oracle $\mathcal{CO}$ to obtain any secret keys in the ring $\vec{\mathbf{pk}}$ (i.e., full key exposure), and the base signature oracle $\mathcal{BSO}$ to obtain any message-base signature pair, including σ_1 and σ_2 of $\mathcal{I}_{i_1}$ and $\mathcal{I}_{i_2}$ on $\vec{\mathsf{m}}$ (i.e., full signature exposure).

Definition 5.10 (Issuer Anonymity)**.** *A ring referral scheme* RR *satisfies* issuer anonymity*, if for any PPT adversary $\mathcal{A}$:*

$$\left|\Pr\left[\begin{array}{c} i_1 \neq i_2 \\ \wedge\ b' = b \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda), \\ (\vec{\mathbf{pk}}, \vec{\mathbf{sk}}) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}], \\ (i_1, i_2, \mathsf{C}, \vec{\mathsf{m}} \in \mathcal{M}_{\mathsf{C}}) \leftarrow \mathcal{A}[\mathsf{pp}, \vec{\mathbf{pk}}], \\ \sigma_1 \leftarrow \mathtt{Sig.Sign}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}_{i_1}], \\ \sigma_2 \leftarrow \mathtt{Sig.Sign}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}_{i_2}], \\ b \xleftarrow{\$} \{1, 2\}, \\ \pi \leftarrow \mathtt{Prove}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}_{\mathsf{C}}, \vec{\mathsf{m}}, \sigma_b], \\ b' \leftarrow \mathcal{A}^{\mathcal{CO}, \mathcal{BSO}, \mathcal{PO}}[\mathsf{pp}, \vec{\mathbf{pk}}, \sigma_1, \sigma_2, \pi] \end{array}\right] - \tfrac{1}{2}\right| \leq \mathrm{negl}(\lambda).$$

► **Strong User Anonymity.** This property unlinks the message, signature and the ring referral proof, and hence prevents the issuers and the verifier from jointly correlating a ring referral with a particular user[4]. The notion of strong user anonymity is captured by the following security game:

1) First, a security game instance is set up by generating a public parameter pp and a ring $\vec{\mathbf{pk}}$.
2) Adversary $\mathcal{A}$ picks an issuer $\mathcal{I}_i$, a commitment value C and two distinct messages $\vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2 \in \mathcal{M}_{\mathsf{C}}$.
3) $\mathcal{I}_i$ produces base signatures σ_1 on $\vec{\mathsf{m}}_1$ and σ_2 on $\vec{\mathsf{m}}_2$.
4) Without $\mathcal{A}$'s knowledge, a random $b \xleftarrow{\$} \{1, 2\}$ is selected and a ring referral proof π is generated from $(\mathcal{M}_{\mathsf{C}}, \vec{\mathsf{m}}_b, \sigma_b)$.
5) $\mathcal{A}$ wins, if $\mathcal{A}$ can guess b from $(\sigma_1, \sigma_2, \pi)$. Strong user anonymity is satisfied, if the net probability of $\mathcal{A}$ winning as compared with random guessing is negligible.
6) Note that $\mathcal{A}$ can access the corruption oracle $\mathcal{CO}$ to obtain any secret keys in the ring $\vec{\mathbf{pk}}$ (i.e., full key exposure), and the base signature oracle $\mathcal{BSO}$ to obtain any message-base signature pair, including σ_1 and σ_2 on $\vec{\mathsf{m}}_1$ and $\vec{\mathsf{m}}_2$, respectively (i.e., full signature exposure).

Definition 5.11 (Strong User Anonymity)**.** *A ring referral scheme* RR *satisfies* user anonymity*, if for any PPT adversary $\mathcal{A}$:*

$$\left|\Pr\left[\begin{array}{c} \vec{\mathsf{m}}_1 \neq \vec{\mathsf{m}}_2 \\ \wedge\ b' = b \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda), \\ (\vec{\mathbf{pk}}, \vec{\mathbf{sk}}) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}], \\ (i, \mathsf{C}, \vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2 \in \mathcal{M}_{\mathsf{C}}) \leftarrow \mathcal{A}[\mathsf{pp}, \vec{\mathbf{pk}}], \\ \sigma_1 \leftarrow \mathtt{Sig.Sign}[\mathsf{pp}, \vec{\mathsf{m}}_1, \mathsf{sk}_i], \\ \sigma_2 \leftarrow \mathtt{Sig.Sign}[\mathsf{pp}, \vec{\mathsf{m}}_2, \mathsf{sk}_i], \\ b \xleftarrow{\$} \{1, 2\}, \\ \pi \leftarrow \mathtt{Prove}[\mathsf{pp}, \vec{\mathbf{pk}}, \mathcal{M}_{\mathsf{C}}, \vec{\mathsf{m}}_b, \sigma_b], \\ b' \leftarrow \mathcal{A}^{\mathcal{CO}, \mathcal{BSO}, \mathcal{PO}}[\mathsf{pp}, \vec{\mathbf{pk}}, \sigma_1, \sigma_2, \pi] \end{array}\right] - \tfrac{1}{2}\right| \leq \mathrm{negl}(\lambda).$$

Remark: The definition of strong user anonymity requires message hiding (i.e., $\mathcal{M}_{\mathsf{C}}$ is not a singleton set). For a scheme without message hiding, strong user anonymity is not satisfiable by default. As compared with issuer anonymity, the adversary needs to guess a randomly chosen message (associated with a specific user), rather than a randomly chosen issuer.

6. Multi-Message BBS Signature Schemes

In this section, we present the base signature scheme for our ring referral scheme, based on multi-message BBS signatures. Fig. 2a presents the standard multi-message BBS signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ in [12], [29], [30]. Note that the verification time is linear in the number of messages (M), because of M $\mathbb{G}_1$ exponentiations for $\vec{\mathbf{H}}^{\vec{\mathsf{m}}}$. Next, we will design a new compressed message-hiding BBS signature scheme by outsourcing the computation of $\vec{\mathbf{H}}^{\vec{\mathsf{m}}}$ to a third party and verifying the third party's computation by Dory, as well as hiding the message from the verifier.

6.1. Compressed Multi-Message BBS Signature

Let $\vec{\mathsf{m}}' \triangleq (\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}}) \in \mathbb{Z}_p^{M+1}$, $\vec{\mathbf{H}}' \triangleq (\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}) \in \mathbb{G}_1^{M+1}$, $\vec{\mathbf{\Gamma}}' \triangleq (\vec{\mathbf{\Gamma}}, P_1) \in \mathbb{G}_1^{M+1}$, $\vec{\mathbf{\Lambda}}' \triangleq (\vec{\mathbf{\Lambda}}, \mathbf{1}_{\mathbb{G}_2}) \in \mathbb{G}_2^{M+1}$. Given $G_2 \in \mathbb{G}_2 \backslash \{\mathbf{1}_{\mathbb{G}_2}\}$, denote $\vec{G}_2 \triangleq (G_2, ..., G_2) \in \mathbb{G}_2^{M+1}$.

We outline the basic idea of a compressed message-hiding multi-message BBS signature scheme as follows:

1) First, the signer produces a valid BBS signature (σ_0, σ_1) on $\vec{\mathsf{m}}$ and hides message $\vec{\mathsf{m}}$ in a Pedersen commitment, defined as $\mathtt{Cm}_{\mathsf{m}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{m}}} \cdot P_1^{\mathsf{r}_{\mathsf{m}}} = \vec{\mathbf{\Gamma}}'^{\vec{\mathsf{m}}'}$.

4. We only consider message-level anonymity, but not network-level anonymity, which should be protected by anonymous routing.

$\mathtt{Sig}_{\mathtt{bbs}} = (\mathtt{Setup}_{\mathtt{bbs}}, \mathtt{KeyGen}_{\mathtt{bbs}}, \mathtt{Sign}_{\mathtt{bbs}}, \mathtt{VfySig}_{\mathtt{bbs}})$

- $\mathtt{Setup}_{\mathtt{bbs}}[1^\lambda] \mapsto \mathsf{pp}$:
  $G_1 \xleftarrow{\$} \mathbb{G}_1,\ G_2 \xleftarrow{\$} \mathbb{G}_2^*,\ \vec{\mathbf{H}} \xleftarrow{\$} \mathbb{G}_1^M,\ \mathsf{pp} \triangleq (G_1, G_2, \vec{\mathbf{H}})$
  RETURN pp
- $\mathtt{KeyGen}_{\mathtt{bbs}}[\mathsf{pp}] \mapsto (\mathsf{sk}, \mathsf{pk})$:
  $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_p, \quad \mathsf{pk} \triangleq G_2^{\mathsf{sk}} \in \mathbb{G}_2$
  RETURN $(\mathsf{sk}, \mathsf{pk})$
- $\mathtt{Sign}_{\mathtt{bbs}}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}] \mapsto \sigma$:
  $R \triangleq G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}} \in \mathbb{G}_1, \quad \sigma_1 \xleftarrow{\$} \mathbb{Z}_p, \quad \sigma_0 \triangleq R^{\frac{1}{\mathsf{sk}+\sigma_1}} \in \mathbb{G}_1$
  RETURN $\sigma \triangleq (\sigma_0, \sigma_1)$
- $\mathtt{VfySig}_{\mathtt{bbs}}[\mathsf{pp}, \mathsf{pk}, \vec{\mathsf{m}}, \sigma = (\sigma_0, \sigma_1)] \mapsto \{0,1\}$:
  RETURN $\mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}},\ G_2) \in \{0,1\}$

(a) Standard multi-message BBS signature [12], [29], [30].

$\mathtt{Sig}_{\mathtt{c.bbs}} = (\mathtt{Setup}_{\mathtt{c.bbs}}, \mathtt{KeyGen}_{\mathtt{c.bbs}}, \mathtt{Sign}_{\mathtt{c.bbs}}, \mathtt{VfySig}_{\mathtt{c.bbs}})$

- $\mathtt{Setup}_{\mathtt{c.bbs}}[1^\lambda] \mapsto \mathsf{pp}$:
  $G_1, P_1 \xleftarrow{\$} \mathbb{G}_1,\ G_2 \xleftarrow{\$} \mathbb{G}_2^*,\ \vec{\mathbf{H}}, \vec{\mathbf{\Gamma}} \xleftarrow{\$} \mathbb{G}_1^M,\ \vec{\mathbf{\Lambda}} \xleftarrow{\$} \mathbb{G}_2^M$
  $\mathtt{D}_1 \triangleq \mathsf{e}(\vec{\mathbf{H}}, \vec{\mathbf{\Lambda}}) \in \mathbb{G}_T, \quad \mathsf{pp} \triangleq (G_1, G_2, P_1, \vec{\mathbf{H}}, \vec{\mathbf{\Gamma}}, \vec{\mathbf{\Lambda}})$
  RETURN pp
- $\mathtt{KeyGen}_{\mathtt{c.bbs}}[\mathsf{pp}] \mapsto (\mathsf{sk}, \mathsf{pk})$:
  RETURN $\mathtt{KeyGen}_{\mathtt{bbs}}[\mathsf{pp}]$
- $\mathtt{Sign}_{\mathtt{c.bbs}}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}] \mapsto \sigma$:
  RETURN $\mathtt{Sign}_{\mathtt{bbs}}[\mathsf{pp}, \vec{\mathsf{m}}, \mathsf{sk}]$
- $\mathtt{Compress}_{\mathtt{c.bbs}}[\mathsf{pp}, \vec{\mathsf{m}}] \mapsto (\mathtt{Cm}_{\mathsf{m}}, \pi)$:
  $\mathsf{r}_{\mathsf{m}},\ \mathsf{r}_{\mathtt{D}},\ \mathsf{r}_{\mathtt{R}} \xleftarrow{\$} \mathbb{Z}_p^*,\ \vec{\mathsf{m}}' \triangleq (\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}}) \in \mathbb{Z}_p^{M+1},\ \vec{\mathbf{H}}' \triangleq (\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}) \in \mathbb{G}_1^{M+1}$
  $\mathtt{Cm}_{\mathsf{m}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{m}}} \cdot P_1^{\mathsf{r}_{\mathsf{m}}},\ \mathtt{D}_0 \triangleq \mathsf{e}(\vec{\mathbf{H}}'^{\vec{\mathsf{m}}'}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}},\ \mathtt{R} \triangleq \mathtt{Q}^{\mathsf{r}_{\mathtt{R}}},\ \mathtt{D}_2 \triangleq \mathsf{e}(\mathtt{Cm}_{\mathsf{m}}, G_2)$
  $\pi' \triangleq \mathcal{P}_{\mathrm{do.ip}}[M{+}1, (\vec{\mathbf{\Gamma}}, P_1), (\vec{\mathbf{\Lambda}}, \mathbf{1}_{\mathbb{G}_2}), \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2;\ \vec{G}_2^{\circ \vec{\mathsf{m}}'}, \vec{\mathbf{H}}', \mathsf{r}_{\mathtt{D}}, 0, 0]$
  // Prove the same $\vec{\mathsf{m}}'$ committed in both $\mathtt{D}_0, \mathtt{D}_2$ by Dory
  $\theta \triangleq \mathtt{Hash}[\mathtt{Cm}_{\mathsf{m}}, \mathtt{D}_0, \mathtt{R}, \pi']$ *// Create public-coin challenge via Fiat-Shamir*
  $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{D}} + \theta \cdot \mathsf{r}_{\mathtt{R}}, \quad \pi \triangleq (\mathtt{D}_0, \mathtt{R}, \pi', \mathsf{r}')$
  RETURN $(\mathtt{Cm}_{\mathsf{m}}, \pi)$
- $\mathtt{VfyCSig}_{\mathtt{c.bbs}}[\mathsf{pp}, \mathsf{pk}, \mathtt{Cm}_{\mathsf{m}}, \sigma = (\sigma_0, \sigma_1), \pi = (\mathtt{D}_0, \mathtt{R}, \pi', \mathsf{r}')] \mapsto \{0,1\}$:
  $\mathtt{D}_2 \triangleq \mathsf{e}(\mathtt{Cm}_{\mathsf{m}}, G_2), \quad \theta \triangleq \mathtt{Hash}[\mathtt{Cm}_{\mathsf{m}}, \mathtt{D}_0, \mathtt{R}, \pi']$
  RETURN $\Big(\mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}'} \stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathtt{D}_0 \cdot \mathtt{R}^{\theta} \wedge$
  $\mathcal{V}_{\mathrm{do.ip}}[M{+}1, (\vec{\mathbf{\Gamma}}, P_1), (\vec{\mathbf{\Lambda}}, \mathbf{1}_{\mathbb{G}_2}), \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2, \pi']\Big) \in \{0,1\}$

(b) Compressed message-hiding multi-msg BBS signature scheme.

Figure 2: Multi-message BBS signature schemes.

2) Rather than standard BBS signature verification, the verifier outsources the task of computing $\mathtt{D}_0 \triangleq \mathsf{e}(\vec{\mathbf{H}}'^{\vec{\mathsf{m}}'}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}}$. To relate $\mathtt{Cm}_{\mathsf{m}}$ and $\mathtt{D}_0$, we use Dory to check the following:

$$\begin{cases} \mathtt{D}_0 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{H}}', \vec{G}_2^{\circ \vec{\mathsf{m}}'}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}} = \mathsf{e}(\vec{\mathbf{H}}'^{\vec{\mathsf{m}}'}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}}, \\ \mathtt{D}_1 \triangleq \mathsf{e}(\vec{\mathbf{H}}', \vec{\mathbf{\Lambda}}') \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{H}}, \vec{\mathbf{\Lambda}}), \\ \mathtt{D}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ \vec{\mathsf{m}}'}) = \mathsf{e}(\mathtt{Cm}_{\mathsf{m}}, G_2), \end{cases} \tag{2}$$

where $\vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}'$ are known generators and $\vec{\mathbf{H}}', \vec{G}_2^{\circ \vec{\mathsf{m}}'}$ are the witness. Note that $\mathtt{D}_1 \triangleq \mathsf{e}(\vec{\mathbf{H}}, \vec{\mathbf{\Lambda}})$ can be precomputed.

3) Given a public-coin challenge θ (via Fiat-Shamir transformation), the signer provides $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{D}} + \theta \cdot \mathsf{r}_{\mathtt{R}}$.

4) Finally, the verification of a BBS signature with respect to $\mathtt{Cm}_{\mathsf{m}}$ and pk can be attained by checking the following:

$$\begin{aligned} \mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}'} &\stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathtt{D}_0 \cdot \mathtt{R}^{\theta} \\ &= \mathsf{e}(G_1, G_2) \cdot \mathsf{e}(\vec{\mathbf{H}}'^{\vec{\mathsf{m}}'}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}} \cdot \mathtt{Q}^{\theta \mathsf{r}_{\mathtt{R}}}, \\ \Rightarrow\ \mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) &\stackrel{?}{=} \mathsf{e}(G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}},\ G_2), \text{ i.e. BBS check} \end{aligned}$$

The detailed compressed message-hiding multi-message BBS signature scheme $\mathtt{Sig}_{\mathtt{c.bbs}}$ is presented in Fig. 2b.

Theorem 6.1. *Assume the standard scheme $\mathtt{Sig}_{\mathtt{bbs}}$ is complete and satisfies EU-CMA; AFGHO and Pedersen commitments are complete, perfectly hiding and computationally binding; Dory argument is complete and satisfies CWE, SHVZH; Hash is modeled as a random oracle. Then, the compressed scheme $\mathtt{Sig}_{\mathtt{c.bbs}}$ is complete and satisfies unforgeability, SHVZK in the random oracle model.*

Proof. See Appendix F. □

The verification of $\mathtt{Sig}_{\mathtt{c.bbs}}$ takes 3 pairings and $10 \log M$ $\mathbb{G}_T$ exponentiations, with signature-independent precomputation, which is a significant improvement over linearly verifiable standard multi-message BBS signature.

6.2. Application to User-Binding Credentials

For decentralized identity, we particularly apply multi-message BBS signature scheme to sign user-binding credentials. The basic idea is that an issuer signs a credential (i.e., a multi-message commitment) containing a secret key that is only known to a user. The user can later prove that the credential is binding to her by a proof-of-knowledge of the secret key. First, each user generates a key pair $(\mathsf{upk}, \mathsf{usk}) \in \mathbb{G}_1 \times \mathbb{Z}_p^*$, where $\mathsf{usk} \xleftarrow{\$} \mathbb{Z}_p^*$ and $\mathsf{upk} \triangleq G^{\mathsf{usk}}$ for a public parameter $G \xleftarrow{\$} \mathbb{G}_1$. Then, an issuer is asked for a multi-message signature by a user with public key upk, he applies the method $\mathtt{Sign}_{\mathtt{bbs}}$ in Fig. 2a, but returns the signature $\sigma' \triangleq (\sigma_0', \sigma_1)$, where $\sigma_0' \triangleq (\mathsf{upk} \cdot R)^{\frac{1}{\mathsf{sk}+\sigma_1}} \in \mathbb{G}_1$. Since $\mathsf{upk} \cdot R = G_1 \cdot G^{\mathsf{usk}} \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}}$, σ' is a BBS signature on the extended multi-message $(\mathsf{usk}, \vec{\mathsf{m}})$, for which the user can produce a ring referral proof from our multi-message ring referral scheme.

7. Single-Message Ring Referral Scheme

For the clarity of presentation, this section only presents the single-message ring referral scheme considering single-message BBS signatures. It basically consists of two parts: (1) a proof-of-knowledge for a BBS signature that the user possesses a valid tuple of public key, signed message and corresponding signature; and (2) a proof-of-knowledge for a ring that a privately known public key is within a ring.

7.1. Proof-of-Knowledge for BBS Signature

The prover knows a public key pk, a signed message m and the corresponding signature $\sigma = (\sigma_0, \sigma_1)$. She needs to prove that $(\mathsf{pk}, \mathsf{m}, \sigma)$ is a valid single-message BBS signature tuple. We outline the basic idea of a proof-of-knowledge protocol:

1) First, the prover commits $(\mathsf{pk}, \mathsf{m}, \sigma_0, \sigma_1)$ to $\mathtt{cm}_{\mathsf{pk}} \triangleq \mathsf{e}(G_1, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}}$, $\mathtt{cm}_{\mathsf{m}} \triangleq \mathsf{e}(H, G_2)^{\mathsf{m}} \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{m}}}$, $\mathtt{cm}_{\sigma_0} \triangleq \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}}$, $\mathtt{cm}_{\sigma_1} \triangleq \mathsf{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}$, and additional commitments $\mathtt{Z}_1 \triangleq \mathsf{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}}$, $\mathtt{Z}_2 \triangleq \mathsf{e}(\sigma_0, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_2}}$, which enables the checking of BBS signature.
2) The verifier then checks the well-formedness of these commitments. First, the verifier can perform Schnorr proof-of-knowledge check on $\mathtt{cm}_{\mathsf{m}}$ and $\mathtt{cm}_{\sigma_1}$ using protocol Π_{chkcm} (see Sec. 7.4). Next, the verifier can invoke Dory scalar-product protocol to check as follows:
$$\begin{cases} \mathtt{D}_0 \triangleq \mathtt{Z}_1 \stackrel{?}{=} \mathsf{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}}, \\ \mathtt{D}_1 \triangleq \mathtt{cm}_{\sigma_0} \stackrel{?}{=} \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}}, \\ \mathtt{D}_2 \triangleq \mathtt{cm}_{\sigma_1} \stackrel{?}{=} \mathsf{e}(G_1, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}, \end{cases} \tag{3}$$
where G_1, G_2 are known generators and $\sigma_0, G_2^{\sigma_1}$ are the witness. Since we fix the generators to be G_1, G_2, and $\mathtt{cm}_{\sigma_1}$ has been checked via Schnorr proof-of-knowledge, the well-formedness of $\mathtt{Z}_1$ and $\mathtt{cm}_{\sigma_0}$ follows from the knowledge soundness of Dory. Similarly, the verifier can check the well-formedness of $\mathtt{cm}_{\mathsf{pk}}$ and $\mathtt{Z}_2$ by Dory scalar-product protocol. We also need to ensure that the commitment $\mathtt{cm}_{\mathsf{pk}}$ contains a public key pk for a BBS signature in $\Pi_{\mathsf{bbs.pms}}$. This is done in the first scalar Dory call $\Pi_{\mathsf{do.sp}}$ in $\Pi_{\mathsf{bbs.pms}}$, which checks pk in $\mathtt{cm}_{\mathsf{pk}}$ and $\mathtt{Z}_2$.
3) Finally, the verification of a BBS signature with respect to $\mathtt{Cm}_{\mathsf{m}}$, $\mathtt{Z}_2$ and $\mathtt{Z}_1$ can be checked as follows:
$$\begin{aligned} \mathtt{Z}_1 \cdot \mathtt{Z}_2 &\stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathtt{cm}_{\mathsf{m}} \cdot \mathtt{Q}^{\mathsf{r}'}, \\ \Rightarrow\ \mathsf{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathsf{e}(\sigma_0, \mathsf{pk}) &\stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathsf{e}(H, G_2)^{\mathsf{m}}, \\ \Rightarrow\ \mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) &\stackrel{?}{=} \mathsf{e}(G_1 \cdot H^{\mathsf{m}},\ G_2), \end{aligned}$$
where $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{Z}_1} + \mathsf{r}_{\mathtt{Z}_2} - \mathsf{r}_{\mathsf{m}}$ is provided by the prover.

The detailed protocol $\Pi_{\mathsf{bbs.pms}}$ is described in Fig. 3, which checks the proof-of-knowledge of a valid tuple of $(\mathsf{pk}, \mathsf{m}, \sigma)$ in single-message BBS signature scheme.

7.2. Proof-of-Knowledge for a Ring

Given $\mathtt{cm}_{\mathsf{pk}}$, the prover should prove that the committed pk is one of the ring $\vec{\mathsf{pk}} \triangleq (\mathsf{pk}_i)_{i\in[n]}$, such that $\mathsf{pk} = \mathsf{pk}_{i^*}$. We outline the basic idea of a proof-of-knowledge protocol:

1) In the setup, let $\mathsf{h}_i \triangleq \mathtt{Hash}[\mathsf{pk}_i]$ for $i \in [n]$ and $\vec{\mathbf{K}} \triangleq (K_i \triangleq \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})_{i\in[n]}$. The setting of $\vec{\mathbf{K}}$ is to prevent an attack of manipulating $\vec{\mathsf{pk}}$ to create a false proof.
2) The prover defines a selector vector $\vec{\mathbf{b}} = (b_i)_{i\in[n]} \in \{0,1\}^n$, where $b_{i^*} = 1$ and $b_i = 0$, $\forall i \neq i^*$. Then she commits $\vec{\mathbf{b}}$ to an AFGHO commitment with commitment keys $(\vec{\mathbf{K}}, \mathtt{Q})$ as $\mathtt{cm}_1 \triangleq \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}})^{\vec{\mathbf{b}}} \cdot \mathtt{Q}^{\mathsf{r}_1}$.

$\Pi_{\mathsf{bbs.pms}}\Big[\mathtt{cm}_{\mathsf{pk}} \in \mathbb{G}_T, \mathtt{cm}_{\mathsf{m}} \in \mathbb{G}_T, \mathtt{cm}_{\sigma_0} \in \mathbb{G}_T, \mathtt{cm}_{\sigma_1} \in \mathbb{G}_T;$
$\mathsf{pk} \in \mathbb{G}_2, \mathsf{m} \in \mathbb{Z}_p, \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p\Big]$

$\mathcal{P}$: PARSE $\mathtt{cm}_{\mathsf{pk}} = \mathsf{e}(G_1, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}}$, $\mathtt{cm}_{\mathsf{m}} = \mathsf{e}(H, G_2)^{\mathsf{m}} \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{m}}}$
$\mathtt{cm}_{\sigma_0} = \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}}$, $\mathtt{cm}_{\sigma_1} = \mathsf{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}$
$\mathsf{r}_{\mathtt{Z}_1}, \mathsf{r}_{\mathtt{Z}_2} \stackrel{\$}{\leftarrow} \mathbb{Z}_p^*$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{Z}_1 \triangleq \mathsf{e}(\sigma_0, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}} \in \mathbb{G}_T$, $\mathtt{Z}_2 \triangleq \mathsf{e}(\sigma_0, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_2}} \in \mathbb{G}_T$
$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{chkcm}}[\mathtt{cm}_{\mathsf{m}}, \mathsf{e}(H, G_2), \mathtt{Q};\ \mathsf{m}, \mathsf{r}_{\mathsf{m}}]$ *// Check* m *in* $\mathtt{cm}_{\mathsf{m}}$
RUN $\Pi_{\mathsf{chkcm}}[\mathtt{cm}_{\sigma_1}, \mathsf{e}(G_1, G_2), \mathtt{Q};\ \sigma_1, \mathsf{r}_{\sigma_1}]$ *// Check* σ_1 *in* $\mathtt{cm}_{\sigma_1}$
RUN $\Pi_{\mathsf{do.sp}}[G_1, G_2, \mathtt{Z}_1, \mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1};\ \sigma_0, G_2^{\sigma_1}, \mathsf{r}_{\mathtt{Z}_1}, \mathsf{r}_{\sigma_0}, \mathsf{r}_{\sigma_1}]$
// Check $(\sigma_0, G_2^{\sigma_1})$ *in* $\mathtt{Z}_1$ *given* $\mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1}$
RUN $\Pi_{\mathsf{do.sp}}[G_1, G_2, \mathtt{Z}_2, \mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\mathsf{pk}};\ \sigma_0, \mathsf{pk}, \mathsf{r}_{\mathtt{Z}_2}, \mathsf{r}_{\sigma_0}, \mathsf{r}_{\mathsf{pk}}]$
// Check (σ_0, pk) *in* $\mathtt{Z}_2$ *given* $\mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\mathsf{pk}}$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{Z}_1} + \mathsf{r}_{\mathtt{Z}_2} - \mathsf{r}_{\mathsf{m}}$
$\mathcal{V}$: CHECK $\mathtt{Z}_1 \cdot \mathtt{Z}_2 \stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathtt{cm}_{\mathsf{m}} \cdot \mathtt{Q}^{\mathsf{r}'}$
// Equivalently checking $\mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(G_1 \cdot H^{\mathsf{m}},\ G_2)$

Figure 3: Protocol $\Pi_{\mathsf{bbs.pms}}$ for the proof-of-knowledge of $(\mathsf{pk}, \mathsf{m}, \sigma)$ in single-message BBS signature scheme.

3) The prover then proves that $\vec{\mathbf{b}}$ is a unit vector committed in $\mathtt{cm}_1$ using protocol Π_{chkuv} via Dory (see Sec. 7.4). This proves the knowledge of the index $i^* \in [n]$, and it remains to prove the knowledge of pk_{i^*}.
4) The prover also commits h_{i^*} as $\mathtt{cm}_2 \triangleq \mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}} \cdot \mathtt{Q}^{\mathsf{r}_2}$. The verifier performs Schnorr proof-of-knowledge to check $\mathtt{cm}_2$ using protocol Π_{chkcm}.
5) The verification of $\mathsf{pk} = \mathsf{pk}_{i^*}$ can be checked as follows:
$$\begin{aligned} \mathtt{cm}_1 &\stackrel{?}{=} \mathtt{cm}_{\mathsf{pk}} \cdot \mathtt{cm}_2 \cdot \mathtt{Q}^{\mathsf{r}'_1}, \\ \Rightarrow\ \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}})^{\vec{\mathbf{b}}} &\stackrel{?}{=} \mathsf{e}(G_1, \mathsf{pk}) \cdot \mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}}, \\ \Rightarrow\ \prod\nolimits_{i\in[n]} \mathsf{e}\big(G_1, (\mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})^{b_i}\big) &\stackrel{?}{=} \mathsf{e}(G_1, \mathsf{pk} \cdot G_2^{\mathsf{h}_{i^*}}), \end{aligned}$$
where $\mathsf{r}'_1 \triangleq \mathsf{r}_1 - \mathsf{r}_{\mathsf{pk}} - \mathsf{r}_2$ is provided by the prover.

7.3. Interactive Ring Referral Protocol

Given a ring of issuers with public keys $\vec{\mathsf{pk}} \triangleq (\mathsf{pk}_i)_{i\in[n]}$, the prover/user obtains a BBS signature $\sigma = (\sigma_0, \sigma_1)$ on a message m from an issuer $\mathsf{pk}_{i^*}, i^* \in [n]$. Define the relation for ring referral of single-message BBS signature by
$$\mathscr{R}_{\mathsf{RR.bbs}} \triangleq \Big\{ \vec{\mathsf{pk}} \in \mathbb{G}_2^n;\quad \mathsf{m} \in \mathbb{Z}_p, \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p,$$
$$i^* \in [n], \mathsf{pk}_{i^*} \in \vec{\mathsf{pk}} \ \Big|\ \mathsf{e}(\sigma_0, G_2^{\sigma_1} \cdot \mathsf{pk}_{i^*}) = \mathsf{e}(G_1 \cdot H^{\mathsf{m}}, G_2) \Big\}.$$
We combine the proof-of-knowledge for a BBS signature and a proof-of-knowledge for a ring in protocol $\Pi_{\mathsf{RR.bbs}}$ in Fig. 3, as a proof-of-knowledge for $\mathscr{R}_{\mathsf{RR.bbs}}$.

7.4. Auxiliary Protocols for Ring Referral

We complete the ring referral protocol by describing two auxiliary protocols:

$\Pi_{\mathsf{RR.bbs}}\left[\vec{\mathsf{pk}} \in \mathbb{G}_2^n;\ \mathsf{m} \in \mathbb{Z}_p, \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p,\ i^* \in [n], \mathsf{pk}_{i^*} \in \vec{\mathsf{pk}}\right]$

$\textsc{Setup}: (\mathsf{h}_i \triangleq \mathtt{Hash}[\mathsf{pk}_i] \in \mathbb{Z}_p)_{i\in[n]}$

$\vec{\mathbf{K}} \triangleq (K_i \triangleq \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})_{i\in[n]} \in \mathbb{G}_2^n$

$\mathcal{P}: \mathsf{r}_{\mathsf{pk}},\ \mathsf{r}_{\mathsf{m}},\ \mathsf{r}_{\sigma_0},\ \mathsf{r}_{\sigma_1} \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}: \mathtt{cm}_{\mathsf{pk}} = \mathsf{e}(G_1, \mathsf{pk}_{i^*}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}} \in \mathbb{G}_T$

$\mathtt{cm}_{\mathsf{m}} = \mathsf{e}(H, G_2)^{\mathsf{m}} \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{m}}} \in \mathbb{G}_T$

$\mathtt{cm}_{\sigma_0} = \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}} \in \mathbb{G}_T$

$\mathtt{cm}_{\sigma_1} = \mathsf{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}} \in \mathbb{G}_T$

$\mathcal{V}\ \&\ \mathcal{P}: \textsc{Run}\ \Pi_{\mathsf{bbs.pms}}[\mathtt{cm}_{\mathsf{pk}}, \mathtt{cm}_{\mathsf{m}}, \mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1};\ \mathsf{pk}_{i^*}, \mathsf{m}, \sigma = (\sigma_0, \sigma_1)]$

$\mathcal{P}: \vec{\mathbf{b}} \triangleq (b_i)_{i\in[n]},$ where $b_i \triangleq \begin{cases} 0, \text{ if } i \neq i^* \\ 1, \text{ if } i = i^* \end{cases}$

$\mathsf{r}_1,\ \mathsf{r}_2 \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}: \mathtt{cm}_1 \triangleq \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1} \in \mathbb{G}_T$

$\mathtt{cm}_2 \triangleq \mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}} \cdot \mathtt{Q}^{\mathsf{r}_2} \in \mathbb{G}_T$

$\mathsf{r}'_1 \triangleq \mathsf{r}_1 - \mathsf{r}_{\mathsf{pk}} - \mathsf{r}_2 \in \mathbb{Z}_p$

$\mathcal{V}\ \&\ \mathcal{P}: \textsc{Run}\ \Pi_{\mathtt{chkuv}}[\mathtt{cm}_1, \vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}, \mathtt{Q};\ \vec{G}_1^{\circ\vec{\mathbf{b}}}, \mathsf{r}_1]$

// Check $\vec{\mathbf{b}}$ being a unit basis vector

$\textsc{Run}\ \Pi_{\mathtt{chkcm}}[\mathtt{cm}_2, \mathsf{e}(G_1, G_2), \mathtt{Q};\ \mathsf{h}_{i^*}, \mathsf{r}_2]$ *// Check h_{i^*} in $\mathtt{cm}_2$*

$\mathcal{V}: \textsc{Check}\ \mathtt{cm}_1 \stackrel{?}{=} \mathtt{cm}_{\mathsf{pk}} \cdot \mathtt{cm}_2 \cdot \mathtt{Q}^{\mathsf{r}'_1}$

// Equivalently checking $\mathsf{e}(\vec{G}_1, \vec{\mathbf{K}})^{\vec{\mathbf{b}}} \stackrel{?}{=} \mathsf{e}(G_1, \mathsf{pk}_{i^*} \cdot G_2^{\mathsf{h}_{i^*}})$

Figure 4: Ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ for single-message BBS signature.

- **Protocol $\Pi_{\mathtt{chkcm}}$:** This is a standard Schnorr proof-of-knowledge protocol for checking the knowledge of a committed value in a Pedersen or AFGHO commitment. The protocol is described in Fig. 5a.
- **Protocol $\Pi_{\mathtt{chkuv}}$:** This protocol checks if a committed vector $\vec{\mathbf{b}}$ is a unit basis vector, satisfying the following:

$$\begin{cases} \langle \vec{\mathbf{b}},\ \vec{\mathbf{1}} \rangle \stackrel{?}{=} 1, \\ \vec{\mathbf{b}} \circ (\vec{\mathbf{b}} - \vec{\mathbf{1}}) \stackrel{?}{=} \vec{\mathbf{0}}. \end{cases} \tag{4}$$

We adopt the technique from LLRing [19]. Note that Eqn. (4) is equivalent to the following via bilinear pairing:

$$\begin{cases} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{G}_2) = \mathsf{e}(G_1, G_2)^{\langle \vec{\mathbf{b}}, \vec{\mathbf{1}} \rangle} \stackrel{?}{=} \mathsf{e}(G_1, G_2), \\ \mathsf{e}(\vec{\mathbf{\Gamma}}^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}) = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}})^{\vec{\mathbf{b}}\circ\vec{\mathbf{b}}} \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}})^{\vec{\mathbf{b}}}, \end{cases} \tag{5}$$

where $\vec{\mathbf{\Gamma}} \xleftarrow{\$} \mathbb{G}_1^n$, $G_1 \xleftarrow{\$} \mathbb{G}_1$, $G_2 \xleftarrow{\$} \mathbb{G}_2$, and $\vec{\mathbf{K}} \xleftarrow{\$} \mathbb{G}_2^n$ are public parameters. We write $\vec{G}_1 \triangleq (G_1, ...G_1) \in \mathbb{G}_1^n$ and $\vec{G}_2 \triangleq (G_2, ...G_2) \in \mathbb{G}_2^n$. Eqn. (5) can be re-expressed as the following three sets of inner-product relations:

$$\begin{cases} \mathtt{D}_0 \triangleq \mathsf{e}(G_1, G_2) \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{G}_2) = \mathsf{e}(G_1, G_2)^{\langle \vec{\mathbf{b}}, \vec{\mathbf{1}} \rangle} \\ \mathtt{D}_1 \triangleq \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathtt{D}_2 \triangleq \mathsf{e}(\vec{\mathbf{\Gamma}},\ \vec{G}_2). \end{cases} \tag{6}$$

$$\begin{cases} \mathtt{D}'_0 \triangleq \mathtt{cm}'' \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}) \cdot \mathtt{Q}^{\mathsf{r}''} = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}})^{\vec{\mathbf{b}}\circ\vec{\mathbf{b}}} \cdot \mathtt{Q}^{\mathsf{r}''} \\ \mathtt{D}'_1 \triangleq \mathtt{cm}'' \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}''}, \\ \mathtt{D}'_2 \triangleq \mathtt{cm}'' \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}},\ \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}) \cdot \mathtt{Q}^{\mathsf{r}''}. \end{cases} \tag{7}$$

$\Pi_{\mathtt{chkcm}}\left[\mathtt{cm} \in \mathbb{G}, P \in \mathbb{G}, Q \in \mathbb{G};\ x \in \mathbb{Z}_p, \mathsf{r} \in \mathbb{Z}_p\right]$

$\mathcal{P}: x' \xleftarrow{\$} \mathbb{Z}_p,\quad \mathsf{r}' \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}: \mathtt{cm}' \triangleq P^{x'} \cdot Q^{\mathsf{r}'} \in \mathbb{G}$

$\mathcal{P} \Leftarrow \mathcal{V}: \alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}: z \triangleq \alpha \cdot x + x',\quad \mathsf{r}'_z \triangleq \alpha \cdot \mathsf{r} + \mathsf{r}'$

$\mathcal{V}: \textsc{Check}\ P^z \cdot Q^{\mathsf{r}'_z} \stackrel{?}{=} \mathtt{cm}^\alpha \cdot \mathtt{cm}'$

(a) Protocol $\Pi_{\mathtt{chkcm}}$ for checking the knowledge of committed value in Pedersen commitment.

$\Pi_{\mathtt{chkuv}}\left[\mathtt{cm} \in \mathbb{G}_T, \vec{\mathbf{\Gamma}} \in \mathbb{G}_1^n, \vec{\mathbf{K}} \in \mathbb{G}_2^n, \mathtt{Q} \in \mathbb{G}_T;\ \vec{\mathbf{b}} \in \mathbb{Z}_p^n, \mathsf{r} \in \mathbb{Z}_p\right]$

$\mathcal{P}: \mathsf{r}'' \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P} \Rightarrow \mathcal{V}: \mathtt{cm}'' \triangleq \mathsf{e}(\vec{\mathbf{\Gamma}}^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}''}$

$\mathcal{V}\ \&\ \mathcal{P}: \textsc{Run}\ \Pi_{\mathtt{do.ip}}[n, \vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}, \mathsf{e}(G_1, G_2), \mathtt{cm}, \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2);\ \vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{G}_2, 0, \mathsf{r}, 0]$

$\textsc{Run}\ \Pi_{\mathtt{do.ip}}[n, \vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}, \mathtt{cm}'', \mathtt{cm}'', \mathtt{cm}'';\ \vec{\mathbf{\Gamma}}^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}, \mathsf{r}'', \mathsf{r}'', \mathsf{r}'']$

$\textsc{Run}\ \Pi_{\mathtt{do.ip}}[n, \vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}, \mathtt{cm}, \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}}), \mathtt{cm}'';\ \vec{G}_1, \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}, \mathsf{r}, 0, \mathsf{r}'']$

// Can be batched into a single Dory with generators $(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}})$

(b) Protocol $\Pi_{\mathtt{chkuv}}$ for checking the well-formedness of a unit basis vector.

Figure 5: Auxiliary protocols.

$$\begin{cases} \mathtt{D}''_0 \triangleq \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathtt{D}''_1 \triangleq \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}}), \\ \mathtt{D}''_2 \triangleq \mathtt{cm}'' \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}},\ \vec{\mathbf{K}}^{\circ\vec{\mathbf{b}}}) \cdot \mathtt{Q}^{\mathsf{r}''}. \end{cases} \tag{8}$$

Each of the above sets of inner-product relations can be checked by a Dory argument. Each of the terms $\mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2) = \mathsf{e}(\prod_{i\in[n]} \Gamma_i, G_2)$ and $\mathsf{e}(\vec{G}_1, \vec{\mathbf{K}}) = \mathsf{e}(G_1, \prod_{i\in[n]} K_i)$ can be pre-computed using 1 pairing and n group multiplications, instead of naïvely applying n pairings to compute pairing between vectors. Note that Eqns. (6)-(8) share the common generators $(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}})$. Hence, we can batch the three Dory arguments for checking Eqns. (6)-(8) into a single Dory argument. The full protocol is described in Fig. 5b.

7.5. Security Proofs for Ring Referral Protocol

We prove that as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs}}$, the protocol $\Pi_{\mathsf{RR.bbs}}$ satisfies CWE and SHVZK. Given these properties, we can prove the ring referral security properties of $\Pi_{\mathsf{RR.bbs}}$.

Lemma 7.1. *The protocol $\Pi_{\mathtt{chkcm}}$ satisfies completeness, and knowledge soundness and SHVZK. The protocol $\Pi_{\mathtt{chkuv}}$ satisfies completeness, CWE and SHVZK.*

Proof. See Appendix F. □

Theorem 7.2. *Assume the BBS signature scheme $\mathtt{Sig}_{\mathtt{bbs}}$ is complete and satisfies EU-CMA; the AFGHO and Pedersen commitment schemes are complete, perfectly hiding and computationally binding; the scalar and recursive Dory arguments satisfies completeness, CWE and SHVZK; the*

Schnorr proofs of knowledge in the commitment checking protocols satisfies completeness, knowledge soundness and SHVZK; and $\mathtt{Hash}$ *is a collision-resistant pseudo-random hash function. Then, the ring referral protocol* $\Pi_{\mathsf{RR.bbs}}$ *satisfies completeness, unforgeability, issuer anonymity and strong user anonymity.*

Proof. See Appendix F. □

Optimization by Batching. $\Pi_{\mathsf{RR.bbs}}$ calls the sub-protocol Π_{chkuv} once as a batched argument of three recursive Dory arguments; the protocol Π_{chkcm} is used three times for which we can batch the two using generator $\mathtt{e}(G_1, G_2)$ together; and finally, the two scalar Dory arguments $\Pi_{\mathsf{do.sp}}$ can be batched into a single argument. Overall, $\Pi_{\mathsf{RR.bbs}}$ calls the n-dimensional recursive Dory argument once, the Π_{chkcm} protocol 2 times, and the scalar Dory argument once.

8. Multi-Message Ring Referral Scheme

Building on Sec. 7, we present a full ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ (Fig. 7) for the multi-message BBS signature (Fig. 2a). $\Pi_{\mathsf{RR.bbs.m}}$ consists of two parts: (1) a proof-of-knowledge for a *multi-message* BBS signature that the user possess a valid tuple of public key, signed multi-message and corresponding signature; and (2) a proof-of-knowledge for a ring that a privately known public key is within a ring.

As the standard BBS signature (Fig. 2a) requires M $\mathbb{G}_1$ group exponentiation operations to verify signature on multi-message of size M, a naïve extension of the single-message ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ to multi-message case will result in a protocol whose proof size and verification scale linearly in M. Building upon the compressed multi-message BBS signature scheme in Fig. 2b, we are able to construct a more efficient ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ which has logarithmic proof size and logarithmic verification cost in terms of both the number of messages and the ring size.

8.1. Proof-of-Knowledge for Multi-Msg BBS

The prover knows a public key pk, a signed multi-message $\vec{\mathsf{m}}$ and the corresponding signature σ. We outline a proof-of-knowledge for the prover to prove that $(\mathsf{pk}, \vec{\mathsf{m}}, \sigma)$ is a valid multi-message BBS signature tuple as follows:

1) First, the prover commits $(\mathsf{pk}, \sigma_0, \sigma_1)$ to $\mathtt{cm}_{\mathsf{pk}} \triangleq \mathtt{e}(G_1, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}}$, $\mathtt{cm}_{\sigma_0} \triangleq \mathtt{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}}$, $\mathtt{cm}_{\sigma_1} \triangleq \mathtt{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}$. For the multi-message $\vec{\mathsf{m}}$, the prover uses Pedersen vector commitment to commit $\vec{\mathsf{m}}$ to $\mathtt{cm}_{\mathsf{m}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{m}}} \cdot \Gamma'^{\mathsf{r}_{\mathsf{m}}}$. The prover also provides additional commitments $\mathtt{Z}_1 \triangleq \mathtt{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}}$, $\mathtt{Z}_2 \triangleq \mathtt{e}(\sigma_0, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_2}}$ and $\mathtt{D}_0 \triangleq \mathtt{e}(\vec{\mathbf{H}}^{\vec{\mathsf{m}}}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}}$ which enables the checking of multi-message BBS signature σ.

   The key to achieve efficient verification is that the verifier can compute $\mathtt{D}_2 \triangleq \mathtt{e}(\mathtt{cm}_{\mathsf{m}}, G_2) = \mathtt{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ \vec{\mathsf{m}}'})$ using only 1 pairing operation, then use $\mathtt{D}_2$ in a Dory argument to check that the prover has computed $\mathtt{D}_0$ correctly. The cost of M group exponentiation operations to compute $\mathtt{D}_0$ is outsourced to the prover.

2) The verifier can use an $(M+1)$-dimensional recursive Dory argument to check that $\mathtt{e}(\vec{\mathbf{H}}^{\vec{\mathsf{m}}}, G_2) = \mathtt{e}(\vec{\mathbf{H}}, \vec{G}_2^{\circ \vec{\mathsf{m}}})$ is in $\mathtt{D}_0$ as follows:

$$\begin{cases} \mathtt{D}_0 \overset{?}{=} \mathtt{e}((\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}), \vec{G}_2^{\circ(\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}})}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{H}}}, \\ \mathtt{D}_1 \triangleq \mathtt{H} \overset{?}{=} \mathtt{e}((\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}), \vec{\mathbf{\Lambda}}'), \\ \mathtt{D}_2 \overset{?}{=} \mathtt{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ(\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}})}), \end{cases} \tag{9}$$

   where $\vec{\mathbf{\Gamma}}'$, $\vec{\mathbf{\Lambda}}'$ are known vector generators and $(\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}), \vec{G}_2^{\circ(\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}})})$ are the witness. Since $\mathtt{H}$ is publicly precomputed, and the verifier computes $\mathtt{D}_2$ herself, soundness of Dory implies that $\mathtt{e}(\vec{\mathbf{H}}, \vec{G}_2^{\circ \vec{\mathsf{m}}})$ is in $\mathtt{D}_0$.

   The verifier uses a Schnorr proof-of-knowledge to check the well-formedness of the commitment $\mathtt{cm}_{\sigma_1}$. Then, she can use a Dory scalar-product argument to check that $\mathtt{e}(\sigma_0, G_2^{\sigma_1})$ is in $\mathtt{Z}_1$ as follows:

$$\begin{cases} \mathtt{D}_0 \triangleq \mathtt{Z}_1 \overset{?}{=} \mathtt{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}}, \\ \mathtt{D}_1 \triangleq \mathtt{cm}_{\sigma_0} \overset{?}{=} \mathtt{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_0}}, \\ \mathtt{D}_2 \triangleq \mathtt{cm}_{\sigma_1} \overset{?}{=} \mathtt{e}(G_1, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}, \end{cases} \tag{10}$$

   where G_1, G_2 are known generators and $\sigma_0, G_2^{\sigma_1}$ are the witness. As it has been checked that $\mathtt{cm}_{\sigma_1}$ is well-formed, the knowledge soundness of Dory implies that $\mathtt{e}(\sigma_0, G_2^{\sigma_1})$ is in $\mathtt{Z}_1$ where σ_0, σ_1 are committed in $\mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1}$, respectively. Similarly, the verifier can also check the well-formedness of $\mathtt{Z}_2$ using Dory scalar-product check.

3) Finally, the verification of a multi-message BBS signature with respect to $\mathtt{cm}_{\mathsf{m}}$, $\mathtt{Z}_2$ and $\mathtt{Z}_1$ can be checked by:

$$\begin{aligned} \mathtt{Z}_1 \cdot \mathtt{Z}_2 &\overset{?}{=} \mathtt{e}(G_1, G_2) \cdot \mathtt{D}_0 \cdot \mathtt{Q}^{\mathsf{r}'}, \\ \Rightarrow\ \mathtt{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{e}(\sigma_0, \mathsf{pk}) &\overset{?}{=} \mathtt{e}(G_1, G_2) \cdot \mathtt{e}(\vec{\mathbf{H}}^{\vec{\mathsf{m}}}, G_2), \\ \Rightarrow\ \mathtt{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) &\overset{?}{=} \mathtt{e}(G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}},\ G_2), \end{aligned}$$

   where $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{Z}_1} + \mathsf{r}_{\mathtt{Z}_2} - \mathsf{r}_{\mathtt{D}}$ is provided by the prover.

The detailed protocol $\Pi_{\mathsf{bbs.pms.m}}$ is described in Fig. 6, which checks the proof-of-knowledge of a valid tuple of $(\mathsf{pk}, \vec{\mathsf{m}}, \sigma)$ in multi-message BBS signature scheme.

8.2. Proof-of-Knowledge for a Ring

For BBS signature (Fig. 2a), the multi-message and single-message scheme uses the same public key format $\mathsf{pk} = G_2^{\mathsf{sk}} \in \mathbb{G}_2$. Consequently, a similar proof-of-knowledge of a public key in a ring as in the single-message BBS signature case (Sec. 7.2) can also be used for multi-message BBS signature.

8.3. Interactive Ring Referral Protocol

Define the relation for ring referral of multi-message BBS signature by

$$\mathscr{R}_{\mathsf{RR.bbs.m}} \triangleq \Big\{ \vec{\mathsf{pk}} \in \mathbb{G}_2^n;\ \vec{\mathsf{m}} \in \mathbb{Z}_p^M, \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p,\ i^* \in [n], \mathsf{pk}_{i^*} \in \vec{\mathsf{pk}} \ \Big|\ \mathtt{e}(\sigma_0, G_2^{\sigma_1} \cdot \mathsf{pk}_{i^*}) = \mathtt{e}(G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}}, G_2) \Big\}.$$

$\Pi_{\mathsf{bbs.pms.m}}\big[\mathtt{cm}_{\sigma_0} \in \mathbb{G}_T,\ \mathtt{cm}_{\sigma_1} \in \mathbb{G}_T,\ \mathtt{cm}_{\mathsf{pk}} \in \mathbb{G}_T,\ \mathtt{cm}_{\mathsf{m}} \in \mathbb{G}_T;$
$\mathsf{pk} \in \mathbb{G}_2,\ \vec{\mathsf{m}} \in \mathbb{Z}_p^M,\ \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p\big]$

$\mathcal{P}$: PARSE $\vec{\mathsf{m}}' = (\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}}) \in \mathbb{Z}_p^{M+1},\quad \mathtt{H} = \mathsf{e}(\vec{\mathbf{H}}, \vec{\mathbf{\Lambda}}) \in \mathbb{G}_T$

$\mathtt{cm}_{\sigma_0} = \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}\sigma_0} \in \mathbb{G}_T$

$\mathtt{cm}_{\sigma_1} = \mathsf{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}\sigma_1} \in \mathbb{G}_T$

$\mathtt{cm}_{\mathsf{pk}} = \mathsf{e}(G_1, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}} \in \mathbb{G}_T,$

$\mathtt{cm}_{\mathsf{m}} = \vec{\mathbf{\Gamma}}^{\vec{\mathsf{m}}} \cdot \Gamma'^{\mathsf{r}_{\mathsf{m}}} \in \mathbb{G}_1$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{chkcm}}\big[\mathtt{cm}_{\sigma_1}, \mathsf{e}(G_1, G_2), \mathtt{Q};\ \sigma_1, \mathsf{r}_{\sigma_1}\big]$ *// Check σ_1 in $\mathtt{cm}_{\sigma_1}$*

$\mathcal{P}$: $\mathsf{r}_{\mathtt{Z}_1},\ \mathsf{r}_{\mathtt{Z}_2},\ \mathsf{r}_{\mathtt{D}} \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{Z}_1 \triangleq \mathsf{e}(\sigma_0, G_2^{\sigma_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}1}} \in \mathbb{G}_T,\quad \mathtt{Z}_2 \triangleq \mathsf{e}(\sigma_0, \mathsf{pk}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}2}} \in \mathbb{G}_T$

$\mathtt{D}_0 \triangleq \mathsf{e}(\vec{\mathbf{H}}^{\vec{\mathsf{m}}}, G_2) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{D}}} \in \mathbb{G}_T$

$\mathcal{V}$: $\mathtt{D}_2 \triangleq \mathsf{e}(\mathtt{cm}_{\mathsf{m}}, G_2) = \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ\vec{\mathsf{m}}'}) \in \mathbb{G}_T$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{do.ip}}\big[M{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{D}_0, \mathtt{H}, \mathtt{D}_2;\ (\vec{\mathbf{H}}, \mathbf{1}_{\mathbb{G}_1}), \vec{G}_2^{\circ\vec{\mathsf{m}}'}, \mathsf{r}_{\mathtt{D}}, 0, 0\big]$

// Check the same $\vec{G}_1^{\circ\vec{\mathsf{m}}'}$ is in $\mathtt{D}_0, \mathtt{D}_2$

RUN $\Pi_{\mathsf{do.sp}}\big[G_1, G_2, \mathtt{Z}_1, \mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1};\ \sigma_0, G_2^{\sigma_1}, \mathsf{r}_{\mathtt{Z}1}, \mathsf{r}_{\sigma_0}, \mathsf{r}_{\sigma_1}\big]$

// Check $\mathsf{e}(\sigma_0, G_2^{\sigma_1})$ in $\mathtt{Z}_1$, σ_0 in $\mathtt{cm}_{\sigma_0}$, $G_2^{\sigma_1}$ in $\mathtt{cm}_{\sigma_1}$

RUN $\Pi_{\mathsf{do.sp}}\big[G_1, G_2, \mathtt{Z}_2, \mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\mathsf{pk}};\ \sigma_0, \mathsf{pk}, \mathsf{r}_{\mathtt{Z}2}, \mathsf{r}_{\sigma_0}, \mathsf{r}_{\mathsf{pk}}\big]$

// Check $\mathsf{e}(\sigma_0, \mathsf{pk})$ in $\mathtt{Z}_2$, σ_0 in $\mathtt{cm}_{\sigma_0}$, pk in $\mathtt{cm}_{\mathsf{pk}}$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{Z}_1} + \mathsf{r}_{\mathtt{Z}_2} - \mathsf{r}_{\mathtt{D}}$

$\mathcal{V}$: CHECK $\mathtt{Z}_1 \cdot \mathtt{Z}_2 \stackrel{?}{=} \mathsf{e}(G_1, G_2) \cdot \mathtt{D}_0 \cdot \mathtt{Q}^{\mathsf{r}'}$

// Equivalently checking $\mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \stackrel{?}{=} \mathsf{e}(G_1 \cdot \vec{\mathbf{H}}^{\vec{\mathsf{m}}},\ G_2)$

Figure 6: Protocol $\Pi_{\mathsf{bbs.pms.m}}$ for the proof-of-knowledge of $(\mathsf{pk}, \vec{\mathsf{m}}, \sigma)$ in multi-message BBS signature scheme.

$\Pi_{\mathsf{RR.bbs.m}}\big[\vec{\mathsf{pk}} \in \mathbb{G}_2^n, M;\ \vec{\mathsf{m}} \in \mathbb{Z}_p^M, \sigma = (\sigma_0, \sigma_1) \in \mathbb{G}_1 \times \mathbb{Z}_p,$
$i^* \in [n], \mathsf{pk}_{i^*} \in \vec{\mathsf{pk}}\big]$

SETUP : $(\mathsf{h}_i \triangleq \mathtt{Hash}[\mathsf{pk}_i] \in \mathbb{Z}_p)_{i\in[n]}$

$\vec{\mathbf{K}} \triangleq (K_i \triangleq \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})_{i\in[n]} \in \mathbb{G}_2^n$

$\vec{\mathbf{\Gamma}}' \triangleq (\vec{\mathbf{\Gamma}}, \Gamma') \xleftarrow{\$} \mathbb{G}_1^{M+1},\quad \vec{\mathbf{\Lambda}}' \triangleq (\vec{\mathbf{\Lambda}}, \Lambda') \xleftarrow{\$} \mathbb{G}_2^{M+1}$

$\mathcal{P}$: $\mathsf{r}_{\sigma_0},\ \mathsf{r}_{\sigma_1},\ \mathsf{r}_{\mathsf{pk}},\ \mathsf{r}_{\mathsf{m}} \xleftarrow{\$} \mathbb{Z}_p,\quad \vec{\mathsf{m}}' \triangleq (\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}}) \in \mathbb{Z}_p^{M+1}$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}_{\sigma_0} \triangleq \mathsf{e}(\sigma_0, G_2) \cdot \mathtt{Q}^{\mathsf{r}\sigma_0} \in \mathbb{G}_T$

$\mathtt{cm}_{\sigma_1} \triangleq \mathsf{e}(G_1, G_2)^{\sigma_1} \cdot \mathtt{Q}^{\mathsf{r}\sigma_1} \in \mathbb{G}_T$

$\mathtt{cm}_{\mathsf{pk}} \triangleq \mathsf{e}(G_1, \mathsf{pk}_{i^*}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}} \in \mathbb{G}_T$

$\mathtt{cm}_{\mathsf{m}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{m}}} \cdot \Gamma'^{\mathsf{r}_{\mathsf{m}}} \in \mathbb{G}_1$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{bbs.pms.m}}[\mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1}, \mathtt{cm}_{\mathsf{pk}}, \mathtt{cm}_{\mathsf{m}};\ \mathsf{pk}_{i^*}, \vec{\mathsf{m}}, \sigma]$

$\mathcal{P}$: $\vec{\mathbf{b}} = (b_i)_{i\in[n]},$ where $b_i \triangleq \begin{cases} 0, \text{ if } i \neq i^* \\ 1, \text{ if } i = i^* \end{cases}$

$\mathsf{r}_1,\ \mathsf{r}_2 \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}_1 \triangleq \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1} \in \mathbb{G}_T$

$\mathtt{cm}_2 \triangleq \mathsf{e}(G_1, G_2)^{\mathsf{h}_{i^*}} \cdot \mathtt{Q}^{\mathsf{r}_2} \in \mathbb{G}_T$

$\mathsf{r}_1' \triangleq \mathsf{r}_1 - \mathsf{r}_{\mathsf{pk}} - \mathsf{r}_2 \in \mathbb{Z}_p$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{chkuv}}[\mathtt{cm}_1, \vec{\mathbf{\Omega}}, \vec{\mathbf{K}}, \mathtt{Q}; \vec{G}_1^{\circ\vec{\mathbf{b}}}, \mathsf{r}_1]$ *// Check $\vec{\mathbf{b}}$ a unit vector*

RUN $\Pi_{\mathsf{chkcm}}[\mathtt{cm}_2, \mathsf{e}(G_1, G_2), \mathtt{Q};\ \mathsf{h}_{i^*}, \mathsf{r}_2]$ *// Check h_{i^*} in $\mathtt{cm}_2$*

$\mathcal{V}$: CHECK $\mathtt{cm}_1 \stackrel{?}{=} \mathtt{cm}_{\mathsf{pk}} \cdot \mathtt{cm}_2 \cdot \mathtt{Q}^{\mathsf{r}_1'}$

// Check $\mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{b}}}, \vec{\mathbf{K}}) \stackrel{?}{=} \mathsf{e}(G_1, \mathsf{pk}_{i^*} \cdot G_2^{\mathsf{h}_{i^*}})$

Figure 7: Ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ for multi-message BBS signature.

We combine the proof-of-knowledge for a multi-message BBS signature and proof-of-knowledge for the ring in protocol $\Pi_{\mathsf{RR.bbs.m}}$ in Fig. 6 as a proof-of-knowledge for $\mathscr{R}_{\mathsf{RR.bbs.m}}$.

Theorem 8.1. *Assume the multi-message BBS signature scheme is complete and satisfies EU-CMA; the AFGHO and Pedersen commitment scheme are complete, perfectly hiding and computationally binding; the recursive Dory arguments are complete and satisfied CWE and SHVZK; the scalar Dory arguments and Schnorr proof of knowledge satisfies completeness, knowledge soundness and SHVZK; and* Hash *is a collision-resistant pseudorandom function. Then, the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ satisfies completeness, unforgeability, issuer anonymity and strong user anonymity.*

Proof. See Appendix F. □

Optimization by Batching. $\Pi_{\mathsf{RR.bbs.m}}$ calls the sub-protocol Π_{chkuv} once which is a batched version of three recursive Dory arguments of dimension n; the committed value checking protocol Π_{chkcm} is used twice using the same generators $\mathsf{e}(G_1, G_2)$, hence can be batched into a single call; the two scalar Dory arguments $\Pi_{\mathsf{do.sp}}$ on the generators (G_1, G_2) can also be batched into one; finally, we also call an $(M+1)$-dimensional recursive Dory argument. Overall, $\Pi_{\mathsf{RR.bbs.m}}$ call 1 recursive Dory argument of dimension n, 1 recursive Dory argument of dimension $M+1$, 1 Π_{chkcm} protocol, and 1 scalar Dory argument.

9. Evaluation

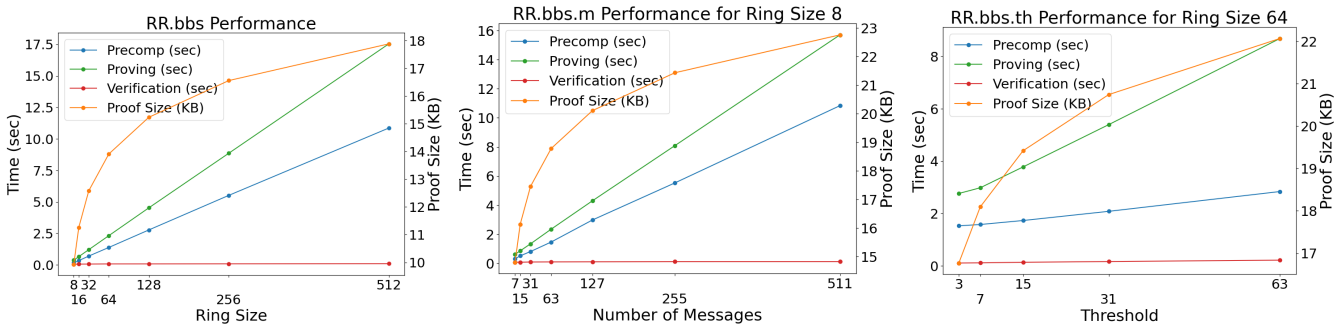
We implemented our schemes [31], using Charm Crypto [32] with curve MNT224. We evaluated the performance with 100 independent trials on a desktop with Intel Core i5 processor (2500 MHz, 16GB of RAM).

TABLE 2: Analytic performance estimation of standard multi-message BBS signature scheme (bbs) and compressed scheme (c.bbs). M is the number of messages. $|\mathbb{Z}_p|$ means field elements, $|\mathbb{G}_*|$ means group elements, $\mathbb{G}_*$ means group exponentiations, $\mathbb{P}$ means pairing operations.

		bbs	c.bbs
Proof Size	$\lvert\mathbb{G}_1\rvert$	1	3
	$\lvert\mathbb{G}_2\rvert$	0	1
	$\lvert\mathbb{G}_T\rvert$	0	$6\log(M+1)+2$
	$\lvert\mathbb{Z}_p\rvert$	$M+1$	2
Verification	$\mathbb{P}$	2	4
	$\mathbb{G}_1$	M	1
	$\mathbb{G}_2$	1	2
	$\mathbb{G}_T$	0	$9\log(M+1)+11$
Proving	$\mathbb{P}$	0	$3M+5$
	$\mathbb{G}_1$	$M+1$	$3(M+1)+2\log(M+1)$
	$\mathbb{G}_2$	0	$2\log(M+1)$
	$\mathbb{G}_T$	0	1
Pre-comp	$\mathbb{P}$	0	$3(M+1)$

9.1. Performance of Compressed BBS Signature

We provide the analytic performance estimation of the dominated terms of standard bbs and compressed c.bbs



(a) Performance evaluation of RR.bbs (b) Performance evaluation of RR.bbs.m (c) Performance evaluation of RR.bbs.th

Figure 8: Performance evaluation results of single-message ring referral (RR.bbs), multi-message ring referral (RR.bbs.m) and threshold ring referral (RR.bbs.th) schemes.

TABLE 3: Evaluation of standard multi-message BBS signature scheme (bbs) and compressed scheme (c.bbs).

M	Sig/Proof Size (KB)		Verification (sec)		Signing/Proving (sec)		Precomp (sec)	
	bbs	c.bbs	bbs	c.bbs	bbs	c.bbs	bbs	c.bbs
127	3.6	5.4	0.5	0.13	0.55	3.6	0	1.8
255	7.2	6.1	1.1	0.14	1.1	7.1	0	3.7
511	14.4	6.7	2.2	0.16	2.2	14.2	0	7.5

schemes in Table 2. We compare the empirical performance in Table 3. We observe that c.bbs has a smaller proof size and significantly faster verification than bbs, when M is large, though it incurs additional precomputation that is only computed once at setup and independent of any signatures.

9.2. Performance of Ring Referral Schemes

Since the issuer-hiding credential schemes [5], [6], [7] rely on private verifiability over a verifier-defined static ring, they have an unfair advantage of performance[5] in n over publicly verifiable schemes. Thus, we use the multi-issuer credential scheme ECA21 [8] as a baseline for comparison, which is a publicly verifiable scheme similar to our schemes. In baseline ECA21[6], an issuer signs the user's public key and a multi-message $(\mathsf{upk}, \vec{\mathsf{m}}) \in \mathbb{G}_1^{1+M}$ using multi-message Groth signature scheme [10] (see Fig. 13 in Appendix).

We provide the analytic performance estimation and evaluation of ring referral and baseline ECA21 in Tables 4-5. We observe that ring referral scheme considerably outperforms ECA21 with only small precomputation overhead.

We highlight the empirical performance as follows:

► ***Single-Message Ring Referral*** (Sec. 7). Fig. 8a shows the performance of the single-message ring referral scheme RR.bbs. We observe that as the ring size increases, the ring-dependent precomputation time and proving time increase linearly. In contrast, our verification time has a clear logarithmic advantage. Similarly, our proof size also exhibits the expected logarithmic relation with the ring size.

► ***Multi-Message Ring Referral*** (Sec. 8). Fig. 8b shows the performance of the multi-message ring referral scheme RR.bbs.m for ring size 8. We observe that as the number of messages increases, the ring-dependent precomputation time and proving time increase linearly. However, our verification time and proof size scale well, exhibiting a logarithmic relation with the number of messages. This makes our scheme more practical for verifying multiple messages in one go, which is beneficial for light-weight verification.

► ***Threshold Ring Referral*** (Appendix C). Fig. 8c shows the performance of the threshold ring referral scheme RR.bbs.th for ring size 64. Similar to RR.bbs.m, the precomputation and proving time increase linearly, while the verification time and proof size scale logarithmically.

TABLE 4: Analytic performance estimation of our schemes and baseline ECA21. n is the ring size, M is the number of messages, k is the threshold.

		RR.bbs	RR.bbs.th	RR.bbs.m	ECA21
Proof Size	$\lvert\mathbb{G}_1\rvert$	2	4	4	$4M$
	$\lvert\mathbb{G}_2\rvert$	2	3	3	$8\log n + 4M$
	$\lvert\mathbb{G}_T\rvert$	$6\log n$	$6\log n$	$6(\log n + \log M)$	0
Verification	$\mathbb{P}$	1	5	4	$26M$
	$\mathbb{G}_1$	2	2	3	0
	$\mathbb{G}_2$	2	2	3	$2n + 16M$
	$\mathbb{G}_T$	$9\log n$	$9\log n + 2k$	$9\log n + 9\log M$	0
Proving	$\mathbb{P}$	$3n$	$3n + 10k$	$3n + 3M$	0
	$\mathbb{G}_1$	$2\log n$	$2\log n + 6k$	$2\log n + 2M$	$8M$
	$\mathbb{G}_2$	$2\log n$	$2\log n + 3k$	$2\log n + 2\log M$	$2n + 9M$
	$\mathbb{G}_T$	15	9	16	0
Pre-comp	$\mathbb{P}$	$3n$	$3n + 6k$	$3n + 4M$	0
	$\mathbb{G}_2$	n	n	n	0

TABLE 5: Evaluation of our scheme and baseline ECA21.

(n, M)	Proof Size (KB)		Verification (sec)		Proving (sec)		Precomp (sec)
	ECA21	RR.bbs.m	ECA21	RR.bbs.m	ECA21	RR.bbs.m	RR.bbs.m
$(8, 127)$	87	7.4	17	0.18	4.9	3.1	2.6
$(16, 255)$	170	8.8	34	0.21	9.9	6.2	5.2
$(32, 511)$	335	10.2	69	0.25	20	12.5	10.5

5. In terms of group exponentiations, the verification time of [5], [6], [7] is constant in n, but linear in M. In terms of the number of pairings for (verification, proving) of BEK+21 [5] is $(13, 7)$, BFGP22 [6] is $(8, 0)$, ST23 [7] is $(3, 1)$, ECA21 [8] is $(26M, 0)$, RR.bbs is $(1, 3n)$, where M, n are the numbers of messages and issuers, respectively.

6. Note that [8] originally uses an accumulator with a trusted setup. We use ElGamal commitment with a transparent setup for benchmarking, although our schemes can also be adapted to incorporate accumulator.

10. Conclusion

In this paper, we presented an efficient cryptographic primitive called *ring referral scheme*, by which a user can publicly prove her knowledge of a valid signature for a private message that is signed by one of an ad hoc set of authorized issuers, assuring issuer and strong user anonymity with logarithmic verifiability and transparent setup. Our ring referral scheme supports many distinguishing features over the extant schemes in [5], [6], [7], [8]. In particular, it can be applied to certificate-hiding decentralized identity, privacy-enhancing federated authentication, anonymous endorsement and privacy-preserving referral marketing.

In future work, we will explore lattice-based ring referral schemes and incorporate blind signatures. A real-world evaluation of ring referrals on OpenID will also be pursued.

References

[1] T.-A. Ta, X. Hui, and S. C.-K. Chau, "Ring Referral: Efficient Publicly Verifiable Ad hoc Credential Scheme with Issuer and Strong User Anonymity for Decentralized Identity and More," in *IEEE SP*, 2025.

[2] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT*, 2001.

[3] OpenID-Foundation, "https://openID.net/."

[4] D. Hardt, "RFC 6749: OAuth 2.0," in *RFC*, 2012.

[5] J. Bobolz, F. Eidens, S. Krenn, S. Ramacher, and K. Samelin, "Issuer-hiding attribute-based credentials," in *Cryptology and Network Security*, 2021.

[6] D. Bosk, D. Frey, M. Gestin, and G. Piolle, "Hidden issuer anonymous credential," *Proc. Priv. Enhancing Technol.*, 2022.

[7] O. Sanders and J. Traore, "Efficient issuer-hiding authentication, application to anonymous credential," *Proc. Priv. Enhancing Technol.*, 2014.

[8] K. Elkhiyaoui, A. D. Caro, and E. Androulaki, "Multi-issuer anonymous credentials without a root authority," Cryptology ePrint Archive, Paper 2021/1669.

[9] J. Lee, "Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments," in *Theory of Cryptography (TCC)*, 2021.

[10] J. Groth, "Efficient fully structure-preserving signatures for large messages," in *ASIACRYPT*, 2015.

[11] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *CT-RSA*, 2016.

[12] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *CRYPTO*, 2004.

[13] D. Chaum and E. van Heyst, "Group signatures," in *EUROCRYPT*, 1991.

[14] O. R. Ioanna Karantaidou, F. Baldimtsi, J. K. Julian Loss, and N. Kamarinakis, "Blind multisignatures for anonymous tokens with decentralized issuance," in *ACM CCS*, 2024.

[15] M. Rosenberg, J. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure," in *IEEE SP*, 2023.

[16] J. Groth, "Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures," in *ASIACRYPT*, 2006.

[17] J. Groth and M. Kohlweiss, "One-out-of-many proofs: Or how to leak a secret and spend a coin," in *EUROCRYPT*, 2015.

[18] J. Bootle, K. Elkhiyaoui, J. Hesse, and Y. Manevich, "DualDory: Logarithmic-Verifier Linkable Ring Signatures Through Preprocessing," in *ESORICS*, 2022.

[19] X. Hui and S. C.-K. Chau, "LLRing: Logarithmic Linkable Ring Signatures with Transparent Setup," in *ESORICS*, 2014.

[20] R. W. F. Lai, V. Ronge, T. Ruffing, D. Schröder, S. A. K. Thyagarajan, and J. Wang, "Omniring: Scaling private payments without trusted setup," in *ACM CCS*, 2019.

[21] B. Bunz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *IEEE SP*, 2018.

[22] A. Connolly, P. Lafourcade, and O. Perez Kempner, "Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes," in *PKC*, 2022.

[23] O. Bicer and A. Kupcu, "Versatile ABS: Usage limited, revocable, threshold traceable, authority hiding, decentralized attribute based signatures," Cryptology ePrint Archive, Paper 2019/203.

[24] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," in *NDSS*, 2019.

[25] J. Doerner, Y. Kondi, E. Lee, A. Shelat, and L. Tyner, " Threshold BBS+ Signatures for Distributed Anonymous Credential Issuance ," in *IEEE SP*, 2023.

[26] B. Campbell, C. Mortimore, and M. B. Jones, "Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants," RFC 7522, May 2015.

[27] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, "Structure-preserving signatures and commitments to group elements," in *CRYPTO*, 2010.

[28] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO*, 1986.

[29] M. H. Au, W. Susilo, and Y. Mu, "Constant-Size Dynamic k-TAA," in *Security and Cryptography for Networks*, 2006.

[30] S. Tessaro and C. Zhu, "Revisiting BBS Signatures," in *EUROCRYPT*, 2023.

[31] "Ring Referral Code," https://github.com/sidckchau/RingReferral.

[32] J. A. Akinyele, M. Green, C. U. Lehmann, A. D. Rubin, M. Rushanan, M. D. Smith, and A. D. Yocum, "Charm: A Framework for Rapidly Prototyping Cryptosystems," https://github.com/JHUISI/charm.

[33] J. Katz and Y. Lindell, *Introduction to Modern Cryptography, Second Edition*, 2nd ed. Chapman & Hall/CRC, 2014.

Appendix A. Cryptographic Assumptions

Definition A.1 (Discrete Logarithm (DLog))**.** *The DLog assumption holds for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\vec{\mathbf{x}} \leftarrow \mathcal{A}[\vec{\mathbf{G}}], \; \vec{\mathbf{G}}^{\vec{\mathbf{x}}} = \eta \;\middle|\; \mathbb{G} \leftarrow \mathtt{Setup}[1^\lambda], \; \vec{\mathbf{G}} \xleftarrow{\$} \mathbb{G}\right] \leq \mathrm{negl}(\lambda).$$

As a result of the DLog assumption, non-trivial discrete logarithm relations among random generators $\vec{\mathbf{G}}$ cannot be discovered by any PPT adversary.

Definition A.2 (Computational Diffie–Hellman (CDH))**.** *Given a random generator $G \xleftarrow{\$} \mathbb{G}$ and a tuple (G^a, G^b), where $(a, b) \xleftarrow{\$} \mathbb{Z}_p^{*2}$ are selected at random, the CDH assumption holds, if G^{ab} is computationally hard for any PPT adversary.*

Definition A.3 (Decisional Diffie–Hellman (DDH))**.** *Given a random generator $G \xleftarrow{\$} \mathbb{G}$ and a tuple (G^a, G^b, G^c), where $(a, b, c) \xleftarrow{\$} \mathbb{Z}_p^{*3}$ are selected at random, the DDH assumption holds, if G^c is computationally indistinguishable from G^{ab} for any PPT adversary.*

Definition A.4 (Symmetric External Diffie-Hellman (SXDH))**.** *Given a bilinear pairing $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and random generators $G \xleftarrow{\$} \mathbb{G}_1, H \xleftarrow{\$} \mathbb{G}_2$, the SXDH assumption holds, if the DDH assumption holds for $\mathbb{G}_1$ and $\mathbb{G}_2$, and the following distributions are computationally indistinguishable for any PPT adversary:*

1) *Tuple $(G, G^a, H, H^b, \mathsf{e}(G, H)^{ab})$, where $(a, b) \xleftarrow{\$} \mathbb{Z}_p^{*2}$.*
2) *Tuple (G, G^a, H, H^b, T), where $(a, b) \xleftarrow{\$} \mathbb{Z}_p^{*2}, T \xleftarrow{\$} \mathbb{G}_T$.*

Definition A.5 (Double Pairing (DPair))**.** *Given a bilinear pairing $\mathsf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and a random element vector $\vec{\mathbf{G}} \xleftarrow{\$} \mathbb{G}_1^n$, the DPair assumption holds, if it is computationally hard to produce $\vec{\mathbf{H}} \in \mathbb{G}_2^n$ for any PPT adversary, such that $\mathsf{e}(\vec{\mathbf{G}}, \vec{\mathbf{H}}) = 1$.*

Appendix B. Additional Definitions

Denote a polynomial-time decidable tertiary relation by $\mathscr{R} \subset \{0,1\}^{*3}$. A language dependent on pp is defined as $\mathscr{L}_{\mathscr{R}}^{\mathsf{pp}} \triangleq \{x \mid \exists \omega : (\mathsf{pp}, x, \omega) \in \mathscr{R}\}$, where ω is a witness for a statement x in the relation $(\mathsf{pp}, x, \omega) \in \mathscr{R}$.

Definition B.1 (Completeness)**.** *Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies* completeness*, if for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\begin{matrix}\mathtt{Accept}[\mathsf{tr}] \\ = 1\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (\mathsf{pp}, x, \omega) \in \mathscr{R}, \\ \mathsf{tr} \leftarrow \langle \mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x)\rangle\end{matrix}\right] \geq 1 - \mathrm{negl}(\lambda).$$

We call it perfect completeness*, if* $\mathrm{negl}(\lambda) = 0$.

Definition B.2 (Computational Witness-Extended Emulation (CWE))**.** *Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies CWE, if there exists an expected polynomial-time emulator $\mathcal{E}$, such that for any interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$:*

$$\left|\Pr\left[\mathcal{A}_1[\mathsf{tr}] = 1 \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}], \\ \mathsf{tr} \leftarrow \langle \tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle\end{matrix}\right]\right.$$
$$\left. -\Pr\left[\begin{matrix}\mathcal{A}_1[\mathsf{tr}'] = 1 \\ \wedge (\mathtt{Accept}[\mathsf{tr}'] = 1 \Rightarrow \\ (\mathsf{pp}, x, w') \in \mathscr{R})\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}], \\ (\mathsf{tr}', w') \leftarrow \mathcal{E}^{\mathcal{O}}[\mathsf{pp}, x]\end{matrix}\right]\right| \leq \mathrm{negl}(\lambda),$$

where $\tilde{\mathcal{P}}$ is a deterministic polynomial-time algorithm, $\mathcal{A}_1[\mathsf{tr}]$ recognizes the transcripts that are produced by $\tilde{\mathcal{P}}$, and $\mathcal{O}$ is a rewindable oracle that can rewind the transcript $\langle \tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle$ and control the randomness in $\mathcal{V}$.

Definition B.3 (Public Coin)**.** *Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called* public-coin*, if the verifier chooses her messages uniformly at random, independent from the messages sent by the prover. Let e be the public-coin challenge. The transcript of a public-coin argument system is defined as $\mathsf{tr} = \langle \mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x; e)\rangle$.*

Definition B.4 (Special Honest-Verifier Zero-Knowledge (SHVZK))**.** *A public-coin argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies SHVZK, if there exists an efficient simulator $\mathcal{S}$, such that for any PPT adversary $\mathcal{A}$:*

$$\left|\Pr\left[\begin{matrix}\mathtt{Accept}[\mathsf{tr}] \\ = 1 \\ \wedge\ (\mathsf{pp}, x, \omega) \in \mathscr{R}\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \omega, e) \leftarrow \mathcal{A}[\mathsf{pp}], \\ \mathsf{tr} \leftarrow \langle \mathcal{P}(\mathsf{pp}, x, \omega), \mathcal{V}(\mathsf{pp}, x; e)\rangle\end{matrix}\right]\right.$$
$$\left. -\Pr\left[\begin{matrix}\mathtt{Accept}[\mathsf{tr}] \\ = 1 \\ \wedge\ (\mathsf{pp}, x, \omega) \in \mathscr{R}\end{matrix} \;\middle|\; \begin{matrix}\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \omega, e) \leftarrow \mathcal{A}[\mathsf{pp}], \\ \mathsf{tr} \leftarrow \mathcal{S}[\mathsf{pp}, x; e]\end{matrix}\right]\right| \leq \mathrm{negl}(\lambda).$$

Definition B.5 (Fiat-Shamir Transformation)**.** *A multi-move interactive public-coin argument of knowledge can be converted to a non-interactive argument of knowledge by replacing the public-coin challenges by the output of a cryptographic hash function, which produces seemingly random output and is regarded as a replacement for a verifier.*

In this paper, we focus on *multi-move interactive public-coin protocols* for arguments of knowledge. The Fiat-Shamir transformation can be applied to convert our interactive protocols to non-interactive arguments using the random oracle model in the security proofs [28]. This is especially useful for reducing a logarithmic number of moves to a single move in a publicly verifiable scheme.

B.1. Detailed Dory Protocol

Denote $[\vec{\mathbf{G}}]_{\mathtt{L}} \triangleq (G_1, ..., G_{\frac{n}{2}})$ and $[\vec{\mathbf{G}}]_{\mathtt{R}} \triangleq (G_{\frac{n}{2}+1}, ..., G_n)$ as the left-half and right-half sub-vectors of $\vec{\mathbf{G}}$, respectively. Dory [9] is a compressive protocol based on a recursive folding technique. Dory improves over Bulletproofs [21] with logarithmic verification efficiency and proof size. It checks

$$\mathtt{D}_0 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_0}, \mathtt{D}_1 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \mathtt{D}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_2},$$

given commitments $\mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2 \in \mathbb{G}_T$ and known random generators $\vec{\mathbf{\Gamma}} \xleftarrow{\$} \mathbb{G}_1^n, \vec{\mathbf{\Lambda}} \xleftarrow{\$} \mathbb{G}_2^n$, with some private witness $(\vec{\mathbf{\Omega}} \in \mathbb{G}_1^n, \vec{\mathbf{\Theta}} \in \mathbb{G}_2^n, \mathsf{r}_0, \mathsf{r}_1, \mathsf{r}_2 \in \mathbb{Z}_p^*)$. We describe the Dory protocol $\Pi_{\mathsf{do.ip}}$ using a recursive argument in Fig. 9a. Note that the choices of $\vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}'$ do not matter. One possible setting is $\vec{\mathbf{\Gamma}}' \triangleq [\vec{\mathbf{\Gamma}}]_{\mathtt{L}}, \vec{\mathbf{\Lambda}}' \triangleq [\vec{\mathbf{\Lambda}}]_{\mathtt{L}}$, and hence, $\Delta_{1\mathtt{L}} = \Delta_{2\mathtt{L}}$.

An n-dimensional Dory inner-product argument attains the following:

- *Proof size*: $6 \log n$ $\mathbb{G}_T$ elements, 1 $\mathbb{G}_1$ element and 1 $\mathbb{G}_2$ element;
- *Verification cost*: 1 pairing, $9 \log n + 9$ $\mathbb{G}_T$ exponentiations, 1 $\mathbb{G}_1$ exponentiation and 1 $\mathbb{G}_2$ exponentiation;
- *Proving cost*: $3n$ pairings, $2 \log n$ $\mathbb{G}_1$ exponentiations and $2 \log n$ $\mathbb{G}_2$ exponentiations;
- *Precomputation* includes $3n$ pairings.

Batching. It is possible to batch multiple Dory arguments into a single argument [9]. Given

$$\mathtt{D}_0 = \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_0}, \mathtt{D}_1 = \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \mathtt{D}_2 = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{\Theta}}) \cdot \mathtt{Q}^{\mathsf{r}_2},$$

and

$$\mathtt{D}_0' = \mathsf{e}(\vec{\mathbf{\Omega}}', \vec{\mathbf{\Theta}}') \cdot \mathtt{Q}^{\mathsf{r}_0'}, \mathtt{D}_1' = \mathsf{e}(\vec{\mathbf{\Omega}}', \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}_1'}, \mathtt{D}_2' = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{\Theta}}') \cdot \mathtt{Q}^{\mathsf{r}_2'},$$

with shared generators $(\vec{\mathbf{\Gamma}}, \vec{\mathbf{\Lambda}})$, we define

$\Pi_{\mathsf{do.ip}}\Big[n \in \mathbb{Z}^+, \vec{\Gamma} \in \mathbb{G}_1^n, \vec{\Lambda} \in \mathbb{G}_2^n, \mathtt{D}_0 \in \mathbb{G}_T, \mathtt{D}_1 \in \mathbb{G}_T, \mathtt{D}_2 \in \mathbb{G}_T;$
$\vec{\Omega} \in \mathbb{G}_1^n, \vec{\Theta} \in \mathbb{G}_2^n, \mathtt{r}_0 \in \mathbb{Z}_p^*, \mathtt{r}_1 \in \mathbb{Z}_p^*, \mathtt{r}_2 \in \mathbb{Z}_p^*\Big]$

SETUP : $\mathtt{X} \triangleq \mathsf{e}(\vec{\Gamma}, \vec{\Lambda}) \in \mathbb{G}_T$

$\Delta_{1\mathtt{L}} \triangleq \mathsf{e}([\vec{\Gamma}]_{\mathtt{L}}, \vec{\Lambda}') \in \mathbb{G}_T, \quad \Delta_{1\mathtt{R}} \triangleq \mathsf{e}([\vec{\Gamma}]_{\mathtt{R}}, \vec{\Lambda}') \in \mathbb{G}_T$

$\Delta_{2\mathtt{L}} \triangleq \mathsf{e}(\vec{\Gamma}', [\vec{\Lambda}]_{\mathtt{L}}) \in \mathbb{G}_T, \quad \Delta_{2\mathtt{R}} \triangleq \mathsf{e}(\vec{\Gamma}', [\vec{\Lambda}]_{\mathtt{R}}) \in \mathbb{G}_T$

IF $n = 1$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{do.sp}}\Big[\Gamma, \Lambda, \mathtt{D}_0, \mathtt{D}_1, \mathtt{D}_2;\ \Omega, \Theta, \mathtt{r}_0, \mathtt{r}_1, \mathtt{r}_2\Big]$

ELSE $n > 1$

$\mathcal{P}$: $\mathtt{r}_{1\mathtt{L}}, \mathtt{r}_{1\mathtt{R}}, \mathtt{r}_{2\mathtt{L}}, \mathtt{r}_{2\mathtt{R}}, \mathtt{r}_{\mathtt{W}1}, \mathtt{r}_{\mathtt{W}2} \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{D}_{1\mathtt{L}} \triangleq \mathsf{e}([\vec{\Omega}]_{\mathtt{L}}, \vec{\Lambda}') \cdot \mathtt{Q}^{\mathtt{r}_{1\mathtt{L}}} \in \mathbb{G}_T, \mathtt{D}_{1\mathtt{R}} \triangleq \mathsf{e}([\vec{\Omega}]_{\mathtt{R}}, \vec{\Lambda}') \cdot \mathtt{Q}^{\mathtt{r}_{1\mathtt{R}}} \in \mathbb{G}_T$

$\mathtt{D}_{2\mathtt{L}} \triangleq \mathsf{e}(\vec{\Gamma}', [\vec{\Theta}]_{\mathtt{L}}) \cdot \mathtt{Q}^{\mathtt{r}_{2\mathtt{L}}} \in \mathbb{G}_T, \mathtt{D}_{2\mathtt{R}} \triangleq \mathsf{e}(\vec{\Gamma}', [\vec{\Theta}]_{\mathtt{R}}) \cdot \mathtt{Q}^{\mathtt{r}_{2\mathtt{R}}} \in \mathbb{G}_T$

$\mathcal{P} \Leftarrow \mathcal{V}$: $\beta \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P}$: $\vec{\Omega}^\circ \triangleq \vec{\Omega} \circ \vec{\Gamma}^\beta \in \mathbb{G}_1^n, \quad \vec{\Theta}^\circ \triangleq \vec{\Theta} \circ \vec{\Lambda}^{\beta^{-1}} \in \mathbb{G}_2^n$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{W}_1 \triangleq \mathsf{e}([\vec{\Omega}]_{\mathtt{L}}^\circ, [\vec{\Theta}]_{\mathtt{R}}^\circ) \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{W}1}}, \ \mathtt{W}_2 \triangleq \mathsf{e}([\vec{\Omega}]_{\mathtt{R}}^\circ, [\vec{\Theta}]_{\mathtt{L}}^\circ) \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{W}2}} \in \mathbb{G}_T$

$\tilde{\mathtt{r}}_0 \triangleq \mathtt{r}_0 + \beta \cdot \mathtt{r}_1 + \beta^{-1} \cdot \mathtt{r}_2$

$\mathcal{P} \Leftarrow \mathcal{V}$: $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P}$: $\vec{\Omega}' \triangleq [\vec{\Omega}^\circ]_{\mathtt{L}}^\alpha \circ [\vec{\Omega}^\circ]_{\mathtt{R}} \in \mathbb{G}_1^{\frac{n}{2}}, \ \vec{\Theta}' \triangleq [\vec{\Theta}^\circ]_{\mathtt{L}}^{\alpha^{-1}} \circ [\vec{\Theta}^\circ]_{\mathtt{R}} \in \mathbb{G}_2^{\frac{n}{2}}$

$\mathtt{r}_0' \triangleq \tilde{\mathtt{r}}_0 + \alpha \cdot \mathtt{r}_{\mathtt{W}1} + \alpha^{-1} \cdot \mathtt{r}_{\mathtt{W}2}$

$\mathtt{r}_1' \triangleq \alpha \cdot \mathtt{r}_{1\mathtt{L}} + \mathtt{r}_{1\mathtt{R}}, \quad \mathtt{r}_2' \triangleq \alpha^{-1} \cdot \mathtt{r}_{2\mathtt{L}} + \mathtt{r}_{2\mathtt{R}}$

$\mathcal{V}$: $\mathtt{D}_0' \triangleq \mathtt{D}_0 \cdot \mathtt{X} \cdot \mathtt{D}_1^{\beta^{-1}} \cdot \mathtt{D}_2^{\beta} \cdot \mathtt{W}_1^{\alpha} \cdot \mathtt{W}_2^{\alpha^{-1}} \in \mathbb{G}_T$

$\mathtt{D}_1' \triangleq \mathtt{D}_{1\mathtt{L}}^{\alpha} \cdot \mathtt{D}_{1\mathtt{R}} \cdot \Delta_{1\mathtt{L}}^{\alpha\beta} \cdot \Delta_{1\mathtt{R}}^{\beta} \in \mathbb{G}_T$

$\mathtt{D}_2' \triangleq \mathtt{D}_{2\mathtt{L}}^{\alpha^{-1}} \cdot \mathtt{D}_{2\mathtt{R}} \cdot \Delta_{2\mathtt{L}}^{\alpha^{-1}\beta^{-1}} \cdot \Delta_{2\mathtt{R}}^{\beta^{-1}} \in \mathbb{G}_T$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{do.ip}}[\frac{n}{2}, \vec{\Gamma}', \vec{\Lambda}', \mathtt{D}_0', \mathtt{D}_1', \mathtt{D}_2';\ \vec{\Omega}', \vec{\Theta}', \mathtt{r}_0', \mathtt{r}_1', \mathtt{r}_2']$

(a) Dory protocol $\Pi_{\mathsf{do.ip}}$ for checking inner-product relations $\langle \mathtt{D}_0 \stackrel{?}{=} \mathsf{e}(\vec{\Omega}, \vec{\Theta}) \cdot \mathtt{Q}^{\mathtt{r}_0}, \mathtt{D}_1 \stackrel{?}{=} \mathsf{e}(\vec{\Omega}, \vec{\Lambda}) \cdot \mathtt{Q}^{\mathtt{r}_1}, \mathtt{D}_2 \stackrel{?}{=} \mathsf{e}(\vec{\Gamma}, \vec{\Theta}) \cdot \mathtt{Q}^{\mathtt{r}_2} \rangle$.

$\Pi_{\mathsf{do.sp}}\Big[\Gamma \in \mathbb{G}_1, \Lambda \in \mathbb{G}_2, \mathtt{D}_0 \in \mathbb{G}_T, \mathtt{D}_1 \in \mathbb{G}_T, \mathtt{D}_2 \in \mathbb{G}_T;$
$\Omega \in \mathbb{G}_1, \Theta \in \mathbb{G}_2, \mathtt{r}_0 \in \mathbb{Z}_p^*, \mathtt{r}_1 \in \mathbb{Z}_p^*, \mathtt{r}_2 \in \mathbb{Z}_p^*\Big]$

SETUP : $\mathtt{X} \triangleq \mathsf{e}(\Gamma, \Lambda) \in \mathbb{G}_T$

$\mathcal{P}$: $\Omega' \xleftarrow{\$} \mathbb{G}_1, \quad \Theta' \xleftarrow{\$} \in \mathbb{G}_2, \quad \mathtt{r}_{\mathtt{P}1}, \mathtt{r}_{\mathtt{P}2}, \mathtt{r}_{\mathtt{S}}, \mathtt{r}_{\mathtt{R}} \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{P}_1 \triangleq \mathsf{e}(\Omega', \Gamma) \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{P}1}} \in \mathbb{G}_T, \quad \mathtt{P}_2 \triangleq \mathsf{e}(\Lambda, \Theta') \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{P}2}} \in \mathbb{G}_T$

$\mathtt{S} \triangleq \mathsf{e}(\Omega', \Theta) \cdot \mathsf{e}(\Omega, \Theta') \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{S}}} \in \mathbb{G}_T, \quad \mathtt{R} \triangleq \mathsf{e}(\Omega', \Theta') \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{R}}} \in \mathbb{G}_T$

$\mathcal{P} \Leftarrow \mathcal{V}$: $\epsilon \xleftarrow{\$} \mathbb{Z}_p^*$

$\mathcal{P} \Rightarrow \mathcal{V}$: $E_1 \triangleq \Omega' \cdot \Omega^\epsilon \in \mathbb{G}_1, \quad E_2 \triangleq \Theta' \cdot \Theta^\epsilon \in \mathbb{G}_2$

$\mathtt{r}_1 \triangleq \mathtt{r}_{\mathtt{P}1} + \epsilon \cdot \mathtt{r}_1 \in \mathbb{Z}_p^*, \quad \mathtt{r}_2 \triangleq \mathtt{r}_{\mathtt{P}2} + \epsilon \cdot \mathtt{r}_2 \in \mathbb{Z}_p^*$

$\mathtt{r}_0 \triangleq \mathtt{r}_{\mathtt{R}} + \epsilon \cdot \mathtt{r}_Q + \epsilon^2 \cdot \mathtt{r}_0 \in \mathbb{Z}_p^*$

$\mathcal{V}$: $\theta \xleftarrow{\$} \mathbb{Z}_p^*, \quad \mathtt{r} \triangleq \mathtt{r}_0 + \theta \cdot \mathtt{r}_2 + \theta^{-1} \cdot \mathtt{r}_1$

CHECK $\mathsf{e}(E_1 \cdot \Gamma^\theta, E_2 \cdot \Lambda^{\theta^{-1}}) \stackrel{?}{=}$

$\mathtt{X} \cdot \mathtt{R} \cdot \mathtt{S}^\epsilon \cdot \mathtt{D}_0^{\epsilon^2} \cdot \mathtt{P}_2^\theta \cdot \mathtt{D}_2^{\theta \cdot \epsilon} \cdot \mathtt{P}_1^{\theta^{-1}} \cdot \mathtt{D}_1^{\theta^{-1} \cdot \epsilon} \cdot \mathtt{Q}^{\mathtt{r}}$

(b) Dory protocol $\Pi_{\mathsf{do.sp}}$ for checking scalar-product relations $\langle \mathtt{D}_0 \stackrel{?}{=} \mathsf{e}(\Omega, \Theta) \cdot \mathtt{Q}^{\mathtt{r}_0}, \mathtt{D}_1 \stackrel{?}{=} \mathsf{e}(\Omega, \Lambda) \cdot \mathtt{Q}^{\mathtt{r}_1}, \mathtt{D}_2 \stackrel{?}{=} \mathsf{e}(\Gamma, \Theta) \cdot \mathtt{Q}^{\mathtt{r}_2} \rangle$.

Figure 9: Interactive Dory protocols

$\Pi_{\mathsf{chkuv}}^{\mathsf{th}}\Big[n, k, \mathtt{cm} \in \mathbb{G}_T, \vec{\Omega} \in \mathbb{G}_1^n, \vec{\mathbf{K}} \in \mathbb{G}_2^n, \mathtt{Q} \in \mathbb{G}_T;\ \vec{\mathbf{d}} \in \mathbb{Z}_p^n, \mathtt{r} \in \mathbb{Z}_p\Big]$

$\mathcal{P}$: $\mathtt{r}'' \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}'' \triangleq \mathsf{e}(\vec{\Omega}^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathtt{r}''}$

$\mathcal{V}$ & $\mathcal{P}$: RUN $\Pi_{\mathsf{do.ip}}\big[n, \vec{\Omega}, \vec{\mathbf{K}}, \mathsf{e}(G_1, G_2)^k, \mathtt{cm}, \mathsf{e}(\vec{\Omega}, \vec{G}_2);\ \vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{G}_2, 0, \mathtt{r}, 0\big]$

RUN $\Pi_{\mathsf{do.ip}}\big[n, \vec{\Omega}, \vec{\mathbf{K}}, \mathtt{cm}'', \mathtt{cm}'', \mathtt{cm}'';\ \vec{\Omega}^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}^{\circ\vec{\mathbf{d}}}, \mathtt{r}'', \mathtt{r}'', \mathtt{r}''\big]$

RUN $\Pi_{\mathsf{do.ip}}\big[n, \vec{\Omega}, \vec{\mathbf{K}}, \mathtt{cm}, \mathsf{e}(\vec{G}_1, \vec{\mathbf{K}}), \mathtt{cm}'';\ \vec{G}_1, \vec{\mathbf{K}}^{\circ\vec{\mathbf{d}}}, \mathtt{r}, 0, \mathtt{r}''\big]$

// Can be batched into a single Dory with generators $(\vec{\Omega}, \vec{\mathbf{K}})$

Figure 10: Protocol $\Pi_{\mathsf{chkuv}}^{\mathsf{th}}$ for checking the well-formedness of a binary vector of weight k.

$$\mathtt{X} \triangleq \mathsf{e}(\vec{\Omega}, \vec{\Theta}') \cdot \mathsf{e}(\vec{\Omega}', \vec{\Theta}) \cdot \mathtt{Q}^{\mathtt{r}_{\mathtt{X}}},$$

and

$$\mathtt{D}_0'' \triangleq \mathtt{D}_0^{\gamma^2} \cdot \mathtt{X}^{\gamma} \cdot \mathtt{D}_0', \mathtt{D}_1'' \triangleq \mathtt{D}_1^{\gamma} \cdot \mathtt{D}_1', \mathtt{D}_2'' \triangleq \mathtt{D}_2^{\gamma} \cdot \mathtt{D}_2',$$

where $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$. Then the new witnesses to $(\mathtt{D}_0'', \mathtt{D}_1'', \mathtt{D}_2'')$ are

$$\vec{\Omega}'' \triangleq \vec{\Omega} \circ \vec{\Omega}'^{\gamma}, \vec{\Theta}'' \triangleq \vec{\Theta} \circ \vec{\Theta}'^{\gamma},$$

and

$$\mathtt{r}_0'' \triangleq \gamma^2 \mathtt{r}_0 + \gamma \mathtt{r}_{\mathtt{X}} + \mathtt{r}_0', \mathtt{r}_1'' \triangleq \gamma \mathtt{r}_1 + \mathtt{r}_1', \mathtt{r}_2'' \triangleq \gamma \mathtt{r}_2 + \mathtt{r}_2'.$$

Theorem B.1 ([9]). $\Pi_{\mathsf{do.ip}}$ *satisfies perfect completeness, SHVZK and CWE (or it breaks the SXDH assumption).*

As we also use the 1-dimensional Dory scalar-product arguments several times in our ring referral schemes, we recall it in the protocol $\Pi_{\mathsf{do.sp}}$ (Fig. 9b) for convenience. Dory scalar-product protocol $\Pi_{\mathsf{do.sp}}$ attains the following:

- *Proof size*: 4 $\mathbb{G}_T$ elements, 1 $\mathbb{G}_1$ element and 1 $\mathbb{G}_2$ element;
- *Verification cost*: 1 pairing, 7 $\mathbb{G}_T$ exponentiations, 1 $\mathbb{G}_1$ exponentiations and 1 $\mathbb{G}_2$ exponentiations;
- *Proving cost*: 4 pairings, 1 $\mathbb{G}_1$ exponentiations, 1 $\mathbb{G}_2$ exponentiations, and $4\mathbb{G}_T$ exponentiations;
- *Precomputation* of 1 pairing.

Appendix C. Threshold Ring Referral Scheme

We give a concise description of a threshold variant of ring referral for single-message BBS signature. In a k-out-of-n threshold ring referral scheme, the prover needs to show that she has k valid signatures from k different issuers in a ring of size n. The construction of our threshold ring referral scheme consists of two parts: (1) a proof-of-knowledge for k BBS signatures that the user has a collection of k valid tuples of public key, signed message and corresponding BBS signature; and (2) a proof-of-knowledge for a ring that a collection of k different public keys, which are privately known only to the prover, is a subset

$\Pi_{\mathsf{bbs.pms.th}}\Big[\mathtt{cm}_{\sigma_0} \in \mathbb{G}_T, \mathtt{cm}_{\sigma_1} \in \mathbb{G}_1, \mathtt{cm}_{\mathsf{pk}} \in \mathbb{G}_T, \mathtt{cm}_{\mathsf{m}} \in \mathbb{G}_2;$
$\vec{\mathsf{pk}}^{\star} \in \mathbb{G}_2^k,\ \vec{\mathsf{m}} \in \mathbb{Z}_p^k, \vec{\sigma} \in \mathbb{G}_1^k \times \mathbb{Z}_p^k\Big]$

$\mathcal{P}$: PARSE $\mathtt{cm}_{\sigma_0} = \mathsf{e}(\vec{\sigma}_0, \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}\sigma_0} \in \mathbb{G}_T$
$\mathtt{cm}_{\sigma_1} = \vec{\mathbf{\Gamma}}^{\vec{\sigma}_1} \cdot \Gamma^{\mathsf{r}\sigma_1} \in \mathbb{G}_1$
$\mathtt{cm}_{\mathsf{pk}} = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathsf{pk}}^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}} \in \mathbb{G}_T$
$\mathtt{cm}_{\mathsf{m}} = \vec{\mathbf{\Lambda}}^{\vec{\mathsf{m}}} \cdot \Lambda^{\mathsf{r}_{\mathsf{m}}} \in \mathbb{G}_2$
$\vec{\sigma} = (\vec{\sigma}_0, \vec{\sigma}_1) \in \mathbb{G}_1^k \times \mathbb{Z}_p^k$
$\mathcal{P}$: $\vec{\sigma}'_0 \triangleq (\vec{\sigma}_0, \mathbf{1}_{\mathbb{G}_1}) \in \mathbb{G}_1^{k+1}, \quad \vec{\sigma}'_1 \triangleq (\vec{\sigma}_1, \mathsf{r}_{\sigma_1}) \in \mathbb{Z}_p^{k+1}$
$\vec{\mathsf{m}}' \triangleq (\vec{\mathsf{m}}, \mathsf{r}_{\mathsf{m}}) \in \mathbb{Z}_p^{k+1}, \quad \vec{\mathsf{pk}}'^{\star} \triangleq (\vec{\mathsf{pk}}^{\star}, \mathbf{1}_{\mathbb{G}_2}) \in \mathbb{G}_2^{k+1}$
$\mathcal{P} \Leftarrow \mathcal{V}$: $\vec{\theta} \triangleq (\theta_j)_{j\in[k]} \xleftarrow{\$} \mathbb{Z}_p^{*k}, \quad \vec{\theta}' \triangleq (\vec{\theta}, 0) \in \mathbb{Z}_p^{*k+1}$
$\mathcal{V}$: $\mathtt{D}_0 \triangleq \prod_{j\in[k]} \mathsf{e}(\Gamma_j, \Lambda_j)^{\theta_j} = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{\Lambda}}^{\circ\vec{\theta}}) \in \mathbb{G}_T$
$\mathtt{D}_1 \triangleq \mathsf{e}(H_1, \mathtt{cm}_{\mathsf{m}}) = \mathsf{e}(\vec{H}_1^{\circ\vec{\mathsf{m}}'}, \vec{\mathbf{\Lambda}}') \in \mathbb{G}_T$
$\mathtt{D}_2 \triangleq \mathsf{e}(\mathtt{cm}_{\sigma_1}, G_2) = \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ\vec{\sigma}'_1}) \in \mathbb{G}_T$
$\mathtt{D}_3 \triangleq \prod_{j\in[k]} \mathsf{e}(\Gamma_j, G_2)^{\theta_j} = \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ\vec{\theta}'}) \in \mathbb{G}_T$
$\mathcal{P}$: $\mathsf{r}_{\mathtt{Z}_1}, \mathsf{r}_{\mathtt{Z}_2}, \mathsf{r}_{\mathtt{Z}_3}, \mathsf{r}'_{\sigma_0} \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}'_{\sigma_0} = \mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}'\sigma_0} \in \mathbb{G}_T$
$\mathtt{Z}_1 \triangleq \mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{G}_2^{\circ\vec{\sigma}_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}} \in \mathbb{G}_T$
$\mathtt{Z}_2 \triangleq \mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{\mathsf{pk}}^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_2}} \in \mathbb{G}_T$
$\mathtt{Z}_3 \triangleq \mathsf{e}(\vec{H}_1^{\circ\vec{\mathsf{m}}}, \vec{G}_2^{\circ\vec{\theta}}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_3}} \in \mathbb{G}_T$
$\mathcal{V}\ \&\ \mathcal{P}$: Run $\Pi_{\mathsf{do.ip}}\big[k{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{cm}'_{\sigma_0}, \mathtt{cm}_{\sigma_0}, \mathtt{D}_0;\ \vec{\sigma}'_0, \vec{\mathbf{\Lambda}}'^{\circ\vec{\theta}'}, \mathsf{r}'_{\sigma_0}, \mathsf{r}_{\sigma_0}, 0\big]$
// Check $\vec{\sigma}_0^{\circ\vec{\theta}}$ in $\mathtt{cm}'_{\sigma_0}$
Run $\Pi_{\mathsf{do.ip}}\big[k{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{Z}_1, \mathtt{cm}'_{\sigma_0}, \mathtt{D}_2;\ \vec{\sigma}'_0{}^{\circ\vec{\theta}'}, \vec{G}_2^{\circ\vec{\sigma}'_1}, \mathsf{r}_{\mathtt{Z}_1}, \mathsf{r}'_{\sigma_0}, 0\big]$
// Check $\mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{G}_2^{\circ\vec{\sigma}_1})$ in $\mathtt{Z}_1$
Run $\Pi_{\mathsf{do.ip}}\big[k{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{Z}_2, \mathtt{cm}'_{\sigma_0}, \mathtt{cm}_{\mathsf{pk}};\ \vec{\sigma}'_0{}^{\circ\vec{\theta}'}, \vec{\mathsf{pk}}'^{\star}, \mathsf{r}_{\mathtt{Z}_2}, \mathsf{r}'_{\sigma_0}, \mathsf{r}_{\mathsf{pk}}\big]$
// Check $\mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{\mathsf{pk}}^{\star})$ in $\mathtt{Z}_2$
Run $\Pi_{\mathsf{do.ip}}\big[k{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{Z}_3, \mathtt{D}_1, \mathtt{D}_3;\ \vec{H}_1^{\circ\vec{\mathsf{m}}'}, \vec{G}_2^{\circ\vec{\theta}'}, \mathsf{r}_{\mathtt{Z}_3}, 0, 0\big]$
// Check $\mathsf{e}(\vec{H}_1^{\circ\vec{\mathsf{m}}}, \vec{G}_2^{\circ\vec{\theta}})$ in $\mathtt{Z}_3$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathsf{r}' \triangleq \mathsf{r}_{\mathtt{Z}_1} + \mathsf{r}_{\mathtt{Z}_2} - \mathsf{r}_{\mathtt{Z}_3} \in \mathbb{Z}_p$
$\mathcal{V}$: CHECK $\mathtt{Z}_1 \cdot \mathtt{Z}_2 \stackrel{?}{=} \mathsf{e}(G_1, G_2)^{\sum_{j\in[k]} \theta_k} \cdot \mathtt{Z}_3 \cdot \mathtt{Q}^{\mathsf{r}'}$
// Check $\mathsf{e}(\sigma_{j,0}, G_2^{\sigma_{j,1}} \cdot \mathsf{pk}_{i_j}) = \mathsf{e}(G_1 \cdot H_1^{\mathsf{m}_j}, G_2),\ \forall j \in [k]$

Figure 11: Proof-of-knowledge of k-tuple of public key, message and signature of BBS signature.

of this ring. The relation for threshold ring referral of BBS signature is defined as

$$\mathscr{R}_{\mathsf{RR.bbs.th}} \triangleq \Big\{ \vec{\mathsf{pk}} \in \mathbb{G}_2^n;\ \vec{\mathsf{m}} \in \mathbb{Z}_p^M, \vec{\sigma} \in \mathbb{G}_1^k \times \mathbb{Z}_p^k, \{i_j\}_{j\in[k]} \subset [n], \vec{\mathsf{pk}}^{\star} = \big(\mathsf{pk}_{i_j} \in \vec{\mathsf{pk}}\big)_{j\in[k]} \in \mathbb{G}_2^k \ \Big|\ \mathsf{e}(\sigma_{j,0}, G_2^{\sigma_{j,1}} \cdot \mathsf{pk}_{i_j}) = \mathsf{e}(G_1 \cdot H_1^{\mathsf{m}_j}, G_2),\ \forall j \in [k] \Big\}.$$

Proof-of-Knowledge for k BBS signatures. The prover has a collection of k single-message BBS signatures $\vec{\sigma} = (\vec{\sigma}_0, \vec{\sigma}_1) \in \mathbb{G}_1^k \times \mathbb{Z}_p^k$ on messages $\vec{\mathsf{m}} = (\mathsf{m}_1, \ldots, \mathsf{m}_k)$ from k different issuers with public keys $\vec{\mathsf{pk}}^{\star} = (\mathsf{pk}_{i_j})_{j\in[k]}$ in the ring. The prover needs to prove that these k signatures are

$\Pi_{\mathsf{RR.bbs.th}}\Big[\ \vec{\mathsf{pk}} \in \mathbb{G}_2^n, k \in [n];\ \vec{\mathsf{m}} \in \mathbb{Z}_p^k,\ \vec{\sigma} = (\vec{\sigma}_0, \vec{\sigma}_1) \in \mathbb{G}_1^k \times \mathbb{Z}_p^k,$
$\{i_j\}_{j\in[k]} \subset [n], \vec{\mathsf{pk}}^{\star} = (\mathsf{pk}_{i_j} \in \vec{\mathsf{pk}})_{j\in[k]} \in \mathbb{G}_2^k\Big]$

SETUP : $\vec{\mathsf{h}} \triangleq (\mathsf{h}_i \triangleq \mathsf{Hash}[\mathsf{pk}_i])_{i\in[n]} \in \mathbb{Z}_p^n$
$\vec{\mathbf{K}} \triangleq (K_i \triangleq \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i}) \in \mathbb{G}_2^n$
$\vec{\mathbf{\Gamma}}' \triangleq (\vec{\mathbf{\Gamma}}, \Gamma) \xleftarrow{\$} \mathbb{G}_1^{k+1}, \quad \vec{\mathbf{\Lambda}}' \triangleq (\vec{\mathbf{\Lambda}}, \Lambda) \xleftarrow{\$} \mathbb{G}_2^{k+1}$
$\mathcal{P}$: $\mathsf{r}_{\sigma_0}, \mathsf{r}_{\sigma_1}, \mathsf{r}_{\mathsf{pk}}, \mathsf{r}_{\mathsf{m}} \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}_{\sigma_0} \triangleq \mathsf{e}(\vec{\sigma}_0, \vec{\mathbf{\Lambda}}) \cdot \mathtt{Q}^{\mathsf{r}\sigma_0} \in \mathbb{G}_T, \quad \mathtt{cm}_{\sigma_1} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\sigma}_1} \cdot \Gamma^{\mathsf{r}\sigma_1} \in \mathbb{G}_1$
$\mathtt{cm}_{\mathsf{pk}} \triangleq \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathsf{pk}}^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}} \in \mathbb{G}_T, \quad \mathtt{cm}_{\mathsf{m}} \triangleq \vec{\mathbf{\Lambda}}^{\vec{\mathsf{m}}} \cdot \Lambda^{\mathsf{r}_{\mathsf{m}}} \in \mathbb{G}_2$
$\mathcal{V}\ \&\ \mathcal{P}$: RUN $\Pi_{\mathsf{bbs.pms.th}}\big[\mathtt{cm}_{\sigma_0}, \mathtt{cm}_{\sigma_1}, \mathtt{cm}_{\mathsf{pk}}, \mathtt{cm}_{\mathsf{m}};\ \vec{\mathsf{pk}}^{\star}, \vec{\mathsf{m}}, \vec{\sigma}\big]$
$\mathcal{P}$: $\vec{\mathbf{d}} \triangleq (d_i)_{i\in[n]}$, where $d_i \triangleq \begin{cases} 0, \text{ if } \mathsf{pk}_i \notin \vec{\mathsf{pk}}^{\star} \\ 1, \text{ if } \mathsf{pk}_i \in \vec{\mathsf{pk}}^{\star} \end{cases}$
$\vec{\mathsf{h}}^{\star} \triangleq (\mathsf{h}_{i_1}, \ldots, \mathsf{h}_{i_k}) \in \mathbb{Z}_p^k, \quad \vec{\mathbf{K}}^{\star} \triangleq (K_{i_1}, \ldots, K_{i_k}) \in \mathbb{G}_2^k$
$\vec{G}'_1 \triangleq (\vec{G}_1, \mathbf{1}_{\mathbb{G}_1}) \in \mathbb{G}_1^{k+1}, \quad \vec{\mathbf{K}}'^{\star} \triangleq (\vec{\mathbf{K}}^{\star}, \mathbf{1}_{\mathbb{G}_2}) \in \mathbb{G}_2^{k+1}$
$\mathsf{r}_1, \mathsf{r}_2, \mathsf{r}_{\mathtt{D}}, \mathsf{r}'_{\mathtt{D}} \xleftarrow{\$} \mathbb{Z}_p$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\mathtt{cm}_1 \triangleq \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1} \in \mathbb{G}_T, \quad \mathtt{cm}_2 \triangleq \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_2} \in \mathbb{G}_T$
$\mathtt{cm}_{\mathsf{h}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{h}}^{\star}} \cdot \Gamma^{\mathsf{r}_{\mathtt{D}}} \in \mathbb{G}_1, \quad \mathtt{R} \triangleq \mathsf{e}(\Gamma, G_2)^{\mathsf{r}'_{\mathtt{D}}} \in \mathbb{G}_T$
$\mathsf{r}'_2 \triangleq \mathsf{r}_2 - \mathsf{r}_{\mathsf{pk}} \in \mathbb{Z}_p$
$\mathcal{V}\ \&\ \mathcal{P}$: RUN $\Pi^{\mathsf{th}}_{\mathsf{chkuv}}[n, k, \mathtt{cm}_1, \vec{\mathbf{\Omega}}, \vec{\mathbf{K}}, \mathtt{Q}; \vec{G}_1^{\circ\vec{\mathbf{d}}}, \mathsf{r}_1]$
// Check $\vec{\mathbf{d}}$ being a binary vector of weight k
RUN $\Pi_{\mathsf{do.ip}}\big[k{+}1, \vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}', \mathtt{cm}_1, \mathsf{e}(\vec{G}_1, \vec{\mathbf{\Lambda}}), \mathtt{cm}_2;\ \vec{G}'_1, \vec{\mathbf{K}}'^{\star}, \mathsf{r}_1, 0, \mathsf{r}_2\big]$
// Check $\vec{\mathbf{K}}^{\star}$ in $\mathtt{cm}_2$
$\mathcal{P} \Leftarrow \mathcal{V}$: $\varepsilon \xleftarrow{\$} \mathbb{Z}_p^*$
$\mathcal{P} \Rightarrow \mathcal{V}$: $\widetilde{\mathsf{r}}_{\mathtt{D}} \triangleq \mathsf{r}_{\mathtt{D}} + \varepsilon \cdot \mathsf{r}'_{\mathtt{D}} \in \mathbb{Z}_p$
$\mathcal{V}$: $W \triangleq \mathsf{e}(\mathtt{cm}_{\mathsf{h}}, G_2) \cdot \mathtt{R}^{\varepsilon} \cdot \mathsf{e}(\Gamma, G_2)^{-\widetilde{\mathsf{r}}_{\mathtt{D}}} \in \mathbb{G}_T$
CHECK $\mathtt{cm}_2 \stackrel{?}{=} \mathtt{cm}_{\mathsf{pk}} \cdot W \cdot \mathtt{Q}^{\mathsf{r}'_2}$
// Check $\mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}^{\star}) \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathsf{pk}}^{\star}) \cdot \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2^{\circ\vec{\mathsf{h}}^{\star}})$

Figure 12: Threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$ for BBS signature.

valid simultaneously:

$$\mathsf{e}(\sigma_{j,0}, G_2^{\sigma_{j,1}} \cdot \mathsf{pk}_{i_j}) = \mathsf{e}(G_1 \cdot H_1^{\mathsf{m}_j}, G_2),\ \forall j \in [k], \quad (11)$$

where $\vec{\sigma}_0 = (\sigma_{j,0})_{j\in[k]}$ and $\vec{\sigma}_1 = (\sigma_{j,1})_{j\in[k]}$, which can be batchec into a single check as

$$\mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{G}_2^{\circ\vec{\sigma}_1}) \cdot \mathsf{e}(\vec{\sigma}_0^{\circ\vec{\theta}}, \vec{\mathsf{pk}}^{\star}) \stackrel{?}{=} \mathsf{e}(\vec{G}_1 \cdot \vec{H}_1^{\circ\vec{\mathsf{m}}}, \vec{G}_2^{\circ\vec{\theta}}), \quad (12)$$

with verifier's random $\vec{\theta} = (\theta_1, \ldots, \theta_k) \xleftarrow{\$} \mathbb{Z}_p^{*k}$. Our proof-of-knowledge for k BBS signatures consists of four Dory checks

$$\begin{cases} \mathtt{cm}'_{\sigma_0} \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'_0, \vec{\mathbf{\Lambda}}'^{\circ\vec{\theta}'}) \cdot \mathtt{Q}^{\mathsf{r}'_{\sigma_0}}, \\ \mathtt{cm}_{\sigma_1} \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'_0, \vec{\mathbf{\Lambda}}') \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}, \\ \mathtt{D}_0 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}'^{\circ\vec{\theta}'}). \end{cases} \quad \begin{cases} \mathtt{Z}_1 \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'^{\circ\vec{\theta}'}_0, \vec{G}_2^{\circ\vec{\sigma}'_1}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_1}}, \\ \mathtt{cm}'_{\sigma_0} \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'^{\circ\vec{\theta}'}_0, \vec{\mathbf{\Lambda}}') \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}, \\ \mathtt{D}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ\vec{\sigma}'_1}). \end{cases}$$

$$\begin{cases} \mathtt{Z}_2 \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'^{\circ\vec{\theta}'}_0, \vec{\mathsf{pk}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_2}}, \\ \mathtt{cm}'_{\sigma_0} \stackrel{?}{=} \mathsf{e}(\vec{\sigma}'^{\circ\vec{\theta}'}_0, \vec{\mathbf{\Lambda}}') \cdot \mathtt{Q}^{\mathsf{r}_{\sigma_1}}, \\ \mathtt{cm}_{\mathsf{pk}} \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{\mathsf{pk}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathsf{pk}}}. \end{cases} \quad \begin{cases} \mathtt{Z}_3 \stackrel{?}{=} \mathsf{e}(\vec{H}_1^{\circ\vec{\mathsf{m}}'}, \vec{G}_2^{\circ\vec{\theta}'}) \cdot \mathtt{Q}^{\mathsf{r}_{\mathtt{Z}_3}}, \\ \mathtt{D}_1 \stackrel{?}{=} \mathsf{e}(\vec{H}_1^{\circ\vec{\mathsf{m}}'}, \vec{\mathbf{\Lambda}}'), \\ \mathtt{D}_3 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{G}_2^{\circ\vec{\theta}'}). \end{cases}$$

The batched verification of k BBS signatures is

$$\mathtt{Z}_1 \cdot \mathtt{Z}_2 \stackrel{?}{=} \mathsf{e}(G_1, G_2)^{\sum_{j\in[k]} \theta_k} \cdot \mathtt{Z}_3 \cdot \mathtt{Q}^{\mathsf{r}'}.$$

Proof-of-Knowledge of k-out-of-n in a Ring. Given the commitment $\mathtt{cm}_{\mathsf{pk}}$ of k public keys in $\vec{\mathsf{pk}}^{\star}$, the prover needs to prove that the committed vector $\vec{\mathsf{pk}}^{\star}$ is a subset of k elements in the ring $\vec{\mathsf{pk}} \triangleq (\mathsf{pk}_i)_{i\in[n]}$. We need to prove knowledge of a binary vector $\vec{\mathbf{d}}$ such that $\vec{\mathbf{d}}\circ(\vec{\mathbf{d}}-\vec{\mathbf{1}}) = \vec{\mathbf{0}}$ and $\langle\vec{\mathbf{d}}, \vec{\mathbf{1}}\rangle = k$, and the public setup of the ring is well-formed. This is done with two Dory checks

$$\begin{cases} \mathsf{E}_0 \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{G}_2), \\ \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathsf{E}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{G}_2). \end{cases} \quad \begin{cases} \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1', \vec{\mathbf{K}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathsf{F}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1', \vec{\mathbf{\Lambda}}'), \\ \mathtt{cm}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{\mathbf{K}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_2}. \end{cases}$$

We construct a proof-of-knowledge for k-out-of-n public keys in a ring as follows:

1) In the setup, let $\mathsf{h}_i \triangleq \mathtt{Hash}[\mathsf{pk}_i]$ for $i \in [n]$ and $\vec{\mathbf{K}} \triangleq (K_i \triangleq \mathsf{pk}_i \cdot G_2^{\mathsf{h}_i})_{i\in[n]}$.
2) The prover defines a selector vector $\vec{\mathbf{d}} \triangleq (d_i)_{i\in[n]} \in \{0,1\}^n$, where $d_i = 1$ if pk_i appears in $\vec{\mathsf{pk}}^{\star}$, and $d_i = 0$ otherwise. Note that $\sum_{i\in[n]} d_i = k$. Then the prover commits $\vec{\mathbf{d}}$ to $\mathtt{cm}_1 \triangleq \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1}$.
3) We can slightly modify the unit basis checking protocol Π_{chkuv} in Sec. 7.4 to prove the knowledge of $\vec{\mathbf{d}}$ by checking
$$\begin{cases} \vec{\mathbf{d}} \circ (\vec{\mathbf{d}} - \vec{\mathbf{1}}) = \vec{\mathbf{0}}, \\ \langle \vec{\mathbf{d}}, \vec{\mathbf{1}} \rangle = k, \end{cases} \tag{13}$$
We can equivalently check the relation $\langle\vec{\mathbf{d}}, \vec{\mathbf{1}}\rangle = k$ as
$$\mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{G}_2) = \mathsf{e}(G_1, G_2)^{\langle\vec{\mathbf{d}},\vec{\mathbf{1}}\rangle} \stackrel{?}{=} \mathsf{e}(G_1, G_2)^k, \tag{14}$$
using the following Dory check
$$\begin{cases} \mathsf{E}_0 \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{G}_2), \\ \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{\mathbf{K}}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathsf{E}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{G}_2), \end{cases} \tag{15}$$
with $(\vec{\mathbf{\Omega}}, \vec{\mathbf{K}})$ are known generators, $(\vec{G}_1^{\circ\vec{\mathbf{d}}}, \vec{G}_2)$ are the witness; and $\mathsf{E}_0 \triangleq \mathsf{e}(G_1, G_2)^k$ and $\mathsf{E}_3 \triangleq \mathsf{e}(\vec{\mathbf{\Omega}}, \vec{G}_2)$ can be precomputed.
For checking the relation $\vec{\mathbf{d}} \circ (\vec{\mathbf{d}} - \vec{\mathbf{1}}) = \vec{\mathbf{0}}$, we can use the same approach as in Π_{chkuv} for checking unit vector, which requires two additional Dory arguments with the same random generators $(\vec{\mathbf{\Omega}}, \vec{\mathbf{K}})$. We obtain the protocol $\Pi^{\mathsf{th}}_{\mathsf{chkuv}}\left[n, k, \mathtt{cm}_1, \vec{\mathbf{\Omega}}, \vec{\mathbf{K}}, \mathtt{Q}; \vec{G}_1^{\circ\vec{\mathbf{d}}}, \mathsf{r}_1\right]$ for checking the relations in Eqn. (13) from the unit basis checking protocol Π_{chkuv} in Fig. 5b by changing the term $\mathsf{e}(G_1, G_2)$ in Π_{chkuv} to $\mathsf{e}(G_1, G_2)^k$.
4) The prover also commits $\vec{\mathbf{K}}^{\star} = (K_{i_1}, \ldots, K_{i_k})$, $\vec{h}^{\star} = (\mathsf{h}_{i_1}, \ldots, \mathsf{h}_{i_k})$ to $\mathtt{cm}_2 \triangleq \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_2}$, $\mathtt{cm}_{\mathsf{h}} \triangleq \vec{\mathbf{\Gamma}}^{\vec{\mathsf{h}}^{\star}} \cdot \Gamma^{\mathsf{r}_{\mathsf{h}}}$. She also sends an additional term $\mathtt{R} \triangleq \mathsf{e}(\Gamma, G_2)^{\mathsf{r}'_{\mathsf{h}}}$ to the verifier. The verifier can check that $\vec{\mathbf{K}}^{\star}$ is committed in $\mathtt{cm}_2$ by using the following Dory check
$$\begin{cases} \mathtt{cm}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1', \vec{\mathbf{K}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_1}, \\ \mathsf{F}_1 \stackrel{?}{=} \mathsf{e}(\vec{G}_1', \vec{\mathbf{\Lambda}}'), \\ \mathtt{cm}_2 \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}', \vec{\mathbf{K}}'^{\star}) \cdot \mathtt{Q}^{\mathsf{r}_2}, \end{cases} \tag{16}$$
where $(\vec{\mathbf{\Gamma}}', \vec{\mathbf{\Lambda}}')$ are known generators, $\vec{G}_1' \triangleq (\vec{G}_1, \mathbf{1}_{\mathbb{G}_1}) \in \mathbb{G}_1^{k+1}$ and $\vec{\mathbf{K}}'^{\star} \triangleq (\vec{\mathbf{K}}^{\star}, \mathbf{1}_{\mathbb{G}_2})$ are the witness; and $\mathsf{F}_1 \triangleq \mathsf{e}(\vec{G}_1', \vec{\mathbf{\Lambda}}')$ can be precomputed. Next, the verifier can compute $\mathsf{e}(\mathtt{cm}_{\mathsf{h}}, G_2) = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2^{\circ\vec{\mathsf{h}}^{\star}}) \cdot \mathsf{e}(\Gamma, G_2)^{\mathsf{r}_{\mathsf{h}}}$ herself. The prover sends to the verifier $\widetilde{\mathsf{r}}_{\mathsf{h}} \triangleq \mathsf{r}_{\mathsf{h}} + \varepsilon \cdot \mathsf{r}'_{\mathsf{h}}$ where $\varepsilon \xleftarrow{\$} \mathbb{Z}_p^*$ is provided by the verifier. Using $\mathtt{R}$ and $\widetilde{\mathsf{r}}_{\mathsf{h}}$, the verifier can compute $W \triangleq \mathsf{e}(\mathtt{cm}_{\mathsf{h}}, G_2) \cdot \mathtt{R}^{\varepsilon} \cdot \mathsf{e}(\Gamma, G_2)^{-\widetilde{\mathsf{r}}_{\mathsf{h}}} = \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2^{\circ\vec{\mathsf{h}}^{\star}})$.
5) The verification of k public keys in $\vec{\mathsf{pk}}^{\star}$ can be checked as follows:
$$\mathtt{cm}_2 \stackrel{?}{=} \mathtt{cm}_{\mathsf{pk}} \cdot W \cdot \mathtt{Q}^{\mathsf{r}'_2},$$
$$\Rightarrow \ \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathbf{K}}^{\star}) \stackrel{?}{=} \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{\mathsf{pk}}^{\star}) \cdot \mathsf{e}(\vec{\mathbf{\Gamma}}, \vec{G}_2^{\circ\vec{\mathsf{h}}^{\star}}),$$
where $\mathsf{r}'_2 \triangleq \mathsf{r}_2 - \mathsf{r}_{\mathsf{pk}}$ is provided by the prover.

Optimization by Batching. Using batching for Dory protocols, $\Pi_{\mathsf{RR.bbs.th}}$ is optimized to calling only 2 batched Dory arguments of dimension n and $(k+1)$. Note that for threshold k and ring size n, the proof size and verification cost of $\Pi_{\mathsf{RR.bbs.m}}$ is logarithmic in n, but scales linearly in k.

$$\Pi_{\mathsf{eca21}}\Big[[\mathsf{ipk}_1, \ldots, \mathsf{ipk}_n] \in \mathbb{G}_2^n,\ \vec{\mathsf{m}} \in \mathbb{Z}_p^M,\ \mathsf{msg} \in \{0,1\}^*,\ \sigma_0 \in \mathbb{G}_2;$$
$$i^* \in [n],\ \mathsf{ipk}_{i^*},\ \mathsf{upk}, \mathsf{usk}, \sigma_1, (\tau_i)_{i\in[1+M]}\Big]$$

- -

$\mathcal{P}: \mathsf{r}, \mathsf{t} \xleftarrow{\$} \mathbb{Z}_p$

$\mathcal{P} \Rightarrow \mathcal{V}: \vec{\mathsf{c}} \triangleq (\mathsf{ipk}_{i^*} \cdot \mathsf{Y}^{\mathsf{r}}, G_2^{\mathsf{r}}) \in \mathbb{G}_2^2, \quad \mathsf{upk}' \triangleq \mathsf{upk}^{\mathsf{t}} \in \mathbb{G}_1$

$\mathcal{V}\ \&\ \mathcal{P}: \vec{\mathsf{c}}_i \triangleq (\frac{\vec{\mathsf{c}}[1]}{\mathsf{ipk}_{i^*}}, \vec{\mathsf{c}}[2]) \in \mathbb{G}_2^2,\ i = 1, \ldots, n.$

$\mathcal{P} \Rightarrow \mathcal{V}: \pi_{1/n} \leftarrow \Pi_{1/n}[n, (\vec{\mathsf{c}}_1, \ldots, \vec{\mathsf{c}}_n);\ (i^*, \mathsf{r})]$

$\pi^1_{\mathsf{gs}} \leftarrow \Pi_{\mathsf{gs}\text{-}\mathsf{G1}}[m = 1, n' = 0, T_1 = \mathsf{upk}';\ X_1 = \mathsf{upk}, b_1 = \mathsf{t}]$

$\pi^2_{\mathsf{gs}} \leftarrow \Pi_{\mathsf{gs}\text{-}\mathsf{PPE}}[m = 1, n = 1, B_1 = \sigma_0, A_1 = G_1,$
$T = \mathsf{e}(\mathsf{Y}_1, G_2);\ \ X_1 = \sigma_1, \mathsf{Y}_1 = \mathsf{ipk}_{i^*}]$

$\pi^3_{\mathsf{gs}} \leftarrow \Pi_{\mathsf{gs}\text{-}\mathsf{PPE}}[m = 2, n = 1, B_1 = \sigma_0, B_2 = G_2, A_1 = \mathsf{Y}_1,$
$T = \mathbf{1}_{\mathbb{G}_T};\ \ X_1 = \tau_1, X_2 = \mathsf{upk}, \mathsf{Y}_1 = \mathsf{ipk}_{i^*}]$

$\pi^{4,k}_{\mathsf{gs}} \leftarrow \Pi_{\mathsf{gs}\text{-}\mathsf{PPE}}[m = 2, n = 1, B_1 = \sigma_0, B_2 = G_2, A_1 = \mathsf{Y}_{k+1},$
$T = \mathbf{1}_{\mathbb{G}_T};\ \ X_1 = \tau_{k+1}, X_2 = \mathsf{m}_k, \mathsf{Y}_1 = \mathsf{ipk}_{i^*}], \quad k \in [M]$

$\mathcal{V}:$ CHECK $\Pi_{1/n}.\mathsf{Verify}[\pi_{1/n}] \stackrel{?}{=} 1,\ \Pi_{\mathsf{gs}\text{-}\mathsf{PPE}}.\mathsf{Verify}[\pi^i_{\mathsf{gs}}] \stackrel{?}{=} 1,\ i = 2, 3,$
$\Pi_{\mathsf{gs}\text{-}\mathsf{G}_1}.\mathsf{Verify}[\pi^1_{\mathsf{gs}}] \stackrel{?}{=} 1, \quad \Pi_{\mathsf{gs}\text{-}\mathsf{PPE}}.\mathsf{Verify}[\pi^{4,k}_{\mathsf{gs}}] \stackrel{?}{=} 1,\ k \in [M]$

Figure 13: Multi-issuer anonymous credential scheme ECA21 ([8]) with transparent setup.

Appendix D. The multi-issuer anonymous credential scheme ECA21

We recall the version with transparent setup of the ECA21 multi-issuer anonymous credential scheme from [8] in Fig.13. We refer to the original paper [8] for details on the primitives and the Groth-Sahai proof used in this scheme.

Appendix E.
Security of Digital Signature Schemes

In this section, we first recall the security properties of a standard digital signature scheme in Definition 5.1. This is standard material and can be found, for example, in the book [33]. Then, we present the security model for our compressed message-hiding signature scheme in Definition 5.2, which is new in this work. In particular, we need to extend the unforgeability property in this case to account for an additional attack vector induced by compression proof.

E.1. Standard Digital Signature

The security model of a digital signature scheme in Definition 5.1 consists of Completeness and Existential Unforgeability under Chosen Message Attack (EU-CMA) properties [33].

Definition E.1 (Completeness)**.** *A digital signature scheme* $\mathtt{Sig} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Sign}, \mathtt{VfySig})$ *satisfies* completeness*, if for any message* $\vec{\mathsf{m}}$*:*

$$\Pr\left[\begin{array}{c}\mathtt{VfySig}[\mathsf{pp},\mathsf{pk},\vec{\mathsf{m}},\sigma]\\=1\end{array}\;\middle|\;\begin{array}{c}\mathsf{pp}\leftarrow\mathtt{Setup}[1^\lambda],\\(\mathsf{pk},\mathsf{sk})\leftarrow\mathtt{KeyGen}[\mathsf{pp}],\\\sigma\leftarrow\mathtt{Sign}[\mathsf{pp},\vec{\mathsf{m}};\mathsf{sk}]\end{array}\right]\geq 1-\mathrm{negl}(\lambda).$$

An adversary against the signature unforgeability property is given access to a signing oracle.

Definition E.2 (Signing Oracle $\mathcal{SO}$)**.** *Initialize the set of queried messages as* $\mathcal{M}_{\mathcal{SO}}=\varnothing$*. When* $\mathcal{SO}$ *is queried with a message* $\vec{\mathsf{m}}$*, the oracle uses the secret key* sk *to sign* $\sigma=\mathtt{Sign}[\mathsf{pp},\vec{\mathsf{m}};\mathsf{sk}]$*. It returns* σ *to the adversary, and adds* $\vec{\mathsf{m}}$ *to* $\mathcal{M}_{\mathcal{SO}}(\mathsf{pk})$*. The adversary can adaptively make polynomially many signing queries.*

Definition E.3 (EU-CMA)**.** *A digital signature scheme* $\mathtt{Sig} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Sign}, \mathtt{VfySig})$ *satisfies* existential unforgeability under chosen message attack (EU-CMA) *property, if for any PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[\begin{array}{c}\mathtt{VfySig}[\mathsf{pp},\mathsf{pk},\vec{\mathsf{m}}^*,\sigma^*]=1\\\wedge\ \vec{\mathsf{m}}^*\notin\mathcal{M}_{\mathcal{SO}}(\mathsf{pk})\end{array}\;\middle|\;\begin{array}{c}\mathsf{pp}\leftarrow\mathtt{Setup}[1^\lambda],\\(\mathsf{pk},\mathsf{sk})\leftarrow\mathtt{KeyGen}[\mathsf{pp}],\\(\vec{\mathsf{m}}^*,\sigma^*)\leftarrow\mathcal{A}^{\mathcal{SO}}[\mathsf{pp},\mathsf{pk}]\end{array}\right]\leq\mathrm{negl}(\lambda).$$

E.2. Compressed Message-hiding Signature

In Definition 5.2, we define a compressed message-hiding signature $\mathtt{CSig} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Sign}, \mathtt{Compress}, \mathtt{VfyCSig})$ by adding a compression method and changing the verification accordingly.

Definition E.4 (Completeness)**.** *A compressed message-hiding signature scheme* $\mathtt{CSig}$ *satisfies* completeness*, if*

$$\Pr\left[\begin{array}{c}\mathtt{VfyCSig}[\mathsf{pp},\mathsf{pk},\mathtt{Cm}_{\mathtt{m}},\sigma,\pi]\\=1\end{array}\;\middle|\;\begin{array}{c}\mathsf{pp}\leftarrow\mathtt{Setup}(1^\lambda),\\(\mathsf{pk},\mathsf{sk})\leftarrow\mathtt{KeyGen}[\mathsf{pp}],\\\sigma\leftarrow\mathtt{Sign}[\mathsf{pp},\vec{\mathsf{m}};\mathsf{sk}],\\\mathtt{Cm}_{\mathtt{m}},\pi\leftarrow\mathtt{Compress}[\mathsf{pp},\vec{\mathsf{m}}]\end{array}\right]\geq 1-\mathrm{negl}(\lambda).$$

In this case, due to the message-hiding property the signing oracle $\mathcal{BSO}$ of the underlying signature returns both the message $\vec{\mathsf{m}}$ and the queried signature σ. It is required that the adversary must produce a fresh, unqueried signature to win. In addition, the unforgeability adversary against the compressed message-hiding signature $\mathtt{CSig}$ also has access to a proving oracle which returns valid compression proof.

Definition E.5 (Base Signature Oracle)**.** *Initialize the set of queried signature as* $\mathbf{\Sigma}_{\mathcal{BSO}}=\varnothing$*. When* $\mathcal{BSO}$ *is queried with a message* $\vec{\mathsf{m}}$*, the oracle uses the secret key* sk *to sign* $\sigma=\mathtt{Sign}[\mathsf{pp},\vec{\mathsf{m}};\mathsf{sk}]$*. It returns* σ *to the adversary, and adds* $(\vec{\mathsf{m}},\sigma)$ *to* $\mathbf{\Sigma}_{\mathcal{BSO}}(\mathsf{pk})$*.*

Definition E.6 (Proving Oracle $\mathcal{PO}$)**.** *Initialize the set of queries by* $\mathbf{\Pi}_{\mathcal{PO}}=\varnothing$*. When* $\mathcal{PO}$ *is queried with a message* $\vec{\mathsf{m}}$*, it runs the compression method to generate* $\mathtt{Cm}_{\mathtt{m}},\pi$ *on* $\mathsf{pp},\vec{\mathsf{m}}$*.* $\mathcal{PO}$ *then returns* $\mathtt{Cm}_{\mathtt{m}},\pi$ *to the adversary, and adds* $(\mathtt{Cm}_{\mathtt{m}},\pi)$ *to* $\mathbf{\Pi}_{\mathcal{PO}}$*.*

Definition E.7 (Unforgeability)**.** *A compressed message-hiding signature* $\mathtt{CSig}$ *satisfies* unforgeability*, if for any PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[\begin{array}{c}\mathtt{VfyCSig}[\mathsf{pp},\mathsf{pk},\mathtt{Cm}_{\mathtt{m}}{}^*,\sigma^*,\pi^*]=1\\\wedge(\cdot,\sigma^*)\notin\mathbf{\Sigma}_{\mathcal{BSO}}\\\wedge\ (\mathtt{Cm}_{\mathtt{m}}{}^*,\pi^*)\notin\mathbf{\Pi}_{\mathcal{PO}}\end{array}\;\middle|\;\begin{array}{c}\mathsf{pp}\leftarrow\mathtt{Setup}[1^\lambda],\\(\mathsf{pk},\mathsf{sk})\leftarrow\mathtt{KeyGen}[\mathsf{pp}],\\(\sigma^*,\mathtt{Cm}_{\mathtt{m}}{}^*,\pi^*)\leftarrow\mathcal{A}^{\mathcal{BSO},\mathcal{PO}}[\mathsf{pp},\mathsf{pk}]\end{array}\right]$$
$$\leq\mathrm{negl}(\lambda).$$

Note that if the underlying signature scheme is EU-CMA, the commitment scheme is binding and hiding, and the compression proof satisfies soundness and SHVZK properties, then the unforgeability property of $\mathtt{CSig}$ follows by a generic security reduction. We will provide details of this security reduction for our compressed message-hiding multi-message BBS signature in Section F.2.

Appendix F.
Security Proofs

We provide security proofs for the compressed multi-message BBS signature scheme $\mathtt{Sig}_{\mathsf{bbs.c}}$, the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ for single-message BBS signature, the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ for multi-message BBS signature, and the threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$ for BBS signature. The high level ideas of the proofs are: (1) Firstly, we will prove soundness and zero-knowledge properties of the underlying arguments of knowledge; (2) Secondly, we will prove security properties using soundness and zero-knowledge properties with standard security reduction arguments; (3) Finally, the security properties for non-interactive versions of our protocols follows from the general security of the Fiat-Shamir transform in the random oracle model.

F.1. Relaxed Soundness

Among our four protocols, only $\Pi_{\mathsf{RR.bbs}}$ has the witness extractability property, meaning that we can construct a polynomial time algorithm to extract the witness of the corresponding relation $\mathscr{R}_{\mathsf{RR.bbs}}$ from a valid transcript by rewinding the prover. The other three protocols $\mathtt{Sig}_{\mathsf{bbs.c}}$, $\Pi_{\mathsf{RR.bbs.m}}$, $\Pi_{\mathsf{RR.bbs.th}}$ satisfies a weaker soundness property which we call *relaxed soundness*. In particular, instead

of extracting all elements of the witness w, we consider $w = (w_1, \vec{w}_2)$ for some $\vec{w}_2 \in \mathbb{Z}_p^q$ for which a protocol has the relaxed soundness property if there is a polynomial time algorithm which can extract w_1 and $\vec{G}' \in \mathbb{G}^q$ such that $\vec{G}' = \vec{G}^{\circ \vec{w}_2} = (G^{w_{2,j}})_{j\in[q]} \in \mathbb{G}^q$ for a public group $\mathbb{G}$, a group generator G and some unknown $\vec{w}_2 \in \mathbb{Z}_p^q$. Assume that $\mathbb{G}$ is cyclic and the Dlog problem is hard for $\mathbb{G}$, then the relaxed soundness property will be enough for proving unforgeability property of $\mathtt{Sig}_{\mathsf{bbs.c}}$, $\Pi_{\mathsf{RR.bbs.m}}$ and $\Pi_{\mathsf{RR.bbs.th}}$.

Definition F.1 (Relaxed Soundness)**.** *Argument system $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ satisfies relaxed soundness with respect to $G \in \mathbb{G}$, if there exists an expected polynomial-time extractor $\mathcal{E}$, such that for any interactive adversaries $\mathcal{A}_1, \mathcal{A}_2$:*

$$\left|\Pr\left[\mathcal{A}_1[\mathsf{tr}] = 1 \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}], \\ \mathsf{tr} \leftarrow \langle \tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle \end{array}\right]\right.$$

$$\left. -\Pr\left[\begin{array}{c} \mathcal{A}_1[\mathsf{tr}'] = 1 \\ \wedge \vec{G}' = \vec{G}^{\circ \vec{w}_2} \\ \wedge \big(\mathtt{Accept}[\mathsf{tr}'] = 1 \;\Rightarrow \\ (\mathsf{pp}, x, (w_1, \vec{w}_2)) \in \mathscr{R}\big) \end{array} \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda), \\ (x, \tilde{w}, \tilde{\mathcal{P}}) \leftarrow \mathcal{A}_2[\mathsf{pp}], \\ (\mathsf{tr}', (w_1, \vec{G}')) \leftarrow \mathcal{E}^{\mathcal{O}}[\mathsf{pp}, x] \end{array}\right]\right| \leq \mathrm{negl}(\lambda),$$

where $\tilde{\mathcal{P}}$ is a deterministic polynomial-time algorithm, $\mathcal{A}_1[\mathsf{tr}]$ recognizes the transcripts that are produced by $\tilde{\mathcal{P}}$, and $\mathcal{O}$ is a rewindable oracle that can rewind the transcript $\langle \tilde{\mathcal{P}}(\mathsf{pp}, x, \tilde{w}), \mathcal{V}(\mathsf{pp}, x)\rangle$ and control the randomness in $\mathcal{V}$.

F.2. Security Proofs of $\mathtt{Sig}_{\mathsf{bbs.c}}$

Lemma F.1 (Relaxed Soundness of the Compressed Message-hiding Signature $\mathtt{Sig}_{\mathsf{bbs.c}}$)**.** *Assume the standard signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ is complete and satisfies EU-CMA; AFGHO and Pedersen vector commitments are complete and computational binding; Dory argument is complete and satisfies CWE;* Hash *is a collision resistant hash function. Then, the interactive protocol underlying $\mathtt{Sig}_{\mathsf{bbs.c}}$ satisfies relaxed soundness property with respect to $G_2 \in \mathbb{G}_2$.*

Proof. Consider the underlying interactive protocol of $\mathtt{Sig}_{\mathsf{bbs.c}}$, we describe a polynomial time algorithm $\mathtt{Ext}_{\mathsf{bbs.c}}$ which after rewinding the prover in $\mathtt{Sig}_{\mathsf{bbs.c}}$ polynomially many times on a valid transcript tr will output a valid relaxed witness $w = \vec{\mathsf{m}} \in \mathbb{Z}_p^M$ for the verification relation of $\mathtt{Sig}_{\mathsf{bbs.c}}$.

Since the $(M+1)$-dimensional recursive Dory argument $\Pi_{\mathsf{do.ip}}$ the CWE property in $\mathtt{Compress}_{\mathsf{bbs.c}}$, $\mathtt{Ext}_{\mathsf{bbs.c}}$ can run its witness emulator to obtain $\vec{G}'_2 \in \mathbb{G}_2^{M+1}$, $\vec{\boldsymbol{H}}' \in \mathbb{G}_1^{M+1}$ and $\mathsf{r}_{\mathtt{D}} \in \mathbb{Z}_p^*$. That the transcript tr is valid for the recursive Dory argument also implies that $\vec{G}'_2 = \vec{G}_2^{\circ \vec{\mathsf{m}}'} \in \mathbb{G}_2^{M+1}$ for some unknown $\vec{\mathsf{m}}' \in \mathbb{Z}_p^{M+1}$ committed in $\mathtt{D}_0$ and $\mathtt{D}_2$. By rewinding the randomness θ once, $\mathtt{Ext}_{\mathsf{bbs.c}}$ can extract the values of $\mathsf{r}_{\mathtt{D}}$ and $\mathsf{r}_{\mathtt{R}}$. The final equation check guarantees that the extracted elements satisfy the verification equation of BBS signature and pass the verification of Dory argument. Thus, $\mathtt{Ext}_{\mathsf{bbs.c}}$ has extracted a relaxed witness of $\mathtt{Compress}_{\mathsf{bbs.c}}$ for the transcript tr by rewinding the prover polynomially many times. □

Lemma F.2 (SHVZK of $\mathtt{Sig}_{\mathsf{bbs.c}}$)**.** *Assume the standard signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ is complete and satisfies EU-CMA; AFGHO and Pedersen vector commitments are complete and perfectly hiding; Dory argument is complete and satisfies SHVZH;* Hash *is a pseudo random hash function. Then, the interactive protocol underlying $\mathtt{Sig}_{\mathsf{bbs.c}}$ satisfies SHVZK property.*

Proof. We construct a polynomial time simulator algorithm $\mathtt{Sim}_{\mathsf{bbs.c}}$ which, without knowing the witness, will have output distribution that is indistinguishable from the distribution of real executions of $\mathtt{Compress}_{\mathsf{bbs.c}}$.

$\mathtt{Sim}_{\mathsf{bbs.c}}$ first samples $\widehat{\mathtt{Cm}}_{\mathsf{m}} \xleftarrow{\$} \mathbb{G}_1$, $\widehat{\mathtt{D}}_0, \widehat{\mathtt{D}}_1 \xleftarrow{\$} \mathbb{G}_T$, and computes $\widehat{\mathtt{D}}_2 \triangleq \mathsf{e}(\widehat{\mathtt{Cm}}_{\mathsf{m}}, G_2)$. Using the SHVZK property of Dory, $\mathtt{Sim}_{\mathsf{bbs.c}}$ can simulate the proof $\widehat{\pi}'$ of the $(M+1)$-dimensional Dory argument on input $(\vec{\boldsymbol{\Gamma}}, P_1), (\vec{\boldsymbol{\Lambda}}, \mathbf{1}_{\mathbb{G}_2}), \widehat{\mathtt{D}}_0, \widehat{\mathtt{D}}_1, \widehat{\mathtt{D}}_2$. Next, $\mathtt{Sim}_{\mathsf{bbs.c}}$ samples $\widehat{\mathsf{r}}', \widehat{\theta} \xleftarrow{\$} \mathbb{Z}_p$, and computes $\widehat{R} \triangleq \big(\mathsf{e}(\sigma_0,\ G_2^{\sigma_1} \cdot \mathsf{pk}) \cdot \mathtt{Q}^{\widehat{\mathsf{r}}'} \cdot \mathsf{e}(G_1, G_2)^{-1} \cdot \widehat{\mathtt{D}}_0^{-1}\big)^{\widehat{\theta}^{-1}}$. It then defines $\widehat{\pi} \triangleq (\widehat{\mathtt{D}}_0, \widehat{\mathtt{R}}, \widehat{\pi}', \widehat{\mathsf{r}}')$.

By the definition of $\widehat{\mathtt{R}}$ and $\widehat{\pi}'$, the output of $\mathtt{Sim}_{\mathsf{bbs.c}}$ passes the verification in $\mathtt{VfyCSig}_{\mathsf{bbs.c}}$. The hiding property of commitment schemes implies that $\widehat{\mathtt{Cm}}_{\mathsf{m}}$ and $\widehat{\mathtt{D}}_0$ are indistinguishable from real commitments. The pseudorandom property of Hash implies that $\widehat{\theta}$ is indistinguishable from the distribution of θ. By the SHVZK property of Dory, $\widehat{\pi}'$ is indistinguishable from the distribution of the real transcript of the recursive Dory argument. We conclude that $\mathtt{Sim}_{\mathsf{bbs.c}}$ is a valid SHVZK simulator for $\mathtt{Compress}_{\mathsf{bbs.c}}$. □

Theorem F.3. *Assume the standard signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ is complete and satisfies EU-CMA; AFGHO and Pedersen vector commitments are complete, perfectly hiding and computational binding; Dory argument is complete and satisfies CWE, SHVZH;* Hash *is modeled as a random oracle. Then, the compressed signature scheme $\mathtt{Sig}_{\mathsf{bbs.c}}$ is complete and satisfies unforgeability and SHVZK in the random oracle model.*

Proof. **Completeness.** As the recursive Dory argument is complete, we can check that the verification will almost surely return accept, if we execute the protocol in $\mathtt{Sig}_{\mathsf{bbs.c}}$ honestly.

Unforgeability. We recall the security game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ for the unforgeability property (Definition E.7) as follows.

- $\mathcal{C}$ generates the public parameters $\mathsf{pp} \leftarrow \mathtt{Setup}_{\mathsf{bbs.c}}[1^\lambda]$, and a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KeyGen}_{\mathsf{bbs.c}}[\mathsf{pp}, \mathsf{rc}]$ using random coin rc. $\mathcal{C}$ then sends pp and pk to $\mathcal{A}$.
- $\mathcal{C}$ initializes the set of base signature queries $\boldsymbol{\Sigma}_{\mathcal{BSO}}(\mathsf{pk}) \triangleq \varnothing$, and the set of compression proof queries $\boldsymbol{\Pi}_{\mathcal{PO}} \triangleq \varnothing$.
- $\mathcal{A}$ can make signing and proving queries polynomially many times to which $\mathcal{C}$ responds as follows:
  - $\mathcal{BSO}[\vec{\mathsf{m}}]$: $\mathcal{C}$ signs the message $\vec{\mathsf{m}}$ as $\sigma = \mathtt{Sign}_{\mathsf{bbs.c}}[\mathsf{pp}, \vec{\mathsf{m}}; \mathsf{sk}]$ using the secret key sk. It returns σ to $\mathcal{A}$, and adds $(\vec{\mathsf{m}}, \sigma)$ to $\boldsymbol{\Sigma}_{\mathcal{BSO}}(\mathsf{pk})$.

– $\mathcal{PO}[\vec{\mathsf{m}}]$: $\mathcal{C}$ runs the compression method $\mathtt{Compress}_{\mathsf{bbs.c}}[\mathsf{pp}, \vec{\mathsf{m}}]$ to generate $\mathtt{Cm}_{\mathsf{m}}, \pi$ on $\vec{\mathsf{m}}$. $\mathcal{C}$ then returns $\mathtt{Cm}_{\mathsf{m}}, \pi$ to the adversary $\mathcal{A}$, and adds $(\mathtt{Cm}_{\mathsf{m}}, \pi)$ to $\mathbf{\Pi}_{\mathcal{PO}}$.

- $\mathcal{A}$ outputs $(\sigma^*, {\mathtt{Cm}_{\mathsf{m}}}^*, \pi^*)$.
If $\mathtt{VfyCSig}_{\mathsf{bbs.c}}[\mathsf{pp}, \mathsf{pk}, {\mathtt{Cm}_{\mathsf{m}}}^*, \sigma^*, \pi^*] = 1$, σ^* has not been queried on any message before, and $({\mathtt{Cm}_{\mathsf{m}}}^*, \pi^*) \notin \mathbf{\Pi}_{\mathcal{PO}}$, then we say that the adversary $\mathcal{A}$ wins the unforgeability game.

SHVZK. Since the interactive protocol underlying $\mathtt{Sig}_{\mathsf{bbs.c}}$ satisfies SHVZK, the challenger $\mathcal{C}$ can replace transcripts of all honest executions of the protocol by output of the zero-knowledge simulator $\mathtt{Sim}_{\mathsf{bbs.c}}$ which the adversary $\mathcal{A}$ can notice the difference with only negligible probability.

Suppose that we have some adversary $\mathcal{A}$ who wins against this challenger $\mathcal{C}$ with non-negligible probability in the unforgeability game above. Consider a winning round of $\mathcal{A}$, we can fix the randomness of $\mathcal{A}$ up to this round, and rewinds $\mathcal{A}$ using new randomness starting from this round. The adversary $\mathcal{A}$ will output a new valid signature-compression proof with non-negligible probability.

At this point, we use the relaxed soundness property of $\mathtt{Sig}_{\mathsf{bbs.c}}$: by using this rewinding procedure polynomial number of times, we obtain enough valid transcripts for the relaxed witness extractor $\mathtt{Ext}_{\mathsf{bbs.c}}$ of $\mathtt{Sig}_{\mathsf{bbs.c}}$ to extract a relaxed witness $\vec{G}_2' = \vec{G}_2^{\circ\vec{\mathsf{m}}'}$. Note that the map $\vec{\mathsf{m}} \mapsto \vec{G}_2^{\circ\vec{\mathsf{m}}}$ is one to one, and computational binding under the Dlog assumption in $\mathbb{G}_2$.

But the transcript has been simulated in zero-knowledge without using witness at the beginning, so the adversary $\mathcal{A}$ must either break the EU-CMA of the base BBS signature scheme, or break the discrete logarithm assumption in $\mathbb{G}_2$ to find an uncorrupted multi-message $\vec{\mathsf{m}}'$ such that $\vec{G}_2' = \vec{G}_2^{\circ\vec{\mathsf{m}}'}$, or break the binding property of the Pedersen commitment scheme and CWE property of Dory to forge a new compression proof, with non-negligible probability. The unforgeability property of the non-interactive compressed multi-message signature scheme $\mathtt{Sig}_{\mathsf{bbs.c}}$ follows in the random oracle model by applying the Fiat-Shamir transform. □

F.3. Security Proofs of $\Pi_{\mathsf{RR.bbs}}$

We show that as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs}}$, the protocol $\Pi_{\mathsf{RR.bbs}}$ has CWE and SHVZK properties. We then use these properties to prove the ring referral security of $\Pi_{\mathsf{RR.bbs}}$.

Lemma F.4 (CWE of $\Pi_{\mathsf{RR.bbs}}$)**.** *Assume that in the setting of the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$, AFGHO and Pedersen commitments are complete and computationally binding; the scalar and vector Dory arguments are complete and satisfies CWE; the Schnorr protocol for commitment checks is complete and satisfies knowledge soundness, and* $\mathtt{Hash}$ *is a collision resistant hash function. Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs}}$, $\Pi_{\mathsf{RR.bbs}}$ satisfies CWE.*

Proof. We describe a polynomial time extractor algorithm $\mathtt{Ext}_{\mathsf{RR.bbs}}$ which, after rewinding the prover in $\Pi_{\mathsf{RR.bbs}}$ polynomially many times on a valid transcript tr, will output a valid witness for the relation $\mathscr{R}_{\mathsf{RR.bbs}}$.

First, we will use the fact that the sub-protocol $\Pi_{\mathsf{bbs.pms}}$ also accepts its part of the transcript tr to extract the part of witness in $\Pi_{\mathsf{bbs.pms}}$. Using the knowledge soundness assumption on the Schnorr protocols Π_{chkcm} for checking commitment values, $\mathtt{Ext}_{\mathsf{RR.bbs}}$ can extract m in $\mathtt{cm}_{\mathsf{m}}$ and σ_1 in $\mathtt{cm}_{\sigma_1}$. Since the two scalar Dory arguments $\Pi_{\mathsf{do.sp}}$ also have the CWE property, $\mathtt{Ext}_{\mathsf{RR.bbs}}$ can run their witness emulator to extract σ_0, pk in $\mathtt{cm}_{\sigma_0}$, $\mathtt{cm}_{\mathsf{pk}}$. Due to the binding property of the commitment schemes, the extracted values from various protocols are consistent. The final equation check in $\Pi_{\mathsf{bbs.pms}}$ guarantees that the extracted elements satisfy the verification equation of single-message BBS signature.

Second, in the main protocol $\Pi_{\mathsf{RR.bbs}}$, Π_{chkuv} is a batched recursive Dory argument of dimension n which has the CWE property, so $\mathtt{Ext}_{\mathsf{RR.bbs}}$ can run the witness emulator of the Dory argument to obtain a unit basis vector $\vec{\boldsymbol{b}} \in \{0,1\}^n$ such that $\vec{L}^{\circ\vec{\boldsymbol{b}}}$ is committed in $\mathtt{cm}_1$. Then, $\mathtt{Ext}_{\mathsf{RR.bbs}}$ extracts the index i^* as the non-zero index of $\vec{\boldsymbol{b}}$. Next, $\mathtt{Ext}_{\mathsf{RR.bbs}}$ can run the witness extractor of the Schnorr protocol Π_{chkcm} to extract h from $\mathtt{cm}_2$.

Finally, the final checking equation of $\Pi_{\mathsf{RR.bbs}}$ ensures that for the extracted index i^*, public key pk, and h, we have $K_{i^*} = \mathsf{pk} \cdot G_2^h$ which shows that pk is indeed the public key of the issuer with index i^*: $\mathsf{pk} = \mathsf{pk}_{i^*}$ and $h = h_{i^*}$, by the assumption that $\mathtt{Hash}$ is collision resistant.

We conclude that $\mathtt{Ext}_{\mathsf{RR.bbs}}$ has extracted a valid witness for $\mathscr{R}_{\mathsf{RR.bbs}}$ for the transcript tr by rewinding the prover polynomially many times. □

Lemma F.5 (SHVZK of $\Pi_{\mathsf{RR.bbs}}$)**.** *Assume that in the setting of the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$, AFGHO and Pedersen commitments are complete and perfectly hiding, the scalar and vector Dory arguments and Schnorr protocols for commitment checks are complete and have SHVZK property, and* $\mathtt{Hash}$ *is a collision-resistant pseudo random hash function.*

Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs}}$, $\Pi_{\mathsf{RR.bbs}}$ satisfies SHVZK.

Proof. We construct a polynomial time simulator algorithm $\mathtt{Sim}_{\mathsf{RR.bbs}}$ which, without knowing the witness, will have output distribution that is indistinguishable from the distribution of real executions of $\Pi_{\mathsf{RR.bbs}}$.

First, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ samples $\widehat{i^*} \xleftarrow{\$} [n]$, picks the public key $\mathsf{pk}_{\widehat{i^*}}$ of the issuer $\widehat{i^*}$ from the ring $\vec{\mathsf{pk}}$, and computes $\widehat{h} \triangleq \mathtt{Hash}[\mathsf{pk}_{\widehat{i^*}}]$. In the sub-protocol $\Pi_{\mathsf{bbs.pms}}$, the simulator $\mathtt{Sim}_{\mathsf{RR.bbs}}$ uniformly randomly samples $\widehat{\sigma}_0 \xleftarrow{\$} \mathbb{G}_1$, $\widehat{\mathsf{m}} \xleftarrow{\$} \mathbb{Z}_p$. Then, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ samples uniformly random scalar $\widehat{\mathsf{r}}_{\sigma_0}$, $\widehat{\mathsf{r}}_{\mathsf{pk}}$, $\widehat{\mathsf{r}}_{\mathsf{m}}$ from $\mathbb{Z}_p^*$, and computes the corresponding AFGHO and Pedersen commitment values $\widehat{\mathtt{cm}}_{\sigma_0}$, $\widehat{\mathtt{cm}}_{\mathsf{pk}}$, $\widehat{\mathtt{cm}}_{\mathsf{m}}$. Note that σ_1 and $\widehat{\mathtt{cm}}_{\sigma_1}$ are not yet defined. The simulator $\mathtt{Sim}_{\mathsf{RR.bbs}}$ then samples a new randomness $\widehat{\mathsf{r}}'_{\mathsf{pk}} \xleftarrow{\$} \mathbb{Z}_p^*$ and computes $\widehat{\mathtt{cm}}'_{\mathsf{pk}} \triangleq \mathsf{e}(\widehat{\sigma}_0, \widehat{\vec{\mathsf{pk}}}) \cdot \mathtt{Q}^{\widehat{\mathsf{r}}'_{\mathsf{pk}}}$. Next, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ samples $\widehat{\mathsf{r}}' \xleftarrow{\$} \mathbb{Z}_p^*$, and defines $\widehat{\mathtt{cm}}'_{\sigma_1} \triangleq \mathsf{e}(G_1, G_2) \cdot \widehat{\mathtt{cm}}_{\mathsf{m}} \cdot \mathtt{Q}^{\widehat{\mathsf{r}}'} \cdot (\widehat{\mathtt{cm}}'_{\mathsf{pk}})^{-1}$.

Second, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ uses the SHVZK simulator of the two scalar Dory arguments to simulate their transcripts on $\widehat{\mathsf{cm}}_{\sigma_0}$, $\widehat{\mathsf{cm}}_{\sigma_1}$, $\widehat{\mathsf{cm}}'_{\sigma_1}$, $\widehat{\mathsf{cm}}_{\mathsf{pk}}$ and $\widehat{\mathsf{cm}}'_{\mathsf{pk}}$. In addition, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ also uses the SHVZK simulator of the two Schnorr protocols Π_{chkcm} on $\widehat{\mathsf{cm}}_{\mathsf{m}}$ and $\widehat{\mathsf{cm}}_{\sigma_1}$. Thanks to the hiding property of the commitment schemes and the SHVZK property of the sub-protocols, the distribution of the simulated transcripts defined by $\mathtt{Sim}_{\mathsf{RR.bbs}}$ above is indistinguishable from the distribution of the transcripts of real executions of $\Pi_{\mathsf{bbs.pms}}$.

Finally, the simulator $\mathtt{Sim}_{\mathsf{RR.bbs}}$ uses $\widehat{i^*}$ to define the unit basis vector $\widehat{\vec{\boldsymbol{b}}}$, then uses $\widehat{\vec{\boldsymbol{b}}}$ in the place of $\vec{\boldsymbol{b}}$ to compute $\widehat{\mathsf{cm}}_1$ with a new random $\widehat{\mathsf{r}}_1$ which $\mathtt{Sim}_{\mathsf{RR.bbs}}$ samples from $\mathbb{Z}_p^*$. The hiding property of commitment schemes implies that the distribution of $\widehat{\mathsf{cm}}_1$ is indistinguishable from random. Then, by using the SHVZK property of the batched n-dimensional Dory argument Π_{chkuv}, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ uses its SHVZK simulator to simulate the transcript of Π_{chkuv} on input $\widehat{\mathsf{cm}}_1$, $\widehat{\vec{\boldsymbol{b}}}$, $\widehat{\mathsf{r}}_1$. Next, $\mathtt{Sim}_{\mathsf{RR.ps}}$ samples $\widehat{\mathsf{r}}'_1 \xleftarrow{\$} \mathbb{Z}_p^*$, and defines $\widehat{\mathsf{cm}}_2 \triangleq \widehat{\mathsf{cm}}_1 \cdot \widehat{\mathsf{cm}}_{\mathsf{pk}}^{-1} \cdot \mathtt{Q}^{-\widehat{\mathsf{r}}'_1}$ which makes the final checking equation valid. By using the SHVZK property of Schnorr protocol Π_{chkcm}, $\mathtt{Sim}_{\mathsf{RR.bbs}}$ can simulate the transcript of Π_{chkcm} on input $\widehat{\mathsf{cm}}_2$ which completes the full description of the simulator $\mathtt{Sim}_{\mathsf{RR.bbs}}$. □

Theorem F.6 (Security of the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$)**.** *Assume in the setting of the ring referral protocol* $\Pi_{\mathsf{RR.bbs}}$ *(Fig. 4), the single-message BBS signature scheme* $\mathtt{Sig}_{\mathsf{bbs}}$ *is correct and EU-CMA; AFGHO and Pedersen commitment schemes are complete, computationally binding and perfectly hiding; the scalar and vector Dory arguments are complete and have CWE and SHVZK properties; the Schnorr protocol for commitment checks is complete, and has knowledge soundness and SHVZK properties, and* $\mathtt{Hash}$ *is a collision-resistant pseudo random hash function.*

Then, the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ *is complete, unforgeable, and has issuer anonymity against exposed signature and message-hiding user anonymity properties.*

Proof. **Completeness.** In $\Pi_{\mathsf{RR.bbs}}$, suppose that σ is a valid single-message BBS signature from an issuer $\mathsf{pk}_{i^*} \in \vec{\boldsymbol{pk}}$ on a message m, and all computations are performed correctly as prescribed in $\Pi_{\mathsf{RR.bbs}}$. Then, the checks by Dory arguments and the Schnorr protocols for commitment check will return 1 by the completeness. The equation checking in the sub-protocol $\Pi_{\mathsf{bbs.pms}}$ returns 1 as the signature is valid. The final checking equation of $\Pi_{\mathsf{RR.bbs}}$ also passes since K_{i^*} is correctly generated from the public key pk_{i^*}.

Unforgeability. We recall the security game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ for the ring referral unforgeability property as follows:

- $\mathcal{C}$ setups the game by generating the public parameters $\mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda)$, and key pairs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}, \mathsf{rc}_i]$ for each issuer $\mathcal{I}_i$, $i \in [n]$, using random coin rc_i. Then, $\mathcal{C}$ defines $\vec{\boldsymbol{pk}} \triangleq (\mathsf{pk}_i)_{i\in[n]}$, and initializes the set of corruption queries $\vec{\boldsymbol{pk}}_{\mathcal{CO}} \triangleq \varnothing$, the set of base signature queries $\boldsymbol{\Sigma}_{\mathcal{BSO}} \triangleq \varnothing$, and the set of ring referral proof queries $\boldsymbol{\Pi}_{\mathcal{PO}} \triangleq \varnothing$.
- $\mathcal{C}$ sends the public parameters pp and the ring of issuers $\vec{\boldsymbol{pk}}$ to $\mathcal{A}$.
- $\mathcal{A}$ can make corruption, base signature and proving queries polynomially many times to which $\mathcal{C}$ responds as follows:
  - $\mathcal{CO}[\mathsf{pk}_i]$: $\mathcal{C}$ returns the random coin rc_i to $\mathcal{A}$, and adds pk_i to $\vec{\boldsymbol{pk}}_{\mathcal{CO}}$.
  - $\mathcal{BSO}[\mathsf{pp}, \mathsf{pk}_i, \vec{\mathsf{m}}]$: $\mathcal{C}$ uses the secret key sk_i of the issuer in pk_i to sign $\vec{\mathsf{m}}$, and returns a valid signature σ to $\mathcal{A}$. Then, $\mathcal{C}$ adds $(\mathsf{pp}, \mathsf{pk}_i, \vec{\mathsf{m}}, \sigma)$ to $\boldsymbol{\Sigma}_{\mathcal{BSO}}$.
  - $\mathcal{PO}[\mathsf{pp}, \vec{\boldsymbol{pk}}, \mathcal{M}^*]$: $\mathcal{C}$ uses the secret key of a random issuer in the ring $\boldsymbol{pk}$ to sign a random message $\vec{\mathsf{m}} \in \mathcal{M}^*$, then runs the ring referral protocol on this message, signature, issuer tuple, and returns a valid ring referral proof π to $\mathcal{A}$. Then, $\mathcal{C}$ adds $(\mathsf{pp}, \vec{\boldsymbol{pk}}, \mathcal{M}^*, \pi)$ to $\boldsymbol{\Pi}_{\mathcal{PO}}$.
- $\mathcal{A}$ outputs a forged ring referral proof π' with respected to a sub-ring $\vec{\boldsymbol{pk}}' \subset \vec{\boldsymbol{pk}}$ and a message space $\mathcal{M}^*$. If $\mathtt{Verify}[\mathsf{pp}, \vec{\boldsymbol{pk}}', \mathcal{M}^*, \pi'] = 1$; and $\vec{\boldsymbol{pk}}' \subset \vec{\boldsymbol{pk}} \backslash \vec{\boldsymbol{pk}}_{\mathcal{CO}}$; and $(\mathsf{pp}, \mathsf{pk}, \vec{\mathsf{m}}, \cdot) \notin \boldsymbol{\Sigma}_{\mathcal{BSO}}$ for any $\mathsf{pk} \in \vec{\boldsymbol{pk}}'$ and any $\vec{\mathsf{m}} \in \mathcal{M}$; and $(\mathsf{pp}, \vec{\boldsymbol{pk}}', \mathcal{M}^*, \pi') \notin \boldsymbol{\Pi}_{\mathcal{PO}}$, then we say that the adversary $\mathcal{A}$ wins the unforgeability game.

Since the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ satisfies SHVZK, the challenger $\mathcal{C}$ can replace transcripts of all honest executions of the protocol by output of the zero-knowledge simulator $\mathtt{Sim}_{\mathsf{RR.bbs}}$ with negligible probability of being noticed by the adversary $\mathcal{A}$.

Suppose that we have some adversary $\mathcal{A}$ who wins against this challenger $\mathcal{C}$ with non-negligible probability in the unforgeability game above. After checking for corrupted public keys and base signature queries, under the DLog and unforgeability of the base signature assumptions, the adversary $\mathcal{A}$ gives an algorithm to output valid, unqueried ring referral proofs with non-negligible advantage. Consider a winning round of $\mathcal{A}$, we can fix the randomness of $\mathcal{A}$ up to this round, and rewinds $\mathcal{A}$ using new randomness starting from this round. The adversary $\mathcal{A}$ will output a new valid transcript of the ring referral protocol with non-negligible probability.

At this point, we use the CWE property of $\Pi_{\mathsf{RR.bbs}}$: by using this rewinding procedure polynomial number of times, we obtain enough valid transcripts for the witness extractor $\mathtt{Ext}_{\mathsf{RR.bbs}}$ of the ring referral protocol to extract a witness of the relation $\mathscr{R}_{\mathsf{RR.bbs}}$. But the transcript has been simulated without using witness at the beginning, so the adversary $\mathcal{A}$ is able to guess valid message, signature and public key tuples of the ring referral scheme with non-negligible probability, which breaks either the hiding property of the commitment scheme, or the EU-CMA of the BBS signature, or the DLog assumption.

Issuer Anonymity. We recall the security game between challenger $\mathcal{C}$ and adversary $\mathcal{A}$ for issuer anonymity against exposed signatures as follows:

- $\mathcal{C}$ setups the game by generating the public parameters $\mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda)$, and key pairs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}, \mathsf{rc}_i]$ for each issuer $\mathcal{I}_i$, $i \in [n]$, using

random coin rc_i. Then, $\mathcal{C}$ defines $\vec{\boldsymbol{pk}} \triangleq \left(\mathsf{pk}_i\right)_{i\in[n]}$, and initializes the set of corruption queries $\vec{\boldsymbol{pk}}_{\mathcal{CO}} \triangleq \varnothing$, the set of base signature queries $\boldsymbol{\Sigma}_{\mathcal{BSO}} \triangleq \varnothing$, and the set of ring referral proof queries $\boldsymbol{\Pi}_{\mathcal{PO}} = \varnothing$.
- $\mathcal{C}$ sends the public parameters pp and the ring of issuers $\vec{\boldsymbol{pk}}$ to $\mathcal{A}$.
- $\mathcal{A}$ picks two distinct indices i_1, i_2 from $[n]$ and a message $\vec{\mathsf{m}} \in \mathcal{M}^*$, then sends $(i_1, i_2, \vec{\mathsf{m}}, \mathcal{M}^*)$ to $\mathcal{C}$.
- $\mathcal{C}$ signs the message $\vec{\mathsf{m}}$ using secret keys $\mathsf{sk}_{i_1}, \mathsf{sk}_{i_2}$ to obtain signatures σ_1, σ_2, respectively. $\mathcal{C}$ sends σ_1, σ_2 to $\mathcal{A}$.
- $\mathcal{C}$ randomly samples $b \xleftarrow{\$} \{1, 2\}$, and produces the transcript π of the ring referral protocol on the signature σ_b with respected to the ring $\vec{\boldsymbol{pk}}$ and message space $\mathcal{M}^*$. Then $\mathcal{C}$ sends π to $\mathcal{A}$.
- $\mathcal{A}$ can make corruption, base signature and proving queries polynomially many times to which $\mathcal{C}$ replies as specified in the unforgeability proof above.
- $\mathcal{A}$ outputs a guess $b' \in \{1, 2\}$. If $i_1 \neq i_2$ and $b' = b$, then we say that the adversary $\mathcal{A}$ wins the issuer anonymity game.

Next, we show that the issuer anonymity property follows from the SHVZK property of the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$. In the issuer anonymity game above, independently of the bit b, the challenger $\mathcal{C}$ simulates valid transcripts of the protocol and sends them to the adversary $\mathcal{A}$. As the protocol transcript is generated without using the secret bit b, $\mathcal{A}$ cannot win better than random guessing. Note that in this case, even when the signatures are given to $\mathcal{A}$, it still cannot distinguish between transcripts of the simulated and real executions, which implies the anonymity property against full signature exposure.

User Anonymity. We recall the security game for the message-hiding user anonymity property as follows:

- $\mathcal{C}$ setups the game by generating the public parameters $\mathsf{pp} \leftarrow \mathtt{Setup}(1^\lambda)$, and key pairs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathtt{Sig.Gen}[\mathsf{pp}, \mathsf{rc}_i]$ for each issuer $\mathcal{I}_i$, $i \in [n]$, using random coin rc_i. Then, $\mathcal{C}$ defines $\vec{\boldsymbol{pk}} \triangleq \left(\mathsf{pk}_i\right)_{i\in[n]}$, and initializes the set of corruption queries $\vec{\boldsymbol{pk}}_{\mathcal{CO}} \triangleq \varnothing$, the set of base signature queries $\boldsymbol{\Sigma}_{\mathcal{BSO}} \triangleq \varnothing$, and the set of ring referral proof queries $\boldsymbol{\Pi}_{\mathcal{PO}} = \varnothing$.
- $\mathcal{C}$ sends the public parameters pp and the ring of issuers $\vec{\boldsymbol{pk}}$ to $\mathcal{A}$.
- $\mathcal{A}$ picks an index $i \in [n]$ and two distinct messages $\vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2 \in \mathcal{M}^*$. Then $\mathcal{A}$ sends $(i, \mathcal{M}^*, \vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2)$ to $\mathcal{C}$.
- $\mathcal{C}$ signs the messages $\vec{\mathsf{m}}_1, \vec{\mathsf{m}}_2$ using the secret key sk_i to obtain signatures σ_1, σ_2, respectively. Then $\mathcal{C}$ sends σ_1, σ_2 to $\mathcal{A}$.
- $\mathcal{C}$ randomly samples $b \xleftarrow{\$} \{1, 2\}$ and produces the transcript π of the ring referral protocol on the message-signature pair $(\vec{\mathsf{m}}_b, \sigma_b)$ with respected to the ring $\vec{\boldsymbol{pk}}$ and message space $\mathcal{M}^*$. Then, $\mathcal{C}$ sends π to $\mathcal{A}$.
- $\mathcal{A}$ can make corruption, base signature and proving queries polynomially many times to which $\mathcal{C}$ replies as specified in the unforgeability proof above.
- $\mathcal{A}$ outputs a guess $b' \in \{1, 2\}$. If $\vec{\mathsf{m}}_1 \neq \vec{\mathsf{m}}_2$ and $b' = b$, then we say that the adversary $\mathcal{A}$ wins the user anonymity game.

The user anonymity property also follows from the SHVZK property of the ring referral protocol $\Pi_{\mathsf{RR.bbs}}$ as the message is also part of the witness. This is because in the user anonymity game, independently of the bit b, the challenger $\mathcal{C}$ can simulate valid transcripts of the protocol to send to the adversary $\mathcal{A}$ without knowing m_b. As the protocol transcript is generated without using the secret bit b , $\mathcal{A}$ cannot win better than random guessing. □

F.4. Security Proofs of $\Pi_{\mathsf{RR.bbs.m}}$

We show that as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.m}}$, the protocol $\Pi_{\mathsf{RR.bbs.m}}$ has relaxed soundness and SHVZK properties. The ring referral security properties of $\Pi_{\mathsf{RR.bbs.th}}$ then follows from these properties by blackbox security reductions similar to the proofs in Appendices F.2 and F.3.

Lemma F.7 (Relaxed Soundness of $\Pi_{\mathsf{RR.bbs.m}}$)**.** *Assume that in the setting of the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ for multi-message BBS signature, AFGHO and Pedersen commitments are complete and computationally binding; the scalar and vector Dory arguments are complete and satisfies CWE; the Schnorr protocol for commitment checks is complete and satisfies knowledge soundness; and* $\mathtt{Hash}$ *is a collision resistant hash function. Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.m}}$, $\Pi_{\mathsf{RR.bbs.m}}$ satisfies relaxed soundness property with respect to $G_2 \in \mathbb{G}_2$.*

Proof. We can build a polynomial time extractor algorithm $\mathtt{Ext}_{\mathsf{RR.bbs.m}}$ in a similar way as in the proofs of Lemma F.1 and Lemma F.4, with the exception that by using the CWE property of the $(M+1)$-dimensional recursive Dory argument in $\Pi_{\mathsf{bbs.pms.m}}$, one can extract only the exponentiation vector $\vec{G}_2' = \vec{G}_2^{\vec{\circ}\vec{\mathsf{m}}'} \in \mathbb{G}_2^{M+1}$, not the multi-message $\vec{\mathsf{m}}$ itself. □

Lemma F.8 (SHVZK of $\Pi_{\mathsf{RR.bbs.m}}$)**.** *Assume that in the setting of the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ for multi-message BBS signature, AFGHO and Pedersen commitments are complete and perfectly hiding, the scalar and vector Dory arguments and Schnorr protocols for commitment checks are complete and have SHVZK property, and* $\mathtt{Hash}$ *is a collision-resistant pseudo random hash function.*

Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.m}}$, $\Pi_{\mathsf{RR.bbs.m}}$ satisfies SHVZK.

Proof. We can construct a polynomial time SHVZK simulator algorithm $\mathtt{Sim}_{\mathsf{RR.bbs.m}}$ in two steps: (i) we can construct a SHVZK simulator for the sub-protocol $\Pi_{\mathsf{bbs.pms.m}}$ in a similar way as in the proof of the SHVZK property of $\mathtt{Sig}_{\mathsf{bbs.c}}$ in Lemma F.2; (ii) a similar construction to the SHVZK simulation of the main protocol of $\Pi_{\mathsf{RR.bbs}}$ in the proof of Lemma F.5 can simulate the remaining part in the main protocol of $\Pi_{\mathsf{RR.bbs.m}}$. □

Theorem F.9 (Security of the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$)**.** *Assume in the setting of the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ (Fig. 7), the multi-message BBS signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ is correct and EU-CMA; AFGHO and Pedersen commitment schemes are complete, computationally binding and perfectly hiding; the scalar and vector Dory arguments are complete and have CWE and SHVZK properties; the Schnorr protocol for commitment checks is complete, and has knowledge soundness and SHVZK properties, and* Hash *is a collision-resistant pseudo random hash function. In addition, we also assume the Dlog problem on $\mathbb{G}_2$ is intractable.*

Then, the ring referral protocol $\Pi_{\mathsf{RR.bbs.m}}$ is complete, unforgeable, and has issuer Anonymity against exposed signature and message-hiding user Anonymity properties.

Proof. The security proofs follows the black box security reductions based on the relaxed soundness and SHVZK properties of $\Pi_{\mathsf{RR.bbs.m}}$ in a similar way as in the proofs of Theorem F.6 and Theorem F.3 (for the use of relaxed soundness property instead of CWE). □

F.5. Security Proofs of $\Pi_{\mathsf{RR.bbs.th}}$

We show that as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.th}}$, the protocol $\Pi_{\mathsf{RR.bbs.th}}$ has relaxed soundness and SHVZK properties. The ring referral security properties of $\Pi_{\mathsf{RR.bbs.th}}$ then follows from arguments similar to the previous proofs in this appendix.

Lemma F.10 (Relaxed Soundness of $\Pi_{\mathsf{RR.bbs.th}}$)**.** *Assume that in the setting of the threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$, AFGHO and Pedersen commitments are complete and computationally binding; the scalar and vector Dory arguments are complete and satisfies CWE; the Schnorr protocol for commitment checks is complete and satisfies knowledge soundness; and* Hash *is a collision-resistant hash function. Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.th}}$, $\Pi_{\mathsf{RR.bbs.th}}$ satisfies relaxed soundness property with respect to $H_1 \in \mathbb{G}_1$ and $G_2 \in \mathbb{G}_2$.*

Proof. We note that that the sub-protocol $\Pi_{\mathsf{bbs.pms.th}}$ is similar to batched version of k protocol $\Pi_{\mathsf{bbs.pms}}$ for single-message signature, and the part just before the final verification check in the main protocol $\Pi_{RR.bbs.th}$ is basically a Schnorr protocol. Therefore, we can build a polynomial time extractor algorithm $\mathtt{Ext}_{\mathsf{RR.bbs.th}}$ in a similar way as in the proofs of Lemma F.1 and Lemma F.4, with the exception that by using the CWE property of the $(k+1)$-dimensional recursive Dory arguments in $\Pi_{\mathsf{bbs.pms.th}}$, one can extract only the exponentiation vectors $\vec{H}_1' = \vec{H}_1^{\circ\vec{\mathsf{m}}'} \in \mathbb{G}_1^{k+1}$ and $\vec{G}_2' = \vec{G}_2^{\circ\vec{\mathsf{m}}'} \in \mathbb{G}_2^{k+1}$, not the vectors $\vec{\mathsf{m}}$ and $\vec{\sigma}_1$ themself. □

Lemma F.11 (SHVZK of $\Pi_{\mathsf{RR.bbs.th}}$)**.** *Assume that in the setting of the threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$ for BBS signature, AFGHO and Pedersen commitments are complete and perfectly hiding, the scalar and vector Dory arguments and Schnorr protocols for commitment checks are complete and have SHVZK property, and* Hash *is a collision-resistant pseudo random hash function.*

Then, as a multi-round interactive argument of knowledge for the relation $\mathscr{R}_{\mathsf{RR.bbs.th}}$, $\Pi_{\mathsf{RR.bbs.th}}$ satisfies SHVZK.

Proof. We can construct a polynomial time SHVZK simulator algorithm $\mathtt{Sim}_{\mathsf{RR.bbs.th}}$ by (i) constructing a SHVZK simulator for the sub-protocol $\Pi_{\mathsf{bbs.pms.th}}$ in a similar way as in the proof of the SHVZK property of $\mathtt{Sig}_{\mathsf{bbs.c}}$ in Lemma F.2; (ii) then simulating the remaining part in the main protocol of $\Pi_{\mathsf{RR.bbs.th}}$ using the SHVZK of the recursive Dory arguments and Schnorr protocol. □

Theorem F.12 (Security of the ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$)**.** *Assume in the setting of the threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$ (Fig. 11), the single-message BBS signature scheme $\mathtt{Sig}_{\mathsf{bbs}}$ is correct and EU-CMA; AFGHO and Pedersen commitment schemes are complete, computationally binding and perfectly hiding; the scalar and vector Dory arguments are complete and have CWE and SHVZK properties; the Schnorr protocol for commitment checks is complete, and has knowledge soundness and SHVZK properties, and* Hash *is a collision-resistant pseudo random hash function. In addition, we also assume the Dlog problem on $\mathbb{G}_1$ and $\mathbb{G}_2$ is intractable.*

Then, the threshold ring referral protocol $\Pi_{\mathsf{RR.bbs.th}}$ is complete, unforgeable, and has issuer Anonymity against exposed signature and message-hiding user Anonymity properties.

Proof. The security proofs follows the security reductions based on the relaxed soundness and SHVZK properties of $\Pi_{\mathsf{RR.bbs.th}}$ in a similar way as in the proofs of Theorem F.6 and Theorem F.3. Here, we use the intractability of Dlog problem on $\mathbb{G}_1$ and $\mathbb{G}_2$ for deducing the unforgeability property of $\Pi_{\mathsf{RR.bbs.th}}$ from the relaxed soundness property instead of CWE. □