Short Pairing-Free Blind Signatures with Exponential Security

Stefano Tessaro and Chenzhi Zhu

Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
{tessaro,zhucz20}@cs.washington.edu

Abstract. This paper proposes the first practical pairing-free three-move blind signature schemes that (1) are concurrently secure, (2) produce short signatures (i.e., *three* or *four* group elements/scalars), and (3) are provably secure either in the generic group model (GGM) or the algebraic group model (AGM) under the (plain or one-more) discrete logarithm assumption (beyond additionally assuming random oracles). We also propose a partially blind version of one of our schemes.

Our schemes do not rely on the hardness of the ROS problem (which can be broken in polynomial time) or of the mROS problem (which admits sub-exponential attacks). The only prior work with these properties is Abe's signature scheme (EUROCRYPT '02), which was recently proved to be secure in the AGM by Kastner et al. (PKC '22), but which also produces signatures twice as long as those from our scheme.

The core of our proofs of security is a new problem, called *weighted fractional* ROS (WFROS), for which we prove (unconditional) exponential lower bounds.

1 Introduction

Blind signatures [Cha81] allow a *user* to interact with a *signer* to produce a valid signature that cannot be linked back by the signer to the interaction that produced it. Blind signatures are used in several applications, such as e-cash systems [Cha81, CFN90], anonymous credentials (e.g., [CL04]), privacy-preserving ad-click measurement [PCM], and various forms of anonymous tokens [HIP+21, Tru]. They are also covered by an RFC draft [DJW21].

This paper develops the first practical pairing-free three-move blind signature schemes that (1) are concurrently secure, (2) produce short signatures (i.e., *three* or *four* group elements/scalars), and (3) are provably secure either in the *generic group model* (GGM) [Sho97, Mau05] or in the *algebraic group model* (AGM) [FKL18] under the discrete logarithm (DL) or the one-more discrete logarithm (OMDL) assumption (in addition to assuming *random oracles* [BR93]). Our DL-based scheme also admits a *partially blind* version [AF96], roughly following a paradigm by Abe and Okamoto [AO00], that targets applications where signatures need to depend on some public input (e.g., an issuing date) known to the signer. An overview of our schemes is given in Table 1.

Unlike blind Schnorr [CP93], Okamoto-Schnorr [PS00], and other other generic constructions based on identification schemes [HKL19], we do not rely on the hardness of the ROS problem, for which a polynomial-time attack has recently been presented [BLL+21]. Also, unlike Clause Blind Schnorr (CBS) signatures [FPS20], we do not rely on the assumed hardness of the mROS problem, which is subject to (mildly) sub-exponential attacks and we can thus support smaller group sizes.[1] In fact, our schemes all admit tight bounds, and this suggests that they can achieve $(\lambda/2)$-bit of security on λ-bit elliptic curves, supporting an instantiation with 256-bit curves. Our security proofs rely on a reduction to a new variant of the ROS problem, called *weighted fractional* ROS (WFROS), for which we prove an exponential, unconditional lower bound. Therefore, another benefit over CBS, beyond concrete parameters, is that we do not need to rely on an additional assumption.

[1] The best known attack against mROS [FPS20] runs in time $2^{\ell+\log(\ell+1)+\lambda/(1+\log(\ell+1))}$, where λ is the security parameter and ℓ corresponds to the number of concurrent sessions. The worst ℓ gives a $2^{O(\lambda/\log\lambda)}$ attack, and in practice, this suggests a choice of $\lambda = 512$ to achieve 128-bit security for all ℓ's.

Scheme	PK size	Sig. size	Assumption	Communication
BS_1 (Section 4)	$1\ \mathbb{G}$	$3\ \mathbb{Z}_p$	GGM	$2\ \mathbb{G} + 3\ \mathbb{Z}_p$
BS_2 (Appendix C)	$1\ \mathbb{G}$	$4\ \mathbb{Z}_p$	OMDL	$2\ \mathbb{G} + 4\ \mathbb{Z}_p$
BS_3 (Section 5.1)	$2\ \mathbb{G}$	$4\ \mathbb{Z}_p$	DL	$2\ \mathbb{G} + 4\ \mathbb{Z}_p$
PBS (Section 6)	$1\ \mathbb{G}$	$4\ \mathbb{Z}_p$	DL	$2\ \mathbb{G} + 4\ \mathbb{Z}_p$
Blind Schnorr [FPS20]	$1\ \mathbb{G}$	$2\ \mathbb{Z}_p$	OMDL + ROS	$1\ \mathbb{G} + 2\ \mathbb{Z}_p$
Clause Blind Schnorr [FPS20]	$1\ \mathbb{G}$	$2\ \mathbb{Z}_p$	OMDL + mROS	$2\ \mathbb{G} + 4\ \mathbb{Z}_p$
Abe [Abe01, KLRX22]	$3\ \mathbb{G}$	$2\ \mathbb{G} + 6\ \mathbb{Z}_p$	DL	λ bits $+ 3\ \mathbb{G} + 6\ \mathbb{Z}_p$

Table 1. Overview of our results. The four schemes proposed in this paper compared to pairing-free schemes that admit GGM/AGM security proofs in the literature. All schemes are three-move and secure assuming the ROM; All schemes except BS_1 admit AGM security proofs; further $p = |\mathbb{G}|$. As in plain Schnorr signatures, most schemes allow replacing one element in $\mathbb{Z}_p$ with a group element in the signature. The ROS assumption can be broken in polynomial time unless the scheme is restricted to tolerate only a very small number of sessions. Also, the mROS assumption admits sub-exponential attacks, which require the choice of a larger order p over all schemes (roughly 512-bit for 128-bit security [FPS20]).

Perhaps as a testament of the unsatisfactory status of pairing-free schemes, the *only* other scheme known to achieve exponential, concurrent, security is Abe's scheme [Abe01]. Although its original (standard-model) proof was found to be flawed, proofs were then given both in the GGM [OA03] and the AGM [KLRX22], along with a proof for the restricted setting of sequential security [BL13]. Still, it produces longer signatures and public keys, and is overall less efficient. Also, it only offers computational blindness (under DDH), whereas our scheme provides perfect blindness.

Discrete-Logarithm based blind signatures. We stress that our focus here is making pairing-free schemes as practical and as secure as possible. Indeed, very simple pairing-based blind signature schemes in the ROM can be obtained from BLS signatures [BLS01, Bol03]. Blind BLS offers a different trade-off: signatures are short (i.e., one group element) and signing requires only *two* moves, but signature verification requires a more expensive (and more complex) pairing evaluation. Indeed, the current blind signature RFC draft [DJW21] favors RSA over BLS, also due to lesser availability of pairings implementations. In particular, several envisioned applications of blind signatures are inherently browser-based, and the available cryptographic libraries (e.g., NSS for Firefox and BoringSSL for Chrome) do not yet offer pairing-friendly curve implementations.

In contrast, (non-blind) Schnorr signatures [Sch90, Sch91] (such as EdDSA [BDL+12]) are short, can rely on standard libraries, and outperform RSA. Though their blind evaluation requires three rounds, this may be less concerning in applications where verification cost is the dominating factor and the signing application can easily keep state. Indeed, [DJW21] identifies CBS as the only plausible alternative to RSA, and our schemes improve upon CBS by avoiding the mROS assumption. Once the group order is adjusted to resist sub-exponential attacks, we achieve comparable signature size, more efficient signing, and accommodate for partial blindness. (No partially blind version of CBS is known to the best of our knowledge.)

Finally, note that it is easier to prove security of pairing-free schemes under sequential access to the signer. For example, Kastner et al. [KLRX22] prove that plain blind Schnorr signatures are secure in this case, in the AGM, assuming the hardness of OMDL. Also, Baldimtsi and Lysyanskaya [BL13] (implicitly) prove sequential security of Abe's scheme. However, many applications, like PCM, easily enable concurrent attacks.

On ideal models. The use of the AGM or the GGM, along with the ROM, still appears necessary for the most practical pairing-free schemes with concurrent security. As of now, solutions solely assuming the ROM can only handle bounded concurrency [HKL19] or, alternatively, their communication and computation costs grow with the number of signing sessions [KLR21, CAL22, WHL22].

A number of other schemes [GRS+11, BFPV13, GG14, FHS15, FHKS16, Gha17, KNYY21] partially or completely avoid ideal models, some of which are fairly practical. However, they do not yet appear suitable for at-scale deployment.

1.1 A Scheme in the GGM

Our simplest scheme only admits a proof in the generic-group model (GGM) but best illustrates our ideas, in particular, how we bypass ROS-style attacks. It is slightly less efficient than Schnorr signatures, i.e., a signature that consists of *three* scalars mod p (or alternatively, two scalars and a group element). Nonetheless, it has a very similar flavor (in particular, signature verification can be built on top of a suitable implementation of Schnorr signatures in a black-box way).

Preface: Blind Schnorr Signatures and ROS. Recall that we seek an interactive scheme (1) that is one-more unforgeable (i.e., no adversary should be able to generate $\ell + 1$ signatures by interacting only ℓ times with the signer), and (2) for which interaction can be blinded. It is helpful to illustrate the main technical barrier behind proving (1) for *interactive* Schnorr signatures. Recall that the verification key is $X = g^x$ for a generator g of a cyclic group $\mathbb{G}$ of prime order p, and a signing key x. The signer starts the session by sending $A = g^a$, for a random $a \in \mathbb{Z}_p$. Then, the user sends a challenge $c = H(A, m)$ for a hash function H and a message m to be signed. Finally, the signer responds with $s = a + c \cdot x$, and the signature is $\sigma = (c, s)$.

Let us now consider an adversary that obtains ℓ initial messages $A_1, \ldots, A_\ell$ from the signer, where $A_i = g^{a_i}$. By solving the so-called *ROS problem* [Sch01, HKL19, FPS20], the attacker can find $\ell + 1$ vectors $\vec{\alpha}_1, \ldots, \vec{\alpha}_{\ell+1} \in \mathbb{Z}_p^\ell$ and a vector $(c_1, \ldots, c_\ell) \in \mathbb{Z}_p^\ell$ such that

$$\sum_{j=1}^{\ell} \alpha_i^{(j)} \cdot c_j = c_i^* \tag{1}$$

for all $i \in [\ell + 1]$, where $c_i^* = H(\prod_{j=1}^{\ell} A_j^{\alpha_i^{(j)}}, m_i^*)$, for some message $m_i^* \in \{0,1\}^*$. (Here, $\alpha_i^{(j)}$ is the j-th component of $\vec{\alpha}_i$.) Then, the attacker can obtain $s_j = a_j + c_j x$ from the signer for all $j \in [\ell]$ by completing the ℓ signing sessions. It is now easy to verify that (c_i^*, s_i^*) is a valid signature for m_i^* for all $i \in [\ell+1]$, where $s_i^* = \sum_{j=1}^{\ell} \alpha_i^{(j)} \cdot s_j$. Benhamouda et al. [BLL+21] recently gave a simple polynomial-time algorithm to solve the ROS problem for the case $\ell > \log(p)$, which thus breaks one-more unforgeability.[2]

Fuchsbauer et al. [FPS20] propose a different interactive signing process for Schnorr signatures that is one-more unforgeable (in the AGM + ROM) assuming that a variant of the ROS problem, called mROS, is hard. The mROS problem, however, admits sub-exponential attacks, and as it gives approximately only 70 bits of security from an implementation on a 256-bit curve, it effectively forces the use of 512-bit curves.[3]

Our first scheme. We take a different path which completely avoids the ROS and mROS problems to obtain our first scheme, BS_1. Again, we present a non-blind version – the scheme can be made blind via fairly standard tricks, as we explain in the body of the paper below. Again, the public key is $X = g^x$ for a secret key x. Then, the signer and the user engage in the following protocol to sign $m \in \{0,1\}^*$:

1. The signer sends $A = g^a$ and $Y = X^y$ for random $a, y \in \mathbb{Z}_p$.
2. The user responds with $c = H(A, Y, m)$
3. The signer returns a pair (s, y), where $s = a + cxy$.
4. The user accepts the signature $\sigma = (c, s, y)$ iff $g^s = A \cdot Y^c$ and $Y = X^y$.

Verification simply checks that $H(g^s X^{-yc}, X^y, M) = c$. In particular, note that (c, s) is a valid Schnorr signature with respect to the public-key X^y – this can be leveraged to implement the verification algorithm on top of an existing implementation of basic Schnorr signatures that also hash the public key (EdDSA does exactly this).[4] Further, as in Schnorr signatures, we could replace c with A in σ, and our results would be unaffected.

[2] Many envisioned implementations allow for $\ell > \log(p)$. Still, is worth noting that the scheme retains some security for $\ell < \log(p)$ even in the standard model [HKL19].

[3] mROS depends on a parameter ℓ, with a similar role as in ROS – sub-exponential attacks require $\ell < \log(p)$, but a one-more unforgeability attack for a small ℓ implies one for any $\ell' > \ell$ simply by generating $(\ell' - \ell)$ additional valid signatures.

[4] Note that this only superficially resembles key-blinding for Schnorr signatures [Hop13]. Here, the "blinding" y is actually public and part of the signature.

SECURITY INTUITION. To gather initial insights about the security of BS_1, it is instructive to *attempt* an ROS-style attack. The attacker opens ℓ sessions and obtains pairs $(A_1, Y_1), \ldots, (A_\ell, Y_\ell)$, where $A_i = g^{a_i}$ and $Y_i = X^{y_i} = g^{xy_i}$ for all $i \in [\ell]$. One natural extension of the ROS attack is to find $\ell + 1$ vectors $\vec{\alpha}_i \in \mathbb{Z}_p^\ell$ along with messages $m_1^*, m_2^*, \ldots \in \{0,1\}^*$ such that

$$c_i^* = H\left(\prod_{j=1}^{\ell} A_j^{\alpha_i^{(j)}}, \prod_{j=1}^{\ell} Y_j^{\alpha_i^{(j)}}, m_i^*\right)$$

for all $i \in [\ell+1]$ and then find $(c_1, \ldots, c_\ell) \in \mathbb{Z}_p^\ell$ such that

$$\sum_{j=1}^{\ell} \alpha_i^{(j)} \cdot y_j \cdot c_j = c_i^* \cdot \sum_{j=1}^{\ell} \alpha_i^{(j)} \cdot y_j \ , \tag{2}$$

for all $i \in [\ell+1]$. Indeed, if this succeeded, the adversary could complete the ℓ sessions to learn (s_j, y_j) by inputting c_j, where y_j is random and $s_j = a_j + c_j \cdot x \cdot y_j$. One could generate $\ell + 1$ signatures (c_i^*, s_i^*, y_i^*) for $i \in [\ell+1]$ by setting $s_i^* = \sum_{j=1}^{\ell} \alpha_i^{(j)} s_j$ and $y_i^* = \sum_{j=1}^{\ell} \alpha_i^{(j)} \cdot y_j$. These would be valid because

$$\begin{aligned} g^{s_i^*} &= g^{\sum_{j=1}^{\ell} \alpha_i^{(j)}(a_j + c_j x y_j)} \\ &= \prod_{j=1}^{\ell} A_j^{\alpha_i^{(j)}} \cdot X^{\sum_{j=1}^{\ell} \alpha_i^{(j)} c_j y_j} \overset{(2)}{=} \prod_{j=1}^{\ell} A_j^{\alpha_i^{(j)}} \cdot \left(\prod_{j=1}^{\ell} Y_j^{\alpha_i^{(j)}}\right)^{c_i^*} . \end{aligned}$$

However, finding $(c_1, \ldots, c_\ell)$ that satisfy (2) for $\ell + 1$ i's simultaneously is *much harder* than ROS. An initial intuition here is that X^y *completely hides* y to the point where y is revealed later in the session, where it appears like a random and fresh weight in the sum, *independent of* c_i. This intuition is however not correct, as an attacker can use the group element X^y and can try to gain information about y, but our proof will show (among other things) that in the GGM no useful information is obtained about y, and y is (close to) uniform when it is later revealed.

THE WFROS PROBLEM. The above attack paradigm is in fact generalized in terms of a new ROS-like problem that we call WFROS (this stands for *Weighted Fractional ROS*), for which we prove an unconditional lower bound. WFROS considers a game with two oracles that can be invoked adaptively in an interleaved way:

- The first oracle, H, accepts as input a pair of vectors $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2\ell+1}$, which are then associated with a random $\delta \in \mathbb{Z}_p^*$.
- The second oracle, S, allows to bind, for some $i \in [\ell]$, chosen input $c_i \in \mathbb{Z}_p$ with a random *weight* $y_i \in \mathbb{Z}_p^*$. During the course of the game, this latter oracle must be called *exactly* once for each $i \in [\ell]$.

The adversary finally commits to a subset of $\ell + 1$ prior H queries and wins if for each query in the subset, which has defined a pair of vector $\vec{\alpha}, \vec{\beta}$ and returned δ, we have $A/B = \delta$, where

$$A = \alpha^{(0)} + \sum_{i \in [\ell]} y_i(\alpha^{(2i-1)} + c_i \cdot \alpha^{(2i)}) \ , \quad B = \beta^{(0)} + \sum_{i \in [\ell]} y_i(\beta^{(2i-1)} + c_i \cdot \beta^{(2i)}) \ .$$

Here, $v^{(i)}$ denotes the i-th component of vector $\vec{v}$. Our main result (Theorem 1) says that no adversary making Q_{H} queries to H can win this game with probability better than $(Q_{\mathrm{H}}^2 + 2\ell Q_{\mathrm{H}})/(p-1)$, or, in other words, $Q_{\mathrm{H}} \geqslant \min\{\sqrt{p}, p/\ell\}$ is needed to win with constant probability. Note that $\ell \ll \sqrt{p}$ is generally true, as for our usage, ℓ is bounded by the number of signing sessions.

Our GGM proof for BS_1 transforms any generic attacker into one breaking the WFROS problem. This transformation is actually not immediate because a one-more unforgeability attacker can learn functions of the secret key x when obtaining the second message from the signer. A similar challenge occurs in proving hardness of the OMDL problem in the GGM, which was recently resolved by Bauer et al. [BFP21], and we rely on their techniques.

1.2 AGM Security and Partial Blindness

The Algebraic Group Model (AGM) [FKL18] can begin seen as a weaker idealization than the GGM. In particular, AGM proofs deal with actual groups (as opposed to representing group elements with random labels) and proceed via *reductions* that apply only to "algebraic adversaries", which provide representation of the group elements they output to the reduction. AGM has become a very popular model for validating security of a number of practical group-based protocols.

The main barrier to proving one-more unforgeability of BS_1 in the AGM is that the representation of X^y could leak some information about y that would not be available in the GGM, and thus we would not be able to apply our argument showing that y is still (close to) random looking when it is later revealed – our reduction in the GGM security proof crucially relies on this. To overcome this issue, for the two schemes BS_2 and BS_3, we replace X^y with a *hiding* commitment to y. In particular, we propose two different ways of achieving this:

Scheme BS_2. Here, X^y is replaced by $g^t X^y$. Later, the signer responds to challenge c with (s, y, t), where $s = a + c \cdot y \cdot x$. A signature is $\sigma = (c, s, y, t)$.

Scheme BS_3. Here, $g^t X^y$ is replaced by $g^t Z^y$, where Z is an extra random group element included in the verification key.

We consider BS_2 mostly for pedagogical reasons. Indeed, we can prove security of BS_3 in the AGM based *solely* on the discrete logarithm problem (DL). In contrast, BS_2 relies on the hardness of the (stronger) *one-more* DL problem (OMDL) [BNPS03], which asks for the hardness of breaking $\ell + 1$ DL instances given access to an oracle that can solve at most ℓ (adaptively chosen) DL instances. While we know that OMDL is generally not easier than DL [BFP21], a prudent instantiation may prefer relying on the (non-interactive) DL problem. While BS_3 requires a longer key, one could mitigate this by obtaining Z as the output of a hash function (assumed to be a random oracle) evaluated on some public input.

The proof of security for both schemes consists of showing that any adversary breaking one-more unforgeability can be transformed into one breaking either OMDL or DL (depending on the scheme) *or* into one breaking the WFROS problem. For the latter, however, we can resort to our unconditional hardness lower bound (Theorem 1).

Adding Partial Blindness. Finally, we note that it is not too hard to add partial blindness to BS_3, which is another reason to consider this scheme. In particular, to obtain the resulting PBS scheme, we can adopt a framework by Abe and Okamoto [AO00]. The main idea is simply to use a hash function (modeled as a random oracle) to generate the extra group element Z in a way that is dependent on a public input upon which the signature depends. We target in particular a stronger notion of one-more unforgeability, which shows that if the protocol is run ℓ times for a public input, then no $\ell+1$ signatures can be generated for that public input regardless of how many signatures have been generated for *different* public inputs. We defer more detail to Section 6.

Outline of the Paper

Section 2 will introduce some basic preliminaries. Section 3 will then introduce the WFROS problem, and prove a lower bound for it. We will then discuss our GGM-based scheme in Section 4, whereas variants secure in the AGM are presented in Section 5. Finally, we give a partially blind instantiation of our AGM scheme in Section 6.

2 Preliminaries

Notation. For positive integer n, we write $[n]$ for $\{1, \ldots, n\}$. We use λ to denote the security parameter. We use $\mathbb{G}$ to denote an (asymptotic) family of cyclic groups $\mathbb{G} := \{\mathbb{G}_\lambda\}_{\lambda>0}$, where $|\mathbb{G}_\lambda| > 2^\lambda$. We use $g(\mathbb{G}_\lambda)$ to denote the generator of $\mathbb{G}_\lambda$, and we will work over prime-order groups. We tacitly assume standard group

Game $\mathrm{OMUF}_{\mathsf{BS}}^{\mathcal{A}}(\lambda)$:	Oracle S_1 :
$par \leftarrow \mathsf{BS.Setup}(1^\lambda)$	$\mathrm{sid} \leftarrow \mathrm{sid} + 1$
$(sk, pk) \leftarrow \mathsf{BS.KG}(par)$	$(\mathsf{st}^s_{\mathrm{sid}}, \mathsf{msg}_1) \leftarrow \mathsf{BS.S}_1(sk)$
$\mathrm{sid} \leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{I}_{\mathrm{fin}} \leftarrow \varnothing$	Return $(\mathrm{sid}, \mathsf{msg}_1)$
$\{(m_k^*, \sigma_k^*)\}_{k \in [\ell+1]} \leftarrow\$ \mathcal{A}^{\mathrm{S}_1, \mathrm{S}_2}(pk)$	Oracle $\mathrm{S}_2(i, c_i)$:
If $\exists\, k_1 \neq k_2$ such that $(m_{k_1}^*, \sigma_{k_1}) = (m_{k_2}^*, \sigma_{k_2})$	If $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
then return 0	$\mathsf{msg}_2 \leftarrow \mathsf{BS.S}_2(\mathsf{st}^s_i, c_i)$
If $\exists\, k \in [\ell+1]$ such that $\mathsf{BS.Ver}(pk, \sigma_k, m_k^*) = 0$	$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
then return 0	$\ell \leftarrow \ell + 1$
Return 1	Return msg_2

Fig. 1. The OMUF security game for a blind signature scheme BS.

operations can be performed in time polynomial in λ in $\mathbb{G}_\lambda$ and adopt multiplicative notation. We will often compute over the finite field $\mathbb{Z}_p$ (for a prime p) – we usually do not write modular reduction explicitly when it is clear from the context. We write $\mathbb{Z}_p^* = \mathbb{Z}_p \backslash \{0\}$. We often need to consider vectors $\vec{\alpha} \in \mathbb{Z}_p^\ell$ and usually refer to the i-th component of $\vec{\alpha}$ as $\alpha^{(i)} \in \mathbb{Z}_p$.

Blind signatures. This paper focuses on *three-move* blind signature schemes, and our notation is similar to that of prior works (e.g., [HKL19, FPS20]). Formally, a (three-move) *blind signature scheme* BS is a tuple of efficient (randomized) algorithms

$$\mathsf{BS} = (\mathsf{BS.Setup}, \mathsf{BS.KG}, \mathsf{BS.S}_1, \mathsf{BS.S}_2, \mathsf{BS.U}_1, \mathsf{BS.U}_2, \mathsf{BS.Ver}) ,$$

with the following behavior:

- The *parameter generation* algorithm $\mathsf{BS.Setup}(1^\lambda)$ outputs a string of parameters par, whereas the *key generation* algorithm $\mathsf{BS.KG}(par)$ outputs a key-pair (sk, pk), where sk is the *secret* (or *signing*) key and pk is the *public* (or *verification*) key.
- The interaction between the user and the signer to sign a message $m \in \{0,1\}^*$ with key-pair (pk, sk) is defined by the following experiment:

$$\begin{aligned} (\mathsf{st}^s, \mathsf{msg}_1) \leftarrow \mathsf{BS.S}_1(sk) \ , \ \ (\mathsf{st}^u, \mathsf{chl}) \leftarrow \mathsf{BS.U}_1(pk, \mathsf{msg}_1, m) \ , \\ \mathsf{msg}_2 \leftarrow \mathsf{BS.S}_2(\mathsf{st}^s, \mathsf{chl}) \ , \ \ \sigma \leftarrow \mathsf{BS.U}_2(\mathsf{st}^u, \mathsf{msg}_2) \ . \end{aligned} \tag{3}$$

  Here, σ is either the resulting *signature* or an *error message* $\perp$.
- The (deterministic) *verification algorithm* outputs a bit $\mathsf{BS.Ver}(pk, \sigma, m)$.

We say that BS is (perfectly) *correct* if for every message $m \in \{0,1\}^*$, with probability one over the sampling of parameters and the key pair (pk, sk), the experiment in (3) returns σ such that $\mathsf{BS.Ver}(pk, \sigma, m) = 1$. All of our schemes are going to be perfectly correct.

One-more unforgeability. The standard notion of security for blind signatures is *one-more unforgeability* (OMUF). OMUF ensures that no adversary playing the role of a user interacting with the signer ℓ times, in an arbitrarily concurrent fashion, can issue $\ell + 1$ signatures (or more, of course). The $\mathrm{OMUF}_{\mathsf{BS}}^{\mathcal{A}}$ game for a blind signature scheme BS is defined in Figure 1. The corresponding advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathsf{BS}}^{\mathrm{omuf}}(\mathcal{A}, \lambda) := \mathsf{Pr}[\mathrm{OMUF}_{\mathsf{BS}}^{\mathcal{A}}(\lambda) = 1]$. All of our analyses will further assume one or more random oracles, which are modeled as an additional oracle to which the adversary $\mathcal{A}$ is given access.

Blindness. We also consider the standard notion of blindness against a malicious server that can, in particular, attempt to publish a malformed public key. The corresponding game $\mathrm{Blind}_{\mathsf{BS}}^{\mathcal{A}}$ is defined in Figure 2, and for any adversary $\mathcal{A}$, we define its advantage as $\mathsf{Adv}_{\mathsf{BS}}^{\mathrm{blind}}(\mathcal{A}, \lambda) := \left|\mathsf{Pr}[\mathrm{Blind}_{\mathsf{BS}}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|$. We say the scheme is perfectly blind if and only if $\mathsf{Adv}_{\mathsf{BS}}^{\mathrm{blind}}(\mathcal{A}, \lambda) = 0$ for any $\mathcal{A}$ and all λ.

Game $\mathrm{Blind}_{\mathsf{BS}}^{\mathcal{A}}(\lambda)$:
$par \leftarrow \mathsf{BS.Setup}(1^\lambda)$
$b \leftarrow\$ \{0,1\}$; $b_0 \leftarrow b$; $b_1 \leftarrow 1-b$
$b' \leftarrow\$ \mathcal{A}^{\mathrm{INIT},\mathrm{U}_1,\mathrm{U}_2}(par)$
If $b' = b$ then return 1
Return 0
 	
Oracle $\mathrm{INIT}(\tilde{pk}, \tilde{m_0}, \tilde{m_1})$:
$\mathsf{sess}_0 \leftarrow \mathtt{init}$
$\mathsf{sess}_1 \leftarrow \mathtt{init}$
$pk \leftarrow \tilde{pk}$
$m_0 \leftarrow \tilde{m_0}$; $m_1 \leftarrow \tilde{m_1}$
	Oracle $\mathrm{U}_1(i, \mathsf{msg}_1^{(i)})$:
	If $i \notin \{0,1\}$ or $\mathsf{sess}_i \neq \mathtt{init}$ then return $\perp$
	$\mathsf{sess}_i \leftarrow \mathtt{open}$
	$(\mathsf{st}_i^u, \mathsf{chl}^{(i)}) \leftarrow \mathsf{BS.U}_1(pk, \mathsf{msg}_1^{(i)}, m_{b_i})$
	Return $\mathsf{chl}^{(i)}$
	
	Oracle $\mathrm{U}_2(i, \mathsf{msg}_2^{(i)})$:
	If $i \notin \{0,1\}$ or $\mathsf{sess}_i \neq \mathtt{open}$ then return $\perp$
	$\mathsf{sess}_i \leftarrow \mathtt{closed}$
	$\sigma_{b_i} \leftarrow \mathsf{BS.U}_2(\mathsf{st}_i^u, \mathsf{msg}_2^{(i)})$
	If $\mathsf{sess}_0 = \mathsf{sess}_1 = \mathtt{closed}$ then
	  If $\sigma_0 = \perp$ or $\sigma_1 = \perp$ then return $(\perp, \perp)$
	  Return (σ_0, σ_1)
	Return $(i, \mathtt{closed})$

Fig. 2. The Blind security game for a blind signature scheme BS.

Game $\mathrm{WFROS}_{\ell,p}^{\mathcal{A}}$:
$\mathrm{hid} \leftarrow 0$; $\mathcal{I}_{\mathrm{fin}} \leftarrow \varnothing$
$\mathcal{J} \leftarrow \mathcal{A}^{\mathrm{H},\mathrm{S}}(p)$
If $\mathcal{J} \nsubseteq [\mathrm{hid}]$ or $\lvert\mathcal{J}\rvert \leqslant \ell$ or $\mathcal{I}_{\mathrm{fin}} \neq [\ell]$ then
  Return 0
For each $j \in \mathcal{J}$,
  $A_j \leftarrow \alpha_j^{(0)} + \sum_{i \in [\ell]} y_i(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)})$
  $B_j \leftarrow \beta_j^{(0)} + \sum_{i \in [\ell]} y_i(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)})$
If $\forall j \in \mathcal{J} : (A_j = \delta_j B_j \ \wedge\ B_j \neq 0)$ then
  Return 1
Return 0
	Oracle $\mathrm{H}(\vec{\alpha}, \vec{\beta})$:
	$\mathrm{hid} \leftarrow \mathrm{hid} + 1$
	$\vec{\alpha}_{\mathrm{hid}} \leftarrow \vec{\alpha}$; $\vec{\beta}_{\mathrm{hid}} \leftarrow \vec{\beta}$
	$\delta_{\mathrm{hid}} \leftarrow\$ \mathbb{Z}_p^*$
	Return $\delta_{\mathrm{hid}}, \mathrm{hid}$
	
	Oracle $\mathrm{S}(i, c_i)$:
	If $i \notin [\ell] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
	$y_i \leftarrow\$ \mathbb{Z}_p^*$
	$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
	Return y_i

Fig. 3. The WFROS problem. Here, $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2\ell+1}$, which is indexed as $\vec{\alpha} = (\alpha^{(0)}, \ldots, \alpha^{(2\ell)})$ and $\vec{\beta} = (\beta^{(0)}, \ldots, \beta^{(2\ell)})$.

Game-playing proofs. Several of our proofs adopt a lightweight variant of the standard "Game-Playing Framework" by Bellare and Rogaway [BR06].

3 The Weighted Fractional ROS Problem

This section introduces and analyzes an unconditionally hard problem underlying all of our proofs, which we call the *Weighted Fractional ROS* problem (WFROS). It is a variant of the original ROS problem [Sch01, HKL19, FPS20], which, in turn, stands for *Random inhomogeneities in a Overdetermined Solvable system of linear equations*. While ROS can be solved in polynomial time [BLL+21] and its mROS variant can be solved in sub-exponential time [FPS20], we are going to prove an *exponential* lower bound for WFROS.

The WFROS problem. The problem is defined via the game $\mathrm{WFROS}_{\ell,p}^{\mathcal{A}}$, described in Figure 3, which involves an adversary $\mathcal{A}$ and depends on two integer parameters ℓ and p, where p is a prime. The adversary here interacts with two oracles, H and S. The first oracle allows the adversary to link a vector pair $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2\ell+1}$ with a random inhomogeneous part $\delta \in \mathbb{Z}_p^*$ – each such query defines implicitly an equation $A/B = \delta$ in the unknowns $\mathsf{C}_1, \ldots, \mathsf{C}_\ell$ and $\mathsf{Y}_1, \ldots, \mathsf{Y}_\ell$. A call to $\mathrm{S}(i, c_i)$ lets us set the value of C_i to c_i and set Y_i to a random value y_i. The second oracle $\mathrm{S}(i, \cdot)$ must be called once for every $i \in [\ell]$. It is noteworthy to stress that the c_i's can be chosen arbitrarily, whereas the corresponding y_i's are random and independent.

In the end, the adversary wins the game if a subset of $\ell+1$ equations defined by the H queries is satisfied by the assignment defined by querying S. In particular, we define

$$\mathsf{Adv}^{\mathrm{wfros}}_{\ell,p}(\mathcal{A}) = \mathsf{Pr}\left[\mathrm{WFROS}^{\mathcal{A}}_{\ell,p} = 1\right] . \tag{4}$$

Note that it would be possible to carry out some of the following security proofs using restricted versions of the WFROS game, but the above formulation lets us handle all schemes via a single notion.

A Lower Bound for WFROS. The following theorem, our main result on WFROS, shows that any adversary winning WFROS with constant probability requires $Q_H = \Omega(\min\{\sqrt{p}, p/\ell\})$ queries. (Also, note that all applications of interest assume $\ell \ll \sqrt{p}$.)

Theorem 1 (Lower bound for WFROS). *For any $\ell > 0$, any prime number p, and any adversary $\mathcal{A}$ playing the* $\mathrm{WFROS}^{\ell,p}$ *game that makes at most Q_{H} queries to* H*, we have*

$$\mathsf{Adv}^{\mathrm{wfros}}_{\ell,p}(\mathcal{A}) \leqslant \frac{Q_{\mathrm{H}}(2\ell + Q_{\mathrm{H}})}{p-1} .$$

The proof is given in the next section. To gain some very high-level intuition, we observe that a key contributor to the hardness of WFROS are values y_i, which are defined *after* the c_i's are fixed and hence randomize the A_j and B_j's. Therefore, to satisfy $A_j = \delta_j \cdot B_j$, the adversary is restricted in the way it plays. For example, to satisfy an equation defined by an H query $(\vec{\alpha}_j, \vec{\beta}_j)$, the adversary can pick c_i's such that $(\alpha_j^{(2i-1)} + c_i\alpha_i^{(2j)}) = \delta_j \cdot (\beta_j^{(2i-1)} + c_i\beta_j^{(2i)})$ for all $i \in [\ell]$. Then, the equation $A_j = \delta_j B_j$ is satisfied no matter what the y_i's are. Our proof shows that the adversary *has* to pick c_i's this way – and in fact, it has to follow even more restrictions. Finally, we show that under these restrictions, no set of $\ell+1$ equations can be satisfied simultaneously.

3.1 Proof of Theorem 1

Let $\mathcal{A}$ be an adversary for the WFROS game that makes at most Q_{H} queries to H. Without loss of generality, we assume that $\mathcal{A}$ makes exactly one query (i, c_i) to S for each $i \in [\ell]$ and that $\mathcal{A}$ always outputs $\mathcal{J} \subseteq [Q_{\mathrm{H}}]$.

In the $\mathrm{WFROS}^{\mathcal{A}}_{\ell,p}$ game, for each $j \in [Q_{\mathrm{H}}]$, denote the event W_j as

$$\alpha_j^{(0)} + \sum_{i\in[\ell]} y_i(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)}) = \delta_j \left(\beta_j^{(0)} + \sum_{i\in[\ell]} y_i(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)})\right) \tag{W1}$$

$$\wedge\ \beta_j^{(0)} + \sum_{i\in[\ell]} y_i(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}) \neq 0 . \tag{W2}$$

In other words, W_j is the event that the equation defined by the j-th H query is satisfied. Then, $\mathcal{A}$ wins if and only if $|\mathcal{J}| > \ell$ and W_j occur for each $j \in \mathcal{J}$. Denote $W := (|\mathcal{J}| > \ell) \ \wedge\ \left(\bigwedge_{j\in\mathcal{J}} W_j\right)$ and we have $\mathsf{Adv}^{\mathrm{wfros}}_{\ell,p}(\mathcal{A}) = \mathsf{Pr}[W]$.

To bound $\mathsf{Pr}[W]$, we need notation to refer to some values (formally, random variables) defined in the execution of the $\mathrm{WFROS}^{\mathcal{A}}_{\ell,p}$ game. First, denote as $\mathcal{I}^{(j)}_{\mathrm{fin}}$ the contents of the set $\mathcal{I}_{\mathrm{fin}}$ when the adversary makes the j-th query to H, and let $(\vec{\alpha}_j, \vec{\beta}_j)$ be the input of this query to H, which is answered with δ_j. Also, let $\mathcal{I}^{(j)}_{\mathrm{unk}} := [\ell] \backslash \mathcal{I}^{(j)}_{\mathrm{fin}}$, i.e., the set of indices $i \in [\ell]$ for which $\mathcal{A}$ has not yet made any query $(i, \cdot)$ to S when the j-th query to H is made. Further, $c_1, \ldots, c_\ell$ and $y_1, \ldots, y_\ell$ are the values defined by querying S.

Now, for each $j \in [Q_{\mathrm{H}}]$, we define the following events:

Event $E_j^{(1)}$. First, let $E_{1,j}^{(1)}$ be the event that $\beta_j^{(0)} + \sum_{i\in\mathcal{I}^{(j)}_{\mathrm{fin}}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) \neq 0$. For each $i \in \mathcal{I}^{(j)}_{\mathrm{unk}}$, also let $E_{2,(j,i)}^{(1)}$ be the event that $\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} \neq \delta_j \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right)$. Finally, let $E_j^{(1)} := E_{1,j}^{(1)} \ \vee\ \left(\bigvee_{i\in[\mathcal{I}^{(j)}_{\mathrm{unk}}]} E_{2,(j,i)}^{(1)}\right)$.

Event $E_j^{(2)}$. We denote the event $E_j^{(2)}$ as the event where

$$\forall\, i \in \mathcal{I}_{\text{unk}}^{(j)} : \alpha_j^{(2i)} \cdot \beta_j^{(2i-1)} = \alpha_j^{(2i-1)} \cdot \beta_j^{(2i)} \ . \tag{5}$$

Note that events $E_j^{(1)}$ and $E_j^{(2)}$ are, by themselves, not necessarily unlikely – the adversary can certainly provoke them. However, we intend to show that this has implications on the ability to satisfy the j-th equation. In particular, we prove the following two lemmas in Sections 3.2 and 3.3 below, respectively.

Lemma 1. $\mathsf{Pr}[W_j \ \wedge \ E_j^{(1)}] \leqslant \frac{\ell+1}{p-1}$.

Lemma 2. $\mathsf{Pr}[W_j \ \wedge \ (\neg E_j^{(1)}) \ \wedge \ E_j^{(2)}] \leqslant \frac{\ell}{p-1}$.

Now, if we denote $E^{(1)} := \bigvee_{j\in[Q_{\mathrm{H}}]}(W_j \ \wedge \ E_j^{(1)})$ and $E^{(2)} := \bigvee_{j\in[Q_{\mathrm{H}}]}(W_j \ \wedge \ (\neg E_j^{(1)}) \ \wedge \ E_j^{(2)})$, the union bound yields $\mathsf{Pr}[E^{(1)}] \leqslant \frac{Q_{\mathrm{H}}(\ell+1)}{p-1}$ and $\mathsf{Pr}[E^{(2)}] \leqslant \frac{Q_{\mathrm{H}}\cdot\ell}{p-1}$. Our final lemma (proved in Section 3.4) is then the following:

Lemma 3. $\mathsf{Pr}[W \ \wedge \ (\neg E^{(1)}) \ \wedge \ (\neg E^{(2)})] \leqslant \frac{Q_{\mathrm{H}}(Q_{\mathrm{H}}-1)}{p-1}$.

The three lemmas can be combined to obtain

$$\mathsf{Pr}[W] \leqslant \mathsf{Pr}[E^{(1)}] + \mathsf{Pr}[E^{(2)}] + \mathsf{Pr}[W \ \wedge \ (\neg E^{(1)}) \ \wedge \ (\neg E^{(2)})] \leqslant \frac{Q_{\mathrm{H}}(2\ell + Q_{\mathrm{H}})}{p-1} \ .$$

which concludes the proof. In the next three sections, we prove the three perceding lemmas.

3.2 Proof of Lemma 1

Throughout this proof, let us fix $j \in [Q_{\mathrm{H}}]$. We first define a sequence of random variables $(D_0, D_1, \ldots, D_n, X_1, \ldots, X_n)$, where $n = \ell + 1$, such that $E_j^{(1)}$ implies one of $D_0, \ldots, D_n$ is not equal to 0 and $D_0 + \sum_{k\in[n]} D_k X_k = 0$. Further, we also ensure that X_k is uniformly distributed over $\mathbb{Z}_p^*$ independent of $(D_0, D_1, \ldots, D_k, X_1, \ldots, X_{k-1})$ for each $k \in [n]$ and use this to bound $\mathsf{Pr}[E_j^{(1)}]$. More concretely:

- Let
$$D_0 := \alpha_j^{(0)} + \sum_{i\in\mathcal{I}_{\text{fin}}^{(j)}} y_i \left(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i-1)}\right) \ ,$$
$$X_1 = -\delta_j \ , \ D_1 := \beta_j^{(0)} + \sum_{i\in\mathcal{I}_{\text{fin}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) \ ,$$
and note that $E_{1,j}^{(1)}$ is equivalent to $D_1 \neq 0$.
- Further, for $1 \leqslant k \leqslant |\mathcal{I}_{\text{unk}}^{(j)}|$, denote $i_k \in \mathcal{I}_{\text{unk}}^{(j)}$ as the index such that (i_k, c_{i_k}) is the k-th query made to S among the indexes in $\mathcal{I}_{\text{unk}}^{(j)}$ and let
$$X_{k+1} = y_{i_k} \ , \ D_{k+1} := \alpha_j^{(2i_k-1)} + c_{i_k} \cdot \alpha_j^{(2i_k)} - \delta_j \left(\beta_j^{(2i_k-1)} + c_i \cdot \beta_j^{(2i_k)}\right) \ ,$$
we have $E_{2,(j,i_k)}^{(1)}$ occurs is equivalent to $D_{k+1} \neq 0$.
- For $|\mathcal{I}_{\text{unk}}^{(j)}| + 1 < k \leqslant n$, let $D_k = 0$ and X_k be a random variable uniformly distributed in $\mathbb{Z}_p^*$ independent of $(D_0, D_1, \ldots, D_k, X_1, \ldots, X_{k-1})$. [5]

[5] For $|\mathcal{I}_{\text{unk}}^{(j)}| + 1 < k \leqslant n$, D_k, X_k act as placeholders so that we can apply Lemma 4 for an a priori fixed value n instead of a random variable $|\mathcal{I}_{\text{unk}}^{(j)}| + 1$.

Note that

$$
\begin{aligned}
D_0 + \sum_{k=1}^{N} D_k X_k = {} & \alpha_j^{(0)} + \sum_{i \in \mathcal{I}_{\text{fin}}^{(j)}} y_i \left(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} \right) \\
& - \delta_j \left(\beta_j^{(0)} + \sum_{i \in \mathcal{I}_{\text{fin}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) \right) \\
& + \sum_{i \in \mathcal{I}_{\text{unk}}^{(j)}} y_i \left(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} - \delta_j \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) \right) \\
= {} & \alpha_j^{(0)} + \sum_{i \in [\ell]} y_i \left(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} \right) \\
& - \delta_j \left(\beta_j^{(0)} + \sum_{i \in [\ell]} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) \right) .
\end{aligned}
$$

Therefore, by (W1), we know W_j occurs implies $D_0 + \sum_{i=1}^{n} D_i X_i = 0$. Thus, the event $W_j \wedge E_j^{(1)}$ implies, in addition, that one of $D_0, \ldots, D_n$ is not equal to 0. Also, we prove the following claim.

Claim 1 *For each $k \in [n]$, X_k is uniformly distributed over $\mathbb{Z}_p^*$ independent of $(D_0,\ldots,D_k,X_1,\ldots,X_{k-1})$.*

Proof (of Claim 1). For $k = 1$, we have $X_1 = -\delta_j$. Consider the step when δ_j is generated. Since $\mathcal{A}$ has made the j-th query to H, we know $\mathcal{I}_{\text{unk}}^{(j)}$, $\vec{\beta}_j$, $\vec{\alpha}_j$, and $\{y_i, c_i\}_{i \in \mathcal{I}_{\text{fin}}^{(j)}}$ are already determined, which implies D_0 and D_1 are also determined. Since δ_j is picked uniformly at random from $\mathbb{Z}_p^*$, we know $X_1 = -\delta_j$ is uniformly distributed over $\mathbb{Z}_p^*$ independent of (D_0, D_1).

For $2 \leqslant k \leqslant |\mathcal{I}_{\text{unk}}^{(j)}| + 1$, we have $X_k = y_{i_{k-1}}$. Consider the step when $y_{i_{k-1}}$ is generated. We know $\mathcal{A}$ has made the query $(i_{k-1}, c_{i_{k-1}})$ to S and the values i_{k-1}, $c_{i_{k-1}}$ are determined. Since $i_{k-1} \in \mathcal{I}_{\text{unk}}^{(j)}$, we know $\mathcal{A}$ has made the j-th query to H, and thus the values $\vec{\beta}_j$, $\vec{\alpha}_j$, δ_j, and (D_0, D_1) are determined. For $1 \leqslant k' < k-1$, since the query $(i_{k'}, c_{i_{k'}})$ to S has returned, we know the values $i_{k'}, c_{i_{k'}}, y_{i_{k'}}$ are determined, which implies $D_{k'+1}$ and $X_{k'+1}$ are determined. Also, since i_{k-1}, $c_{i_{k-1}}$ are determined, we know D_k is determined. Therefore, since $y_{i_{k-1}}$ is picked uniformly at random from $\mathbb{Z}_p^*$, we know $X_k = y_{i_{k-1}}$ is uniformly distributed over $\mathbb{Z}_p^*$ independent of $(D_0, \ldots, D_k, X_1, \ldots, X_{k-1})$.

For $|\mathcal{I}_{\text{unk}}^{(j)}| + 1 < k \leqslant n$, by the definition of X_k, we know X_k is uniformly distributed over $\mathbb{Z}_p^*$ independent of $(D_0, \ldots, D_k, X_1, \ldots, X_{k-1})$. Therefore, the claim holds. □

Now, we can show the upper bound $\Pr[W_j \wedge E_j^{(1)}] \leqslant \frac{\ell+1}{p-1}$ by the following lemma,[6] which we prove in Appendix A.

Lemma 4. *Let p be prime. Let $D_0, D_1, \ldots, D_n, X_1, \ldots, X_n \in \mathbb{Z}_p$ be random variables such that for all $k \in [n]$, X_k is uniform over $U_k \subseteq \mathbb{Z}_p$ and independent of $(D_0, \ldots, D_k, X_1, \ldots, X_{k-1})$. Then,*

$$
\Pr\left[\exists\, i \in \{0, \ldots, n\} \ : \ D_i \neq 0 \ \wedge \ D_0 + \sum_{j=1}^{n} D_j X_j = 0 \right] \leqslant \sum_{i=1}^{n} \frac{1}{|U_i|} .
$$

[6] Note that this lemma cannot be directly derived from the Schwartz-Zippel lemma by viewing $D_0 + \sum_{j=1}^{n} D_j X_j = 0$ as a polynomial of $X_1, \ldots, X_n$, since we cover for example the case where $D_0, D_1, \ldots, D_n$ are adaptively chosen, i.e., each D_i can depend on $X_1 \ldots, X_{i-1}$.

3.3 Proof of Lemma 2

It is easier to introduce a new event F_j and show that $W_j \wedge (\neg E_j^{(1)})$ implies F_j. We will then bound $\mathsf{Pr}[F_j \wedge E_j^{(2)}]$. In particular, define the event F_j as

$$\forall\, i \in \mathcal{I}_{\text{unk}}^{(j)} \;:\; \alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} - \delta_j \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = 0 \tag{F1}$$

$$\wedge \sum_{i \in \mathcal{I}_{\text{unk}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) \neq 0 \,, \tag{F2}$$

and we have the following lemma.

Lemma 5. *If $W_j \wedge (\neg E_j^{(1)})$ occurs, then the event F_j occurs.*

Proof (of Lemma 5). By the definition of F_j, we need only show that if $W_j \wedge (\neg E_j^{(1)})$ occurs, then (F1) and (F2) hold for j.

Suppose W_j occurs but $E_j^{(1)}$ does not occur. Since $E_j^{(1)} = E_{1,j}^{(1)} \vee \left(\bigvee_{i \in [\mathcal{I}_{\text{unk}}^{(j)}]} E_{2,(j,i)}^{(1)}\right)$, we know all of $E_{1,j}^{(1)}$ and $\{E_{2,(j,i)}^{(1)}\}_{i \in [\mathcal{I}_{\text{unk}}^{(j)}]}$ do not occur. Since the event $E_{2,(j,i)}^{(1)}$ does not occur implies

$$\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} - \delta_j \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = 0 \,,$$

we know (F1) holds for j.

Also, since the event $E_{1,j}^{(1)}$ does not occur, we have

$$\beta_j^{(0)} + \sum_{i \in \mathcal{I}_{\text{fin}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = 0.$$

Since W_j occurs, we know (W2) holds and, by the above equation, we have

$$\sum_{i \in \mathcal{I}_{\text{unk}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = \beta_j^{(0)} + \sum_{i \in [\ell]} y_i (\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}) \neq 0 \,.$$

Therefore, we know (F2) holds for j. □

We also denote

$$\mathcal{D}_j := \left\{ \frac{\alpha_j^{(2i)}}{\beta_j^{(2i)}} \mid i \in \mathcal{I}_{\text{unk}}^{(j)}, \beta_j^{(2i)} \neq 0 \right\} \cup \left\{ \frac{\alpha_j^{(2i-1)}}{\beta_j^{(2i-1)}} \mid i \in \mathcal{I}_{\text{unk}}^{(j)}, \beta_j^{(2i)} = 0, \beta_j^{(2i-1)} \neq 0 \right\}.$$

We have $|\mathcal{D}_j| \leqslant |\{i \in \mathcal{I}_{\text{unk}}^{(j)} \mid \beta_j^{(2i)} \neq 0\} \cup \{i \in \mathcal{I}_{\text{unk}}^{(j)} \mid \beta_j^{(2i)} = 0\}| = |\mathcal{I}_{\text{unk}}^{(j)}|$.

Claim 2 *The event $F_j \wedge E_j^{(2)}$ implies $\delta_j \in \mathcal{D}_j$.*

Proof (of Claim 2). Suppose $F_j \wedge E_j^{(2)}$ occurs but $\delta_j \notin \mathcal{D}_j$. We are going to show that $\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} = 0$ for each $i \in \mathcal{I}_{\text{unk}}^{(j)}$. Then, since F_j occurs, we know (F2) holds, which yields a contradiction, and thus the claim holds.

For $i \in \mathcal{I}_{\text{unk}}^{(j)}$, if $\beta_j^{(2i)} \neq 0$, since $\delta_j \notin \mathcal{D}_j$, we have $\delta_j \neq \frac{\alpha_j^{(2i)}}{\beta_j^{(2i)}}$, which implies $\alpha_j^{(2i)} - \delta_j \cdot \beta_j^{(2i)} \neq 0$. Since F_j occurs, by (F1), we have $c_i = -\frac{\alpha_j^{(2i-1)} - \delta_j \cdot \beta_j^{(2i-1)}}{\alpha_j^{(2i)} - \delta_j \cdot \beta_j^{(2i)}}$. Since $E_j^{(2)}$ occurs, by (5), we have $\alpha_j^{(2i)} \cdot \beta_j^{(2i-1)} = \alpha_j^{(2i-1)} \cdot \beta_j^{(2i)}$, and thus

$$\begin{aligned}
\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} &= \beta_j^{(2i-1)} - \frac{\alpha_j^{(2i-1)} \cdot \beta_j^{(2i)} - \delta_j \cdot \beta_j^{(2i-1)} \cdot \beta_j^{(2i)}}{\alpha_j^{(2i)} - \delta_j \cdot \beta_j^{(2i)}} \\
&= \beta_j^{(2i-1)} - \frac{\alpha_j^{(2i)} \cdot \beta_j^{(2i-1)} - \delta_j \cdot \beta_j^{(2i-1)} \cdot \beta_j^{(2i)}}{\alpha_j^{(2i)} - \delta_j \cdot \beta_j^{(2i)}} \\
&= \beta_j^{(2i-1)} - \beta_j^{(2i-1)} = 0 \ .
\end{aligned}$$

Otherwise, suppose $\beta_j^{(2i)} = 0$. Then, if $\beta_j^{(2i-1)} = 0$, we also have $\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} = 0$. If $\beta_j^{(2i-1)} \neq 0$, since $\alpha_j^{(2i)} \cdot \beta_j^{(2i-1)} = \alpha_j^{(2i-1)} \cdot \beta_j^{(2i)} = 0$, we have $\alpha_j^{(2i)} = 0$. Since $\beta_j^{(2i)} = 0$, $\beta_j^{(2i-1)} \neq 0$, and $\delta_j \notin \mathcal{D}_j$, we have $\delta_j \neq \frac{\alpha_j^{(2i-1)}}{\beta_j^{(2i-1)}}$ and thus we have

$$\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} - \delta_j \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = \alpha_j^{(2i-1)} - \delta_j \cdot \beta_j^{(2i-1)} \neq 0,$$

which contradicts (F1). Therefore, it is impossible that $\beta_j^{(2i)} = 0$ and $\beta_j^{(2i-1)} \neq 0$.

Therefore, from the above arguments, we have $\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} = 0$ for any $i \in \mathcal{I}_{\text{unk}}^{(j)}$, and thus $\sum_{i \in \mathcal{I}_{\text{unk}}^{(j)}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}\right) = 0$. However, since F_j occurs, we know (F2) holds, which yields a contradiction, and thus the claim holds. $\square$

Note that δ_j is generated uniformly at random, independently of $\mathcal{D}_j$, since the latter is defined by the j-th H query. Therefore, Lemma 5 and Claim 2 yield

$$\begin{aligned}
\Pr[W_j \ \wedge \ (\neg E_j^{(1)}) \ \wedge \ E_j^{(2)}] &\leqslant \Pr[F_j \ \wedge \ E_j^{(2)}] \\
&\leqslant \Pr[\delta_j \in \mathcal{D}_j] \leqslant \frac{|\mathcal{I}_{\text{unk}}^{(j)}|}{p-1} \leqslant \frac{\ell}{p-1} \ .
\end{aligned}$$

3.4 Proof of Lemma 3

To conclude the analysis, we introduce yet another event, $E^{(3)}$. We will show below that $W \ \wedge \ (\neg E^{(1)}) \ \wedge \ (\neg E^{(2)})$ implies $E^{(3)}$, and thus it is enough to upper bound the probability of $E^{(3)}$ occurring. Concretely, $E^{(3)}$ is defined as follows (the definition of the following events $F_{j'}$ is given in Section 3.3).

Event $E^{(3)}$. For each $j_1, j_2 \in [Q_{\mathrm{H}}]$ and $j_1 < j_2$, denote the event $E_{(j_1,j_2)}^{(3)}$ as

$$\exists \ i \in \mathcal{I}_{\text{unk}}^{(j_1)} \cap \mathcal{I}_{\text{unk}}^{(j_2)} : \alpha_{j_1}^{(2i)} \cdot \beta_{j_1}^{(2i-1)} \neq \alpha_{j_1}^{(2i-1)} \cdot \beta_{j_1}^{(2i)} \ \wedge \ \alpha_{j_2}^{(2i)} \cdot \beta_{j_2}^{(2i-1)} \neq \alpha_{j_2}^{(2i-1)} \cdot \beta_{j_2}^{(2i)} \ .$$

Denote $E'^{(3)}_{(j_1,j_2)} := E^{(3)}_{(j_1,j_2)} \ \wedge \ F_{j_1} \ \wedge \ F_{j_2}$ and $E^{(3)} := \bigvee_{j_1,j_2 \in [Q_{\mathrm{H}}], j_1 < j_2} E'^{(3)}_{(j_1,j_2)}$.

To see why the above implication is true, assume that W indeed occurs, but both $E^{(1)}$ and $E^{(2)}$ do not occur. We now fix some $j \in \mathcal{J}$. We know W_j occurs, but both $E_j^{(1)}$ and $E_j^{(2)}$ do not occur. In particular, by the definition of $E_j^{(2)}$, we know there exists $i \in \mathcal{I}_{\text{unk}}^{(j)}$ such that $\alpha_j^{(2i)} \cdot \beta_j^{(2i-1)} \neq \alpha_j^{(2i-1)} \cdot \beta_j^{(2i)}$.

Let $i_{\min}^{(j)}$ be the smallest index in $\mathcal{I}_{\mathrm{unk}}^{(j)}$ such that $\alpha_j^{(2i_{\min}^{(j)})} \cdot \beta_j^{(2i_{\min}^{(j)}-1)} \neq \alpha_j^{(2i_{\min}^{(j)}-1)} \cdot \beta_j^{(2i_{\min}^{(j)})}$. Since W occurs, we know $|\mathcal{J}| > \ell$. Then, since $i_{\min}^{(j)} \in \mathcal{I}_{\mathrm{unk}}^{(j)} \subseteq [\ell]$ for each $j \in \mathcal{J}$ and $|\mathcal{J}| > \ell$, by the pigeonhole principle, we know there exists $j_1, j_2 \in \mathcal{J}$ such that $j_1 < j_2$ and $i_{\min}^{(j_1)} = i_{\min}^{(j_2)}$, which implies $E_{(j_1,j_2)}^{(3)}$ occurs. Also, since we know both $W_{j_1} \wedge (\neg E_{j_1}^{(1)})$ and $W_{j_2} \wedge (\neg E_{j_2}^{(1)})$ occur, by Lemma 5, we have F_{j_1} and F_{j_2} both occur. Therefore, we know $E'^{(3)}_{(j_1,j_2)} = E^{(3)}_{(j_1,j_2)} \wedge F_{j_1} \wedge F_{j_2}$ occurs, which implies $E^{(3)}$ occurs.

Therefore, we have

$$\mathsf{Pr}\left[W \wedge (\neg E^{(1)}) \wedge (\neg E^{(2)})\right] \leqslant \mathsf{Pr}[E^{(3)}] \leqslant \sum_{j_1,j_2 \in [Q_{\mathrm{H}}], j_1 < j_2} \mathsf{Pr}[E'^{(3)}_{(j_1,j_2)}] .$$

We now just need to bound $\mathsf{Pr}[E'^{(3)}_{(j_1,j_2)}]$ for any $j_1 < j_2$.

To gain insight, suppose $E'^{(3)}_{(j_1,j_2)}$ occurs. We can show that there exists $i \in \mathcal{I}_{\mathrm{unk}}^{(j_1)} \cap \mathcal{I}_{\mathrm{unk}}^{(j_2)}$ such that $\alpha_{j_1}^{(2i)} - \delta_{j_1}\beta_{j_1}^{(2i)} \neq 0$ and $\alpha_{j_2}^{(2i)} - \delta_{j_2}\beta_{j_2}^{(2i)} \neq 0$. Then, since F_{j_1} and F_{j_2} occur, by (F1), it holds that

$$\frac{\alpha_{j_1}^{(2i-1)} - \delta_{j_1} \cdot \beta_{j_1}^{(2i-1)}}{\alpha_{j_1}^{(2i)} - \delta_{j_1} \cdot \beta_{j_1}^{(2i)}} = c_i = \frac{\alpha_{j_2}^{(2i-1)} - \delta_{j_2} \cdot \beta_{j_2}^{(2i-1)}}{\alpha_{j_2}^{(2i)} - \delta_{j_2} \cdot \beta_{j_2}^{(2i)}} .$$

However, this can occur with only small probability since δ_{j_1} and δ_{j_2} are sampled independently. The following claim, proved in Section 3.5, makes this formal.

Claim 3 *For any $j_1, j_2 \in [Q_{\mathrm{H}}]$ such that $j_1 < j_2$, suppose $E'^{(3)}_{(j_1,j_2)}$ occurs. Let i_{dif} be the smallest index in $\mathcal{I}_{\mathrm{unk}}^{(j_1)} \cap \mathcal{I}_{\mathrm{unk}}^{(j_2)}$ such that $\alpha_{j_1}^{(2i_{\mathrm{dif}})} \cdot \beta_{j_1}^{(2i_{\mathrm{dif}}-1)} \neq \alpha_{j_1}^{(2i_{\mathrm{dif}}-1)} \cdot \beta_{j_1}^{(2i_{\mathrm{dif}})}$ and $\alpha_{j_2}^{(2i_{\mathrm{dif}})} \cdot \beta_{j_2}^{(2i_{\mathrm{dif}}-1)} \neq \alpha_{j_2}^{(2i_{\mathrm{dif}}-1)} \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})}$. Then, we have*

$$\alpha_{j_1}^{(2i_{\mathrm{dif}})} - \delta_{j_1}\beta_{j_1}^{(2i_{\mathrm{dif}})} \neq 0.$$

Moreover, let $T = \frac{\alpha_{j_1}^{(2i_{\mathrm{dif}}-1)} - \delta_{j_1} \cdot \beta_{j_1}^{(2i_{\mathrm{dif}}-1)}}{\alpha_{j_1}^{(2i_{\mathrm{dif}})} - \delta_{j_1} \cdot \beta_{j_1}^{(2i_{\mathrm{dif}})}}$, and we have

$$\beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})} \neq 0 \text{ and } \delta_{j_2} = \frac{\alpha_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \alpha_{j_2}^{(2i_{\mathrm{dif}})}}{\beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})}} . \tag{6}$$

Let T and i_{dif} be the values defined in the above claim. Consider the step when δ_{j_2} is generated. We know the j_2-th query to H has been made, and thus $\vec{\alpha}_{j_2}$ and $\vec{\beta}_{j_2}$ are determined. Also, since $j_1 < j_2$, the j_1-th query to H has returned, and thus $\vec{\alpha}_{j_1}$, $\vec{\alpha}_{j_2}$, and δ_j are determined. Therefore, we know i_{dif} and T are also determined. Thus, we know δ_{j_2} is picked uniformly at random from $\mathbb{Z}_p^*$ independent of i_{dif}, $\vec{\alpha}_{j_1}$, $\vec{\alpha}_{j_2}$, $\vec{\beta}_{j_1}$, $\vec{\beta}_{j_2}$, δ_{j_1}, and T. Then, by the above claim,

$$\begin{aligned}
\mathsf{Pr}[E'^{(3)}_{(j_1,j_2)}] &\leqslant \mathsf{Pr}\left[\begin{matrix} \alpha_{j_1}^{(2i_{\mathrm{dif}})} - \delta_{j_1}\beta_{j_1}^{(2i_{\mathrm{dif}})} \neq 0 \\ \wedge\ \beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})} \neq 0 \end{matrix} \ \wedge\ \delta_{j_2} = \frac{\alpha_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \alpha_{j_2}^{(2i_{\mathrm{dif}})}}{\beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})}}\right] \\
&\leqslant \mathsf{Pr}\left[\delta_{j_2} = \frac{\alpha_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \alpha_{j_2}^{(2i_{\mathrm{dif}})}}{\beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})}} \ \middle|\ \begin{matrix} \alpha_{j_1}^{(2i_{\mathrm{dif}})} - \delta_{j_1}\beta_{j_1}^{(2i_{\mathrm{dif}})} \neq 0 \\ \wedge\ \beta_{j_2}^{(2i_{\mathrm{dif}}-1)} - T \cdot \beta_{j_2}^{(2i_{\mathrm{dif}})} \neq 0 \end{matrix}\right] \\
&\leqslant \frac{1}{p-1} .
\end{aligned}$$

3.5 Proof of Claim 3

This proof relies on the following simple lemma, which we first state and prove.

Lemma 6. *Let p be a prime number. Let $a, b, c, d \in \mathbb{Z}_p$ be arbitrary values such that $a \cdot d \neq c \cdot b$. Then, for any $T \in \mathbb{Z}_p$ such that $a + T \cdot b = 0$, we have $c + T \cdot d \neq 0$.*

Proof. Since $a + T \cdot b = 0$ and $a \cdot d \neq c \cdot b$, we have

$$0 = d(a + T \cdot b) = a \cdot d + T \cdot b \cdot d \neq b \cdot c + T \cdot b \cdot d = b(c + T \cdot d),$$

which implies $c + T \cdot d \neq 0$. □

Proof (of Claim 3). Consider $j_1, j_2 \in [Q_{\mathrm{H}}]$ such that $j_1 < j_2$. Suppose $E'^{(3)}_{j_1,j_2}$ occurs. We know the events $E^{(3)}_{(j_1,j_2)}$, F_{j_1}, and F_{j_2} occur. Since $E^{(3)}_{j_1,j_2}$ occurs, let i_{dif} be the smallest index in $\mathcal{I}^{(j_1)}_{\mathrm{unk}} \cap \mathcal{I}^{(j_2)}_{\mathrm{unk}}$ such that $\alpha^{(2i_{\mathrm{dif}})}_{j_1} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_1} \neq \alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}$ and $\alpha^{(2i_{\mathrm{dif}})}_{j_2} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_2} \neq \alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}$.

We first show that $\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1}\beta^{(2i_{\mathrm{dif}})}_{j_1} \neq 0$. Suppose $\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1}\beta^{(2i_{\mathrm{dif}})}_{j_1} = 0$. Since $\alpha^{(2i_{\mathrm{dif}})}_{j_1} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_1} \neq \alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}$, by Lemma 6, we know

$$\alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} - \delta_{j_1}\beta^{(2i_{\mathrm{dif}}-1)}_{j_1} \neq 0 \,.$$

Therefore, we have

$$\begin{aligned}
&\alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} + c_{i_{\mathrm{dif}}} \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1}\left(\beta^{(2i_{\mathrm{dif}}-1)}_{j_1} + c_{i_{\mathrm{dif}}} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}\right) \\
&= \alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_1} + c_{i_{\mathrm{dif}}}\left(\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}\right) \\
&= \alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} - \delta_{j_1}\beta^{(2i_{\mathrm{dif}}-1)}_{j_1} \neq 0 \,.
\end{aligned}$$

However, since F_{j_1} occurs, we know (F1) holds for $j = j_1$, which yields a contradiction. Thus, we have $\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1}\beta^{(2i_{\mathrm{dif}})}_{j_1} \neq 0$.

Similarly, we have $\alpha^{(2i_{\mathrm{dif}})}_{j_2} - \delta_{j_2}\beta^{(2i_{\mathrm{dif}})}_{j_2} \neq 0$. Then, since F_{j_1} and F_{j_2} both occur, we know (F1) holds for $j = j_1$ and $j = j_2$, and thus

$$\frac{\alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_1}}{\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}} = c_{i_{\mathrm{dif}}} = \frac{\alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - \delta_{j_2} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_2}}{\alpha^{(2i_{\mathrm{dif}})}_{j_2} - \delta_{j_2} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}} \,.$$

Denote $T = \frac{\alpha^{(2i_{\mathrm{dif}}-1)}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_1}}{\alpha^{(2i_{\mathrm{dif}})}_{j_1} - \delta_{j_1} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_1}}$ and we have

$$\alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_2} - \delta_{j_2}(\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}) = 0 \,. \tag{7}$$

We now show that $\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2} \neq 0$. Suppose $\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2} = 0$. Since $\alpha^{(2i_{\mathrm{dif}})}_{j_2} \cdot \beta^{(2i_{\mathrm{dif}}-1)}_{j_2} \neq \alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}$, by Lemma 6, we know $\alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_2} \neq 0$ and

$$\alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_2} - \delta_{j_2}(\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}) = \alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_2} \neq 0,$$

which contradicts (7). Therefore, we have

$$\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2} \neq 0 \,,$$

and from (7), it holds that

$$\delta_{j_2} = \frac{\alpha^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \alpha^{(2i_{\mathrm{dif}})}_{j_2}}{\beta^{(2i_{\mathrm{dif}}-1)}_{j_2} - T \cdot \beta^{(2i_{\mathrm{dif}})}_{j_2}} \,.$$

□

Algorithm $\mathsf{BS}_1.\mathsf{Setup}(1^\lambda)$: $p \leftarrow \lvert\mathbb{G}_\lambda\rvert$ Let g be the generator of $\mathbb{G}_\lambda$ Select $\mathrm{H} : \{0,1\}^* \to \mathbb{Z}_p$ Return $par \leftarrow (p, g, \mathrm{H})$ Algorithm $\mathsf{BS}_1.\mathsf{KG}(par)$: $(p, g, \mathrm{H}) \leftarrow par$ $x \leftarrow\$\ \mathbb{Z}_p^*$; $X \leftarrow g^x$ $sk \leftarrow x$; $pk \leftarrow X$ Return (sk, pk) Algorithm $\mathsf{BS}_1.\mathsf{S}_1(sk)$: $x \leftarrow sk$; $X \leftarrow g^x$ $a \leftarrow\$\ \mathbb{Z}_p$; $y \leftarrow\$\ \mathbb{Z}_p^*$ $A \leftarrow g^a$; $Y \leftarrow X^y$ $\mathsf{st}^s \leftarrow (a, y, x)$; $\mathsf{msg}_1 \leftarrow (A, Y)$ Return $(\mathsf{st}^s, \mathsf{msg}_1)$ Algorithm $\mathsf{BS}_1.\mathsf{S}_2(\mathsf{st}^s, c)$: $(a, y, x) \leftarrow \mathsf{st}^s$ $s \leftarrow a + c \cdot y \cdot x$ Return $\mathsf{msg}_2 \leftarrow (s, y)$	Algorithm $\mathsf{BS}_1.\mathsf{U}_1(pk, \mathsf{msg}_1, m)$: $X \leftarrow pk$; $(A, Y) \leftarrow \mathsf{msg}_1$ $r_1, r_2 \leftarrow\$\ \mathbb{Z}_p$; $\gamma \leftarrow\$\ \mathbb{Z}_p^*$ $Y' \leftarrow Y^\gamma$ $A' \leftarrow g^{r_1} \cdot A^\gamma \cdot Y'^{r_2}$ $c' \leftarrow H(A' \,\Vert\, Y' \,\Vert\, m)$ $c \leftarrow c' + r_2$ $\mathsf{st}^u \leftarrow (c, c', r_1, \gamma, X, Y, A)$ Return (st^u, c) Algorithm $\mathsf{BS}_1.\mathsf{U}_2(\mathsf{st}^u, \mathsf{msg}_2)$: $(c, c', r_1, \gamma, X, Y, A) \leftarrow \mathsf{st}^u$ $(s, y) \leftarrow \mathsf{msg}_2$ If $y = 0$ or $Y \neq X^y$ or $g^s \neq A \cdot Y^c$ then return $\perp$ $s' \leftarrow \gamma \cdot s + r_1$ $y' \leftarrow \gamma \cdot y$ Return $\sigma \leftarrow (c', s', y')$ Algorithm $\mathsf{BS}_1.\mathsf{Ver}(pk, \sigma, m)$: $(c, s, y) \leftarrow \sigma$ If $y = 0$ then return 0 $Y \leftarrow X^y$; $A \leftarrow g^s \cdot Y^{-c}$ If $c \neq \mathrm{H}(A \,\Vert\, Y \,\Vert\, m)$ then return 0 Return 1

Fig. 4. The blind signature scheme $\mathsf{BS}_1 = \mathsf{BS}_1[\mathbb{G}]$.

4 Efficient Blind Signatures in the GGM

This section introduces our first scheme, BS_1, which relies on a prime-order cyclic group and a hash function H. We describe this scheme formally in Figure 4. Roughly, it extends (blind) Schnorr Signatures by sending an additional group element $Y = X^y$ in the first round. Then, the signer's final response to challenge c reveals y along with $s = a + cxy$. We also note that we could consider a variant of the scheme where the signature consists of $\sigma = (A', s', y')$, where A' replaces c'.

Security analysis. First off, we observe that the protocol is blind.

Theorem 2. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. Then, the blind signature scheme $\mathsf{BS}_1[\mathbb{G}]$ is perfectly blind.*

Proof (of Theorem 2). Let $\mathcal{A}$ be an adversary playing the $\mathrm{Blind}^{\mathcal{A}}_{\mathsf{BS}_1[\mathbb{G}]}$ game. Without loss of generality, we can assume the randomness of $\mathcal{A}$ is fixed and $\mathcal{A}$ always finishes both signing sessions and receives valid signatures (σ_0, σ_1). [7]

Define the view of $\mathcal{A}$ after its execution as $\pi = (X, m_0, m_1, T_0, T_1, \sigma_0, \sigma_1)$, where $T_i := (A_i, Y_i, c_i, s_i, y_i)$, denoting the transcripts learned from interactions with the i-th signing session and $\sigma_i = (c_i', s_i', y_i')$. Since the randomness of $\mathcal{A}$ is fixed, the only randomness left is the randomness in U_1 and U_2. Denote $\eta := (r_1^{(0)}, r_2^{(0)}, \gamma^{(0)}, r_1^{(1)}, r_2^{(1)}, \gamma^{(1)})$ as the total randomness. To prove the theorem, we need only show that the distribution of π is identical in both the case $b = 0$ and $b = 1$. We prove this by showing that for any fixed

[7] Since the output of each query to U_1 that does not return $\perp$ is uniformly random over $\mathbb{Z}_p$, we know the behavior of $\mathcal{A}$ is identical in both the case $b = 0$ and $b = 1$ before $\mathcal{A}$ receives the valid signature (σ_0, σ_1). Therefore, we know the probability that $\mathcal{A}$ returns before receiving (σ_0, σ_1) or receives $(\perp, \perp)$ after finishing both signing sessions is equal in both the case $b = 0$ and $b = 1$, which means we consider only the case where $\mathcal{A}$ receives valid signatures.

view Δ such that $\Pr[\pi = \Delta | b = 1] > 0$, there exists a unique value of the randomness η that makes $\pi = \Delta$ for the cases $b = 0$ and $b = 1$.

For both the cases $b = 0$ and $b = 1$, we now show that $\pi = \Delta$ if and only if for each $i \in \{0, 1\}$, it holds that

$$\begin{aligned} \gamma^{(i)} &= {y'_{b_i}}^{\Delta} / y_i^{\Delta} , \\ r_1^{(i)} &= {s'_{b_i}}^{\Delta} - \gamma^{(i)} \cdot s_i^{\Delta} , \\ r_2^{(i)} &= c_i^{\Delta} - {c'_{b_i}}^{\Delta} , \end{aligned} \tag{8}$$

where the superscript $(\cdot)^{\Delta}$ represents the corresponding value in Δ. From the algorithms $\mathsf{BS}_1.\mathsf{U}_1$ and $\mathsf{BS}_1.\mathsf{U}_2$, it is clear that the "only if" part holds. For the "if" part, suppose (8) holds. Since the randomness of $\mathcal{A}$ is fixed, the view of $\mathcal{A}$ can differ only on the outputs c_0, c_1 from the oracle U_1 or the output (σ_0, σ_1) from the oracle U_2. Since both signatures in Δ are valid, we have

$$A_i^{\Delta} = g^{s_i^{\Delta}} {X^{\Delta}}^{-c_i^{\Delta} \cdot y_i^{\Delta}} \ , \ Y_i^{\Delta} = {X^{\Delta}}^{y_i^{\Delta}} \ , \tag{9}$$

$${c'_{b_i}}^{\Delta} = \mathrm{H}(g^{{s'_{b_i}}^{\Delta}} {X^{\Delta}}^{-{y'_{b_i}}^{\Delta} \cdot {c'_{b_i}}^{\Delta}} \,\|\, {X^{\Delta}}^{{y'_{b_i}}^{\Delta}} \,\|\, m_{b_i}^{\Delta}) \ . \tag{10}$$

For c_i where $i \in \{0, 1\}$, suppose the values in the view of $\mathcal{A}$ that have already determined when c_i is generated, which must include (X, m_i, A_i, Y_i), are consistent with Δ. By (8), we have

$$\begin{aligned} c_i &= r_2^{(i)} + \mathrm{H}(g^{r_1^{(i)}} A_i^{\gamma^{(i)}} Y_i^{\gamma^{(i)} \cdot r_2^{(i)}} \,\|\, Y_i^{\gamma^{(i)}} \,\|\, m_{b_i}) \\ &= r_2^{(i)} + \mathrm{H}(g^{r_1^{(i)}} {A_i^{\Delta}}^{\gamma^{(i)}} {Y_i^{\Delta}}^{\gamma^{(i)} \cdot r_2^{(i)}} \,\|\, {Y_i^{\Delta}}^{\gamma^{(i)}} \,\|\, m_{b_i}^{\Delta}) \\ &= r_2^{(i)} + \mathrm{H}(g^{\gamma^{(i)} \cdot s_i^{\Delta} + r_1^{(i)}} {X^{\Delta}}^{-y_i^{\Delta} \cdot \gamma^{(i)} \cdot (c_i^{\Delta} - r_2^{(i)})} \,\|\, {X^{\Delta}}^{y_i^{\Delta} \cdot \gamma^{(i)}} \,\|\, m_{b_i}^{\Delta}) \\ &= r_2^{(i)} + \mathrm{H}(g^{{s'_{b_i}}^{\Delta}} {X^{\Delta}}^{-{y'_{b_i}}^{\Delta} \cdot {c'_{b_i}}^{\Delta}} \,\|\, {X^{\Delta}}^{{y'_{b_i}}^{\Delta}} \,\|\, m_{b_i}^{\Delta}) \\ &= r_2^{(i)} + {c'_{b_i}}^{\Delta} = c_i^{\Delta} \ , \end{aligned}$$

where the third equality is due to (9), the fourth equality is due to (8), and the final equality is due to (10). Then, consider the step when (σ_0, σ_1) is output. Suppose the current view, which contains T_i, is consistent with Δ. By (8), we have

$$\begin{aligned} y'_{b_i} &= \gamma^{(i)} \cdot y_i = \gamma^{(i)} \cdot y_i^{\Delta} = {y'_{b_i}}^{\Delta} \ , \\ s'_{b_i} &= r_1^{(i)} + \gamma^{(i)} \cdot s_i = r_1^{(i)} + \gamma^{(i)} \cdot s_i^{\Delta} = {s'_{b_i}}^{\Delta} \ , \\ c'_{b_i} &= c_i - r_2^{(i)} = c_i^{\Delta} - r_2^{(i)} = {c'_{b_i}}^{\Delta} \ . \end{aligned}$$

which implies $(\sigma_0, \sigma_1) = (\sigma_0^{\Delta}, \sigma_1^{\Delta})$. Therefore, by induction, if (8) holds, we know $\pi = \Delta$. $\square$

Our main result shows OMUF security of BS_1 in the *generic-group model* (GGM) following Shoup's original formalization [Sho97], which encodes every group element with a random label. To this end, we present in Figure 5 a game describing a GGM-version of OMUF security for BS_1, adapting the one from Section 2. We also define a corresponding advantage $\mathsf{Adv}^{\text{omuf-ggm}}_{\mathsf{BS}_1[\mathbb{G}]}(\mathcal{A}, \lambda)$ to measure the probability that $\mathcal{A}$ wins the game. Note that to keep notation homogenous, it is convenient to allow the game to depend on $\mathbb{G}$, although the game itself only makes use of the order of the group. The game also models the hash function H as a random oracle, to which the adversary is given oracle access.

The following theorem states our main result in the form of a reduction to WFROS and is proved in Section 4.1.

Game $\mathrm{OMUF\text{-}GGM}^{\mathcal{A}}_{\mathsf{BS}_1[\mathbb{G}]}(\lambda)$:	Oracle S_1 :

$p \leftarrow \lvert\mathbb{G}_\lambda\rvert$; $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$\mathrm{sid} \leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{I}_{\mathrm{fin}} \leftarrow \varnothing$; $\mathsf{Cur} \leftarrow \varnothing$; $\Xi \leftarrow ()$; $T \leftarrow ()$
$\{(m_k, \sigma_k)\}_{k\in[\ell+1]} \leftarrow\!\!\$\ \mathcal{A}^{\Pi,\mathrm{S}_1,\mathrm{S}_2,\mathrm{H}}(p, \Phi(1), \Phi(x))$
If $\exists\ k_1 \neq k_2$ such that $(m_{k_1}, \sigma_{k_1}) = (m_{k_2}, \sigma_{k_2})$ then
   Return 0
If $\exists\ k \in [\ell+1]$ such that $y_k^* = 0$
   or $c_k \neq \mathrm{H}(\Phi(s_k - c_k \cdot y_k \cdot x) \,\Vert\, \Phi(y_k \cdot x) \,\Vert\, m_i)$
where $(c_k, s_k, y_k) = \sigma_k$ then return 0
Return 1
 	
Oracle $\Phi(v)$:
If $v \in \mathsf{Cur}$ then return $\Xi(v)$
$\Xi(v) \leftarrow\!\!\$\ \{0,1\}^{\log(p)} \backslash \Xi(\mathsf{Cur})$
$\mathsf{Cur} \leftarrow \mathsf{Cur} \cap \{v\}$
Return $\Xi(v)$
 	
Oracle $\Pi(\xi, \xi', b)$:
If $\exists v, v' \in \mathsf{Cur}$ such that $\xi = \Xi(v)$ and $\xi' = \Xi(v')$ then
   Return $\Phi(v + (-1)^b v')$
Else return $\perp$
	$\mathrm{sid} \leftarrow \mathrm{sid} + 1$
	$a_{\mathrm{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p$; $y_{\mathrm{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p^*$
	$\mathsf{st}^s_{\mathrm{sid}} \leftarrow (a_{\mathrm{sid}}, y_{\mathrm{sid}})$
	$\mathsf{msg}_1 \leftarrow (\Phi(a_{\mathrm{sid}}), \Phi(y_{\mathrm{sid}} \cdot x))$
	Return $(\mathrm{sid}, \mathsf{msg}_1)$
	
	Oracle $\mathrm{S}_2(i, c_i)$:
	If $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
	$(a_i, y_i) \leftarrow \mathsf{st}^s_i$
	$s_i \leftarrow a_i + c_i \cdot y_i \cdot x$
	$\mathsf{msg}_2 \leftarrow (s_i, y_i)$
	$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
	$\ell \leftarrow \ell + 1$
	Return msg_2
	
	Oracle $\mathrm{H}(\mathrm{str})$:
	If $T(\mathrm{str}) = \perp$ then
	   $T(\mathrm{str}) \leftarrow\!\!\$\ \mathbb{Z}_p$
	Return $T(\mathrm{str})$

Fig. 5. The OMUF security game in GGM for the blind signature scheme $\mathsf{BS}_1[\mathbb{G}]$.

Theorem 3 (OMUF Security of BS_1). *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. For any adversary $\mathcal{A}$ playing game* OMUF-GGM$^{\mathsf{BS}_1[\mathbb{G}]}(\lambda)$ *making at most Q_Π queries to Π, Q_{S_1} queries to S_1, and Q_{H} queries to the random oracle* H*, there exists an adversary $\mathcal{B}$ for the* $\mathrm{WFROS}_{Q_{\mathrm{S}_1},p}$ *problem, where $p = \lvert\mathbb{G}_\lambda\rvert$, making at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to the random oracle* H *such that*

$$\mathsf{Adv}^{\mathrm{omuf\text{-}ggm}}_{\mathsf{BS}_1[\mathbb{G}]}(\mathcal{A}, \lambda) \leqslant \mathsf{Adv}^{\mathrm{wfros}}_{Q_{\mathrm{S}_1},p}(\mathcal{B}) + \frac{Q_\Phi(Q_\Phi + 2Q_{\mathrm{H}} + 2Q_{\mathrm{S}_1} + 2)}{p - (1 + Q_{\mathrm{S}_1} + Q_\Phi^2)} ,$$

where Q_Φ is the maximum number of queries to Φ during the game OMUF-GGM*, and we have $Q_\Phi = Q_\Pi + 4Q_{\mathrm{S}_1} + 4$.*

By Theorem 1, we have the following corollary.

Corollary 1. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. For any adversary $\mathcal{A}$ playing game* OMUF-GGM$^{\mathsf{BS}_1[\mathbb{G}]}(\lambda)$ *making at most Q_Π queries to Π, Q_{S_1} queries to S_1, and Q_{H} queries to the random oracle* H*, we have*

$$\mathsf{Adv}^{\mathrm{omuf\text{-}ggm}}_{\mathsf{BS}_1[\mathbb{G}]}(\mathcal{A}, \lambda) \leqslant \frac{2Q_\Phi(Q_\Phi + 2Q_{\mathrm{H}} + 2Q_{\mathrm{S}_1} + 2)}{p - (1 + Q_{\mathrm{S}_1} + Q_\Phi^2)} ,$$

where $Q_\Phi = Q_\Pi + 4Q_{\mathrm{S}_1} + 4$.

We note in particular that the concrete security of BS_1 in the GGM is comparable to that of the discrete logarithm problem, in that $Q_\Phi = \Omega(\min\{\sqrt{p}, p/Q_{\mathrm{H}}, p/Q_{\mathrm{S}_1}\})$ is necessary to break security with constant probability.

4.1 Proof of Theorem 3

Let us fix an adversary $\mathcal{A}$ that makes (without loss of generality) exactly Q_Π queries to Π, Q_{S_1} queries to S_1, and Q_{H} queries to the random oracle H. Without loss of generality, assume it also makes exactly one query

Game Game_4:
$p \leftarrow \lvert\mathbb{G}_\lambda\rvert$
$\mathrm{sid} \leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{S} \leftarrow \varnothing$; $\mathsf{Cur} \leftarrow \varnothing$; $\Xi \leftarrow ()$; $T \leftarrow ()$
$\{(m_k, \sigma_k)\}_{k\in[\ell+1]} \leftarrow\!\!\$\ \mathcal{A}^{\Pi,\mathrm{S}_1,\mathrm{S}_2,\mathrm{H}}(p, \Phi(1), \Phi(\mathsf{X}))$
If $\exists\ k_1 \neq k_2$ such that $(m_{k_1}, \sigma_{k_1}) = (m_{k_2}, \sigma_{k_2})$ then
  Return 0
If $\exists\ k \in [\ell+1]$ such that $y_k^* = 0$
  or $c_k \neq \mathrm{H}(\Phi(s_k - c_k \cdot y_k \cdot \mathsf{X}) \,\|\, \Phi(y_k \cdot \mathsf{X}) \,\|\, m_i)$
where $(c_k, s_k, y_k) = \sigma_k$ then return 0
Return 1
 	
Oracle $\Phi(P)$:
If $\exists P' \in \mathsf{Cur}$ such that $P =_L P'$ then
  Return $\Xi(P')$
$\Xi(P) \leftarrow\!\!\$\ \{0,1\}^{\lceil \log(p) \rceil} \backslash \Xi(\mathsf{Cur})$
$\mathsf{Cur} \leftarrow \mathsf{Cur} \cap \{P\}$
Return $\Xi(P)$
 	
Oracle $\Pi(\xi, \xi', b)$:
If $\exists P, P' \in \mathsf{Cur}$ such that $\xi = \Xi(P)$
  and $\xi' = \Xi(P')$ then
  Return $\Phi(P + (-1)^b P')$
Else return $\perp$
	Oracle S_1 :
	$\mathrm{sid} \leftarrow \mathrm{sid} + 1$
	$\mathsf{msg}_1 \leftarrow (\Phi(\mathsf{A}_{\ sid}), \Phi(\mathsf{Y}_{\mathrm{sid}}))$
	Return $(\mathrm{sid}, \mathsf{msg}_1)$
	
	Oracle $\mathrm{S}_2(i, c_i)$:
	If $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
	$s_i \leftarrow\!\!\$\ \mathbb{Z}_p$; $y_i \leftarrow\!\!\$\ \mathbb{Z}_p^*$
	$R_1 \leftarrow \mathsf{A}_i + c_i \mathsf{Y}_i - s_i$
	$R_2 \leftarrow \mathsf{Y}_i - y_i \mathsf{X}$
	$L \leftarrow L \cup \{R_1, R_2\}$
	$\mathsf{msg}_2 \leftarrow (s_i, y_i)$
	If $\exists\ P_1, P_2 \in \mathsf{Cur}$ such that
	  $P_1 \neq P_2$ and $P_1 =_L P_2$
	then **abort game**
	$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
	$\ell \leftarrow \ell + 1$
	Return msg_2
	
	Oracle $\mathrm{H}(\mathrm{str})$:
	If $T(\mathrm{str}) = \perp$ then
	  $T(\mathrm{str}) \leftarrow\!\!\$\ \mathbb{Z}_p$
	Return $T(\mathrm{str})$

Fig. 6. The definition of Game_4. The symbols P and P' denote polynomials over variables X, $\{\mathsf{A}_i, \mathsf{Y}_i\}_{i\in[\mathrm{sid}]}$. Also, a new equality notation, "$=_L$", is used. We say $P_1 =_L P_2$ if and only if $P_1 - P_2$ can be represented as a linear combination of polynomials in L.

(i, c_i) to S_2 for each $i \in [Q_{\mathrm{S}_1}]$. Also, it is clear that the overall number of queries to Φ in $\mathrm{OMUF\text{-}GGM}^{\mathcal{A}}_{\mathsf{BS}_1}$ is at most $Q_\Phi := Q_\Pi + 4Q_{\mathrm{S}_1} + 4$. Then, after $\mathcal{A}$ returns, we know $\ell = Q_{\mathrm{S}_1}$ and $\mathcal{I}_{\mathrm{fin}} = [Q_{\mathrm{S}_1}]$.

We prove the theorem by going through a series of games, from Game_0 to Game_4, where Game_0 is the $\mathrm{OMUF\text{-}GGM}^{\mathcal{A}}_{\mathsf{BS}_1}$ game and Game_4 is an intermediate game that enables an easier reduction to WFROS. Here, however, we first introduce Game_4 and Lemma 7 and then discuss the reduction to WFROS, which is the core of the proof. We leave the definition of the intermediate games between Game_0 to Game_4 to the proof of Lemma 7. The game-hopping argument is non-trivial, but it follows the same blueprint as in [BFP21] and is hence deferred to Appendix B.1.

Definition of Game_4. The pseudocode description of Game_4 is given in Figure 6. The main difference from $\mathrm{OMUF\text{-}GGM}^{\mathcal{A}}_{\mathsf{BS}_1}$ is that the encoding oracle Φ takes as input a polynomial instead of an integer in $\mathbb{Z}_p$. (Note that the adversary cannot query Φ directly, and thus this difference is not directly surfaced.) This essentially captures the algebraic core of our proof.

Also, for a valid query (i, c_i) to S_2, the output values (s_i, y_i) are directly sampled uniformly from $\mathbb{Z}_p \times \mathbb{Z}_p^*$. Furthermore, when this happens, two polynomials, $R_1 = \mathsf{A}_i + c_i \cdot \mathsf{Y}_i - s_i$ and $R_2 = \mathsf{Y}_i - y_i \cdot \mathsf{X}$, are recorded in the set L. Then, in the encoding oracle Φ, two polynomials, P_1 and P_2, are considered to differ if and only if $P_1 \neq_L P_2$, where $P_1 =_L P_2$ means that $P_1 - P_2$ can be generated as a linear combination of polynomials in L. Still, $P_1 \neq_L P_2$ could occur when queries P_1 and P_2 are made to Φ, but they becomes equal (in the sense of "$=_L$") after L is updated. The game aborts when this happens.

Overall, we prove the following lemma in Appendix B.1.

Lemma 7. $\mathsf{Adv}^{\mathrm{omuf\text{-}ggm}}_{\mathsf{BS}_1[\mathbb{G}]}(\mathcal{A}, \lambda) \leqslant \mathsf{Pr}[\mathrm{Game}_4^{\mathcal{A}} = 1] + \frac{Q_\Phi^2}{p-(1+Q_{\mathrm{S}_1}+Q_\Phi^2)}$.

Reduction to WFROS. The core of the proof is to relate the probability of the adversary $\mathcal{A}$ winning Game_4 with the advantage of an adversary $\mathcal{B}$ winning the WFROS problem, as stated in the following lemma. The proof is given in Section 4.2.

Lemma 8. *For every λ, there exists an adversary $\mathcal{B}$ for the* $\mathrm{WFROS}_{Q_{\mathrm{S}_1},p}$ *problem, where $p = |\mathbb{G}_\lambda|$, making at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to* H *such that*

$$\Pr[\mathrm{Game}_4^{\mathcal{A}} = 1] \leqslant \mathsf{Adv}^{\mathrm{wfros}}_{Q_{\mathrm{S}_1},p}(\mathcal{B}) + \frac{(2Q_{\varPhi} + 1)(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)}{p - Q_{\varPhi}} \ . \tag{11}$$

The statement of Theorem 3 follows by combining Lemmas 7 and 8.

4.2 Proof of Lemma 8

We construct $\mathcal{B}$ that interacts with $\mathcal{A}$ by simulating the oracles from Game_4 using the two oracles S and H in WFROS. In particular, we extract suitable vectors $\vec{\alpha}$ and $\vec{\beta}$ to query to H in WFROS, i.e., each RO query str is decomposed as $\mathrm{str} = \xi^A \,\|\, \xi^Y \,\|\, m$, where ξ^A and ξ^Y are encodings of group elements. If both encodings are valid, there must exist P^A, P^Y such that $\varXi(P^A) = \xi^A$ and $\varXi(P^Y) = \xi^Y$; then, $\mathcal{B}$ defines two vectors $\vec{\alpha}$ and $\vec{\beta}$ to make a corresponding query to H in WFROS. The oracle S is also used to simulate the signer's second stage. Finally, when $\mathcal{A}$ outputs $Q_{\mathrm{S}_1} + 1$ different valid message-signature pairs in Game_4, $\mathcal{B}$ tries to map each valid message-signature pair to a query to H in WFROS. We show that this strategy succeeds with probability close to that of $\mathcal{A}$ succeeding.

The adversary $\mathcal{B}$. Specifically, $\mathcal{B}$ initializes the variables sid, Cur, $\mathcal{I}_{\mathrm{fin}}$, $\varXi$, and T as in Game_4. In addition, $\mathcal{B}$ initializes an empty table Hid, used later in the simulation of $\hat{\mathrm{H}}$.

Then, $\mathcal{B}$ runs $\mathcal{A}$ on input $(p, \hat{\varPhi}(1), \hat{\varPhi}(\mathsf{X}))$ and with access to the oracles $\hat{\varPi}$, $\hat{\mathrm{S}}_1$, $\hat{\mathrm{S}}_2$, and $\hat{\mathrm{H}}$. These oracles, along with $\hat{\varPhi}$, operate as follows:

Oracles $\hat{\varPhi}, \hat{\varPi}$: Same as in Game_4. In particular, L is updated by calls to $\hat{\mathrm{S}}_2$.
Oracle $\hat{\mathrm{S}}_1$: Same as in Game_4.
Oracle $\hat{\mathrm{S}}_2$: Same as Game_4 except that instead of sampling y_i randomly, if $i \in [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$, $\mathcal{B}$ makes a query (i, c_i) to S and uses its output as the value y_i.
Oracle $\hat{\mathrm{H}}$: After receiving a query str, if $T(\mathrm{str}) \neq \bot$, the value $T(\mathrm{str})$ is returned. Otherwise, str is decomposed as $\mathrm{str} = \xi^A \,\|\, \xi^Y \,\|\, m$ such that the length of ξ^A and ξ^Y is $\lceil \log(p) \rceil$.

– If there exist $P^A, P^Y \in \mathsf{Cur}$ such that $\varXi(P^A) = \xi^A$ and $\varXi(P^Y) = \xi^Y$, denote the coefficients of P^A, P^Y as

$$P^A = \hat{\alpha}^g + \hat{\alpha}^{\mathsf{X}}\mathsf{X} + \sum_{i \in [\mathrm{sid}]} \hat{\alpha}^{\mathsf{A}_i}\mathsf{A}_i + \sum_{i \in [\mathrm{sid}]} \hat{\alpha}^{\mathsf{Y}_i}\mathsf{Y}_i \ , \tag{12}$$

$$P^Y = \hat{\beta}^g + \hat{\beta}^{\mathsf{X}}\mathsf{X} + \sum_{i \in [\mathrm{sid}]} \hat{\beta}^{\mathsf{A}_i}\mathsf{A}_i + \sum_{i \in [\mathrm{sid}]} \hat{\beta}^{\mathsf{Y}_i}\mathsf{Y}_i \ . \tag{13}$$

Then, $\mathcal{B}$ issues the query $(\vec{\alpha}, \vec{\beta})$ to H, where $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2Q_{\mathrm{S}_1}+1}$ are such that

$$\begin{aligned} \alpha^{(i')} &= \begin{cases} \hat{\alpha}^{\mathsf{X}} \ , & i' = 0 \\ \hat{\alpha}^{\mathsf{Y}_i} \ , & i' = 2i - 1 \ , \ i \in [\mathrm{sid}] \\ -\hat{\alpha}^{\mathsf{A}_i} \ , & i' = 2i \ , \ i \in [\mathrm{sid}] \\ 0 \ , & o.w. \end{cases} \ , \\ \beta^{(i')} &= \begin{cases} -\hat{\beta}^{\mathsf{X}} \ , & i' = 0 \\ -\hat{\beta}^{\mathsf{Y}_i} \ , & i' = 2i - 1 \ , \ i \in [\mathrm{sid}] \\ \hat{\beta}^{\mathsf{A}_i} \ , & i' = 2i \ , \ i \in [\mathrm{sid}] \\ 0 \ , & o.w. \end{cases} \ . \end{aligned} \tag{14}$$

After receiving the output $(\delta_{\mathrm{hid}}, \mathrm{hid})$, $\mathcal{B}$ sets $T(\mathrm{str}) \leftarrow \delta_{\mathrm{hid}}$ and $\mathrm{Hid}(\mathrm{str}) \leftarrow \mathrm{hid}$.

– Otherwise, if $\xi^A \notin T(\mathsf{Cur})$ or $\xi^Y \notin T(\mathsf{Cur})$ (or if the decomposition of str is not possible), $\mathcal{B}$ samples $T(\text{str})$ uniformly from $\mathbb{Z}_p$ and sets $\text{Hid}(\text{str}) = \perp$.

Finally, $\mathcal{B}$ returns $T(\text{str})$.

After $\mathcal{A}$ outputs $\{(m_k^*, \sigma_k^*)\}_{k\in[Q_{\mathrm{S}_1}+1]}$, $\mathcal{B}$ aborts if the signatures are not valid, i.e., one of the following conditions is not satisfied:

$$\forall\ k_1, k_2 \in [Q_{\mathrm{S}_1} + 1] \text{ and } k_1 \neq k_2\ :\ (m_{k_1}^*, \sigma_{k_1}^*) \neq (m_{k_2}^*, \sigma_{k_2}^*)\ , \tag{15}$$

$$\forall\ k \in [Q_{\mathrm{S}_1} + 1]\ :\ y_k^* \neq 0\ \wedge\ c_k^* = \hat{\mathrm{H}}(\text{str}_k^*)\ , \tag{16}$$

where $(c_k^*, s_k^*, y_k^*) = \sigma_k^*$ and $\text{str}_k^* = \hat{\Phi}(s_k^* - c_k^* \cdot y_k^* \cdot \mathsf{X}) \,\|\, \hat{\Phi}(y_k^* \cdot \mathsf{X}) \,\|\, m_k^*$. (Here, $\hat{\mathrm{H}}$ and $\hat{\Phi}$ are the oracles described previously.) Further, $\mathcal{B}$ aborts if the following condition does not hold:

$$\forall\ k \in [Q_{\mathrm{S}_1} + 1]\ :\ \text{Hid}(\text{str}_k^*) \neq \perp\ . \tag{17}$$

Otherwise, $\mathcal{B}$ outputs $\mathcal{J} := \{\text{Hid}(\text{str}_k^*)\}_{k\in[Q_{\mathrm{S}_1}+1]}$.

Analysis of $\mathcal{B}$. Note that $\mathcal{B}$ queries to H at most once when it receives a query to $\hat{\mathrm{H}}$ and makes $Q_{\mathrm{S}_1} + 1$ more queries to $\hat{\mathrm{H}}$ when checking the validity of the output. Therefore, $\mathcal{B}$ makes at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to H. Also, it is clear that $\mathcal{B}$ simulates oracles S_1, S_2 in Game_4 perfectly. For the simulation of $\hat{\mathrm{H}}$, the only difference is that the distribution of δ_{hid} outputting from H in WFROS is uniformly over $\mathbb{Z}_p^*$, where in Game_4 it is always uniformly from $\mathbb{Z}_p$. However, the statistical distance between the two distributions is $1/p$. Since $\mathcal{B}$ makes at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to H, the statistical difference between the view of $\mathcal{A}$ in Game_4 and that in the one simulated by $\mathcal{B}$ is bounded by $(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)/p$.

Denote the event E_1 such that when $\mathcal{B}$ checks the output from $\mathcal{A}$, both (15) and (16) hold. As these are exactly the winning conditions of Game_4, which is simulated statistically closed to perfect, we have

$$\mathsf{Pr}[E_1] + \frac{Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1}{p} \geqslant \mathsf{Pr}[\text{Game}_4^{\mathcal{A}} = 1]\ . \tag{18}$$

Also, let E_2 be the event for which the condition (17) holds immediately afterward. If E_2 does not happen, but E_1 does, then we know $\mathcal{A}$ outputs a valid message-signature pair (m_k^*, σ_k^*) such that $\text{Hid}(\text{str}_k^*) = \perp$, which is unlikely to happen. The following formalizes this, and the proof is in Appendix B.3.

Claim 4 $\mathsf{Pr}[E_1\ \wedge\ (\neg E_2)] \leqslant \frac{2Q_\Phi(Q_{\mathrm{H}}+Q_{\mathrm{S}_1}+1)}{p-Q_\Phi}$.

Then, we can conclude the proof with the following claim.

Claim 5 *If both E_1 and E_2 happen, then $\mathcal{B}$ outputs a valid WFROS solution $\mathcal{J}$, which in turn implies that* $\mathsf{Pr}[E_1\ \wedge\ E_2] \leqslant \mathsf{Adv}^{\text{wfros}}_{Q_{\mathrm{S}_1},p}(\mathcal{B})$.

Before we proceed with a proof, we state a simple lemma for Game_4 that is used in the proof of Claim 5. The proof is immediate and follows from the uniqueness of values returned by the oracle.

Lemma 9. *At any step of* Game_4*, for any two polynomials P and P', suppose we make queries P and P' to Φ. If $\Phi(P) = \Phi(P')$, then $P =_L P'$. Equivalently, if $P \neq_L P'$, then we have $\Phi(P) \neq \Phi(P')$.*

Proof (of Claim 5). Suppose both E_1 and E_2 happen. We first show that for any $k_1, k_2 \in [Q_{\mathrm{S}_1} + 1]$ and $k_1 \neq k_2$, it holds that $\text{str}_{k_1}^* \neq \text{str}_{k_2}^*$, which implies $|\mathcal{J}| = Q_{\mathrm{S}_1} + 1$. We then show that $\mathcal{J}$ is valid for the WFROS game.

For $k_1, k_2 \in [Q_{\mathrm{S}_1} + 1]$ and $k_1 \neq k_2$, suppose $\text{str}_{k_1}^* = \text{str}_{k_2}^*$. Then, we have

$$c_{k_1}^* = \hat{\mathrm{H}}(\text{str}_{k_1}^*) = \hat{\mathrm{H}}(\text{str}_{k_2}^*) = c_{k_2}^*\ ,\ m_{k_1}^* = m_{k_2}^*\ ,$$

$$\hat{\Phi}(s^*_{k_1} - c^*_{k_1} \cdot y^*_{k_1} \cdot \mathsf{X}) = \hat{\Phi}(s^*_{k_2} - c^*_{k_2} \cdot y^*_{k_2} \cdot \mathsf{X}) \ , \ \hat{\Phi}(y^*_{k_1} \cdot \mathsf{X}) = \hat{\Phi}(y^*_{k_2} \cdot \mathsf{X}) \ .$$

By Lemma 9, it holds that $(m^*_{k_1}, (c^*_{k_1}, s^*_{k_1}, y^*_{k_1})) = (m^*_{k_2}, (c^*_{k_2}, s^*_{k_2}, y^*_{k_2}))$. However, since E_1 happens, by (15), we have $(m^*_{k_1}, \sigma^*_{k_1}) \neq (m^*_{k_2}, \sigma^*_{k_2})$, which yields a contradiction. Therefore, we know $\mathrm{str}^*_{k_1} \neq \mathrm{str}^*_{k_2}$. From the simulation of $\hat{\mathrm{H}}$, we have $\mathrm{Hid}(\mathrm{str}^*_{k_1}) \neq \mathrm{Hid}(\mathrm{str}^*_{k_2})$, and thus we have $|\mathcal{J}| = \ell + 1$.

We now show that for each $j \in \mathcal{J}$, it holds that

$$\alpha_j^{(0)} + \sum_{i \in \mathcal{I}_{\mathrm{fin}}} y_i \left(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)} \right) = \delta_j \left(\beta_j^{(0)} + \sum_{i \in \mathcal{I}_{\mathrm{fin}}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) \right) \ , \tag{C1}$$

$$\beta_j^{(0)} + \sum_{i \in \mathcal{I}_{\mathrm{fin}}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) \neq 0 \ , \tag{C2}$$

which implies $\mathcal{J}$ is valid for the WFROS game.

Let us fix a $j \in \mathcal{J}$. Since $j \in \mathcal{J}$, there exists $k \in [Q_{\mathrm{S}_1} + 1]$ such that $\mathrm{Hid}(\mathrm{str}^*_k) = j$, and there exists $P_j^A, P_j^Y \in \mathsf{Cur}$ and m_j such that $\mathrm{str}^*_k = \Xi(P_j^A) \,\|\, \Xi(P_j^Y) \,\|\, m_j$. Let $\hat{\alpha}_j$ and $\hat{\beta}_j$ denote the coefficients of P_j^A and P_j^Y. Since E_1 happens, by (16) and Lemma 9, we have $P_j^A =_L s^*_k - \delta_j \cdot y^*_k \cdot \mathsf{X}$, $P_j^Y =_L y^*_k \cdot \mathsf{X}$, which implies there exists $\{r_{1,i}^{P_j^A}, r_{2,i}^{P_j^A}, r_{1,i}^{P_j^Y}, r_{2,i}^{P_j^Y}\}_{i \in \mathcal{I}_{\mathrm{fin}}}$ such that

$$P_j^A = s^*_k - \delta_j \cdot y^*_k \cdot \mathsf{X} + \sum_{i \in [Q_{\mathrm{S}_1}]} r_{1,i}^{P_j^A} \left(\mathsf{A}_i + c_i \mathsf{Y}_i - s_i \right) + \sum_{i \in [Q_{\mathrm{S}_1}]} r_{2,i}^{P_j^A} \left(\mathsf{Y}_i - y_i \mathsf{X} \right) \ ,$$

$$P_j^Y = y^*_k \cdot \mathsf{X} + \sum_{i \in [Q_{\mathrm{S}_1}]} r_{1,i}^{P_j^Y} \left(\mathsf{A}_i + c_i \mathsf{Y}_i - s_i \right) + \sum_{i \in [Q_{\mathrm{S}_1}]} r_{2,i}^{P_j^Y} \left(\mathsf{Y}_i - y_i \mathsf{X} \right) \ . \tag{19}$$

By looking into the coefficients of X, $\{\mathsf{A}_i, \mathsf{Y}_i\}_{i \in [Q_{\mathrm{S}_1}]}$ on both sides of (19), we have $\hat{\alpha}_j^{\mathsf{A}_i} = r_{1,i}^{P_j^A}$, $\hat{\alpha}_j^{\mathsf{Y}_i} = r_{1,i}^{P_j^A} \cdot c_i + r_{2,i}^{P_j^A}$, $\hat{\beta}_j^{\mathsf{A}_i} = r_{1,i}^{P_j^Y}$, $\hat{\beta}_j^{\mathsf{Y}_i} = r_{1,i}^{P_j^Y}$ and $c_i + r_{2,i}^{P_j^Y}$ for each $i \in [Q_{\mathrm{S}_1}]$, $\hat{\beta}_j^{\mathsf{X}} = y^*_k - \sum_{i \in [Q_{\mathrm{S}_1}]} r_{2,i}^{P_j^Y} \cdot y_i$, and $\hat{\alpha}_j^{\mathsf{X}} = -\delta_j \cdot y^*_k - \sum_{i \in [Q_{\mathrm{S}_1}]} r_{2,i}^{P_j^A} \cdot y_i$. By sorting out the equations, we have

$$y^*_k = \beta_j^{\mathsf{X}} + \sum_{i \in \mathcal{I}_{\mathrm{fin}}} y_i \left(\hat{\beta}_j^{\mathsf{Y}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i} \right) \ ,$$

$$\hat{\alpha}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i \left(\hat{\alpha}_j^{\mathsf{Y}_i} - c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i} \right) = -\delta_j \cdot \left(\hat{\beta}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i \left(\hat{\beta}_j^{\mathsf{Y}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i} \right) \right) \ .$$

By the definition of $\vec{\alpha}$ and $\vec{\beta}$ in (14), we know (C1) holds and

$$y^*_k = \beta_j^{(0)} + \sum_{i \in \mathcal{I}_{\mathrm{fin}}} y_i \left(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)} \right) .$$

Since E_1 happens, by (16), we know $y^*_k \neq 0$, which implies (C2) happens. □

5 Efficient Blind Signatures in the AGM

We now present schemes that are secure in the *algebraic group model* (AGM) [FKL18]. This model considers security for *algebraic adversaries* - these are adversaries that, when used within a reduction, provide a representation of any group element they output in terms of all prior group elements input to the adversary. (We dispense with a more formal definition since the use of the AGM is self-evident in our proofs.)

Algorithm $\mathsf{BS}_3.\mathsf{Setup}(1^\lambda)$: $p \leftarrow \lvert\mathbb{G}_\lambda\rvert$; $g \leftarrow g(\mathbb{G}_\lambda)$ Select $\mathrm{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ Return $par \leftarrow (p, \mathbb{G}_\lambda, g, \mathrm{H})$ Algorithm $\mathsf{BS}_3.\mathsf{KG}(par)$: $(p, \mathbb{G}_\lambda, g, \mathrm{H}) \leftarrow par$ $x \leftarrow\$\ \mathbb{Z}_p$; $X \leftarrow g^x$; $Z \leftarrow\$\ \mathbb{G}_\lambda$ $sk \leftarrow x$; $pk \leftarrow (X, Z)$ Return (sk, pk) Algorithm $\mathsf{BS}_3.\mathsf{S}_1(sk)$: $x \leftarrow sk$; $X \leftarrow g^x$ $a, t \leftarrow\$\ \mathbb{Z}_p$; $y \leftarrow\$\ \mathbb{Z}_p^*$ $A \leftarrow g^a$; $C \leftarrow g^t Z^y$ $\mathsf{st}^s \leftarrow (a, y, t, x)$; $\mathsf{msg}_1 \leftarrow (A, C)$ Return $(\mathsf{st}^s, \mathsf{msg}_1)$ Algorithm $\mathsf{BS}_3.\mathsf{S}_2(\mathsf{st}^s, c)$: If $c = 0$ then return $\perp$ $(a, y, t, x) \leftarrow \mathsf{st}^s$ $s \leftarrow a + c \cdot y \cdot x$ Return $\mathsf{msg}_2 \leftarrow (s, y, t)$	Algorithm $\mathsf{BS}_3.\mathsf{U}_1(pk, \mathsf{msg}_1, m)$: $X \leftarrow pk$; $(A, C) \leftarrow \mathsf{msg}_1$ $r_1, r_2 \leftarrow\$\ \mathbb{Z}_p$; $\gamma_1, \gamma_2 \leftarrow\$\ \mathbb{Z}_p^*$ $A' \leftarrow g^{r_1} \cdot A^{\gamma_1/\gamma_2}$ $C' \leftarrow C^{\gamma_1} g^{r_2}$ $c' \leftarrow H(A' \,\\|\, C' \,\\|\, m)$ $c \leftarrow c' \cdot \gamma_2$ $\mathsf{st}^u \leftarrow (c, c', r_1, r_2, \gamma_1, \gamma_2, X, Z, A, C)$ Return (st^u, c) Algorithm $\mathsf{BS}_3.\mathsf{U}_2(\mathsf{st}^u, \mathsf{msg}_2)$: $(c, c', r_1, r_2, \gamma_1, \gamma_2, X, Z, A, C) \leftarrow \mathsf{st}^u$ $(s, y, t) \leftarrow \mathsf{msg}_2$ If $y = 0$ or $C \neq g^t Z^y$ or $g^s \neq A \cdot X^{c \cdot y}$ then return $\perp$ $s' \leftarrow (\gamma_1/\gamma_2) \cdot s + r_1$ $y' \leftarrow \gamma_1 \cdot y$ $t' \leftarrow \gamma_1 \cdot t + r_2$ Return $\sigma \leftarrow (c', s', y', t')$ Algorithm $\mathsf{BS}_3.\mathsf{Ver}(pk, \sigma, m)$: $(c, s, y, t) \leftarrow \sigma$ If $y = 0$ then return 0 $C \leftarrow g^t Z^y$; $A \leftarrow g^s \cdot X^{-c \cdot y}$ If $c \neq \mathrm{H}(A \,\\|\, C \,\\|\, m)$ then return 0 Return 1

Fig. 7. The blind signature scheme $\mathsf{BS}_3 = \mathsf{BS}_3[\mathbb{G}]$.

5.1 A Protocol Secure under the DL Assumption

In this section, we introduce a scheme, which we refer to as BS_3, that relies on the hardness of the (plain) discrete logarithm (DL) problem, which is formalized in Figure 8. In contrast to BS_1, our new scheme (described in Figure 7) requires an extra group element Z in the public key, and the commitment X^y in is replaced by $g^t Z^y$. (This will necessary result in an additional scalar in the signature.) However, one could generate Z as an output of a hash function (assuming the hash function is a random oracle, which we assume anyways), although, interestingly, our proof for BS_3 will show that blindness holds even when Z is chosen maliciously by the signer (who may consequently also know its discrete logarithm). In Appendix C, we present a slightly simpler alternative protocol, called BS_2, that avoids the need of such an extra group element, at the cost of relying on the hardness of a stronger assumption, the *one-more discrete logarithm* (OMDL) problem. (Needless to say, a scheme based on DL only is seen as more desirable than a scheme based on the OMDL assumption [KM08].)

The additional group element Z will in fact allow us to develop a *partially blind* version of BS_3, which we refer to as PBS, which we discuss in Section 6 below. We note that in fact *all* results about BS_3 can be obtained as a corollary of our analysis of PBS, because a blind signature scheme is of course a special case of a partially blind one. However, we are opting for a separate presentation, as the main ideas behind the reduction are much simpler to understand in (plain) BS_3, and the proof of PBS adds some extra complexity (in particular, in order to obtain a tighter bound), which obfuscates the main ideas.

Security analysis. The following theorem establishes the blindness of BS_3. (Its proof is very similar to the blindness proof of $\mathsf{BS}_1[\mathbb{G}]$, so we defer it to Appendix D.2.)

Theorem 4. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. Then, the blind signature scheme $\mathsf{BS}_3[\mathbb{G}]$ is perfectly blind.*

$$
\begin{array}{|l|}
\hline
\underline{\text{Game DLog}_{\mathbb{G}}^{\mathcal{A}}(\lambda):} \\
p \leftarrow |\mathbb{G}_\lambda|;\ g \leftarrow g(\mathbb{G}_\lambda) \\
X \leftarrow\$\ \mathbb{G}_\lambda \\
y \leftarrow \mathcal{A}(p, g, \mathbb{G}_\lambda, X) \\
\text{If } g^y = X \text{ then return } 1 \\
\text{Return } 0 \\
\hline
\end{array}
$$

Fig. 8. The DLog game.

$$
\begin{array}{|l|l|}
\hline
\underline{\text{Game OMUF}_{\mathsf{BS}_3[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}(\lambda):} & \underline{\text{Oracle S}_1:} \\
p \leftarrow |\mathbb{G}_\lambda|;\ g \leftarrow g(\mathbb{G}_\lambda);\ x \leftarrow\$\ \mathbb{Z}_p;\ X \leftarrow g^x;\ Z \leftarrow \mathbb{G}_\lambda & \text{sid} \leftarrow \text{sid} + 1 \\
\text{sid} \leftarrow 0;\ \ell \leftarrow 0;\ \mathcal{I}_{\text{fin}} \leftarrow \varnothing;\ T \leftarrow ();\ \text{hid} \leftarrow 0;\ \text{Hid} \leftarrow () & a_{\text{sid}}, t_{\text{sid}} \leftarrow\$\ \mathbb{Z}_p;\ y_{\text{sid}} \leftarrow\$\ \mathbb{Z}_p^* \\
\{(m_k^*, \sigma_k^*)\}_{k\in[\ell+1]} \leftarrow\$\ \mathcal{A}_{\text{alg}}^{\text{S}_1,\text{S}_2,\text{H}}(p, g, \mathbb{G}_\lambda, X, Z) & \mathsf{st}_{\text{sid}}^s \leftarrow (a_{\text{sid}}, y_{\text{sid}}, t_{\text{sid}}) \\
\text{If } \exists\ k_1 \neq k_2 \text{ such that } (m_{k_1}^*, \sigma_{k_1}^*) = (m_{k_2}^*, \sigma_{k_2}^*) \text{ then} & A_{\text{sid}} \leftarrow g^{a_{\text{sid}}} \\
\quad \text{Return } 0 & C_{\text{sid}} \leftarrow g^{t_{\text{sid}}} Z^{y_{\text{sid}}} \\
\text{If } \exists\ k \in [\ell+1] \text{ such that } y_k^* = 0 & \mathsf{msg}_1 \leftarrow (A_{\text{sid}}, C_{\text{sid}}) \\
\quad \text{or } c_k^* \neq \text{H}(g^{s_k^*} X^{-c_k^* \cdot y_k^*} \,\|\, g^{t_k^*} Z^{y_k^*} \,\|\, m_k^*) & \text{Return } (\text{sid}, \mathsf{msg}_1) \\
\text{where } (c_k^*, s_k^*, y_k^*, t_k^*) = \sigma_k^* \text{ then return } 0 & \underline{\text{Oracle S}_2(i, c_i):} \\
\text{Return } 1 & \text{If } i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}} \\
\underline{\text{Oracle H}(A \,\|\, C \,\|\, m):} & \quad \text{or } c_i = 0 \text{ then} \\
\text{If } T(A \,\|\, C \,\|\, m) = \perp \text{ then} & \quad \text{Return } \perp \\
\quad T(A \,\|\, C \,\|\, m) \leftarrow\$\ \mathbb{Z}_p & (a_i, y_i, t_i) \leftarrow \mathsf{st}_i^s \\
\quad \text{hid} \leftarrow \text{hid} + 1 & s_i \leftarrow a_i + c_i \cdot y_i \cdot x \\
\quad \text{Hid}(A \,\|\, C \,\|\, m) \leftarrow \text{hid} & \mathsf{msg}_2 \leftarrow (s_i, y_i, t_i) \\
\quad /\!/\ A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} Z^{\hat{\alpha}^{\mathsf{Z}}} \prod_{i\in[\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}} & \mathcal{I}_{\text{fin}} \leftarrow \mathcal{I}_{\text{fin}} \cup \{i\} \\
\quad /\!/\ C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} Z^{\hat{\beta}^{\mathsf{Z}}} \prod_{i\in[\text{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}} & \ell \leftarrow \ell + 1 \\
\quad \delta_{\text{hid}} \leftarrow T(A \,\|\, C \,\|\, m);\ \vec{\hat{\alpha}}_{\text{hid}} \leftarrow \vec{\hat{\alpha}};\ \vec{\hat{\beta}}_{\text{hid}} \leftarrow \vec{\hat{\beta}} & \text{Return } \mathsf{msg}_2 \\
\text{Return } T(A \,\|\, C \,\|\, m) & \\
\hline
\end{array}
$$

Fig. 9. The OMUF security game for the blind signature scheme $\mathsf{BS}_3[\mathbb{G}]$.

The core of the analysis is once again a proof that the scheme is one-more unforgeable in the AGM, i.e., we only prove security against algebraic adversaries. In particular, we model the selected hash function as a random oracle H, to which the adversary is given explicit access.

Theorem 5. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. For any algebraic adversary $\mathcal{A}_{\text{alg}}$ for the game $\text{OMUF}^{\mathsf{BS}_3[\mathbb{G}]}(\lambda)$ making at most Q_{S_1} queries to S_1 and Q_{H} queries to the random oracle H, there exists an adversary $\mathcal{B}_{\text{dlog}}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$
\mathsf{Adv}_{\mathsf{BS}_3[\mathbb{G}]}^{\text{omuf}}(\mathcal{A}_{\text{alg}}, \lambda) \leqslant 2\mathsf{Adv}_{\mathbb{G}}^{\text{dlog}}(\mathcal{B}_{\text{dlog}}, \lambda) + \frac{(Q_{\text{H}} + Q_{\text{S}_1} + 1)(Q_{\text{H}} + 3Q_{\text{S}_1} + 1)}{p-1}\,.
$$

Proof (of Theorem 5). Let us fix an adversary $\mathcal{A}_{\text{alg}}$ that makes at most Q_{S_1} queries to S_1 and Q_{H} queries to the random oracle H. Without loss of generality, assume $\mathcal{A}_{\text{alg}}$ makes exactly Q_{S_1} queries to S_1 and exactly one query (i, c_i) to S_2 for each $i \in [Q_{\text{S}_1}]$. Then, after $\mathcal{A}_{\text{alg}}$ returns, we know $\ell = Q_{\text{S}_1}$ and $\mathcal{I}_{\text{fin}} = [Q_{\text{S}_1}]$.

The $\text{OMUF}_{\mathsf{BS}_3[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game is formally defined in Figure 9. In addition to the original OMUF game (defined in Figure 1), for each query $(A \,\|\, C \,\|\, m)$ to H, its corresponding hid is recorded in $\text{Hid}(A \,\|\, Y \,\|\, m)$, and the output of the query is recorded as δ_{hid}. Also, since $\mathcal{A}_{\text{alg}}$ is algebraic, it also provides the representations of A and C, and the corresponding coefficient $\vec{\hat{\alpha}}$ and $\vec{\hat{\beta}}$ are recorded as $\vec{\hat{\alpha}}_{\text{hid}}$ and $\vec{\hat{\beta}}_{\text{hid}}$.

Denote the event WIN as $\mathcal{A}_{\text{alg}}$ wins the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game, i.e., all output message-signature pairs $\{m^*_k, \sigma^*_k\}_{k\in[Q_{\text{S}_1}+1]}$ are distinct and valid. Furthermore, let us denote $\text{str}^*_k := g^{s^*_k} X^{-c^*_k \cdot y^*_k} \,\|\, g^{t^*_k} Z^{y^*_k} \,\|\, m^*_k$. We let E be the event in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game for which, after the validity of the output is checked, for each $k \in [Q_{\text{S}_1}+1]$ and $j = \text{Hid}(\text{str}^*_k)$,[8] the following conditions hold:

$$\hat{\alpha}^{\mathsf{X}}_j - \sum_{i\in[Q_{\text{S}_1}]} y_i \cdot c_i \cdot \hat{\alpha}^{\mathsf{A}_i}_j = -\delta_j \cdot y^*_k \;, \tag{20}$$

$$\hat{\beta}^{\mathsf{Z}}_j + \sum_{i\in[Q_{\text{S}_1}]} y_i \cdot \hat{\beta}^{\mathsf{C}_i}_j = y^*_k \;. \tag{21}$$

Since $\mathsf{Adv}^{\text{omuf}}_{\mathsf{BS}_3[\mathbb{G}]}(\mathcal{A}_{\text{alg}}, \lambda) = \mathsf{Pr}[\text{WIN}] = \mathsf{Pr}[\text{WIN} \wedge E] + \mathsf{Pr}[\text{WIN} \wedge (\neg E)]$, the theorem follows by combining the following two lemmas with Theorem 1.

Lemma 10. *There exists an adversary $\mathcal{B}_{\text{wfros}}$ for the $\text{WFROS}_{Q_{\text{S}_1},p}$ problem making at most $Q_{\text{H}} + Q_{\text{S}_1} + 1$ queries to the random oracle* H *such that*

$$\mathsf{Adv}^{\text{wfros}}_{Q_{\text{S}_1},p}(\mathcal{B}_{\text{wfros}}) \geqslant \mathsf{Pr}[\text{WIN} \wedge E] \;. \tag{22}$$

Lemma 11. *There exists an adversary $\mathcal{B}_{\text{dlog}}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$\mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}_{\text{dlog}}, \lambda) \geqslant \frac{1}{2}\mathsf{Pr}[\text{WIN} \wedge (\neg E)] \;. \tag{23}$$

□

5.2 Proof of Lemma 10

Proof. We first give a detailed description of $\mathcal{B}_{\text{wfros}}$ playing the game $\text{WFROS}_{Q_{\text{S}_1},p}$. To start with, $\mathcal{B}_{\text{wfros}}$ initializes sid, $\mathcal{I}_{\text{fin}}$, ℓ, T, hid, and Hid as described in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game. In addition, $\mathcal{B}_{\text{wfros}}$ samples x, z uniformly from $\mathbb{Z}_p$, sets X to g^x and Z to g^z.

Then, $\mathcal{B}_{\text{wfros}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X, Z)$, and with access to the oracles $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{S}}_1$: Same as the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game except that instead of sampling y_{sid}, t_{sid} randomly and setting $C_{\text{sid}} \leftarrow g^{t_{\text{sid}}} X^{y_{\text{sid}}}$, $\mathcal{B}_{\text{wfros}}$ samples a new variable t'_{sid} uniformly from $\mathbb{Z}_p$ and sets $C_{\text{sid}} = g^{t'_{\text{sid}}}$.

Oracle $\hat{\mathbf{S}}_2$: After receiving a query (i, c_i) to $\hat{\text{S}}_2$ from $\mathcal{A}_{\text{alg}}$, if $i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}}$ or $c_i = 0$, $\mathcal{B}_{\text{wfros}}$ returns $\perp$. Otherwise, $\mathcal{B}_{\text{wfros}}$ makes a query (i, c_i) to S and uses its output as the value y_i. Also, $\mathcal{B}_{\text{wfros}}$ sets $t_i = t'_i - y_i \cdot z$. With the value (a_i, y_i, t_i), the rest of $\hat{\text{S}}_2$ is the same as S_2 in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game.

Oracle $\hat{\mathbf{H}}$: After receiving a query $(A \,\|\, C \,\|\, m)$ to $\hat{\text{H}}$ from $\mathcal{A}_{\text{alg}}$, if $T(A \,\|\, C \,\|\, m) \neq \perp$, the value $T(A \,\|\, C \,\|\, m)$ is returned. Otherwise, since $\mathcal{A}_{\text{alg}}$ is algebraic, $\mathcal{B}_{\text{wfros}}$ also knows the coefficient $\vec{\hat{\alpha}}$ and $\vec{\hat{\beta}}$ such that

$$A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} \prod_{i\in[\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}} \;,\; C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} \prod_{i\in[\text{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}} \;.$$

Then, $\mathcal{B}_{\text{wfros}}$ issues the query $(\vec{\alpha}, \vec{\beta})$ to H, where $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2Q_{\text{S}_1}+1}$ are such that

$$\alpha^{(i')} = \begin{cases} \hat{\alpha}^{\mathsf{X}} \;, & i' = 0 \\ -\hat{\alpha}^{\mathsf{A}_i} \;, & i' = 2i \;,\; i \in [\text{sid}] \\ 0 \;, & o.w. \end{cases} \;,$$
$$\beta^{(i')} = \begin{cases} -\hat{\beta}^{\mathsf{Z}} \;, & i' = 0 \\ -\hat{\beta}^{\mathsf{C}_i} \;, & i' = 2i-1 \;,\; i \in [\text{sid}] \\ 0 \;, & o.w. \end{cases} \;. \tag{24}$$

[8] Here, $\text{Hid}(\text{str}^*_k)$ must be defined since a query str^*_k is made to H when checking the validity of the output (m^*_k, σ^*_k).

After receiving the output $(\delta_{\text{hid}}, \text{hid})$, $\mathcal{B}_{\text{wfros}}$ sets $T(A \,\|\, C \,\|\, m) \leftarrow \delta_{\text{hid}}$ and $\text{Hid}(A \,\|\, C \,\|\, m) \leftarrow \text{hid}$. Finally, $\mathcal{B}_{\text{wfros}}$ returns $T(A \,\|\, C \,\|\, m)$.

After $\mathcal{A}_{\text{alg}}$ outputs $\{(m_k^*, \sigma_k^*)\}_{k \in [Q_{\text{S}_1}+1]}$, $\mathcal{B}_{\text{wfros}}$ aborts if the conditions from the event $\text{WIN} \wedge E$ do not occur. Otherwise, $\mathcal{B}_{\text{wfros}}$ outputs $\mathcal{J} := \{\text{Hid}(\text{str}_k^*) \mid k \in [Q_{\text{S}_1}+1]\}$. Since $\mathcal{B}_{\text{wfros}}$ simulates the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game perfectly, the probability that $\text{WIN} \wedge E$ occurs when running $\mathcal{B}_{\text{wfros}}$ is the same as in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game with $\mathcal{A}_{\text{alg}}$.

Following the similar analysis of $\mathcal{B}$ in the GGM proof (Section 4.2), we know $\mathcal{B}_{\text{wfros}}$ makes at most $Q_{\text{H}} + Q_{\text{S}_1} + 1$ queries to H.

It is left to show that if $\text{WIN} \wedge E$ occurs within the simulation, then $\mathcal{B}_{\text{wfros}}$ wins the WFROS game. We first show that $|\mathcal{J}| = Q_{\text{S}_1} + 1$. Suppose $|\mathcal{J}| \leqslant Q_{\text{S}_1}$. Since $\mathcal{A}_{\text{alg}}$ wins the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game, we know there exists $k_1, k_2 \in [Q_{\text{S}_1}+1]$ such that $k_1 \neq k_2$ and $\text{Hid}(\text{str}_{k_1}^*) = \text{Hid}(\text{str}_{k_2}^*)$, which implies $\text{str}_{k_1}^* = \text{str}_{k_2}^*$. Therefore, we have

$$g^{s_{k_1}^*} X^{-c_{k_1}^* \cdot y_{k_1}^*} = g^{s_{k_2}^*} X^{-c_{k_2}^* \cdot y_{k_2}^*} \; , \; g^{t_{k_1}^*} Z^{y_{k_1}^*} = g^{t_{k_2}^*} Z^{y_{k_2}^*} \; , \; m_{k_1}^* = m_{k_2}^* \; . \tag{25}$$

Also, let $j = \text{Hid}(\text{str}_{k_1}^*) = \text{Hid}(\text{str}_{k_2}^*)$. Since E occurs in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game simulated by $\mathcal{B}_{\text{wfros}}$, by (20), we have

$$y_{k_1}^* = \hat{\beta}_j^{\mathsf{X}} + \sum_{i \in [Q_{\text{S}_1}]} y_i(\hat{\beta}_j^{\mathsf{C}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i}) = y_{k_2}^* \; .$$

Since $y_{k_1}^* = y_{k_2}^*$ and $c_{k_1}^* = c_{k_2}^*$, by (25), we have

$$t_{k_1}^* = t_{k_2}^* \; , \; s_{k_1}^* = s_{k_2}^* \; .$$

However, since $(m_{k_1}^*, \sigma_{k_1}^*)$ and $(m_{k_2}^*, \sigma_{k_2}^*)$ are different message-signature pairs, we have

$$(m_{k_1}^*, c_{k_1}^*, s_{k_1}^*, y_{k_1}^*, t_{k_1}^*) \neq (m_{k_2}^*, c_{k_2}^*, s_{k_2}^*, y_{k_2}^*, t_{k_2}^*),$$

which yields a contradiction. Therefore, we have $|\mathcal{J}| = Q_{\text{S}_1} + 1$.

Then, since in particular E occurs, by (20) and (21), it holds that for any $j \in \mathcal{J}$

$$\alpha_j^{\mathsf{X}} - \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot c_i \cdot \alpha_j^{\mathsf{A}_i} = -\delta_j \left(\hat{\beta}_j^{\mathsf{Z}} + \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot \hat{\beta}_j^{\mathsf{C}_i} \right) .$$

From the simulation of $\hat{\text{H}}$, by (24), we have for any $j \in \mathcal{J}$

$$\alpha_j^{(0)} + \sum_{i \in [Q_{\text{S}_1}]} y_i(\alpha_j^{(2i-1)} + c_i \cdot \alpha_j^{(2i)}) = \delta_j \left(\beta_j^{(0)} + \sum_{i \in [Q_{\text{S}_1}]} y_i(\beta_j^{(2i-1)} + c_i \cdot \beta_j^{(2i)}) \right) .$$

Therefore, $\mathcal{B}_{\text{wfros}}$ wins the $\text{WFROS}_{Q_{\text{S}_1}, p}$ game, which concludes the proof. □

5.3 Proof of Lemma 11

Proof. We first partition the event $\text{WIN} \wedge (\neg E)$ into two cases. Denote F_1 as the event in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game that there exists $k \in [Q_{\text{S}_1}+1]$ such that (20) does not hold, and denote F_2 as the event that there exists $k \in [Q_{\text{S}_1}+1]$ such that (21) does not hold. Then, if E does not occur, we know either F_1 or F_2 occurs. Therefore, we have $\text{WIN} \wedge (\neg E) = (\text{WIN} \wedge F_1) \vee (\text{WIN} \wedge F_2)$. We then prove the following two claims.

Claim 6 *There exists $\mathcal{B}^{(0)}_{\text{dlog}}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$\mathsf{Pr}[\text{WIN} \wedge F_1] \leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(0)}_{\text{dlog}}, \lambda) \; . \tag{26}$$

Claim 7 *There exists $\mathcal{B}^{(1)}_{\text{dlog}}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$\mathsf{Pr}[\text{WIN} \ \wedge \ F_2] \leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(1)}_{\text{dlog}}, \lambda) \ . \tag{27}$$

By the above two claims, we can construct an adversary $\mathcal{B}_{\text{dlog}}$ for the DLog problem that runs either $\mathcal{B}^{(0)}_{\text{dlog}}$ or $\mathcal{B}^{(1)}_{\text{dlog}}$ with $1/2$ probability, and we can conclude the lemma since

$$\begin{aligned}\mathsf{Pr}[\text{WIN} \ \wedge \ (\neg E)] &\leqslant \mathsf{Pr}[\text{WIN} \ \wedge \ F_1] + \mathsf{Pr}[\text{WIN} \ \wedge \ F_2] \\ &\leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(0)}_{\text{dlog}}, \lambda) + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(1)}_{\text{dlog}}, \lambda) = 2\mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}_{\text{dlog}}, \lambda).\end{aligned}$$

□

Proof (of Claim 6). We first give a detailed description of $\mathcal{B}^{(0)}_{\text{dlog}}$ playing the $\text{DLog}_{\mathbb{G}}$ game.

THE ADVERSARY $\mathcal{B}^{(0)}_{\text{dlog}}$. Initially, $\mathcal{B}^{(0)}_{\text{dlog}}$ initializes sid, $\mathcal{I}_{\text{fin}}$, ℓ, T, hid, and Hid as described in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game. After $\mathcal{B}^{(0)}_{\text{dlog}}$ receives $(p, g, \mathbb{G}_\lambda, W)$ from the $\text{DLog}_{\mathbb{G}}$ game, $\mathcal{B}^{(0)}_{\text{dlog}}$ samples z uniformly from $\mathbb{Z}_p$ and sets $X \leftarrow W, Z \leftarrow g^z$. Then, $\mathcal{B}^{(0)}_{\text{dlog}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{S}}_1$: $\mathcal{B}^{(0)}_{\text{dlog}}$ samples $s_{\text{sid}}, t'_{\text{sid}}$ uniformly from $\mathbb{Z}_p$ and y'_{sid} unifomly from $\mathbb{Z}_p^*$ and sets $A_{\text{sid}} = g^{s_{\text{sid}}} X^{-y'_{\text{sid}}}$ and $C_{\text{sid}} = g^{t'_{\text{sid}}}$. Then, $\mathcal{B}^{(0)}_{\text{dlog}}$ returns $(\text{sid}, A_{\text{sid}}, C_{\text{sid}})$.

Oracle $\hat{\mathbf{S}}_2$: Same as in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game if $i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}}$ or $c_i = 0$. Otherwise, after receving a query (i, c_i) to $\hat{\text{S}}_2$ from $\mathcal{A}_{\text{alg}}$, $\mathcal{B}^{(0)}_{\text{dlog}}$ sets $y_i \leftarrow y'_i / c_i$ and $t_i \leftarrow t'_i - y_i \cdot z$. Then, $\mathcal{B}^{(0)}_{\text{dlog}}$ returns (s_i, y_i, t_i) to $\mathcal{A}_{\text{alg}}$.

Oracle $\hat{\mathbf{H}}$: Same as in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game.

After receiving the output $\{(m^*_k, \sigma^*_k)\}_{k \in [Q_{\text{S}_1}+1]}$, $\mathcal{B}^{(0)}_{\text{dlog}}$ aborts if the event WIN $\wedge$ F_1 does not occur.

It is clear that $\mathcal{B}^{(0)}_{\text{dlog}}$ simulates the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game perfectly. Therefore, it is left to show that if the event WIN $\wedge$ F_1 occurs within the simulation, $\mathcal{B}^{(0)}_{\text{dlog}}$ can compute the discrete log of X, which equals to W.

Suppose WIN $\wedge$ F_1 occurs. There exists $k \in [Q_{\text{S}_1} + 1]$ and $j = \text{Hid}(\text{str}^*_k)$ such that (20) does not hold. Since $\text{Hid}(\text{str}^*_k) = j$ and $\delta_j = c^*_k$, we have

$$g^{s^*_k} X^{-\delta_j \cdot y^*_k} = g^{s^*_k} X^{-c^*_k \cdot y^*_k} = g^{\hat{\alpha}^g_j} X^{\hat{\alpha}^{\mathsf{X}}_j} Z^{\hat{\alpha}^{\mathsf{Z}}_j} \prod_{i \in [\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}_j} C_i^{\hat{\alpha}^{\mathsf{C}_i}_j} \ . \tag{28}$$

Similar to the preceding case, since $\mathcal{B}^{(0)}_{\text{dlog}}$ knows the discrete log of Z as z, by substituting $A_i = g^{s_i} X^{-c_i \cdot y_i}$, $C_i = g^{t_i} Z^{y_i}$, and $Z = g^z$ into (28), we have

$$g^{s^*_k} X^{-\delta_j \cdot y^*_k} = g^{\hat{\alpha}^g_j + \hat{\alpha}^{\mathsf{Z}}_j \cdot z + \sum_{i \in [Q_{\text{S}_1}]} (\hat{\alpha}^{\mathsf{A}_i}_j \cdot s_i + \hat{\alpha}^{\mathsf{C}_i}_j \cdot (t_i + y_i \cdot z))} X^{\hat{\alpha}^{\mathsf{X}}_j - \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot c_i \cdot \hat{\alpha}^{\mathsf{A}_i}_j} \ .$$

Since (20) does not hold, $\mathcal{B}^{(0)}_{\text{dlog}}$ can compute the discrete log of X as

$$x := \frac{s^*_k - \hat{\alpha}^g_j - \hat{\alpha}^{\mathsf{Z}}_j \cdot z - \sum_{i \in [Q_{\text{S}_1}]} (\hat{\alpha}^{\mathsf{A}_i}_j \cdot s_i + \hat{\alpha}^{\mathsf{C}_i}_j \cdot (t_i + y_i \cdot z))}{\hat{\alpha}^{\mathsf{X}}_j - \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot c_i \cdot \hat{\alpha}^{\mathsf{A}_i}_j + \delta_j \cdot y^*_k} \ .$$

□

Proof (of Claim 7). We first give a detailed description of $\mathcal{B}^{(1)}_{\text{dlog}}$ playing the $\text{DLog}_{\mathbb{G}}$ game.

THE ADVERSARY $\mathcal{B}^{(1)}_{\text{dlog}}$. Initially, $\mathcal{B}^{(1)}_{\text{dlog}}$ initializes sid, $\mathcal{I}_{\text{fin}}$, ℓ, T, hid, and Hid as described in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game. After $\mathcal{B}^{(1)}_{\text{dlog}}$ receives $(p, g, \mathbb{G}_\lambda, W)$ from the $\text{DLog}_{\mathbb{G}}$ game, $\mathcal{B}^{(1)}_{\text{dlog}}$ samples x uniformly from $\mathbb{Z}_p$ and sets $X \leftarrow g^x, Z \leftarrow W$. Then, $\mathcal{B}^{(1)}_{\text{dlog}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$. Since $\mathcal{B}^{(1)}_{\text{dlog}}$ knows $X = g^x$, $\mathcal{B}^{(1)}_{\text{dlog}}$ can simulate all the oracles $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$ the same as in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game. After receiving the output $\{(m^*_k, \sigma^*_k)\}_{k \in [Q_{\text{S}_1}+1]}$, $\mathcal{B}^{(1)}_{\text{dlog}}$ aborts if the event $\text{WIN} \wedge F_2$ does not occur.

It is clear that $\mathcal{B}^{(1)}_{\text{dlog}}$ simulates the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_3[\mathbb{G}]}$ game perfectly. Therefore, it is left to show that if the event $\text{WIN} \wedge F_2$ occurs within the simulation, $\mathcal{B}^{(1)}_{\text{dlog}}$ can compute the discrete log of Z, which equals to W.

Suppose $\text{WIN} \wedge F_2$ occurs. There exists $k \in [Q_{\text{S}_1} + 1]$ and $j = \text{Hid}(\text{str}^*_k)$ such that (21) does not hold. Since $\text{Hid}(\text{str}^*_k) = j$, we have

$$g^{t^*_k} Z^{y^*_k} = g^{\hat{\beta}^g_j} X^{\hat{\beta}^{\mathsf{X}}_j} Z^{\hat{\beta}^{\mathsf{Z}}_j} \prod_{i \in [\text{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}_j} C_i^{\hat{\beta}^{\mathsf{C}_i}_j} \ . \tag{29}$$

From the simulation of $\hat{\text{S}}_2$, for each $i \in [Q_{\text{S}_1}]$, we have

$$g^{s_i} = A_i X^{c_i \cdot y_i} \ , \ g^{t_i} = C_i Z^{-y_i} \ .$$

Also, $\mathcal{B}^{(1)}_{\text{dlog}}$ knows the discrete log of X as x. By substituting $A_i = g^{s_i} X^{-c_i \cdot y_i}$, $C_i = g^{t_i} Z^{y_i}$, and $X = g^x$ into (29), we have

$$g^{t^*_k} Z^{y^*_k} = g^{\hat{\beta}^g_j + \hat{\beta}^{\mathsf{X}}_j \cdot x + \sum_{i \in [Q_{\text{S}_1}]} (\hat{\beta}^{\mathsf{A}_i}_j \cdot (s_i - c_i \cdot y_i \cdot x) + \hat{\beta}^{\mathsf{C}_i}_j \cdot t_i)} Z^{\hat{\beta}^{\mathsf{Z}}_j + \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot \hat{\beta}^{\mathsf{C}_i}} \ .$$

Since (21) does not hold, $\mathcal{B}^{(1)}_{\text{dlog}}$ can compute the discrete log of Z as

$$z := \frac{t^*_k - \hat{\beta}^g_j - \hat{\beta}^{\mathsf{X}}_j \cdot x - \sum_{i \in [Q_{\text{S}_1}]} (\hat{\beta}^{\mathsf{A}_i}_j \cdot (s_i - c_i \cdot y_i \cdot x) + \hat{\beta}^{\mathsf{C}_i}_j \cdot t_i)}{\hat{\beta}^{\mathsf{Z}}_j + \sum_{i \in [Q_{\text{S}_1}]} y_i \cdot \hat{\beta}^{\mathsf{C}_i} - y^*_k} \ .$$

□

6 Partially Blind Signatures

This section presents our partially blind signature scheme, PBS, which is detailed in Figure 10. The scheme builds on top of the BS_3 scheme by replacing the extra generator Z contained in the public key with the output of a hash function F (also modeled as a random oracle in the OMUF proof) applied to the public input info. We do not formally redefine the syntax of partially blind signatures, but we note that it simply extends that of blind signatures by adding the extra input $\mathsf{info} \in \{0,1\}^*$ to the signer, the user, and the verification algorithm.

BLINDNESS. We first study the blindness of PBS. The $\text{PBlind}^{\mathcal{A}}_{\mathsf{PBS}}$ game is defined in Figure 11. The only difference between PBlind and Blind is that initially, the adversary $\mathcal{A}$ also picks a public information info and interacts with $\mathsf{PBS.U}_1$ and $\mathsf{PBS.U}_2$ for signing (info, m_0) and (info, m_1). Denote the advantage of the adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{pblind}}_{\mathsf{PBS}}(\mathcal{A}, \lambda) := \left| \Pr[\text{PBlind}^{\mathcal{A}}_{\mathsf{PBS}}(\lambda) = 1] - \frac{1}{2} \right| \ .$$

We say a partially blind signature scheme PBS is perfectly blind if and only if $\mathsf{Adv}^{\text{pblind}}_{\mathsf{PBS}}(\mathcal{A}) = 0$ for any $\mathcal{A}$.

Theorem 6. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. The partially blind signature scheme $\mathsf{PBS}[\mathbb{G}]$ is perfectly blind.*

Since the algorithm $\mathsf{PBS.U}_1$ and $\mathsf{PBS.U}_2$ are almost the same as $\mathsf{BS}_3.\mathsf{U}_1$ and $\mathsf{BS}_3.\mathsf{U}_2$, we can use a proof similar to the one for BS_3 (Section 5.1) to show $\mathsf{PBS}[\mathbb{G}]$ is perfectly blind. The only difference is that in BS_3, Z is given in the public key, while in $\mathsf{PBS}[\mathbb{G}]$, Z is given by $\text{F}(\mathsf{info})$.

Algorithm $\mathsf{PBS.Setup}(1^\lambda)$:
$p \leftarrow |\mathbb{G}_\lambda|$; $g \leftarrow g(\mathbb{G}_\lambda)$
Select $\mathrm{H} : \{0,1\}^* \to \mathbb{Z}_p^*$
Select $\mathrm{F} : \{0,1\}^* \to \mathbb{G}_\lambda$
Return $par \leftarrow (p, \mathbb{G}_\lambda, g, \mathrm{H}, \mathrm{F})$

Algorithm $\mathsf{PBS.KG}(par)$:
$(p, \mathbb{G}_\lambda, g, \mathrm{H}, \mathrm{F}) \leftarrow par$
$x \leftarrow_{\$} \mathbb{Z}_p$; $X \leftarrow g^x$
$sk \leftarrow x$; $pk \leftarrow X$
Return (sk, pk)

Algorithm $\mathsf{PBS.S_1}(sk, \mathsf{info})$:
$x \leftarrow sk$; $X \leftarrow g^x$; $Z \leftarrow \mathrm{F}(\mathsf{info})$
$a, t \leftarrow_{\$} \mathbb{Z}_p$; $y \leftarrow_{\$} \mathbb{Z}_p^*$
$A \leftarrow g^a$; $C \leftarrow g^t Z^y$
$\mathsf{st}^s \leftarrow (a, y, t, x)$; $\mathsf{msg}_1 \leftarrow (A, C)$
Return $(\mathsf{st}^s, \mathsf{msg}_1)$

Algorithm $\mathsf{PBS.S_2}(\mathsf{st}^s, c)$:
If $c = 0$ then return $\perp$
$(a, y, t, x) \leftarrow \mathsf{st}^s$
$s \leftarrow a + c \cdot y \cdot x$
Return $\mathsf{msg}_2 \leftarrow (s, y, t)$

Algorithm $\mathsf{PBS.U_1}(pk, \mathsf{msg}_1, \mathsf{info}, m)$:
$X \leftarrow pk$; $(A, C) \leftarrow \mathsf{msg}_1$; $Z \leftarrow \mathrm{F}(\mathsf{info})$
$r_1, r_2 \leftarrow_{\$} \mathbb{Z}_p$; $\gamma_1, \gamma_2 \leftarrow_{\$} \mathbb{Z}_p^*$
$A' \leftarrow g^{r_1} \cdot A^{\gamma_1/\gamma_2}$
$C' \leftarrow C^{\gamma_1} g^{r_2}$
$c' \leftarrow \mathrm{H}(\mathsf{info} \,\|\, A' \,\|\, C' \,\|\, m)$
$c \leftarrow c' \cdot \gamma_2$
$\mathsf{st}^u \leftarrow (c, c', r_1, r_2, \gamma_1, \gamma_2, X, Z, A, C)$
Return (st^u, c)

Algorithm $\mathsf{PBS.U_2}(\mathsf{st}^u, \mathsf{msg}_2)$:
$(c, c', r_1, r_2, \gamma_1, \gamma_2, X, Z, A, C) \leftarrow \mathsf{st}^u$
$(s, y, t) \leftarrow \mathsf{msg}_2$
If $y = 0$ or $C \neq g^t Z^y$ or $g^s \neq A \cdot X^{c \cdot y}$
  then return $\perp$
$s' \leftarrow (\gamma_1/\gamma_2) \cdot s + r_1$
$y' \leftarrow \gamma_1 \cdot y$
$t' \leftarrow \gamma_1 \cdot t + r_2$
Return $\sigma \leftarrow (c', s', y', t')$

Algorithm $\mathsf{PBS.Ver}(pk, \mathsf{info}, \sigma, m)$:
$X \leftarrow pk$; $Z \leftarrow \mathrm{F}(\mathsf{info})$; $(c, s, y, t) \leftarrow \sigma$
If $y = 0$ then return 0
$C \leftarrow g^t Z^y$; $A \leftarrow g^s \cdot X^{-c \cdot y}$
If $c \neq \mathrm{H}(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m)$ then return 0
Return 1

Fig. 10. The partially blind signature scheme $\mathsf{PBS} = \mathsf{PBS}[\mathbb{G}]$.

OMUF security. We next study the OMUF security of PBS. Note that the definition must also be adjusted: The main difference is that the adversary wins as long as it can produce $\ell + 1$ valid message-signature pairs for some info for which it has run only ℓ signing sessions, regardless of how many signing sessions are run with $\mathsf{info}' \neq \mathsf{info}$ (i.e., their number could be higher than ℓ). The corresponding game is defined in Figure 12, for the specific case of the scheme PBS. We prove the following theorem.

Theorem 7. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic adversary for the game $\mathrm{OMUF}^{\mathsf{PBS}[\mathbb{G}]}(\lambda)$ such that for each public information* info*, makes at most Q_{S_1} queries to S_1 and Q_{H} queries to the random oracle* H *that start with* info*. Also, let the total number of distinct public information* info*'s queried by $\mathcal{A}_{\mathrm{alg}}$ to S_1 be bounded by Q_{info}. Then, there exists an adversary $\mathcal{B}_{\mathrm{dlog}}$ for the* DLog *problem running in similar running time as $\mathcal{A}_{\mathrm{alg}}$ such that*

$$\mathsf{Adv}^{\mathrm{omuf}}_{\mathsf{PBS}[\mathbb{G}]}(\mathcal{A}_{\mathrm{alg}}, \lambda) \leqslant 2\mathsf{Adv}^{\mathrm{dlog}}_{\mathbb{G}}(\mathcal{B}_{\mathrm{dlog}}, \lambda) + \frac{Q_{\mathsf{info}}(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)(Q_{\mathrm{H}} + 3Q_{\mathrm{S}_1} + 1) + 2}{p - 1} \,.$$

The proof is very similar to that for BS_3 except we need to additionally perform a hybrid argument over queries to F, guessing which info will be the one leading to a one-more forgery. However, we need to work harder here to ensure the discrete logarithm avantage does not scale with Q_{info}.

We also note that we have no argument supporting the fact that the information-theoretic term in Theorem 7 is tight and the inclusion of info in H is necessary. However, a tighter analysis appears to require studying a more general version of WFROS. We leave this to future work.

6.1 Proof of Theorem 7

Proof. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic adversary described in the theorem. The $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game is formally defined in Figure 12. Without loss of generality, we assume that if $\mathcal{A}_{\mathrm{alg}}$ outputs the public information info^*,

$$
\begin{array}{l|l}
\underline{\text{Game PBlind}^{\mathcal{A}}_{\mathsf{PBS}}(\lambda):} & \underline{\text{Oracle U}_1(i, \mathsf{msg}_1^{(i)}):} \\
par \leftarrow \mathsf{BS.Setup}(1^\lambda) & \text{If } i \notin \{0,1\} \text{ or } \mathsf{sess}_i \neq \texttt{init} \text{ then return } \bot \\
b \leftarrow_\$ \{0,1\};\ b_0 \leftarrow b;\ b_1 \leftarrow 1-b & \mathsf{sess}_i \leftarrow \texttt{open} \\
b' \leftarrow_\$ \mathcal{A}^{\text{INIT},\text{U}_1,\text{U}_2}(par) & (\mathsf{st}_i^u, \mathsf{chl}^{(i)}) \leftarrow \mathsf{PBS.U}_1(pk, \mathsf{msg}_1^{(i)}, \mathsf{info}, m_{b_i}) \\
\text{If } b' = b \text{ then return } 1 & \text{Return } \mathsf{chl}^{(i)} \\
\text{Return } 0 & \underline{\text{Oracle U}_2(i, \mathsf{msg}_2^{(i)}):} \\
\underline{\text{Oracle INIT}(\tilde{pk}, \tilde{\mathsf{info}}, \tilde{m_0}, \tilde{m_1}):} & \text{If } i \notin \{0,1\} \text{ or } \mathsf{sess}_i \neq \texttt{open} \text{ then return } \bot \\
\mathsf{sess}_0 \leftarrow \texttt{init} & \mathsf{sess}_i \leftarrow \texttt{closed} \\
\mathsf{sess}_1 \leftarrow \texttt{init} & \sigma_{b_i} \leftarrow \mathsf{PBS.U}_2(\mathsf{st}_i^u, \mathsf{msg}_2^{(i)}) \\
pk \leftarrow \tilde{pk} & \text{If } \mathsf{sess}_0 = \mathsf{sess}_1 = \texttt{closed} \text{ then} \\
\mathsf{info} \leftarrow \tilde{\mathsf{info}}\ m_0 \leftarrow \tilde{m_0};\ m_1 \leftarrow \tilde{m_1} & \quad \text{If } \sigma_0 = \bot \text{ or } \sigma_1 = \bot \text{ then return } (\bot, \bot) \\
& \quad \text{Return } (\sigma_0, \sigma_1) \\
& \text{Return } (i, \texttt{closed})
\end{array}
$$

Fig. 11. The PBlind security game for a partially blind signature scheme PBS.

$$
\begin{array}{l|l}
\underline{\text{Game OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}(\lambda):} & \underline{\text{Oracle S}_1(\mathsf{info}):} \\
p \leftarrow |\mathbb{G}_\lambda|;\ g \leftarrow g(\mathbb{G}_\lambda);\ x \leftarrow_\$ \mathbb{Z}_p;\ X \leftarrow g^x & Z \leftarrow \text{F}(\mathsf{info}) \\
\text{sid} \leftarrow 0;\ \mathcal{I}_{\text{fin}} \leftarrow \varnothing;\ T_1 \leftarrow ();\ T_2 \leftarrow () & \text{sid} \leftarrow \text{sid}+1;\ \mathsf{info}_{\text{sid}} \leftarrow \mathsf{info} \\
\ell \leftarrow \text{a table where all entry are initially set to } 0 & a_{\text{sid}}, t_{\text{sid}} \leftarrow_\$ \mathbb{Z}_p;\ y_{\text{sid}} \leftarrow_\$ \mathbb{Z}_p^* \\
\text{fid} \leftarrow 0;\ \text{Fid} \leftarrow ();\ \text{Hid} \leftarrow () & \mathsf{st}^s_{\text{sid}} \leftarrow (a_{\text{sid}}, y_{\text{sid}}, t_{\text{sid}}) \\
(\mathsf{info}^*, \{(m_k^*, \sigma_k^*)\}_{k \in [\ell(\mathsf{info}^*)+1]}) \leftarrow_\$ \mathcal{A}_{\text{alg}}^{\text{S}_1,\text{S}_2,\text{H},\text{F}}(p, g, \mathbb{G}_\lambda, X) & A_{\text{sid}} \leftarrow g^{a_{\text{sid}}};\ C_{\text{sid}} \leftarrow g^{t_{\text{sid}}} Z^{y_{\text{sid}}} \\
\text{If } \exists\ k_1 \neq k_2 \text{ such that } (m_{k_1}^*, \sigma_{k_1}^*) = (m_{k_2}^*, \sigma_{k_2}^*) \text{ then} & \mathsf{msg}_1 \leftarrow (A_{\text{sid}}, C_{\text{sid}}) \\
\quad \text{Return } 0 & \text{Return } (\text{sid}, \mathsf{msg}_1) \\
\text{If } \exists\ k \in [\ell(\mathsf{info}^*)+1] \text{ such that } y_k^* = 0 & \underline{\text{Oracle S}_2(i, c_i):} \\
\quad \text{or } c_k^* \neq \text{H}(\mathsf{info}^* \,\|\, g^{s_k^*} X^{-c_k^* \cdot y_k^*} \,\|\, g^{t_k^*} Z^{y_k^*} \,\|\, m_k^*) & \text{If } i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}} \text{ or } c_i = 0 \text{ then} \\
\quad \text{where } (c_k^*, s_k^*, y_k^*, t_k^*) = \sigma_k^* \text{ and } Z = \text{F}(\mathsf{info}^*) & \quad \text{Return } \bot \\
\quad \text{then return } 0 & (a_i, y_i, t_i) \leftarrow \mathsf{st}_i^s \\
\text{Return } 1 & s_i \leftarrow a_i + c_i \cdot y_i \cdot x \\
\underline{\text{Oracle H}(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m):} & \mathsf{msg}_2 \leftarrow (s_i, y_i, t_i) \\
\text{If } T_1(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m) = \bot \text{ then} & \mathcal{I}_{\text{fin}} \leftarrow \mathcal{I}_{\text{fin}} \cup \{i\} \\
\quad T_1(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m) \leftarrow_\$ \mathbb{Z}_p & \mathcal{I}_{\text{fin}}^{(\mathsf{info}_i)} \leftarrow \mathcal{I}_{\text{fin}}^{(\mathsf{info}_i)} \cup \{i\} \\
\quad \text{hid} \leftarrow \text{hid}+1 & \ell(\mathsf{info}) \leftarrow \ell(\mathsf{info})+1 \\
\quad \text{Hid}(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m) \leftarrow \text{hid} & \text{Return } \mathsf{msg}_2 \\
\quad /\!/\ A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} \prod_{i \in [\text{fid}]} Z_i^{\hat{\alpha}^{\mathsf{Z}_i}} \prod_{i \in [\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}} & \underline{\text{Oracle F}(\mathsf{info}):} \\
\quad /\!/\ C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} \prod_{i \in [\text{fid}]} Z_i^{\hat{\beta}^{\mathsf{Z}_i}} \prod_{i \in [\text{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}} & \text{If } T_2(\mathsf{info}) = \bot \text{ then} \\
\quad \delta_{\text{hid}} \leftarrow T_1(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m) & \quad T_2(\mathsf{info}) \leftarrow_\$ \mathbb{G}_\lambda \\
\quad \vec{\alpha}_{\text{hid}} \leftarrow \vec{\alpha};\ \vec{\beta}_{\text{hid}} \leftarrow \vec{\beta} & \quad \text{fid} \leftarrow \text{fid}+1;\ \text{Fid}(\mathsf{info}) \leftarrow \text{fid} \\
\text{Return } T_1(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m) & \quad Z_{\text{fid}} = T_2(\mathsf{info}) \\
& \quad \mathcal{I}_{\text{fin}}^{(\mathsf{info})} \leftarrow \varnothing \\
& \text{Return } T_2(\mathsf{info})
\end{array}
$$

Fig. 12. The OMUF security game for the partially blind signature scheme $\mathsf{PBS}[\mathbb{G}]$.

then $\mathcal{A}_{\text{alg}}$ makes exactly Q_{S_1} queries to S_1 and Q_{S_1} queries to S_2 that do not return $\bot$ for info^*. Then, when $\mathcal{A}_{\text{alg}}$ returns, we know $\ell(\mathsf{info}^*) = Q_{\text{S}_1}$.

In the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game, the corresponding hid for each query $(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m)$ to H is recorded in $\text{Hid}(\mathsf{info} \,\|\, A \,\|\, C \,\|\, m)$, and the output of the query is recorded as δ_{hid}. Also, since $\mathcal{A}_{\text{alg}}$ is algebraic, $\mathcal{A}_{\text{alg}}$ also provides the representation of A and C and the corresponding coefficients $\vec{\alpha}$ and $\vec{\beta}$ are recorded as $\vec{\alpha}_{\text{hid}}$ and

$\vec{\beta}_{\text{hid}}$. The corresponding fid for each new query info to S_1 is recorded in $\mathrm{Fid}(\mathsf{info})$. Also, $\mathcal{I}_{\text{fin}}^{(\mathsf{info})}$ records the subset of $\mathcal{I}_{\text{fin}}$ corresponding to signing sessions with public information info.

Denote the event WIN as $\mathcal{A}_{\text{alg}}$ wins the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game, i.e., all the output message-signature pairs $\{m_k^*, \sigma_k^*\}_{k \in [Q_{\mathrm{S}_1}+1]}$ are distinct and valid for info^*. Furthermore, we denote $\text{str}_k^* := \mathsf{info}^* \parallel g^{s_k^*} X^{-c_k^* \cdot y_k^*} \parallel g^{t_k^*} Z_{\text{Fid}(\mathsf{info}^*)}^{y_k^*} \parallel m_k^*$. We let E be the event in the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game that after the validity of the output is checked, for each $k \in [Q_{\mathrm{S}_1}+1]$, $j = \text{Hid}(\text{str}_k^*)$, and $i^* = \text{Fid}(\mathsf{info}^*)$ [9], the following conditions hold:

$$\hat{\beta}^{\mathsf{Z}_{i^*}} + \sum_{i \in \mathcal{I}_{\text{fin}}^{(\mathsf{info}^*)}} y_i \cdot \hat{\beta}_j^{\mathsf{C}_i} = y_k^* \ , \tag{30}$$

$$\hat{\alpha}_j^{\mathsf{X}} - \sum_{i \in \mathcal{I}_{\text{fin}}} y_i \cdot c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i} = -\delta_j \cdot y_k^* \ , \tag{31}$$

$$\forall i \in [\text{sid}] \backslash \mathcal{I}_{\text{fin}} \ : \ \hat{\alpha}_j^{\mathsf{A}_i} = 0 \ . \tag{32}$$

Since $\mathsf{Adv}_{\mathsf{PBS}[\mathbb{G}]}^{\text{omuf}}(\mathcal{A}_{\text{alg}}, \lambda) = \mathsf{Pr}[\text{WIN}] = \mathsf{Pr}[\text{WIN} \ \wedge \ E] + \mathsf{Pr}[\text{WIN} \ \wedge \ (\neg E)]$, the theorem follows by combining the following two lemmas with Theorem 1.

Lemma 12. *There exists an adversary $\mathcal{B}_{\text{wfros}}$ for the $\text{WFROS}_{Q_{\mathrm{S}_1},p}$ problem making at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to the random oracle* H *such that*

$$\mathsf{Adv}_{Q_{\mathrm{S}_1},p}^{\text{wfros}}(\mathcal{B}_{\text{wfros}}) \geqslant \frac{1}{Q_{\mathsf{info}}} \mathsf{Pr}[\text{WIN} \ \wedge \ E] \ . \tag{33}$$

Lemma 13. *There exists an adversary $\mathcal{B}_{\text{dlog}}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$\mathsf{Pr}[\text{WIN} \ \wedge \ (\neg E)] \leqslant 2\mathsf{Adv}_{\mathbb{G}}^{\text{dlog}}(\mathcal{B}_{\text{dlog}}, \lambda) + \frac{2}{p-1} \ . \tag{34}$$

□

6.2 Proof of Lemma 12

Proof. We first give a detailed description of $\mathcal{B}_{\text{wfros}}$ playing the WFROS game.

THE ADVERSARY $\mathcal{B}_{\text{wfros}}$. To start with, $\mathcal{B}_{\text{wfros}}$ first samples a label $\hat{i}^*$ uniformly from $[Q_{\mathsf{info}}]$. Also, $\mathcal{B}_{\text{wfros}}$ samples x uniformly from $\mathbb{Z}_p$, sets X to g^x, and initializes sid, hid, fid, Hid, Fid, $\mathcal{I}_{\text{fin}}$, T_1, and T_2 as described in the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game. In addition, $\mathcal{B}_{\text{wfros}}$ initializes tfid to 0 and tFid to an empty table, which are used to record the labels of info queries to S_1, and initializes tsid to 0 and tSid to an empty table, which are used to record the labels of session IDs for info such that $\text{tFid}(\mathsf{info}) = \hat{i}^*$.

Then, $\mathcal{B}_{\text{wfros}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\mathrm{F}}$, $\hat{\mathrm{S}}_1$, $\hat{\mathrm{S}}_2$, and $\hat{\mathrm{H}}$. These oracles, operate as follows:

Oracles $\hat{\mathbf{F}}$: Same as in the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game except instead of sampling $T_2(\mathsf{info})$ uniformly from $\mathbb{G}$, if $T_2(\mathsf{info}) = \perp$, $\mathcal{B}_{\text{wfros}}$ samples z_{fid} uniformly from $\mathbb{Z}_p$ and sets $T_2(\mathsf{info}) \leftarrow g^{z_{\text{fid}}}$.

Oracles $\hat{\mathbf{S}}_1$: After receiving a query info to $\hat{\mathrm{S}}_1$ from $\mathcal{A}_{\text{alg}}$, if $\text{tFid}(\mathsf{info}) = \perp$, $\mathcal{B}_{\text{wfros}}$ increases tfid by 1 and sets $\text{tFid}(\mathsf{info}) = \text{tfid}$. Then, there are two cases:

- If $\text{tFid}(\mathsf{info}) \neq \hat{i}^*$, $\mathcal{B}_{\text{wfros}}$ samples $s_{\text{sid}}, t'_{\text{sid}}$ uniformly from $\mathbb{Z}_p$ and samples y'_{sid} uniformly from $\mathbb{Z}_p^*$. Then, $\mathcal{B}_{\text{wfros}}$ sets $A_{\text{sid}} = g^{s_{\text{sid}}} X^{-y'_{\text{sid}}}$ and $C_{\text{sid}} = g^{t'_{\text{sid}}}$.
- If $\text{tFid}(\mathsf{info}) = \hat{i}^*$, $\mathcal{B}_{\text{wfros}}$ samples $a_{\text{sid}}, t'_{\text{sid}}$ uniformly from $\mathbb{Z}_p$ and sets $A_{\text{sid}} = g^{a_{\text{sid}}}$ and $C_{\text{sid}} = g^{t'_{\text{sid}}}$. Also, $\mathcal{B}_{\text{wfros}}$ increases tsid by 1 and sets $\text{tSid}(\text{tsid}) \leftarrow \text{sid}$.

Finally, $\mathcal{B}_{\text{wfros}}$ returns $(\text{sid}, A_{\text{sid}}, C_{\text{sid}})$.

[9] Here, $\text{Hid}(\text{str}_k^*)$ must be defined since a query str_k^* is made to H when checking the validity of the output (m_k^*, σ_k^*).

Oracles $\hat{\mathrm{S}}_2$: After receiving a query (i, c_i) to $\hat{\mathrm{S}}_2$ from $\mathcal{A}_{\mathrm{alg}}$, if $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ or $c_i = 0$, $\mathcal{B}_{\mathrm{wfros}}$ returns $\perp$. Otherwise, there are two cases:

– If $\mathrm{tFid}(\mathsf{info}_i) \neq \hat{\imath}*$, $\mathcal{B}_{\mathrm{wfros}}$ computes $y_i \leftarrow y'_i/c_i$ and $t_i \leftarrow t'_i - y_i \cdot z_{\mathrm{Fid}(\mathsf{info}_i)}$.

– If $\mathrm{tFid}(\mathsf{info}_i) = \hat{\imath}*$, let i' be the index in $[\mathrm{sid}]$ such that $\mathrm{tSid}(i') = i$ and $\mathcal{B}_{\mathrm{wfros}}$ sets $\tilde{c}_{i'} \leftarrow c_i$. Then, $\mathcal{B}_{\mathrm{wfros}}$ makes a query $(i', \tilde{c}_{i'})$ to S. After $\mathcal{B}_{\mathrm{wfros}}$ receives $\tilde{y}_{i'}$ from S, $\mathcal{B}$ sets $y_i \leftarrow \tilde{y}_{i'}$ and $t_i \leftarrow t'_i - y_i \cdot z_{\mathrm{Fid}(\mathsf{info}_i)}$.

Finally, $\mathcal{B}_{\mathrm{wfros}}$ returns (s_i, y_i, t_i).

Oracles $\hat{\mathrm{H}}$: After receiving a query $(\mathsf{info} \parallel A \parallel C \parallel m)$ to $\hat{\mathrm{H}}$ from $\mathcal{A}_{\mathrm{alg}}$, if $\mathrm{tFid}(\mathsf{info}) \neq \hat{\imath}*$ or $T_1(\mathsf{info} \parallel A \parallel C \parallel m) \neq \perp$, then $\hat{\mathrm{H}}$ is the same as H in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game. Otherwise, since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, $\mathcal{B}_{\mathrm{wfros}}$ also knows $\vec{\hat{\alpha}}$ and $\vec{\hat{\beta}}$ such that

$$A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} \prod_{i \in [\mathrm{fid}]} Z_i^{\hat{\alpha}^{\mathsf{Z}_i}} \prod_{i \in [\mathrm{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}} \ , \ C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} \prod_{i \in [\mathrm{fid}]} Z_i^{\hat{\beta}^{\mathsf{Z}_i}} \prod_{i \in [\mathrm{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}} \ .$$

Then, $\mathcal{B}_{\mathrm{wfros}}$ issues the query $(\vec{\alpha}, \vec{\beta})$ to H, where $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2Q_{\mathrm{S}_1}+1}$ such that

$$\alpha^{(i')} = \begin{cases} \hat{\alpha}^{\mathsf{X}} - \sum_{i \in [\mathrm{sid}], \mathrm{tFid}(\mathsf{info}_i) \neq \hat{\imath}*} \hat{\alpha}^{\mathsf{A}_i} \cdot y'_i \ , & i' = 0 \\ -\hat{\alpha}^{\mathsf{A}_{\mathrm{tSid}(i)}} \ , & i' = 2i \ , \ i \in [\mathrm{tsid}] \\ 0 \ , & o.w. \end{cases} \ , \tag{35}$$

$$\beta^{(i')} = \begin{cases} -\hat{\beta}^{\mathsf{Z}_{\hat{\imath}*}} \ , & i' = 0 \\ -\hat{\beta}^{\mathsf{C}_{\mathrm{tSid}(i)}} \ , & i' = 2i-1 \ , \ i \in [\mathrm{tsid}] \\ 0 \ , & o.w. \end{cases} \ . \tag{36}$$

After receiving the output $(\delta_{\mathrm{hid}}, \mathrm{hid})$, $\mathcal{B}_{\mathrm{wfros}}$ sets $T_1(\mathsf{info} \parallel A \parallel C \parallel m) \leftarrow \delta_{\mathrm{hid}}$ and $\mathrm{Hid}(\mathsf{info} \parallel A \parallel C \parallel m) \leftarrow \mathrm{hid}$. Finally, $\mathcal{B}_{\mathrm{wfros}}$ returns $T_1(\mathsf{info} \parallel A \parallel C \parallel m)$.

After receiving the output $\{\mathsf{info}^*, (m_k^*, \sigma_k^*)\}_{k \in [Q_{\mathrm{S}_1}+1]}$ from $\mathcal{A}_{\mathrm{alg}}$, $\mathcal{B}_{\mathrm{wfros}}$ aborts if the conditions from the event $\mathrm{WIN} \wedge E$ do not occur. Otherwise, $\mathcal{B}_{\mathrm{wfros}}$ outputs $\mathcal{J} := \{\mathrm{Hid}(\mathrm{str}_k^*) \mid k \in [Q_{\mathrm{S}_1}+1]\}$.

Analysis of $\mathcal{B}_{\mathrm{wfros}}$. Note that $\mathcal{B}_{\mathrm{wfros}}$ makes a query to H at most once when it receives a query to $\hat{\mathrm{H}}$ for $\mathsf{info}_{\mathrm{tfid}}$ and at most $Q_{\mathrm{S}_1}+1$ more queries to $\hat{\mathrm{H}}$ when checking the validity of the output. Therefore, $\mathcal{B}$ makes at most $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ queries to H. Also, it is clear that $\mathcal{B}$ simulates oracles F, S_1, S_2, H in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game perfectly no matter what label is assigned to tfid. Therefore, the probability that $\mathrm{tFid}(\mathsf{info}^*) = \hat{\imath}*$ and $\mathrm{WIN} \wedge E$ occurs when running $\mathcal{B}_{\mathrm{wfros}}$ is equal to $\frac{1}{Q_{\mathsf{info}}} \mathsf{Pr}[\mathrm{WIN} \wedge E]$.

It is left to show that if $\mathrm{tFid}(\mathsf{info}^*) = \hat{\imath}*$ and $\mathrm{WIN} \wedge E$ occurs within the simulation, then $\mathcal{B}_{\mathrm{wfros}}$ wins the WFROS game. Suppose $\mathrm{WIN} \wedge E$ occurs and $\mathrm{tFid}(\mathsf{info}^*) = \hat{\imath}*$. Following the similar analysis of $\mathcal{B}_{\mathrm{wfros}}$ in the proof of Lemma 10, we have $|\mathcal{J}| = Q_{\mathrm{S}_1} + 1$.

Denote $\mathcal{I}^{\mathrm{tot}}_{\mathrm{fin}}$ and $\mathrm{sid}^{\mathrm{tot}}$ as the values of $\mathcal{I}_{\mathrm{fin}}$ and sid when $\mathcal{A}_{\mathrm{alg}}$ returns. Then, since E occurs, by (30) and (31), for any $j \in \mathcal{J}$ it holds that

$$\hat{\alpha}_j^{\mathsf{X}} - \sum_{i \in \mathcal{I}^{\mathrm{tot}}_{\mathrm{fin}}} y_i \cdot c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i} = -\delta_j \left(\hat{\beta}_j^{\mathsf{Z}_{\hat{\imath}*}} + \sum_{i \in \mathcal{I}^{(\mathsf{info}*)}_{\mathrm{fin}}} y_i \cdot \hat{\beta}_j^{\mathsf{C}_i} \right) \ . \tag{37}$$

$$
\begin{array}{|l|}
\hline
\underline{\text{Game rel-DLog}_{\mathbb{G},n}^{\mathcal{A}}(\lambda):} \\
p \leftarrow |\mathbb{G}_\lambda|;\ g \leftarrow g(\mathbb{G}_\lambda) \\
\{X_i\}_{i\in[n]} \leftarrow\!\!\$\ \mathbb{G}_\lambda \\
y_0, y_1, \ldots, y_n \leftarrow \mathcal{A}(p, g, \mathbb{G}_\lambda, \{X_i\}_{i\in[n]}) \\
\text{If } \forall\ i \in \{1,\ldots,n\}\ :\ y_i = 0 \text{ then return } 0 \\
\text{If } g^{y_0}\prod_{i\in[n]} X_i^{y_i} = 1_{\mathbb{G}_\lambda} \text{ then return } 1 \\
\text{Return } 0 \\
\hline
\end{array}
$$

Fig. 13. The rel-DLog game.

Then, by (32), we have

$$
\begin{aligned}
-\delta_j\left(\hat{\beta}_j^{\mathsf{Z}_{\hat{i}*}} + \sum_{i\in\mathcal{I}_{\text{fin}}^{(\mathsf{info}*)}} y_i\cdot\hat{\beta}_j^{\mathsf{C}_i}\right)x &= \hat{\alpha}_j^{\mathsf{X}} - \sum_{i\in\mathcal{I}_{\text{fin}}^{\text{tot}}} y_i\cdot c_i\cdot\hat{\alpha}_j^{\mathsf{A}_i} \\
&= \hat{\alpha}_j^{\mathsf{X}} - \sum_{i\in\mathcal{I}_{\text{fin}}^{\text{tot}}} y_i\cdot c_i\cdot\hat{\alpha}_j^{\mathsf{A}_i} - \sum_{i\in[\text{sid}^{\text{tot}}]\setminus\mathcal{I}_{\text{fin}}^{\text{tot}}} y_i'\cdot\hat{\alpha}_j^{\mathsf{A}_i} \\
&= \hat{\alpha}_j^{\mathsf{X}} - \sum_{i\in[\text{sid}^{\text{tot}}],\text{tFid}(\mathsf{info}_i)\neq\hat{i}*} y_i'\cdot\hat{\alpha}_j^{\mathsf{A}_i} - \sum_{i\in\mathcal{I}_{\text{fin}}^{(\mathsf{info}*)}} y_i\cdot c_i\cdot\hat{\alpha}_j^{\mathsf{A}_i}\ .
\end{aligned}
$$

Then, from the simulation, by (35), we have for any $j\in\mathcal{J}$

$$
\alpha_j^{(0)} + \sum_{i\in[Q_{\text{S}_1}]}\tilde{y}_i(\alpha_j^{(2i-1)} + \tilde{c}_i\cdot\alpha_j^{(2i)}) = \delta_j\left(\beta_j^{(0)} + \sum_{i\in[Q_{\text{S}_1}]}\tilde{y}_i(\beta_j^{(2i-1)} + \tilde{c}_i\cdot\beta_j^{(2i)})\right).
$$

Therefore, $\mathcal{B}_{\text{wfros}}$ wins the $\text{WFROS}_{Q_{\text{S}_1},p}$ game. □

6.3 Proof of Lemma 13

Proof. We first partition the event $\text{WIN}\ \wedge\ (\neg E)$ into two cases. Denote F_1 as the event in the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game that there exists $k\in[Q_{\text{S}_1}+1]$ such that either (31) or (32) does not hold, and denote F_2 as the event that there exists $k\in[Q_{\text{S}_1}+1]$ such that (30) does not hold. Then, if E does not occur, we know either F_1 or F_2 occurs. Therefore, we have $\text{WIN}\ \wedge\ (\neg E) = (\text{WIN}\ \wedge\ F_1)\ \vee\ (\text{WIN}\ \wedge\ F_2)$. For the case that $\text{WIN}\ \wedge\ F_1$ occurs, we show the following claim.

Claim 8 *There exists $\mathcal{B}_{\text{dlog}}^{(0)}$ for the* DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$
\mathsf{Pr}[\text{WIN}\ \wedge\ F_1] \leqslant \mathsf{Adv}_{\mathbb{G}}^{\text{dlog}}(\mathcal{B}_{\text{dlog}}^{(0)},\lambda) + \frac{1}{p-1}\ . \tag{38}
$$

For the case that $\text{WIN}\ \wedge\ F_2$ occurs, we construct an adversary $\mathcal{B}_{\text{rel-dlog}}$ for the $\text{rel-DLog}_{\mathbb{G},Q_{\text{F}}}$ game (defined in 13) with advantage equals to the probability that $\text{WIN}\ \wedge\ F_2$ occurs, where Q_{F} denotes the maximum number of queries to F issued in the $\text{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\text{alg}}}$ game, and we summarize it into the following claim.

Claim 9 *There exists $\mathcal{B}_{\text{rel-dlog}}$ for the* rel-DLog *problem running in a similar running time as $\mathcal{A}_{\text{alg}}$ such that*

$$
\mathsf{Pr}[\text{WIN}\ \wedge\ F_2] \leqslant \mathsf{Adv}_{\mathbb{G},Q_{\text{F}}}^{\text{rel-dlog}}(\mathcal{B}_{\text{rel-dlog}},\lambda)\ . \tag{39}
$$

The rel-DLog problem is equivalent to the DLog problem, as shown in the following lemma from [JT20].

Lemma 14 (Lemma 3 in [JT20][10]). *For any $n > 0$ and any adversary $\mathcal{B}_{\text{rel-dlog}}$ for the* rel-DLog$_{\mathbb{G},n}$ *game, there exists an adversary $\mathcal{B}_{\text{dlog}}$ for the* DLog$_{\mathbb{G}}$ *game such that*

$$\mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{G},n}(\mathcal{B}_{\text{rel-dlog}}, \lambda) \leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}_{\text{dlog}}, \lambda) + 1/p \ .$$

By the lemma and Claim 9, there exists an adversary $\mathcal{B}^{(1)}_{\text{dlog}}$ for the DLog$_{\mathbb{G}}$ problem such that $\Pr[\text{WIN} \wedge F_1] \leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(1)}_{\text{dlog}}, \lambda) + \frac{1}{p}$. Therefore, together with Claim 8, we can construct an adversary $\mathcal{B}_{\text{dlog}}$ for the DLog$_{\mathbb{G}}$ problem that runs either $\mathcal{B}^{(0)}_{\text{dlog}}$ or $\mathcal{B}^{(1)}_{\text{dlog}}$ with 1/2 probability, and we can conclude the lemma since

$$\begin{aligned} \Pr[\text{WIN} \wedge (\neg E)] &\leqslant \Pr[\text{WIN} \wedge F_1] + \Pr[\text{WIN} \wedge F_2] \\ &\leqslant \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(0)}_{\text{dlog}}, \lambda) + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}^{(1)}_{\text{dlog}}, \lambda) + \frac{2}{p-1} = 2\mathsf{Adv}^{\text{dlog}}_{\mathbb{G}}(\mathcal{B}_{\text{dlog}}, \lambda) + \frac{2}{p-1}. \end{aligned}$$

Proof (of Claim 8). We first give a detailed description of $\mathcal{B}^{(0)}_{\text{dlog}}$ playing the DLog$_{\mathbb{G}}$ game.

THE ADVERSARY $\mathcal{B}^{(0)}_{\text{dlog}}$. To start with, $\mathcal{B}^{(0)}_{\text{dlog}}$ initializes sid, hid, fid, Hid, Fid, $\mathcal{I}_{\text{fin}}$, T_1, and T_2 as described in the OMUF$^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game. After $\mathcal{B}^{(0)}_{\text{dlog}}$ receives $(p, g, \mathbb{G}_\lambda, W)$ from the DLog$_{\mathbb{G}}$ game, sets $X \leftarrow W$. Then, $\mathcal{B}^{(0)}_{\text{dlog}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$, and with access to the oracles $\hat{\text{F}}$, $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{F}}$: Same as in the OMUF$^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game except instead of sampling $T_2(\mathsf{info})$ uniformly from $\mathbb{G}$, if $T_2(\mathsf{info}) = \perp$, $\mathcal{B}^{(0)}_{\text{dlog}}$ samples z uniformly from $\mathbb{Z}_p$ and sets $T_2(\mathsf{info}) \leftarrow g^{z_{\text{fid}}}$.

Oracle $\hat{\mathbf{S}}_1$: After receiving a query info from $\mathcal{A}_{\text{alg}}$, $\mathcal{B}_{\text{wfros}}$ samples $s_{\text{sid}}, t'_{\text{sid}}$ uniformly from $\mathbb{Z}_p$ and samples y'_{sid} uniformly from $\mathbb{Z}^*_p$. Then, $\mathcal{B}_{\text{wfros}}$ sets $A_{\text{sid}} \leftarrow g^{s_{\text{sid}}} X^{-y'_{\text{sid}}}$ and $C_{\text{sid}} \leftarrow g^{t'_{\text{sid}}}$ and returns $(\text{sid}, A_{\text{sid}}, C_{\text{sid}})$.

Oracle $\hat{\mathbf{S}}_2$: After receiving a query (i, c_i) to $\hat{\text{S}}_2$ from $\mathcal{A}_{\text{alg}}$, if $i \notin [\text{sid}]\backslash\mathcal{I}_{\text{fin}}$ or $c_i = 0$, $\mathcal{B}^{(0)}_{\text{dlog}}$ returns $\perp$. Otherwise, let $\hat{i} := \text{Fid}(\mathsf{info}_i)$, and $\mathcal{B}^{(0)}_{\text{dlog}}$ computes $y_i \leftarrow y'_i/c_i$ and $t_i \leftarrow t'_i - y_i \cdot z_{\hat{i}}$. Then, $\mathcal{B}^{(0)}_{\text{dlog}}$ returns (s_i, y_i, t_i).

Oracle $\hat{\mathbf{H}}$: Same as in the OMUF$^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game.

After receiving the output $(\mathsf{info}^*, \{(m^*_k, \sigma^*_k)\}_{k \in [Q_{\text{S}_1}+1]})$, $\mathcal{B}^{(0)}_{\text{dlog}}$ aborts if the event $\text{WIN} \wedge F_1$ does not occur.

It is clear that $\mathcal{B}^{(0)}_{\text{dlog}}$ simulates the OMUF$^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game perfectly, and thus it is left to show that if $\text{WIN} \wedge F_1$ occurs, $\mathcal{B}^{(0)}_{\text{dlog}}$ can compute the discrete log of W except for probability $1/p$.

Suppose $\text{WIN} \wedge F_1$ occurs in the OMUF$^{\mathcal{A}_{\text{alg}}}_{\mathsf{PBS}[\mathbb{G}]}$ game simulated by $\mathcal{B}^{(0)}_{\text{dlog}}$. There exists $k \in [Q_{\text{S}_1} + 1]$ and $j = \text{Hid}(\text{str}^*_k)$ such that either (31) or (32) does not hold. Since $j = \text{Hid}(\text{str}^*_k)$ and $\delta_j = c^*_k$, we have

$$g^{s^*_k} X^{-\delta_j \cdot y^*_k} = g^{s^*_k} X^{-c^*_k \cdot y^*_k} = g^{\hat{\alpha}^g_j} X^{\hat{\alpha}^{\mathsf{X}}_j} \prod_{i \in [\text{fid}]} Z_i^{\hat{\alpha}^{\mathsf{Z}_i}_j} \prod_{i \in [\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}_j} C_i^{\hat{\alpha}^{\mathsf{C}_i}_j} \ . \tag{40}$$

From the simulation of $\hat{\text{S}}_1$, for each $i \in [\text{sid}]$, we have

$$A_i = g^{s_i} X^{-y'_i} \ , \ C_i = g^{t'_i} \ .$$

Also, $\mathcal{B}^{(0)}_{\text{dlog}}$ knows the discrete log of Z_i as z_i for each $i \in [\text{fid}]$. By substituting $A_i = g^{s_i} X^{-y'_i}$, $C_i = g^{t'_i}$, and $Z_i = g^{z_i}$ into (40), we have

$$g^{s^*_k} X^{-\delta_j \cdot y^*_k} = g^{\eta^g_j} X^{\eta^Z_j} \ ,$$

[10] The DLog and rel-DLog games defined in [JT20] differ slightly from our descriptions, but the lemma follows by a similar proof.

where

$$\eta_j^g := \hat{\alpha}_j^g + \sum_{i\in[\mathrm{fid}]} \hat{\alpha}_j^{\mathsf{Z}_i} \cdot z_i + \sum_{i\in[\mathrm{sid}]} (\hat{\alpha}_j^{\mathsf{A}_i} \cdot s_i + \hat{\alpha}_j^{\mathsf{C}_i} \cdot t_i') ,$$

$$\eta_j^{\mathsf{X}} := \hat{\alpha}_j^{\mathsf{X}} - \sum_{i\in[\mathrm{sid}]} y_i' \cdot \hat{\alpha}_j^{\mathsf{A}_i} .$$

If $\eta_j^{\mathsf{X}} \neq -\delta_j \cdot y_k^*$, $\mathcal{B}_{\mathrm{dlog}}^{(0)}$ can compute the discrete log of X, which is also W, as

$$x := \frac{s_k^* - \eta_j^g}{\eta_j^{\mathsf{X}} + \delta_j \cdot y_k^*} .$$

Therefore, it is left to bound the probability that $\eta_j^{\mathsf{X}} = -\delta_j \cdot y_k^*$, and there are the following two cases.
(32) does not hold for k, j. Consider the transcript π^{tot} that the adversary sees before it returns. Given the transcript π^{tot}, since for each $i \in [\mathrm{sid}]\backslash\mathcal{I}_{\mathrm{fin}}$, the adversary sees only A_i but does not know either s_i or y_i', the value y_i' is uniformly distributed over $\mathbb{Z}_p^*$ independent of all other $y_{i'}'$ for $i' \neq i$. Therefore, the probability that $\eta_j^{\mathsf{X}} = -\delta_j \cdot y_k^*$ is $\frac{1}{p-1}$.
(32) holds but (31) does not hold for k, j. Since (32) holds and for each $i \in \mathcal{I}_{\mathrm{fin}}$ it holds that $y_i' = y_i \cdot c_i$, we have

$$\eta_{\hat{j}}^{\mathsf{X}} = \hat{\alpha}_{\hat{j}}^{\mathsf{X}} - \sum_{i\in\mathcal{I}_{\mathrm{fin}}} y_i \cdot c_i \cdot \hat{\alpha}_{\hat{j}}^{\mathsf{A}_i} .$$

Then, since (31) does not hold, we have

$$\eta_{\hat{j}}^{\mathsf{X}} \neq -\delta_{\hat{j}} \cdot y_{\hat{k}}^* ,$$

which means the probability that $\eta_j^{\mathsf{X}} = -\delta_j \cdot y_k^*$ is 0. Therefore, for both cases, the probability that $\eta_j^{\mathsf{X}} = -\delta_j \cdot y_k^*$ is bounded by $\frac{1}{p-1}$. □

Proof (of Claim 9). We first give a detailed description of $\mathcal{B}_{\text{rel-dlog}}$ playing the rel-$\mathrm{DLog}_{\mathbb{G},Q_{\mathrm{F}}}$ game.

THE ADVERSARY $\mathcal{B}_{\text{rel-dlog}}$. To start with, $\mathcal{B}_{\text{rel-dlog}}$ initializes sid, hid, fid, Hid, Fid, $\mathcal{I}_{\mathrm{fin}}$, T_1, and T_2 as described in the $\mathrm{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\mathrm{alg}}}$ game. Also, $\mathcal{B}_{\text{rel-dlog}}$ samples x uniformly from $\mathbb{Z}_p$ and sets $X \leftarrow g^x$. After $\mathcal{B}_{\text{rel-dlog}}$ receives $(p, g, \mathbb{G}_\lambda, Z_1, \ldots, Z_{Q_{\mathrm{F}}})$ from the rel-$\mathrm{DLog}_{\mathbb{G},Q_{\mathrm{F}}}$ game, $\mathcal{B}_{\text{rel-dlog}}$ runs $\mathcal{A}_{\mathrm{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\mathrm{F}}$, $\hat{\mathrm{S}}_1$, $\hat{\mathrm{S}}_2$, and $\hat{\mathrm{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{F}}$: Same as in the $\mathrm{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\mathrm{alg}}}$ game except instead of sampling $T_2(\mathsf{info})$ uniformly from $\mathbb{G}$, if $T_2(\mathsf{info}) = \perp$, $\mathcal{B}_{\text{rel-dlog}}$ sets $T_2(\mathsf{info}) \leftarrow Z_{\mathrm{fid}}$.

Oracle $\hat{\mathbf{S}}_1$, $\hat{\mathbf{S}}_2$, $\hat{\mathbf{H}}$: The same as in the $\mathrm{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\mathrm{alg}}}$ game.

After receiving the output $(\mathsf{info}^*, \{(m_k^*, \sigma_k^*)\}_{k\in[Q_{\mathrm{S}_1}+1]})$, $\mathcal{B}_{\text{rel-dlog}}$ aborts if WIN $\wedge$ F_2 does not occur.

It is clear that $\mathcal{B}_{\text{rel-dlog}}$ simulates the $\mathrm{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\mathrm{alg}}}$ game perfectly, and thus it is left to show that if WIN $\wedge$ F_2 occurs, $\mathcal{B}_{\text{rel-dlog}}$ can win the rel-$\mathrm{DLog}_{\mathbb{G},Q_{\mathrm{F}}}$ game.

Suppose WIN $\wedge$ F_2 occurs in the $\mathrm{OMUF}_{\mathsf{PBS}[\mathbb{G}]}^{\mathcal{A}_{\mathrm{alg}}}$ game simulated by $\mathcal{B}_{\text{rel-dlog}}$. There exists $k \in [Q_{\mathrm{S}_1} + 1]$ and $j = \mathrm{Hid}(\mathrm{str}_k^*)$ such that (30) does not hold. Since $j = \mathrm{Hid}(\mathrm{str}_k^*)$, we have

$$g^{t_k^*} Z_{i^*}^{y_k^*} = g^{\hat{\beta}_j^g} X^{\hat{\beta}_j^{\mathsf{X}}} \prod_{i\in[\mathrm{fid}]} Z_i^{\hat{\beta}_j^{\mathsf{Z}_i}} \prod_{i\in[\mathrm{sid}]} A_i^{\hat{\beta}_j^{\mathsf{A}_i}} C_i^{\hat{\beta}_j^{\mathsf{C}_i}} . \tag{41}$$

From the simulation of $\hat{\mathrm{S}}_1$, for each $i \in [\mathrm{sid}]$, we have

$$A_i = g^{a_i} \ , \ g^{t_i} = C_i Z_i^{-y_i} .$$

Also, $\mathcal{B}_{\text{rel-dlog}}$ knows the discrete log of X as x. By substituting $A_i = g^{a_i}$, $C_i = g^{t_i} Z_i^{y_i}$, and $X = g^x$ into (41), we have

$$g^{t_k^*} Z_{i^*}^{y_k^*} = g^{\hat{\beta}_j^g + \hat{\beta}_j^{\mathsf{X}} \cdot x + \sum_{i \in [\text{sid}]} (\hat{\beta}_j^{\mathsf{A}_i} \cdot a_i + \hat{\beta}_j^{\mathsf{C}_i} \cdot t_i)} \prod_{i \in [\text{fid}]} Z_i^{\hat{\beta}_j^{\mathsf{Z}_i} + \sum_{i' \in [\text{sid}], \text{tSid}(i') = i} y_{i'} \cdot \hat{\beta}^{\mathsf{C}_{i'}}} .$$

Therefore, $\mathcal{B}_{\text{rel-dlog}}$ can compute $(w_0, \ldots, w_{Q_{\mathrm{F}}})$ such that $g^{w_0} \prod_{i \in [Q_{\mathrm{F}}]} W_i^{w_i} = g^{w_0} \prod_{i \in [\text{fid}]} Z_i^{w_i} = 1_{\mathbb{G}_\lambda}$ as

$$w_i := \begin{cases} \hat{\beta}_j^g + \hat{\beta}_j^{\mathsf{X}} \cdot x + \sum_{i \in [\text{sid}]} (\hat{\beta}_j^{\mathsf{A}_i} \cdot a_i + \hat{\beta}_j^{\mathsf{C}_i} \cdot t_i) - t_k^* , & i = 0 \\ \hat{\beta}_j^{\mathsf{Z}_i} + \sum_{i' \in [\text{sid}], \text{tSid}(i') = i} y_{i'} \cdot \hat{\beta}^{\mathsf{C}_{i'}} , & i \in [\text{fid}], i \neq i^* \\ -y_k^* + \hat{\beta}_j^{\mathsf{Z}_i} + \sum_{i' \in [\text{sid}], \text{tSid}(i') = i} y_{i'} \cdot \hat{\beta}_j^{\mathsf{C}_{i'}} , & i = i^* \\ 0 , & o.w. \end{cases}$$

Since (30) does not hold, we have

$$w_{i^*} = -y_k^* + \hat{\beta}_j^{\mathsf{Z}_{i^*}} + \sum_{i \in \mathcal{I}_{\text{fin}}^{(\mathsf{info}^*)}} y_{i'} \cdot \hat{\beta}^{\mathsf{C}_{i'}} \neq 0 .$$

Therefore, $\mathcal{B}_{\text{rel-dlog}}$ wins the rel-$\text{DLog}_{\mathbb{G}, Q_{\mathrm{F}}}$ game by outputting $(w_0, \ldots, w_{Q_{\mathrm{F}}})$ defined above. □

Acknowledgments

The authors wish to thank Christopher A. Wood for extensive discussions. Both authors were partially supported by NSF grants CNS-1930117 (CAREER), CNS-1926324, CNS-2026774, a Sloan Research Fellowship, and a JP Morgan Faculty Award.

References

Abe01. Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 136–151. Springer, Heidelberg, May 2001.

AF96. Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 244–251. Springer, Heidelberg, November 1996.

AO00. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, Heidelberg, August 2000.

BDL+12. Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, September 2012.

BFP21. Balthazar Bauer, Georg Fuchsbauer, and Antoine Plouviez. The one-more discrete logarithm assumption in the generic group model. Cryptology ePrint Archive, Report 2021/866, 2021. https://ia.cr/2021/866.

BFPV13. Olivier Blazy, Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Short blind signatures. *J. Comput. Secur.*, 21(5):627–661, 2013.

BL13. Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 1087–1098. ACM Press, November 2013.

BLL+21. Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Heidelberg, October 2021.

BLS01. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, Heidelberg, December 2001.

BNPS03. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003.

Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, January 2003.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

BR06. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.

CAL22. Rutchathon Chairattana-Apirom and Anna Lysyanskaya. Compact cut-and-choose: Boosting the security of blind signature schemes, compactly. Cryptology ePrint Archive, Report 2022/003, 2022. https://ia.cr/2022/003.

CFN90. David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, Heidelberg, August 1990.

Cha81. David Chaum. Verification by anonymous monitors. In Allen Gersho, editor, *CRYPTO'81*, volume ECE Report 82-04, pages 138–139. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Heidelberg, August 2004.

CP93. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Heidelberg, August 1993.

DJW21. Frank Denis, Frederic Jacobs, and Christopher A. Wood. RSA Blind Signatures. Internet-Draft draft-irtf-cfrg-rsa-blind-signatures-02, Internet Engineering Task Force, August 2021. Work in Progress.

FHKS16. Georg Fuchsbauer, Christian Hanser, Chethan Kamath, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model from weaker assumptions. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 391–408. Springer, Heidelberg, August / September 2016.

FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, August 2015.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Heidelberg, May 2020.

GG14. Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 477–495. Springer, Heidelberg, May 2014.

Gha17. Essam Ghadafi. Efficient round-optimal blind signatures in the standard model. In Aggelos Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 455–473. Springer, Heidelberg, April 2017.

GRS+11. Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 630–648. Springer, Heidelberg, August 2011.

HIP+21. Scott Hendrickson, Jana Iyengar, Tommy Pauly, Steven Valdez, and Christopher A. Wood. Private Access Tokens. Internet-Draft draft-private-access-tokens-01, Internet Engineering Task Force, October 2021. Work in Progress.

HKL19. Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 345–375. Springer, Heidelberg, May 2019.

Hop13. Nicholas Hopper. Proving security of tor's hidden service identity blinding protocol. https://www-users.cse.umn.edu/~hoppernj/basic-proof.pdf, 2013.

JT20. Joseph Jaeger and Stefano Tessaro. Expected-time cryptography: Generic techniques and applications to concrete soundness. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 414–443. Springer, Heidelberg, November 2020.

KLR21. Jonathan Katz, Julian Loss, and Michael Rosenberg. Boosting the security of blind signature schemes. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021*, volume 13093 of *LNCS*, pages 468–492. Springer, December 2021.

KLRX22. Julia Kastner, Julian Loss, Michael Rosenberg, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. PKC 2022, 2022. to appear.

KM08. Neal Koblitz and Alfred Menezes. Another look at non-standard discrete log and diffie-hellman problems. *J. Math. Cryptol.*, 2(4):311–326, 2008.

KNYY21. Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Round-optimal blind signatures in the plain model from classical and quantum standard assumptions. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 404–434. Springer, Heidelberg, October 2021.

Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.

OA03. Miyako Ohkubo and Masayuki Abe. Security of some three-move blind signature schemes reconsidered. In *The 2003 Symposium on Cryptography and Information Security*, 2003.

PCM. PCM: Click fraud prevention and attribution sent to advertiser. https://webkit.org/blog/11940/pcm-click-fraud-prevention-and-attribution-sent-to-advertiser/. Accessed: 2021-09-30.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

Sch90. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.

Sch01. Claus-Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Heidelberg, November 2001.

Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

Tru. Trust tokens. https://developer.chrome.com/docs/privacy-sandbox/trust-tokens/. Accessed: 2022-01-11.

WHL22. Benedikt Wagner, Lucjan Hanzlik, and Julian Loss. Pi-cut-choo! parallel instance cut and choose for practical blind signatures. Cryptology ePrint Archive, Report 2022/007, 2022. https://ia.cr/2022/007.

A Proof of Lemma 4

Proof. For $k \in \{0, \ldots, n\}$, define E_k as

$$\exists\, i \in \{0, \ldots, k\} \text{ such that } D_i \neq 0 \ \wedge\ D_0 + \sum_{j=1}^{k} D_j X_j = 0\ .$$

We will prove the theorem using induction. It is clear that $\mathsf{Pr}[E_0] = 0$. For $k \geqslant 1$, assume $\mathsf{Pr}[E_{k-1}] \leqslant \sum_{i=1}^{k-1} \frac{1}{|U_i|}$. It holds that

$$\begin{aligned}
\mathsf{Pr}[E_k] &= \mathsf{Pr}[E_k | E_{k-1}]\mathsf{Pr}[E_{k-1}] + \mathsf{Pr}[E_k | \neg E_{k-1}]\mathsf{Pr}[\neg E_{k-1}] \\
&\leqslant \mathsf{Pr}[E_{k-1}] + \mathsf{Pr}[E_k | \neg E_{k-1}] \\
&= \mathsf{Pr}[E_{k-1}] + \mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k \neq 0\right] \mathsf{Pr}[D_k \neq 0 | \neg E_{k-1}] \\
&\quad + \mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k = 0\right] \mathsf{Pr}[D_k = 0 | \neg E_{k-1}] \\
&\leqslant \mathsf{Pr}[E_{k-1}] + \mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k \neq 0\right] + \mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k = 0\right]\ .
\end{aligned} \tag{42}$$

It is left to bound $\mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k \neq 0\right]$ and $\mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k = 0\right]$.

Suppose E_{k-1} does not occur and then we have either $D_i = 0$ for all $0 \leqslant i < k$ or $D_0 + \sum_{j=1}^{k-1} D_j X_j = 0$.

If $D_k = 0$, we have either $D_i = 0$ for all $0 \leqslant i \leqslant k$, or $D_0 + \sum_{j=1}^{k} D_j X_j = D_0 + \sum_{j=1}^{k-1} D_j X_j \neq 0$, which means E_k does not occur. Therefore, we have

$$\mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge\ D_k = 0\right] = 0. \tag{43}$$

Otherwise, if $D_k \neq 0$, we know E_k occurs if and only if $D_0 + \sum_{j=1}^{k} D_j X_j \neq 0$. Since X_k is uniformly distributed over U_k independent of $(D_0, \ldots, D_k, X_1, \ldots, X_{k-1})$ given $D_k \neq 0$ and E_{k-1} does not occur, it holds that

$$\begin{aligned}\mathsf{Pr}\left[E_k \,\middle|\, (\neg E_{k-1}) \ \wedge \ D_k \neq 0\right] &= \mathsf{Pr}\left[D_0 + \sum_{j=1}^{k} D_j X_j = 0 \,\middle|\, (\neg E_{k-1}) \ \wedge \ D_k \neq 0\right] \\ &= \mathsf{Pr}\left[X_k = \frac{D_0 + \sum_{j=1}^{k-1} D_j X_j}{D_k} \,\middle|\, (\neg E_{k-1}) \ \wedge \ D_k \neq 0\right] \\ &\leqslant \frac{1}{|U_i|} \ .\end{aligned} \tag{44}$$

Therefore, from (42), (43), and (44), we have

$$\mathsf{Pr}[E_k] \leqslant \mathsf{Pr}[E_{k-1}] + \frac{1}{|U_i|} \leqslant \sum_{i=1}^{k} \frac{1}{|U_i|} \ .$$

Therefore, by induction, we have

$$\mathsf{Pr}\left[\exists\, i \in \{0, \ldots, n\} \ : \ D_i \neq 0 \ \wedge \ D_0 + \sum_{j=1}^{n} D_j X_j = 0\right] = \mathsf{Pr}[E_n] \leqslant \sum_{i=1}^{n} \frac{1}{|U_i|}.$$

□

B Postponed Proofs from Section 4

B.1 Proof of Lemma 7

We prove the lemma by going through a serious of games.

<u>$\text{Game}_0^{\mathcal{A}}$</u>: This is $\text{OMUF-GGM}_{\mathsf{BS}_1}^{\mathcal{A}}$ (Figure 5).

<u>$\text{Game}_1^{\mathcal{A}}$</u>: This is defined in Figure 14 that only contains the dashed box. We introduce variables X, A_1, Y_1, $\ldots$, $\mathsf{A}_{Q_{\mathsf{S}_1}}$, $\mathsf{Y}_{Q_{\mathsf{S}_1}}$ in $\text{Game}_1^{\mathcal{A}}$. Each variable is assigned a value, that is, X is assigned x, A_i is assigned a_i, and Y_i is assigned $y_i \cdot x$. The input to Φ is a polynomial P of variables X, $\{\mathsf{A}_i, \mathsf{Y}_i\}_{i \in [Q_{\mathsf{S}_1}]}$ over $\mathbb{Z}_p$ instead of a single value $v \in \mathbb{Z}_p$ and the set Cur is a set of polynomials. Also, in Φ we check the equality of two polynomials by its evaluation on the assigned values, which is denoted by $=_{\text{eval}}$ (see Definition 1).

Definition 1. *For two ploynomial P and P' of the variables $\mathsf{X}_1, \ldots, \mathsf{X}_n$ over a field $\mathcal{F}$, suppose each X_i is assigned with a value $x_i \in \mathcal{F}$. We say $P =_{\text{eval}} P'$ if and only if $P(\mathsf{X}_1 = x_1, \ldots, \mathsf{X}_n = x_n) = P'(\mathsf{X}_1 = x_1, \ldots, \mathsf{X}_n = x_n)$.*

For convenience, we also have $P =_{\text{eval}} P(\mathsf{X}_1 = x_1, \ldots, \mathsf{X}_n = x_n)$.

It is easy to check that $=_{\text{eval}}$ is an equivalence relation over the polynomials of the variables $\mathsf{X}_1, \ldots, \mathsf{X}_n$.

We first show that the oracle Φ in $\text{Game}_1^{\mathcal{A}}$ is well-defined, that is, for each query P to Φ, there exists at most one $P' \in \mathsf{Cur}$ such that $P =_{\text{eval}} P'$. Suppose there exists $P', P'' \in \mathsf{Cur}$ such that $P' \neq P''$, $P' =_{\text{eval}} P =_{\text{eval}} P''$. Suppose P'' is added to Cur after P'. Consider the query to Φ during which P'' is added to Cur. Since P' is already in Cur when P'' is added, we have $P' \neq_{\text{eval}} P''$, which yields a contradiction. Therefore, for each query to Φ, if there exists $P' \in \mathsf{Cur}$ such that $P =_{\text{eval}} P'$, then P' is the unique polynomial in Cur such that $P =_{\text{eval}} P'$.

We now show that the views of the adversary in Game_0 and Game_1 are identical. Define an intermediate game $\text{Game}_1'^{\mathcal{A}}$ such that it is identical to $\text{Game}_1^{\mathcal{A}}$ except each the polynomial P appear in the game is

Game $\text{Game}_1^{\mathcal{A}}$, $\boxed{\text{Game}_2^{\mathcal{A}}}$, $\boxed{\text{Game}_2'^{\mathcal{A}}}$:

$p \leftarrow |\mathbb{G}_\lambda|$; $x \leftarrow\!\!\$\ \mathbb{Z}_p^*$; assign x to variable X
sid $\leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{I}_{\text{fin}} \leftarrow \varnothing$; $\varXi \leftarrow ()$; $T \leftarrow ()$
$\mathsf{Cur} \leftarrow \varnothing$; $\boxed{L \leftarrow \varnothing}$
$\{(m_k, \sigma_k)\}_{k \in [\ell+1]} \leftarrow\!\!\$\ \mathcal{A}^{\varPi, \mathrm{S}_1, \mathrm{S}_2, \mathrm{H}}(p, \varPhi(1), \varPhi(\mathsf{X}))$
If $\exists\ k_1 \neq k_2$ such that $(m_{k_1}, \sigma_{k_1}) = (m_{k_2}, \sigma_{k_2})$ then
  Return 0
If $\exists\ k \in [\ell+1]$ such that $y_k^* = 0$
  or $c_k \neq \mathrm{H}(\varPhi(s_k - c_k \cdot y_k \cdot \mathsf{X}) \,\|\, \varPhi(y_k \cdot \mathsf{X}) \,\|\, m_i)$
where $(c_k, s_k, y_k) = \sigma_k$ then return 0
Return 1

Oracle $\varPhi(P)$:
$\boxed{\begin{array}{l}\text{If } \exists P' \in \mathsf{Cur} \text{ such that } P =_{\text{eval}} P' \\ \quad \text{and } P \neq_L P' \text{ then } \textbf{abort game}\end{array}}$
If $\exists P' \in \mathsf{Cur}$ such that $P =_{\text{eval}} P'$ then
  Return $\varXi(P')$
If $\exists P' \in \mathsf{Cur}$ such that $P =_L P'$ then
  Return $\varXi(P')$
$\varXi(P) \leftarrow\!\!\$\ \{0,1\}^{\log(p)} \backslash \varXi(\mathsf{Cur})$
$\mathsf{Cur} \leftarrow \mathsf{Cur} \cap \{P\}$
Return $\varXi(P)$

Oracle $\varPi(\xi, \xi', b)$:
If $\exists P, P' \in \mathsf{Cur}$ such that $\xi = \varXi(P)$
  and $\xi' = \varXi(P')$ then
  Return $\varPhi(P + (-1)^b P')$
Else return $\perp$

Oracle S_1 :
sid $\leftarrow$ sid $+ 1$
$a_{\text{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p$; $y_{\text{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$\mathsf{st}_{\text{sid}}^s \leftarrow (a_{\text{sid}}, y_{\text{sid}})$
Assign a_{sid} to variable A_{sid}
Assign $y_{\text{sid}} \cdot x$ to varaible Y_{sid}
$\mathsf{msg}_1 \leftarrow (\varPhi(\mathsf{A}_{\text{sid}}), \varPhi(\mathsf{Y}_{\text{sid}}))$
Return (sid, msg_1)

Oracle $\mathrm{S}_2(i, c_i)$:
If $i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}}$ then return $\perp$
$(a_i, y_i) \leftarrow \mathsf{st}_i^s$
$s_i \leftarrow a_i + c_i \cdot y_i \cdot x$
$\boxed{\begin{array}{l} R_1 \leftarrow \mathsf{A}_i + c_i \mathsf{Y}_i - s_i \\ R_2 \leftarrow \mathsf{Y}_i - y_i \mathsf{X} \\ L \leftarrow L \cup \{R_1, R_2\}\end{array}}$
$\mathsf{msg}_2 \leftarrow (s_i, y_i)$
$\mathcal{I}_{\text{fin}} \leftarrow \mathcal{I}_{\text{fin}} \cup \{i\}$
$\ell \leftarrow \ell + 1$
Return msg_2

Oracle $\mathrm{H}(\text{str})$:
If $T(\text{str}) = \perp$ then
  $T(\text{str}) \leftarrow\!\!\$\ \mathbb{Z}_p$
Return $T(\text{str})$

Fig. 14. The definition for $\text{Game}_1^{\mathcal{A}}$, $\text{Game}_2^{\mathcal{A}}$, and $\text{Game}_2'^{\mathcal{A}}$, where $\text{Game}_1^{\mathcal{A}}$ only contains the dashed box, $\text{Game}_2^{\mathcal{A}}$ contains all but the gray box, and $\text{Game}_2'^{\mathcal{A}}$ contains all but the dashed box.

---

replaced by its evaluation value $P(\mathsf{X} = x, \mathsf{A}_1 = a_1, \mathsf{Y}_1 = y_1 \cdot x, \ldots, \mathsf{A}_{sid} = a_{sid}, \mathsf{Y}_{sid} = y_{sid} \cdot x)$. It is clear that $\text{Game}_1'^{\mathcal{A}}$ is identical to $\text{Game}_0^{\mathcal{A}}$. Also, since in the oracle $\varPhi$ in $\text{Game}_1^{\mathcal{A}}$, a polynomial P is considered equal or not equal to another polynomials by its evaluation value, the view of the adversary in Game_1 and Game_1' are identical. Thus, we know the views of the adversary in Game_0 and Game_1 are identical, which implies

$$\Pr[\text{Game}_0^{\mathcal{A}} = 1] = \Pr[\text{Game}_1^{\mathcal{A}} = 1] \ . \tag{45}$$

$\underline{\text{Game}_2^{\mathcal{A}}}$: This is defined in Figure 14 by ignoring the graybox. A set L is introduced to record the information leaked to the adversary by S_2. For the query (i, c_i) to S_2, polynomials $R_1 = \mathsf{A}_i + c_i \mathsf{Y}_i - s_i$ and $R_2 = \mathsf{Y}_i - y_i \mathsf{X}$ are added to L. Suppose L is also recorded in $\text{Game}_1^{\mathcal{A}}$. In $\text{Game}_1^{\mathcal{A}}$, define the event E_1 as after an query P to $\varPhi$ is made,

$$\exists P' \in \mathsf{Cur} \text{ such that } P =_{\text{eval}} P' \text{ and } P \neq_L P' \ .$$

Then, $\text{Game}_2^{\mathcal{A}}$ is identical to $\text{Game}_1^{\mathcal{A}}$ except it aborts when E_1 occurs and we have

$$\Pr[\text{Game}_1^{\mathcal{A}} = 1] \leqslant \Pr[\text{Game}_2^{\mathcal{A}} = 1] + \Pr[E_1] \ , \tag{46}$$

To bound $\Pr[E_1]$, for each $j \in [Q_\varPhi]$, we denote the event $E_{1,j}$ in $\text{Game}_1^{\mathcal{A}}$ as during the j-th query to $\varPhi$

$$\exists P' \in \mathsf{Cur} \text{ such that } P_j =_{\text{eval}} P' \text{ and } P_j \neq_L P' \ .$$

Then, we have $E_1 = \bigvee_{j\in[Q_\Phi]} E_{1,j}$. Denote $E'_{1,j} := E_{1,j} \bigwedge_{i\in[j]}(\neg E_{1,i})$. We now bound $\mathsf{Pr}[E'_{1,j}]$ for each $j \in [Q_\Phi]$.

We now fix a certain $j \in [Q_\Phi]$. Consider the step when the j-th query to Φ is made during $\mathrm{Game}_1^{\mathcal{A}}$. Denote the transcripts between the oracles and adversarys when the j-th query to Φ is made as π_j, which contains $\Phi(1), \Phi(\mathsf{X})$, and all the inputs and outputs of the queries to S_1, S_2, Π, and H made before the j-th query to Φ. For a certain transcript $\pi_j = \Delta$, for $1 \leqslant k \leqslant j$, denote the k-th query to Φ in Δ as P_k^Δ. From the transcript Δ, one can compute the set $\mathcal{I}_{\mathrm{fin}}$, Cur, and L at the step when the j-th query to Φ is been made. Denote them by $\mathcal{I}_{\mathrm{fin}}^\Delta$, Cur^Δ, and L^Δ. For each $i \in \mathcal{I}_{\mathrm{fin}}^\Delta$, denote the input and output of the query to the S_2 for the session i in the transcript Δ as c_i^Δ and (s_i^Δ, y_i^Δ). Also, from the transcript π_j, one can tell whether $E_{1,k}$ occurs or not for $k \in [j-1]$, since the event $E_{1,k}$ occurs if and only if $P_k \neq_L P'$ for all $P' \in \mathsf{Cur}$ but P_k is not added to Cur. Denote the value of sid when the j-th query to Φ is made as sid^Δ.

Denote $\mathcal{T}_j$ as the set of all transcripts Δ such that $\mathsf{Pr}[\pi_j = \Delta] > 0$ and none of $\{E_{1,k}\}_{k\in[j]}$ occurs given $\pi_j = \Delta$. We just need to bound $\mathsf{Pr}[E'_{1,j}|\pi_j = \Delta]$ for each $\Delta \in \mathcal{T}_j$.

We now fix a certain $\Delta \in \mathcal{T}_j$. For any polynomial P, denote the event F_P as $P =_{\mathrm{eval}} P_j$ and $P \neq_L P_j$. Then we know $E'_{1,j}$ implies one of $\{F_P\}_{P\in\mathsf{Cur}^\Delta}$ occurs and we have

$$\mathsf{Pr}[E'_{1,j}|\pi_j = \Delta] \leqslant \mathsf{Pr}\left[\bigvee_{P\in\mathsf{Cur}^\Delta} F_P|\pi_j = \Delta\right] .$$

Therefore, it is left to bound $\mathsf{Pr}[F_P]$ for each $P \in \mathsf{Cur}^\Delta$.

We now fix a certain $\hat{P} \in \mathsf{Cur}^\Delta$. Since P_j^Δ and L^Δ are fixed in Δ, we can directly check whether $\hat{P} =_{L^\Delta} P_j^\Delta$ or not. If $\hat{P} =_{L^\Delta} P_j^\Delta$, then we have $\mathsf{Pr}[F_{\hat{P}}] = 0$. Therefore, we can assume $\hat{P} \neq_{L^\Delta} P_j^\Delta$. Then, we only need to bound the probability of $\hat{P} =_{\mathrm{eval}} P_j^\Delta$. Since we fix $\pi_j = \Delta$, the only randomness here is the values assigned to the random variables $\mathsf{X}, \{\mathsf{A}_i, \mathsf{Y}_i\}_{i\in[\mathrm{sid}^\Delta]}$. Denote the values as $\vec{\eta} := (x, a_1, y_1 \cdot x, \ldots, a_{\mathrm{sid}^\Delta}, y_{\mathrm{sid}^\Delta} \cdot x) \in \mathbb{Z}_p^{1+2\mathrm{sid}^\Delta}$, where $x, \{a_i, y_i\}_{i\in[\mathrm{sid}^\Delta]}$ are random variables sampled in the game, and we have $P =_{\mathrm{eval}} P(\mathsf{X} = \eta_1, \{\mathsf{A}_i = \eta_{2i}, \mathsf{Y}_i = \eta_{2i+1}\}_{i\in[\mathrm{sid}^\Delta]})$.

To bound $\mathsf{Pr}[\hat{P} =_{\mathrm{eval}} P_j^\Delta|\pi_j = \Delta]$, we first introduce Lemma 15 below. Then the proof structure can be described as follows. We first define a sequence of polynomials $D_0, D_1, \ldots, D_m, B_1, \ldots, B_{q+1}$ over variables $\mathsf{X}, \{\mathsf{A}_i, \mathsf{Y}_i\}_{i\in[\mathrm{sid}^\Delta]}$ such that $B_{q+1} := \hat{P} - P_j^\Delta$. Then, we try to apply Lemma 15 to bound the probability by showing η is uniformly distributed over $\mathcal{C}$, $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} \neq \varnothing$, and $B_{q+1} \notin \mathsf{Span}(\{1, B_1, \ldots, B_q\})$ given $\pi_j = \Delta$, where $\mathcal{C}$ is defined in Lemma 15.

Lemma 15 (Lemma 1 in [BFP21]). *Let $D_1, \ldots, D_m, B_1, \ldots, B_{q+1}$ be polynomials in $\mathbb{Z}_p[\mathsf{X}_1, \ldots, \mathsf{X}_n]$ of degree 1. Let*

$$\mathcal{C} := \left(\bigcap_{i\in[q]} \mathrm{Zero}(B_i)\right) \setminus \left(\bigcup_{i\in[m]} \mathrm{Zero}(D_i)\right) ,$$

where $\mathrm{Zero}(P)$ means the zero set of P. Assume $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} \neq \varnothing$ and $B_{q+1} \notin \mathsf{Span}(\{1, B_1, \ldots, B_q\})$. If $\vec{x}$ is picked uniformly at random from $\mathcal{C}$ then

$$\frac{p-m}{p^2} \leqslant \mathsf{Pr}[B_{q+1}(\vec{x}) = 0] \leqslant \frac{1}{p-m} .$$

Let $m := \mathrm{sid}^\Delta + 1 + |\mathsf{Cur}^\Delta|(|\mathsf{Cur}^\Delta - 1|)$. Denote $D_1 := \mathsf{X}$ and $D_{i+1} := \mathsf{Y}_i$ for $i \in [\mathrm{sid}^\Delta]$. For each $P, P' \in \mathsf{Cur}^\Delta$ such that $P \neq P'$, denote $D_{P,P'} := P - P'$. We can relable $\{D_{P,P'}\}_{P,P'\in\mathsf{Cur}^\Delta, P\neq P'}$ to $D_{\mathrm{sid}^\Delta+2}, \ldots, D_m$.

Let $q := 2|\mathcal{I}_{\mathrm{fin}}^\Delta|$. For each $i \in \mathcal{I}_{\mathrm{fin}}^\Delta$, denote

$$B_{(i,1)} := \mathsf{A}_i + c_i^\Delta \mathsf{Y}_i - s_i^\Delta\ ,\ B_{i,2} := \mathsf{Y}_i - y_i^\Delta \mathsf{X}\ .$$

We can relabel $\{B_{(i,1)}, B_{(i,2)}\}_{i\in\mathcal{I}_{\mathrm{fin}}^\Delta}$ to $B_1, \ldots, B_q$ and denote $B_{q+1} := \hat{P} - P_j^\Delta$. Here one thing to notice is that we have $L^\Delta = \{B_1, \ldots, B_q\}$.

Denote $\mathcal{C} := \left(\bigcap_{i\in[q]} \mathrm{Zero}(B_i)\right) \setminus \left(\bigcup_{i\in[m]} \mathrm{Zero}(D_i)\right)$ and we have the following claim. The proof of the claim is deferred to Appendix B.2.

Claim 10 *In* $\mathrm{Game}_1^{\mathcal{A}}$*, for any* $\Delta \in \mathcal{T}_j$*, given* $\pi_j = \Delta$*, we have* $\vec{\eta}$ *is uniformly distributed over* $\mathcal{C}$*.*

We now continue to show that $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} \neq \varnothing$. If $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} = \varnothing$, since by the above claim η must be in $\mathcal{C}$ given $\pi_j = \Delta$, we know $B_{q+1} =_{\mathrm{eval}} B_{q+1}(\eta) \neq 0$, which implies $\mathsf{Pr}[\hat{P} =_{\mathrm{eval}} P_j | \pi_j = \Delta] = 0$. Therefore, we only need to consider the case when $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} \neq \varnothing$.

We then show that $B_{q+1} \notin \mathsf{Span}(\{1, B_1, \ldots, B_q\})$. Since $\hat{P} \neq_{L^{\mathsf{Cur}}} P_j^{\Delta}$ and $L^{\Delta} = \{B_1, \ldots, B_q\}$, we know $B_{q+1} \notin \mathsf{Span}(\{B_1, \ldots, B_q\})$. If $B_{q+1} \in \mathsf{Span}(\{1, B_1, \ldots, B_q\})$, we know there exists a constant $\delta \in \mathbb{Z}_p$ such that $\delta \neq 0$ and $B_{q+1} + \delta \in \mathsf{Span}(\{B_1, \ldots, B_q\})$. Let $B' = B_{q+1} + \delta$. Then, we have for any $\vec{\eta}_0 \in \mathcal{C}$, $B'(\vec{\eta}) = 0$ and thus $B_{q+1}(\vec{\eta}) = B'(\vec{\eta}) - \delta = -\delta \neq 0$, which means $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} = \varnothing$. This contradicts with the above argument that $\mathrm{Zero}(B_{q+1}) \cap \mathcal{C} \neq \varnothing$. Therefore, we have $B_{q+1} \notin \mathsf{Span}(\{1, B_1, \ldots, B_q\})$.

Then, by the above claim, we can apply Lemma 15 here and we have

$$\mathsf{Pr}[\hat{P} =_{\mathrm{eval}} P_j^{\Delta} | \pi_j = \Delta] = \mathsf{Pr}[B_{q+1}(\vec{\eta}) = 0 \mid \pi_j = \Delta] \leqslant \frac{1}{p - m} \ .$$

Since $m = \mathrm{sid}^{\Delta} + 1 + |\mathsf{Cur}^{\Delta}|(|\mathsf{Cur}^{\Delta} - 1|) \leqslant 1 + Q_{\mathrm{S}_1} + Q_{\Phi}^2$, we have

$$\begin{aligned} \mathsf{Pr}[E'_{1,j}] &= \sum_{\Delta \in \mathcal{T}_j} \mathsf{Pr}[E'_{1,j} \ \wedge \ \pi_j = \Delta] \\ &= \sum_{\Delta \in \mathcal{T}_j} \mathsf{Pr}[\pi_j = \Delta] \sum_{\hat{P} \in \mathsf{Cur}^{\Delta}} \mathsf{Pr}[F_{\hat{P}} \mid \pi_j = \Delta] \leqslant \frac{Q_{\Phi}}{p - (1 + Q_{\mathrm{S}_1} + Q_{\Phi}^2)} \ . \end{aligned}$$

Therefore, we have $\mathsf{Pr}[E_1] = \sum_{j \in [Q_\Phi]} \mathsf{Pr}[E'_{1,j}] \leqslant \frac{Q_\Phi^2}{p-(1+Q_{\mathrm{S}_1}+Q_\Phi^2)}$ and by (46)

$$\mathsf{Pr}[\mathrm{Game}_1^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\mathrm{Game}_2^{\mathcal{A}} = 1] + \frac{Q_{\Phi}^2}{p - (1 + Q_{\mathrm{S}_1} + Q_{\Phi}^2)} \ . \tag{47}$$

$\underline{\mathrm{Game}_2'^{\mathcal{A}}}$: This is defined in Figure 14 by ignoring the dashed box. The only difference between $\mathrm{Game}_2^{\mathcal{A}}$ and $\mathrm{Game}_2'^{\mathcal{A}}$ is that in the oracle Φ the condition "$\exists P' \in \mathsf{Cur}$ such that $P =_{\mathrm{eval}} P'$" is changed to "$\exists P' \in \mathsf{Cur}$ such that $P =_L P'$". We will show that $P =_{\mathrm{eval}} P'$ is equivalent to $P =_L P'$ here in Game_2, and thus we know the view of adversary are identical in these two games.

In $\mathrm{Game}_2^{\mathcal{A}}$, consider an query P to the oracle Φ. Let P' be an arbitrary polynomial in Cur. Consider the step when the condition "$\exists P' \in \mathsf{Cur}$ such that $P =_{\mathrm{eval}} P'$" is checked. We now show that $P =_{\mathrm{eval}} P'$ is if and only if $P =_L P'$. Suppose $P =_{\mathrm{eval}} P'$. Since the game does not abort, it must hold that $P =_L P'$. Therefore, we know $P =_{\mathrm{eval}} P'$ implies $P =_L P'$.

On the other hand, we show the following lemma.

Lemma 16. *In* $\mathrm{Game}_2^{\mathcal{A}}$*, at any step of the execution, we have*

$$\forall \ P \in \mathsf{Span}(L) \ : \ P =_{\mathrm{eval}} 0 \ , \tag{48}$$

which implies for any two polynomials P, P' *of variables* X *and* $\{\mathsf{A}_i, \mathsf{Y}_i\}_{i \in [sid]}$

$$P =_L P' \textit{ implies } P =_{\mathrm{eval}} P' \ . \tag{49}$$

Proof. We just need show that for each $R \in L$, we have $R =_{\mathrm{eval}} 0$. From the description of S_2, we know

$$L = \{\mathsf{A}_i + c_i \mathsf{Y}_i - s_i, \mathsf{Y}_i - y_i \mathsf{X}\}_{i \in \mathcal{I}_{\mathrm{fin}}} \ .$$

For $R = \mathsf{A}_i + c_i \mathsf{Y}_i - s_i$, we have $R =_{\mathrm{eval}} a_i + c_i \cdot y_i \cdot x_i - s_i = 0$, since $s_i = a_i + c_i \cdot y_i \cdot x_i$. For $R = \mathsf{Y}_i - y_i \mathsf{X}$, we have $R =_{\mathrm{eval}} y_i \cdot x - y_i \cdot x = 0$. Therfore, we know the lemma holds. $\square$

Game $\boxed{\mathrm{Game}_2'^{\mathcal{A}}}$, $\boxed{\mathrm{Game}_3^{\mathcal{A}}}$, $\mathrm{Game}_3'^{\mathcal{A}}$:

$p \leftarrow |\mathbb{G}_\lambda|$
$x \leftarrow\!\$\ \mathbb{Z}_p^*$; $\boxed{\text{assign } x \text{ to variable } \mathsf{X}}$
sid $\leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{S} \leftarrow \varnothing$; $\mathsf{Cur} \leftarrow \varnothing$; $\Xi \leftarrow ()$; $T \leftarrow ()$
$\{(m_k, \sigma_k)\}_{k\in[\ell+1]} \leftarrow\!\$\ \mathcal{A}^{\Pi,\mathrm{S}_1,\mathrm{S}_2,\mathrm{H}}(p, \Phi(1), \Phi(\mathsf{X}))$
If $\exists\ k_1 \neq k_2$ such that $(m_{k_1}, \sigma_{k_1}) = (m_{k_2}, \sigma_{k_2})$ then
  Return 0
If $\exists\ k \in [\ell+1]$ such that $y_k^* = 0$
  or $c_k \neq \mathrm{H}(\Phi(s_k - c_k \cdot y_k \cdot \mathsf{X}) \,\|\, \Phi(y_k \cdot \mathsf{X}) \,\|\, m_i)$
where $(c_k, s_k, y_k) = \sigma_k$ then return 0
Return 1

Oracle $\Phi(P)$:
$\boxed{\text{If } \exists P' \in \mathsf{Cur} \text{ such that } P =_{\mathrm{eval}} P' \text{ and } P \neq_L P' \text{ then } \textbf{abort game}}$
If $\exists P' \in \mathsf{Cur}$ such that $P =_L P'$ then
  Return $\Xi(P')$
$\Xi(P) \leftarrow\!\$\ \{0,1\}^{\log(p)} \backslash \Xi(\mathsf{Cur})$
$\mathsf{Cur} \leftarrow \mathsf{Cur} \cap \{P\}$
Return $\Xi(P)$

Oracle $\Pi(\xi, \xi', b)$:
If $\exists P, P' \in \mathsf{Cur}$ such that $\xi = \Xi(P)$
  and $\xi' = \Xi(P')$ then
  Return $\Phi(P + (-1)^b P')$
Else return $\perp$

Oracle S_1 :
sid $\leftarrow$ sid + 1
$a_{\mathrm{sid}} \leftarrow\!\$\ \mathbb{Z}_p$; $y_{\mathrm{sid}} \leftarrow\!\$\ \mathbb{Z}_p^*$
$\mathsf{st}_{\mathrm{sid}}^s \leftarrow (a_{\mathrm{sid}}, y_{\mathrm{sid}})$
$\boxed{\text{Assign } a_{\mathrm{sid}} \text{ to variable } \mathsf{A}_{\mathrm{sid}};\ \text{Assign } y_{\mathrm{sid}} \cdot x \text{ to varaible } \mathsf{Y}_{\mathrm{sid}}}$
$\mathsf{msg}_1 \leftarrow (\Phi(\mathsf{A}_{\mathrm{sid}}), \Phi(\mathsf{Y}_{\mathrm{sid}}))$
Return $(\mathrm{sid}, \mathsf{msg}_1)$

Oracle $\mathrm{S}_2(i, c_i)$:
If $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
$(a_i, y_i) \leftarrow \mathsf{st}_i^s$
$s_i \leftarrow a_i + c_i \cdot y_i \cdot x$
$R_1 \leftarrow \mathsf{A}_i + c_i \mathsf{Y}_i - s_i$
$R_2 \leftarrow \mathsf{Y}_i - y_i \mathsf{X}$
$L \leftarrow L \cup \{R_1, R_2\}$
$\mathsf{msg}_2 \leftarrow (s_i, y_i)$
$\boxed{\text{If } \exists\ P_1, P_2 \in \mathsf{Cur} \text{ such that } P_1 \neq P_2 \text{ and } P_1 =_L P_2 \text{ then } \textbf{abort game}}$
$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
$\ell \leftarrow \ell + 1$
Return msg_2

Oracle $\mathrm{H}(\mathrm{str})$:
If $T(\mathrm{str}) = \perp$ then
  $T(\mathrm{str}) \leftarrow\!\$\ \mathbb{Z}_p$
Return $T(\mathrm{str})$

Fig. 15. The definition for $\mathrm{Game}_3^{\mathcal{A}}$ and its difference from $\mathrm{Game}_2^{\mathcal{A}}$. $\mathrm{Game}_2^{\mathcal{A}}$ contains all but the solid boxes and $\mathrm{Game}_3^{\mathcal{A}}$ contains all but the dashed boxes. We also define an intermediate game $\mathrm{Game}_3'^{\mathcal{A}}$ which contains both dashed and solid boxes.

From the above lemma, we know $P =_L P'$ is equivalent to $P =_{\mathrm{eval}} P'$ at the step in Φ when the condition "$\exists P' \in \mathsf{Cur}$ such that $P =_{\mathrm{eval}} P'$". Therefore, we know the view of adversary are identical in these two games, which implies

$$\Pr[\mathrm{Game}_2^{\mathcal{A}} = 1] = \Pr[\mathrm{Game}_2'^{\mathcal{A}} = 1]\ . \tag{50}$$

$\underline{\mathrm{Game}_3^{\mathcal{A}}}$: $\mathrm{Game}_3^{\mathcal{A}}$ is defined in Figure 15 by ignoring the dashed box, where the only difference from $\mathrm{Game}_2'^{\mathcal{A}}$ is the orinal abort condition is removed from Φ and a new abort condition is added to S_2. Also, in $\mathrm{Game}_3^{\mathcal{A}}$, since the new abort condition only use the information L, we do not need to assign values to the variables anymore.

We first show that the oracle Φ in $\mathrm{Game}_3^{\mathcal{A}}$ is well-defined, that is, for each query P to Φ, there exists at most one $P' \in \mathsf{Cur}$ such that $P =_L P'$. Suppose during a query P to Φ in $\mathrm{Game}_2^{\mathcal{A}}$, the game does not abort and there exists $P', P''The \in \mathsf{Cur}$ such that $P' =_L P =_L P''$. Without loss of generality assume P'' is added to Cur after P'. If L is not updated after P'' is added to Cur, then by the description of Φ, we know $P' \neq_L P''$, which yields a contradiction. Otherwise, if L is updated after P'' is added to Cur. Consider the last time L is updated in S_2. Since $P' =_L P''$ and $P', P'' \in \mathsf{Cur}$, we know $\mathrm{Game}_3^{\mathcal{A}}$ must abort in S_2, which yields a contradiction. Therefore, we know the oracle Φ in $\mathrm{Game}_3^{\mathcal{A}}$ is well-defined.

To show that the probability $\mathcal{A}$ wins $\mathrm{Game}_2'^{\mathcal{A}}$ is bounded by the probability $\mathcal{A}$ wins $\mathrm{Game}_3^{\mathcal{A}}$, we introduce an itermidiate game $\mathrm{Game}_3'^{\mathcal{A}}$ which is defined in Figure 15 containing everything. We first show that the probability $\mathcal{A}$ wins $\text{Game}_2'^{\mathcal{A}}$ is bounded by the probability $\mathcal{A}$ wins $\text{Game}_3'^{\mathcal{A}}$. Denote the event E_2 in $\text{Game}_2'^{\mathcal{A}}$ as during a query to S_2 after L is updated,

$$\exists P_1, P_2 \in \mathsf{Cur} \text{ such that } P_1 \neq P_2 \text{ and } P_1 =_L P_2 \; .$$

Then, we have $\text{Game}_3'^{\mathcal{A}}$ is identical to $\text{Game}_2'^{\mathcal{A}}$ except it aborts when E_2 occurs, which implies

$$\mathsf{Pr}[\text{Game}_2'^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\text{Game}_3'^{\mathcal{A}} = 1] + \mathsf{Pr}[E_2] \; , \tag{51}$$

We now show that $\mathsf{Pr}[E_2] = 0$. Suppose E_2 occurs. Then, we know at some timestep in $\text{Game}_2'^{\mathcal{A}}$ there exists $P_1, P_2 \in \mathsf{Cur}$ such that $P_1 \neq P_2$ and $P_1 =_L P_2$. We first show that $P_1 \neq_{\text{eval}} P_2$. Suppose $P_1 =_{\text{eval}} P_2$. Without loss of generality assume P_1 is added to Cur before P_2. Consider the step when P_2 is added to Cur. Since P_1 is already in Cur, we know $P_1 \neq_L P_2$. However, since $P_2 \neq_L P_1$ but $P_2 =_{\text{eval}} P_1$, the game aborts, which yields a contradiction. Thus, we know $P_1 \neq_{\text{eval}} P_2$. Then, by Lemma 16, we know $P_1 \neq_L P_2$ at any timestep in $\text{Game}_2'^{\mathcal{A}}$, which yields a contradiction. Therefore, we know E_2 never occurs in $\text{Game}_2'^{\mathcal{A}}$, which implies

$$\mathsf{Pr}[\text{Game}_2'^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\text{Game}_3'^{\mathcal{A}} = 1] \; .$$

Also, since the only difference between $\text{Game}_3'^{\mathcal{A}}$ and $\text{Game}_3^{\mathcal{A}}$ is that $\text{Game}_3'^{\mathcal{A}}$ might abort in Φ while $\text{Game}_3^{\mathcal{A}}$ never abort in Φ, we have $\mathsf{Pr}[\text{Game}_3'^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\text{Game}_3^{\mathcal{A}} = 1]$. Therefore, we have

$$\mathsf{Pr}[\text{Game}_2'^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\text{Game}_3'^{\mathcal{A}} = 1] \leqslant \mathsf{Pr}[\text{Game}_3^{\mathcal{A}} = 1] \; . \tag{52}$$

$\underline{\text{Game}_4^{\mathcal{A}}}$: This is defined in Figure 16 by ignoring the dashed box. $\text{Game}_4^{\mathcal{A}}$ is identical to $\text{Game}_3^{\mathcal{A}}$, except the generation of $x, \{a_i, y_i, s_i\}_{i \in [sid]}$ are changed. More precisely, the sampling of x is removed from the main procedure, the sampling of a_{sid}, y_{sid} is removed from S_1, and in S_2, y_i is sampled from $\mathbb{Z}_p^*$ and s_i is sampled from $\mathbb{Z}_p$ instead of computing from a_i and y_i. The oracle Φ in $\text{Game}_4^{\mathcal{A}}$ is well-defined, which can be showed using the same way as in $\text{Game}_3^{\mathcal{A}}$.

We now show that the view of the adversary in Game_3 and Game_4 are identical. Since the value x and a_i are not used in $\text{Game}_3^{\mathcal{A}}$ except the dashed box, we just need to show that the distribution of (s_i, y_i) are identical in $\text{Game}_3^{\mathcal{A}}$ and $\text{Game}_4^{\mathcal{A}}$ for each query (i, c_i) to S_2. Consider the step when the adversary makes a query (i, c_i) to S_2 in $\text{Game}_3^{\mathcal{A}}$ and assume $i \in [sid] \backslash \mathcal{I}_{\text{fin}}$. The value y_i and a_i are not used anywhere in the game yet. Therefore, given the current transcript, the distribution of (s_i, y_i) is uniformly random in $\mathbb{Z}_p \times \mathbb{Z}_p^*$. Since $a_i \leftarrow s_i + c_i \cdot y_i \cdot x$ and s_i is uniformly in $\mathbb{Z}_p$ given y_i, we know the distribution of a_i is uniformly random in $\mathbb{Z}_p$ even given y_i. Therefore, the distribution of (a_i, y_i) is uniformly random in $\mathbb{Z}_p \times \mathbb{Z}_p^*$. Thus, we know the view of the adversary in $\text{Game}_3^{\mathcal{A}}$ and $\text{Game}_4^{\mathcal{A}}$ are identical, which implies

$$\mathsf{Pr}[\text{Game}_3^{\mathcal{A}} = 1] = \mathsf{Pr}[\text{Game}_4^{\mathcal{A}} = 1] \; . \tag{53}$$

B.2 Proof of Claim 10

Proof. Without loss of generality, assume the randomness used in Φ and the randomness of $\mathcal{A}$ are fixed and assume $\mathsf{Pr}[\pi_j = \Delta] > 0$ given those fixed randomness.

The claim is equivalent to show that

$$\forall\, \vec{\eta}_0 \in \mathcal{C} \;:\; \mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0 \mid \pi_j = \Delta] = \frac{1}{|\mathcal{C}|} \; .$$

The probability here is taken over the randomness $x, \vec{a}, \vec{y}$, where $\vec{a} = (a_1, \ldots, a_{\text{sid}^\Delta})$, $\vec{y} = (y_1, \ldots, y_{\text{sid}^\Delta})$. Also, x, $y_1, \ldots, y_{\text{sid}^\Delta}$ are picked uniformly at random from $\mathbb{Z}_p^*$ and $a_1, \ldots, a_{\text{sid}^\Delta}$ are picked uniformly at random from $\mathbb{Z}_p$.

We first show that

$$\pi_j = \Delta \quad \text{implies} \quad \vec{\eta} \in \mathcal{C} \; .$$

Game $\text{Game}_3^{\mathcal{A}}$, $\boxed{\text{Game}_4^{\mathcal{A}}}$:

$p \leftarrow |\mathbb{G}_\lambda|$
$x \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$\mathsf{sid} \leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{S} \leftarrow \varnothing$; $\mathsf{Cur} \leftarrow \varnothing$; $\Xi \leftarrow ()$; $T \leftarrow ()$
$\{(m_k, \sigma_k)\}_{k \in [\ell+1]} \leftarrow\!\!\$\ \mathcal{A}^{\Pi, \mathrm{S}_1, \mathrm{S}_2, \mathrm{H}}(p, \Phi(1), \Phi(\mathsf{X}))$
If $\exists\ k_1 \neq k_2$ such that $(m_{k_1}, \sigma_{k_1}) = (m_{k_2}, \sigma_{k_2})$ then
  Return 0
If $\exists\ k \in [\ell+1]$ such that $y_k^* = 0$
  or $c_k \neq \mathrm{H}(\Phi(s_k - c_k \cdot y_k \cdot \mathsf{X}) \,\|\, \Phi(y_k \cdot \mathsf{X}) \,\|\, m_i)$
where $(c_k, s_k, y_k) = \sigma_k$ then return 0
Return 1

Oracle $\Phi(P)$:
If $\exists P' \in \mathsf{Cur}$ such that $P =_L P'$ then
  Return $\Xi(P')$
$\Xi(P) \leftarrow\!\!\$\ \{0,1\}^{\log(p)} \backslash \Xi(\mathsf{Cur})$
$\mathsf{Cur} \leftarrow \mathsf{Cur} \cap \{P\}$
Return $\Xi(P)$

Oracle $\Pi(\xi, \xi', b)$:
If $\exists P, P' \in \mathsf{Cur}$ such that $\xi = \Xi(P)$
  and $\xi' = \Xi(P')$ then
  Return $\Phi(P + (-1)^b P')$
Else return $\perp$

Oracle S_1 :
$\mathsf{sid} \leftarrow \mathsf{sid} + 1$
$a_{\mathsf{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p$; $y_{\mathsf{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p^*$
$\mathsf{st}_{\mathsf{sid}}^s \leftarrow (a_{\mathsf{sid}}, y_{\mathsf{sid}})$
$\mathsf{msg}_1 \leftarrow (\Phi(\mathsf{A}_{\mathsf{sid}}), \Phi(\mathsf{Y}_{\mathsf{sid}}))$
Return $(\mathsf{sid}, \mathsf{msg}_1)$

Oracle $\mathrm{S}_2(i, c_i)$:
If $i \notin [\mathsf{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then return $\perp$
$(a_i, y_i) \leftarrow \mathsf{st}_i^s$
$s_i \leftarrow a_i + c_i \cdot y_i \cdot x$
$\boxed{s_i \leftarrow\!\!\$\ \mathbb{Z}_p;\ y_i \leftarrow\!\!\$\ \mathbb{Z}_p^*}$
$R_1 \leftarrow \mathsf{A}_i + c_i \mathsf{Y}_i - s_i$
$R_2 \leftarrow \mathsf{Y}_i - y_i \mathsf{X}$
$L \leftarrow L \cup \{R_1, R_2\}$
$\mathsf{msg}_2 \leftarrow (s_i, y_i)$
If $\exists\ P_1, P_2 \in \mathsf{Cur}$ such that
  $P_1 \neq P_2$ and $P_1 =_L P_2$
then **abort game**
$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$; $\ell \leftarrow \ell + 1$
Return msg_2

Oracle $\mathrm{H}(\mathrm{str})$:
If $T(\mathrm{str}) = \perp$ then
  $T(\mathrm{str}) \leftarrow\!\!\$\ \mathbb{Z}_p$
Return $T(\mathrm{str})$

Fig. 16. The definition for $\text{Game}_4^{\mathcal{A}}$ and its difference from $\text{Game}_3^{\mathcal{A}}$. $\text{Game}_3^{\mathcal{A}}$ contains all but the solid box and $\text{Game}_4^{\mathcal{A}}$ contains all but the dashed box.

Suppose $\pi_j = \Delta$ occurs. We just need to show $D_i(\eta) \neq 0$ for each $i \in [m]$ and $B_i(\eta) = 0$ for each $i \in [q]$. For $D_1, \ldots, D_{\mathsf{sid}^\Delta+1}$, since $x \neq 0$ and $y_i \neq 0$ for each $i \in [\mathsf{sid}^\Delta]$, we know $D_1(\eta) = x \neq 0$ and $D_{i+1}(\eta) = y_i \cdot x \neq 0$ for each $i \in [\mathsf{sid}^\Delta]$. For $D_{\mathsf{sid}^\Delta+1}, \ldots, D_m$, we make the argument using the original label $\{D_{P,P'}\}_{P,P' \in \mathsf{Cur}^\Delta, P \neq P'}$. For each $P, P' \in \mathsf{Cur}^\Delta$ such that $P \neq P'$, assume without loss of generality P is added to Cur before P'. When P' is added to Cur, since P is already in Cur, we know $P' \neq_{\mathrm{eval}} P$, which implies $D_{P,P'}(\vec{\eta}) = P'(\vec{\eta}) - P(\vec{\eta}) \neq 0$.

For $B_1, \ldots, B_q$, we also make the argument using the original label $\{B_{(i,1)}, B_{(i,2)}\}_{i \in \mathcal{I}_{\mathrm{fin}}^\Delta}$. For each $i \in \mathcal{I}_{\mathrm{fin}}^\Delta$, consider the query (i, c_i^Δ) made to S_2. Since $\pi_j = \Delta$, we have $s_i^\Delta = a_i + c_i^\Delta \cdot y_i \cdot x$ and $y_i^\Delta = y_i$. Therefore, we have $B_{(i,1)}(\vec{\eta}) = a_i + c_i^\Delta \cdot y_i \cdot x - s_i^\Delta = 0$ and $B_{(i,2)}(\vec{\eta}) = y_i \cdot x - y_i^\Delta \cdot x = 0$.[11] Therefore, we have $\vec{\eta} \in \mathcal{C}$.

We then show that

$$\vec{\eta} \in \mathcal{C} \quad \text{implies} \quad \pi_j = \Delta\,.$$

Since $\Pr[\pi_j = \Delta] > 0$, we know there exists $(x_0, \vec{a}_0, \vec{y}_0) \in \mathbb{Z}_p^{1+2\mathsf{sid}^\Delta}$ such that $\pi_j = \Delta$ when $(x, \vec{a}, \vec{y}) = (x_0, \vec{a}_0, \vec{y}_0)$. We now show that for any $(x_1, \vec{a}_1, \vec{y}_1) \in \mathbb{Z}_p^{1+2\mathsf{sid}^\Delta}$, given $(x, \vec{a}, \vec{y}) = (x_1, \vec{a}_1, \vec{y}_1)$ and $\vec{\eta} \in \mathcal{C}$, it must have $\pi_j = \Delta$.

Denote the case when $(x, \vec{a}, \vec{y}) = (x_0, \vec{a}_0, \vec{y}_0)$ as case 0 and the case when $(x, \vec{a}, \vec{y}) = (x_1, \vec{a}_1, \vec{y}_1)$ as case 1. We will show that the transcripts between the adversary and the oracles are exactly the same in these two cases, which implies $\pi_j = \Delta$ in case 1. We show this by induction. It is clear that the transcripts are the

[11] Note here the value $y_i \cdot x$ is assigned to Y_i

same at the begining. For a time step T, suppose the transcripts are the same prior to this step and we have the following situations:

- Query to Φ, S_1, Π: Suppose the adversary receives $(\Phi(1), \Phi(\mathsf{X}))$ or makes query to S_1 or Π at step T. For the case that the adversary makes query to S_1 or Φ, the transcripts can only differ on the invokation of Φ in S_1 or Π. Therefore, we only need to consider the queries and outputs of each Φ.
  For the k-th query to Φ where $k < j$, since the prior transcripts are the same in these two cases and the adversary is deterministic, we know the query P_k and the set Cur are the same in the two cases. If $P_k \neq_{\text{eval}} P'$ for any $P' \in \mathsf{Cur}$ in case 0, then we know P_k is added to Cur in case 0. Since $\pi_j = \Delta$ occurs in case 0, we know $\{P_k\} \cup \mathsf{Cur} \subseteq \mathsf{Cur}^{\Delta}$. Since $\vec{\eta}_1 \in \mathcal{C}$, we know $P_k(\vec{\eta}_1) \neq P'(\vec{\eta}_1)$ for any $P' \in \mathsf{Cur}^{\Delta}$. Therefore, we have $P_k \neq_{\text{eval}} P'$ for any $P' \in \mathsf{Cur}$ in case 1 too. Then, the outputs of Φ are the same in the two cases.
  Otherwise, if $P_k =_{\text{eval}} P'$ for some $P' \in \mathsf{Cur}$, we know such P' must be unique. Since $E_{1,k}$ does not occur in case 0, we have $P_k =_L P'$ in case 0. Since the current L is the same in the two cases, we know $P_k =_L P'$ in case 1 too. Since $P_k =_L P'$ implies $P_k =_{\text{eval}} P'$, we have $P_k =_{\text{eval}} P'$ in case 1 too. Thus, the output of Φ must be the same in the two cases. Therefore, we know the transcripts in these two cases must be the same after the k-th query to Φ is finished.
- Query to S_2: Suppose the adversary makes query (i, c_i) to S_2 at step T. Since $\pi_j = \Delta$ occurs in case 0, we know $i \in \mathcal{I}_{\text{fin}}^{\Delta}$, $c_i = c_i^{\Delta}$, $y_i = y_{0,i} = y_i^{\Delta}$, and $s_i = a_{0,i} + c_i \cdot y_{0,i} \cdot x_0 = s_i^{\Delta}$ in case 0. Since the transcripts are the same in the two cases prior to T and the adversary is deterministic, we know c_i is the same in both cases. Therefore, we know $c_i = c_i^{\Delta}$ in case 1. Since $\vec{\eta}_1 \in \mathcal{C}$, we have

$$B_{(i,1)}(\vec{\eta}_1) = a_{1,i} + c_i^{\Delta} \cdot y_{1,i} \cdot x_1 - s_i^{\Delta} = 0 \ ,$$

$$B_{(i,2)}(\vec{\eta}_1) = y_{1,i} \cdot x_1 - y_i^{\Delta} \cdot x_1 = 0 \ .$$

  Therefore, we have $y_i = y_{i,1} = y_i^{\Delta}$ and $s_i = a_{1,i} + c_i \cdot y_{1,i} \cdot x_1 = a_{1,i} + c_i^{\Delta} \cdot y_{1,i} \cdot x_1 = s_i^{\Delta}$ in case 1. Since the output (y_i, s_i) is the same in the two cases, we know the transcripts must be the same in these two cases after the query to S_2 is finished.
- Query to H : Since H does not envolve the randomness $x, \vec{a}, \vec{y}$, we know the transcripts are the same in the two cases after the query.

By induction, we know the transcript is the same by the step when the j-the query is made to Φ in the two cases. Therefore, we know $\pi_j = \Delta$ in case 1. Since it holds for any $(x_1, \vec{a}_1, \vec{y}_1) \in \mathbb{Z}_p^{\text{sid}^{\Delta}}$, we know $\vec{\eta} \in \mathcal{C}$ implies $\pi_j = \Delta$. Therefore, $\pi_j = \Delta$ is equivalent to $\vec{\eta} \in \mathcal{C}$, which implies for any $\vec{\eta}_0 \in \mathcal{C}$

$$\mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0 | \pi_j = \Delta] = \mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0 | \vec{\eta} \in \mathcal{C}] \ .$$

It is left to show $\mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0 | \vec{\eta} \in \mathcal{C}] = \frac{1}{|\mathcal{C}|}$ for any $\vec{\eta}_0 \in \mathcal{C}$. Denote $\mathcal{E} := \mathbb{Z}_p^* \times (\mathbb{Z}_p \times \mathbb{Z}_p^*)^{\text{sid}^{\Delta}}$ and we know $(x, a_1, y_1, \ldots, a_{\text{sid}^{\Delta}}, y_{\text{sid}^{\Delta}})$ is uniformly distributed over $\mathcal{E}$. Therefore, $\vec{\eta} = (x, a_1, y_1 \cdot x, \ldots, a_{\text{sid}^{\Delta}} isalsody_{\text{sid}^{\Delta} \cdot x})$ is also uniformly distributed over $\mathcal{E}$, which implies for any $\vec{\eta}_0 \in \mathcal{E}$,

$$\mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0] = \frac{1}{|\mathcal{E}|} \ .$$

Since $\mathcal{C} \subseteq \mathcal{E}$, we have for any $\vec{\eta}_0 \in \mathcal{C}$

$$\mathsf{Pr}_{x,\vec{a},\vec{y}}[\vec{\eta} = \vec{\eta}_0 | \vec{\eta} \in \mathcal{C}] = \frac{1/|\mathcal{E}|}{|\mathcal{C}|/|\mathcal{E}|} = \frac{1}{|\mathcal{C}|} \ .$$

$\square$

B.3 Proof of Claim 4

Proof (of Claim 4). Suppose $E_1 \wedge (\neg E_2)$ occurs. Denote str_j as the input of the j-th query to $\hat{\mathrm{H}}$. Denote the total number of queries to $\hat{\mathrm{H}}$ as $\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}$. Denote the decompositin of str_j as $\mathrm{str}_j = \xi_j^A \parallel \xi_j^Y \parallel m_j$. Denote Cur_j as the set Cur by the step when the j-th query to $\hat{\mathrm{H}}$ is made and denote $\mathsf{Cur}^{\mathrm{tot}}$ as the set Cur after $\mathcal{B}$ finishes the check of the condition (15) and (16). Since $\mathcal{B}$ makes a query str_k^* to $\hat{\mathrm{H}}$ to check the condition (16), there exists $j \in [\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}]$ such that $\mathrm{str}_j = \mathrm{str}_k^*$. Let $j_{\min}$ be the smallest index such that $\mathrm{str}_{j_{\min}} = \mathrm{str}_k^*$. Since $\mathrm{Hid}(\mathrm{str}_k^*) = \perp$, from the simulation of $\hat{\mathrm{H}}$, we know $\xi_{j_{\min}}^A \notin \varXi(\mathsf{Cur}_{j_{\min}})$ or $\xi_{j_{\min}}^Y \notin \varXi(\mathsf{Cur}_{j_{\min}})$. However, since $\xi_{j_{\min}}^A = \hat{\varPhi}(s_k^* - c_k^* \cdot y_k^* \cdot \mathsf{X})$ and $\xi_{j_{\min}}^Y = \hat{\varPhi}(y_k^* \cdot \mathsf{X})$, we know $\xi_{j_{\min}}^A, \xi_{j_{\min}}^Y \in \mathsf{Cur}^{\mathrm{tot}}$. Therefore, denote the set of all ξ_j^Y and ξ_j^A that do not correspond to any encoding of polynomials when the j-th query to $\hat{\mathrm{H}}$ is made as

$$D^{\mathrm{tot}} := \{\xi_j^A | j \in [\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}], \xi_j^A \notin \varXi(\mathsf{Cur}_j)\} \cup \{\xi_j^Y | j \in [\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}], \xi_j^Y \notin \varXi(\mathsf{Cur}_j)\} ,$$

and then we have at least one of $\xi_{j_{\min}}^A$ and $\xi_{j_{\min}}^Y$ is in $D \cap \varXi(\mathsf{Cur}^{\mathrm{tot}})$, which implies $D \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing$. Therefore, we have the event E occurs implies $D \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing$, which means

$$\mathsf{Pr}[E_1 \wedge (\neg E_2)] \leqslant \mathsf{Pr}[D^{\mathrm{tot}} \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing] . \tag{54}$$

It is left to bound $\mathsf{Pr}[D \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing]$.

Denote

$$D_j := \{\xi_{j'}^A | j' \in [j], \xi_{j'}^A \notin \varXi(\mathsf{Cur}_{j'})\} \cup \{\xi_{j'}^Y | j' \in [j], \xi_{j'}^Y \notin \varXi(\mathsf{Cur}_{j'})\} .$$

Denote $\mathsf{Cur}^{(i)}$ as the set Cur after the i-th query to $\hat{\varPhi}$ is finished and $\mathsf{Cur}^{(0)} = \varnothing$. Consider the step when the i-th query to $\hat{\varPhi}$ is made. Denote the number of queries to $\hat{\mathrm{H}}$ before the i-th query to $\hat{\varPhi}$ is made as $\mathrm{num}_{\hat{\mathrm{H}}}^{(i)}$. Denote the event E_i' as $D_{\mathrm{num}_{\hat{\mathrm{H}}}^{(i)}} \cap \varXi(\mathsf{Cur}^{(i-1)}) = \varnothing$ and $D_{\mathrm{num}_{\hat{\mathrm{H}}}^{(i)}} \cap \varXi(\mathsf{Cur}^{(i)}) \neq \varnothing$. We first show that if $D^{\mathrm{tot}} \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing$, then there exists i such that E_i' occurs, and then bound $\mathsf{Pr}[E_i']$ for each i.

Denote the total number of queries to $\hat{\varPhi}$ as $\mathrm{num}_{\hat{\varPhi}}^{\mathrm{tot}}$. Suppose none of $\{E_i'\}_{i \in [\mathrm{num}_{\hat{\varPhi}}^{\mathrm{tot}}]}$ occurs. We show that at any time step, supposing the number of queries to $\hat{\varPhi}$ made so far is i and the number of queries to $\hat{\mathrm{H}}$ made so far is j, we have $D_j \cap T(\mathsf{Cur}^{(i)}) = \varnothing$, which implies $D^{\mathrm{tot}} \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) = \varnothing$. We show the statement by induction. At the begining, we know $i = 0$, $j = 0$, $\mathsf{Cur}^{(0)} = \varnothing$, and $D_0 = \varnothing$. Thus, the statement holds trivially. For any time step with $i > 0$ or $j > 0$, suppose the latest query is made to $\hat{\mathrm{H}}$ and we have $D_{j-1} \cap T(\mathsf{Cur}^{(i)}) = \varnothing$. Consider the step when the j-th query to $\hat{\mathrm{H}}$ is made. If $T(\mathrm{str}_j) \neq \perp$, we have $D_j = D_{j-1}$ and $D_j \cap T(\mathsf{Cur}^{(i)}) = \varnothing$. Otherwise, if $T(\mathrm{str}_j) = \perp$, we have $D_j = D_{j-1} \cup (\{\xi_j^A, \xi_j^Y\} \backslash T(\mathsf{Cur}_j))$. Since $\mathsf{Cur}_j = \mathsf{Cur}^{(i)}$, we have $D_j \cap T(\mathsf{Cur}^{(i)}) = D_{j-1} \cup T(\mathsf{Cur}^{(i)}) = \varnothing$. Therefore, we have $D_j \cap T(\mathsf{Cur}^{(i)}) = \varnothing$. Otherwise, suppose the latest query is made to $\hat{\varPhi}$ and we have $D_j \cap T(\mathsf{Cur}^{(i-1)}) = \varnothing$. Since we have $j = \mathrm{num}_{\hat{\mathrm{H}}}^{(i)}$ and E_i' does not occur, we have $D_j \cap T(\mathsf{Cur}^{(i-1)}) = D_{\mathrm{num}_{\hat{\mathrm{H}}}^{(i)}} \cap \varXi(\mathsf{Cur}^{(i)}) = \varnothing$. Therefore, by induction, the statement holds. Then, considering the step when $\mathcal{B}$ finishes the check of the condition (15) and (16), we have $D^{\mathrm{tot}} \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) = D_{\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}} \cap \varXi(\mathsf{Cur}^{(\mathrm{num}_{\hat{\varPhi}}^{\mathrm{tot}})}) = \varnothing$. Therefore, if $D^{\mathrm{tot}} \cap \varXi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing$, then at least one of $\{E_i'\}_{i \in [\mathrm{num}_{\hat{\varPhi}}^{\mathrm{tot}}]}$ occurs.

Finally, to bound $\mathsf{Pr}[E_i']$, consider the i-th query to $\hat{\varPhi}$. Denote the input of the i-th query to $\hat{\varPhi}$ as P_i. Denote $j = \mathrm{num}_{\hat{\mathrm{H}}}^{(i)}$ for simplicity. Suppose E_i' occurs. We know $\mathsf{Cur}^{(i)} \neq \mathsf{Cur}^{(i-1)}$, which implies $\mathsf{Cur}^{(i)} = \mathsf{Cur}^{(i-1)} \cup \{P_i\}$ and $\varXi(\mathsf{Cur}^{(i)}) = \varXi(\mathsf{Cur}^{(i-1)}) \cup \{\varXi(P_i)\}$. Since $D_j \cap \varXi(\mathsf{Cur}^{(i-1)}) = \varnothing$ and $D_j \cap \varXi(\mathsf{Cur}^{(i)}) \neq \varnothing$, we know $\varXi(P_i) \in D_j$. Therefore, we have

$$\mathsf{Pr}[E_i'] \leqslant \mathsf{p}[\mathsf{Cur}^{(i)} = \mathsf{Cur}^{(i-1)} \cap \{P_i\} \wedge \varXi(P_i) \in D_j] .$$

Game $\mathrm{OMDL}_{\mathbb{G}}^{\mathcal{A}}(\lambda)$:	Oracle $\textsc{Chal}$:
$p \leftarrow \lvert\mathbb{G}_\lambda\rvert$; $g \leftarrow g(\mathbb{G}_\lambda)$	$\mathrm{cid} \leftarrow \mathrm{cid} + 1$
$\mathrm{cid} \leftarrow 0$; $\ell \leftarrow 0$	$x_{\mathrm{cid}} \leftarrow\!\$\ \mathbb{Z}_p$
$\{y_i\}_{i \in [\mathrm{cid}]} \leftarrow \mathcal{A}^{\textsc{Chal},\textsc{DLog}}(p, g, \mathbb{G}_\lambda)$	Return $g^{x_{\mathrm{cid}}}$
If $\ell \geqslant \mathrm{cid}$ then return 0	Oracle $\textsc{DLog}(X)$:
If $\forall\, i \in [\mathrm{cid}] : y_i = x_i$ then	$\ell \leftarrow \ell + 1$
Return 1	Return $\log_g(X)$
Return 0	

Fig. 17. The OMDL game.

Consider the step when $\Xi(P_i)$ is generated. We know D_j is already determined. Therefore, we know $\Xi(P_i)$ is sampled uniformly at random from $\{0,1\}^{\log(p)} \backslash \Xi(\mathsf{Cur}^{(i-1)})$ independent of D_j, which implies

$$\begin{aligned}\Pr[E_i'] &\leqslant \Pr[\mathsf{Cur}^{(i)} = \mathsf{Cur}^{(i-1)} \cup \{P_i\} \ \wedge\ \Xi(P_i) \in D_j] \\ &\leqslant \Pr[\Xi(P_i) \in D_j \mid \mathsf{Cur}^{(i)} = \mathsf{Cur}^{(i-1)} \cup \{P_i\}] \\ &\leqslant \frac{|D_j|}{p - |\mathsf{Cur}^{(i-1)}|} \leqslant \frac{|D^{\mathrm{tot}}|}{p - |\mathsf{Cur}^{\mathrm{tot}}|}\ .\end{aligned}$$

Therefore, we have

$$\Pr[D^{\mathrm{tot}} \cap \Xi(\mathsf{Cur}^{\mathrm{tot}}) \neq \varnothing] \leqslant \Pr\left[\bigvee_{i \in [\mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}}]} E_i'\right] \leqslant \sum_{i \in [\mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}}]} \Pr[E_i'] \leqslant \frac{\mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}} \cdot |D^{\mathrm{tot}}|}{p - |\mathsf{Cur}^{\mathrm{tot}}|}.$$

Since $|D^{\mathrm{tot}}| \leqslant 2\mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}} \leqslant 2(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)$ and $|\mathsf{Cur}^{\mathrm{tot}}| \leqslant \mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}} \leqslant Q_\Phi$, by (54), we have

$$\Pr[E_1 \ \wedge\ (\neg E_2)] \leqslant \frac{2\mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}} \cdot \mathrm{num}_{\hat{\mathrm{H}}}^{\mathrm{tot}}}{p - \mathrm{num}_{\hat{\Phi}}^{\mathrm{tot}}} = \frac{2Q_\Phi(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)}{p - Q_\Phi}\ .$$

□

C A Scheme Secure under OMDL

In this section, we present our second blind signature scheme, BS_2, that is proved secure in AGM assuming the hardness of the one-more discrete logarithm (OMDL) problem [BNPS03], which is formalized in Figure 17. We also denote by $\mathsf{Adv}_{\mathbb{G}}^{\mathrm{omdl}}(\mathcal{A}, \lambda)$ the corresponding advantage that $\mathcal{A}$ wins the game. The adversary is now given access to a powerful oracle that can compute discrete logarithms, but if the adversary queries this oracle ℓ times, it is asked to solve $\ell + 1$ discrete-log instances. While the OMDL game gives more power to an adversary compared to the classical DL problem, its generic concrete security is comparable, as recently proved by Fuchsbauer et al. [BFP21].

The scheme BS_2 is described in Figure 18. It very much resembles BS_1, with the exception that the commitment C is now $g^t X^y$ instead of X^y. This also gives us a more involved blinding method. Still, the resulting scheme is perfectly blind, as shown by the following theorem. (Its proof is very similar to the blindness proof of $\mathsf{BS}_1[\mathbb{G}]$, so we defer it to Appendix D.1.)

Theorem 8. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. Then, the blind signature scheme $\mathsf{BS}_2[\mathbb{G}]$ is perfectly blind.*

The core of our analysis is the following theorem, which asserts the one-more unforgeability of BS_2 in the AGM, assuming random oracles.

Algorithm $\mathsf{BS}_2.\mathsf{Setup}(1^\lambda)$: $p \leftarrow \lvert\mathbb{G}_\lambda\rvert$; $g \leftarrow g(\mathbb{G}_\lambda)$ Select $\mathrm{H} : \{0,1\}^* \to \mathbb{Z}_p$ Return $par \leftarrow (p, \mathbb{G}, g, \mathrm{H})$ Algorithm $\mathsf{BS}_2.\mathsf{KG}(par)$: $(p, \mathbb{G}, g, \mathrm{H}) \leftarrow par$ $x \leftarrow_{\$} \mathbb{Z}_p$; $X \leftarrow g^x$ $sk \leftarrow x$; $pk \leftarrow X$ Return (sk, pk) Algorithm $\mathsf{BS}_2.\mathsf{S}_1(sk)$: $x \leftarrow sk$; $X \leftarrow g^x$ $a, t \leftarrow_{\$} \mathbb{Z}_p$; $y \leftarrow_{\$} \mathbb{Z}_p^*$ $A \leftarrow g^a$; $C \leftarrow g^t X^y$ $\mathsf{st}^s \leftarrow (a, y, t, x)$; $\mathsf{msg}_1 \leftarrow (A, C)$ Return $(\mathsf{st}^s, \mathsf{msg}_1)$ Algorithm $\mathsf{BS}_2.\mathsf{S}_2(\mathsf{st}^s, c)$: $(a, y, t, x) \leftarrow \mathsf{st}^s$ $s \leftarrow a + c \cdot y \cdot x$ Return $\mathsf{msg}_2 \leftarrow (s, y, t)$	Algorithm $\mathsf{BS}_2.\mathsf{U}_1(pk, \mathsf{msg}_1, m)$: $X \leftarrow pk$; $(A, C) \leftarrow \mathsf{msg}_1$ $r_1, r_2, r_3 \leftarrow_{\$} \mathbb{Z}_p$; $\gamma \leftarrow_{\$} \mathbb{Z}_p^*$ $A' \leftarrow g^{r_1} \cdot A^\gamma \cdot C^{r_3 \cdot \gamma}$ $C' \leftarrow C^\gamma g^{r_2}$ $c' \leftarrow H(A' \parallel C' \parallel m)$ $c \leftarrow c' + r_3$ $\mathsf{st}^u \leftarrow (c, c', r_1, r_2, r_3, \gamma, X, Z, A, C)$ Return (st^u, c) Algorithm $\mathsf{BS}_2.\mathsf{U}_2(\mathsf{st}^u, \mathsf{msg}_2)$: $(c, c', r_1, r_2, r_3, \gamma, X, Z, A, C) \leftarrow \mathsf{st}^u$ $(s, y, t) \leftarrow \mathsf{msg}_2$ If $y = 0$ or $C \neq g^t X^y$ or $g^s \neq A \cdot X^{c \cdot y}$ then return $\perp$ $s' \leftarrow \gamma \cdot s + r_1 + r_3 \cdot \gamma \cdot t$ $y' \leftarrow \gamma \cdot y$ $t' \leftarrow \gamma \cdot t + r_2$ Return $\sigma \leftarrow (c', s', y', t')$ Algorithm $\mathsf{BS}_2.\mathsf{Ver}(pk, \sigma, m)$: $(c, s, y, t) \leftarrow \sigma$ If $y = 0$ then return 0 $C \leftarrow g^t X^y$; $A \leftarrow g^s \cdot X^{-c \cdot y}$ If $c \neq \mathrm{H}(A \parallel C \parallel m)$ then return 0 Return 1

Fig. 18. The blind signature scheme $\mathsf{BS}_2 = \mathsf{BS}_2[\mathbb{G}]$.

Theorem 9. *Let $\mathbb{G}$ be an (asymptotic) family of* prime-order *cyclic groups. For any algebraic adversary $\mathcal{A}_{\mathrm{alg}}$ for the game $\mathrm{OMUF}^{\mathsf{BS}_2[\mathbb{G}]}(\lambda)$ making at most Q_{S_1} queries to S_1 and Q_{H} queries to the random oracle H, there exists an adversary $\mathcal{B}_{\mathrm{omdl}}$ running in a similar running time as $\mathcal{A}_{\mathrm{alg}}$ for the* OMDL *problem making at most $2Q_{\mathrm{S}_1} + 1$ queries to* Chal *such that*

$$\mathsf{Adv}^{\mathrm{omuf}}_{\mathsf{BS}_2[\mathbb{G}]}(\mathcal{A}_{\mathrm{alg}}, \lambda) \leqslant \mathsf{Adv}^{\mathrm{omdl}}_{\mathbb{G}}(\mathcal{B}_{\mathrm{omdl}}, \lambda) + \frac{(Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1)(Q_{\mathrm{H}} + 3Q_{\mathrm{S}_1} + 2)}{p - 1}\ .$$

The proof of Theorem 9 resembles the proof of security for BS_1 in Theorem 3, in particular, by relying on the WFROS game.

Proof (Theorem 9). Let us fix an adversary $\mathcal{A}_{\mathrm{alg}}$ making at most Q_{S_1} queries to S_1, and Q_{H} queries to the random oracle H. Without loss of generality, assume $\mathcal{A}_{\mathrm{alg}}$ makes exactly Q_{S_1} queries to S_1 and exactly one query (i, c_i) to S_2 for each $i \in [Q_{\mathrm{S}_1}]$. Then, after $\mathcal{A}_{\mathrm{alg}}$ returns, we know $\ell = Q_{\mathrm{S}_1}$ and $\mathcal{I}_{\mathrm{fin}} = [Q_{\mathrm{S}_1}]$.

The $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game is formally defined in Figure 19. In addition to the original OMUF game (defined in Figure 1), for each query $(A \parallel C \parallel m)$ to H, its corresponding hid is recorded in $\mathrm{Hid}(A \parallel C \parallel m)$ and the output of the query is recorded as δ_{hid}. Also, since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, $\mathcal{A}_{\mathrm{alg}}$ also provides the representations of A and C, and the corresponding coefficients $\vec{\alpha}$ and $\vec{\beta}$ are recorded as $\vec{\alpha}_{\mathrm{hid}}$ and $\vec{\beta}_{\mathrm{hid}}$.

Denote the event WIN as $\mathcal{A}_{\mathrm{alg}}$ wins the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game, i.e., all output message-signature pairs $\{m_k^*, \sigma_k^*\}_{k \in [Q_{\mathrm{S}_1}+1]}$ are distinct and valid. Furthermore, let us denote $\mathrm{str}_k^* := g^{s_k^*} X^{-c_k^* \cdot y_k^*} \parallel g^{t_k^*} X^{y_k^*} \parallel m_k^*$. We let E be the event in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game for which, after the validity of the output is checked, for each

Game $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}(\lambda)$:

$p \leftarrow |\mathbb{G}_\lambda|$; $g \leftarrow g(\mathbb{G}_\lambda)$; $x \leftarrow\!\!\$\ \mathbb{Z}_p$; $X \leftarrow g^x$
sid $\leftarrow 0$; $\ell \leftarrow 0$; $\mathcal{I}_{\mathrm{fin}} \leftarrow \varnothing$; $T \leftarrow ()$; hid $\leftarrow 0$; Hid $\leftarrow ()$
$\{(m_k^*, \sigma_k^*)\}_{k\in[\ell+1]} \leftarrow\!\!\$\ \mathcal{A}_{\mathrm{alg}}^{\mathrm{S}_1,\mathrm{S}_2,\mathrm{H}}(p, g, \mathbb{G}_\lambda, X)$
If $\exists\, k_1 \neq k_2$ such that $(m_{k_1}^*, \sigma_{k_1}^*) = (m_{k_2}^*, \sigma_{k_2}^*)$ then
  Return 0
If $\exists\, k \in [\ell+1]$ such that $y_k^* = 0$
  or $c_k^* \neq \mathrm{H}(g^{s_k^*} X^{-c_k^* \cdot y_k^*} \,\|\, g^{t_k^*} X^{y_k^*} \,\|\, m_k^*)$
where $(c_k^*, s_k^*, y_k^*, t_k^*) = \sigma_k^*$ then return 0
Return 1

Oracle $\mathrm{H}(A \,\|\, C \,\|\, m)$:

If $T(A \,\|\, C \,\|\, m) = \perp$ then
  $T(A \,\|\, C \,\|\, m) \leftarrow\!\!\$\ \mathbb{Z}_p$
  hid $\leftarrow$ hid + 1
  $\mathrm{Hid}(A \,\|\, C \,\|\, m) \leftarrow$ hid
  // $A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} \prod_{i\in[\mathrm{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}}$
  // $C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} \prod_{i\in[\mathrm{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}}$
  $\delta_{\mathrm{hid}} \leftarrow T(\mathcal{A} \,\|\, C \,\|\, m)$; $\vec{\hat{\alpha}}_{\mathrm{hid}} \leftarrow \vec{\hat{\alpha}}$; $\vec{\hat{\beta}}_{\mathrm{hid}} \leftarrow \vec{\hat{\beta}}$
Return $T(A \,\|\, C \,\|\, m)$

Oracle S_1 :

sid $\leftarrow$ sid + 1
$a_{\mathrm{sid}}, t_{\mathrm{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p$; $y_{\mathrm{sid}} \leftarrow\!\!\$\ \mathbb{Z}_p$
$\mathsf{st}^s_{\mathrm{sid}} \leftarrow (a_{\mathrm{sid}}, y_{\mathrm{sid}}, t_{\mathrm{sid}})$
$A_{\mathrm{sid}} \leftarrow g^{a_{\mathrm{sid}}}$
$C_{\mathrm{sid}} \leftarrow g^{t_{\mathrm{sid}}} X^{y_{\mathrm{sid}}}$
$\mathsf{msg}_1 \leftarrow (A_{\mathrm{sid}}, C_{\mathrm{sid}})$
Return $(\mathrm{sid}, \mathsf{msg}_1)$

Oracle $\mathrm{S}_2(i, c_i)$:

If $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$ then
  Return $\perp$
$(a_i, y_i, t_i) \leftarrow \mathsf{st}^s_i$
$s_i \leftarrow a_i + c_i \cdot y_i \cdot x$
$\mathsf{msg}_2 \leftarrow (s_i, y_i, t_i)$
$\mathcal{I}_{\mathrm{fin}} \leftarrow \mathcal{I}_{\mathrm{fin}} \cup \{i\}$
$\ell \leftarrow \ell + 1$
Return msg_2

Fig. 19. The OMUF security game for the blind signature scheme $\mathsf{BS}_2[\mathbb{G}]$ and Game_1 used in the proof of Theorem 9, where $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ contains all but the solid box and $\mathrm{Game}_1^{\mathcal{A}_{\mathrm{alg}}}$ contains all.

$k \in [Q_{\mathrm{S}_1} + 1]$ and $j = \mathrm{Hid}(\mathrm{str}_k^*)$,[12] the following conditions hold:

$$\hat{\alpha}_j^{\mathsf{X}} + \sum_{i\in[Q_{\mathrm{S}_1}]} y_i(\hat{\alpha}_j^{\mathsf{C}_i} - c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i}) = -\delta_j \cdot y_k^* \ , \tag{55}$$

$$\hat{\beta}_j^{\mathsf{X}} + \sum_{i\in[Q_{\mathrm{S}_1}]} y_i(\hat{\beta}_j^{\mathsf{C}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i}) = y_k^* \ . \tag{56}$$

Since $\mathsf{Adv}^{\mathrm{omuf}}_{\mathsf{BS}_2[\mathbb{G}]}(\mathcal{A}_{\mathrm{alg}}, \lambda) = \mathsf{Pr}[\mathrm{WIN}] = \mathsf{Pr}[\mathrm{WIN} \wedge E] + \mathsf{Pr}[\mathrm{WIN} \wedge (\neg E)]$, the theorem follows by combining the following two lemmas with Theorem 1.

Lemma 17. *There exists an adversary* $\mathcal{B}_{\mathrm{wfros}}$ *for the* $\mathrm{WFROS}_{Q_{\mathrm{S}_1},p}$ *problem making at most* $Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1$ *queries to the random oracle* H *such that*

$$\mathsf{Adv}^{\mathrm{wfros}}_{Q_{\mathrm{S}_1},p}(\mathcal{B}_{\mathrm{wfros}}) + \frac{Q_{\mathrm{H}} + Q_{\mathrm{S}_1} + 1}{p} \geqslant \mathsf{Pr}[E_1 \wedge E_2] \ . \tag{57}$$

Lemma 18. *There exists an adversary* $\mathcal{B}_{\mathrm{omdl}}$ *running in similar running time as* $\mathcal{A}_{\mathrm{alg}}$ *for the* OMDL *problem making at most* $2Q_{\mathrm{S}_1} + 1$ *queries to* CHAL, *such that*

$$\mathsf{Adv}^{\mathrm{omdl}}_{\mathbb{G}}(\mathcal{B}_{\mathrm{omdl}}, \lambda) \geqslant \mathsf{Pr}[\mathrm{Game}_1^{\mathcal{A}_{\mathrm{alg}}} = 1] \ . \tag{58}$$

□

C.1 Proof of Lemma 17

The proof is almost the same as the proof of Lemma 10.

Proof. We first give a detailed description of $\mathcal{B}_{\mathrm{wfros}}$ playing the game $\mathrm{WFROS}_{Q_{\mathrm{S}_1},p}$.

[12] Here, $\mathrm{Hid}(\mathrm{str}_k^*)$ must be defined since a query str_k^* is made to H when checking the validity of the output (m_k^*, σ_k^*).

The adversary $\mathcal{B}_{\text{wfros}}$. To start with, $\mathcal{B}_{\text{wfros}}$ initializes sid, $\mathcal{I}_{\text{fin}}$, ℓ, T, hid, and Hid as described in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game. In addition, $\mathcal{B}_{\text{wfros}}$ samples x uniformly from $\mathbb{Z}_p$ and sets X to g^x.

Then, $\mathcal{B}_{\text{wfros}}$ runs $\mathcal{A}_{\text{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\text{S}}_1$, $\hat{\text{S}}_2$, and $\hat{\text{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{S}}_1$: Same as the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game except that instead of sampling y_{sid}, t_{sid} randomly and setting $C_{\text{sid}} \leftarrow g^{t_{\text{sid}}} X^{y_{\text{sid}}}$, $\mathcal{B}_{\text{wfros}}$ samples a new variable t'_{sid} uniformly from $\mathbb{Z}_p$ and sets $C_{\text{sid}} = g^{t'_{\text{sid}}}$.

Oracle $\hat{\mathbf{S}}_2$: After receiving a query (i, c_i) to $\hat{\text{S}}_2$ from $\mathcal{A}_{\text{alg}}$, if $i \notin [\text{sid}] \backslash \mathcal{I}_{\text{fin}}$, $\mathcal{B}_{\text{wfros}}$ returns $\perp$. Otherwise, $\mathcal{B}_{\text{wfros}}$ makes a query (i, c_i) to S and uses its output as the value y_i. Also, $\mathcal{B}_{\text{wfros}}$ sets $t_i = t'_i - y_i \cdot x$. With the value (a_i, y_i, t_i), the rest of $\hat{\text{S}}_2$ is the same as S_2 in the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game.

Oracle $\hat{\mathbf{H}}$: After receiving a query $(A \,\|\, C \,\|\, m)$ to $\hat{\text{H}}$ from $\mathcal{A}_{\text{alg}}$, if $T(A \,\|\, C \,\|\, m) \neq \perp$, the value $T(A \,\|\, C \,\|\, m)$ is returned. Otherwise, since $\mathcal{A}_{\text{alg}}$ is algebraic, $\mathcal{B}_{\text{wfros}}$ also knows the coefficient $\vec{\hat{\alpha}}$ and $\vec{\hat{\beta}}$ such that

$$A = g^{\hat{\alpha}^g} X^{\hat{\alpha}^{\mathsf{X}}} \prod_{i \in [\text{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}} C_i^{\hat{\alpha}^{\mathsf{C}_i}} \;,\; C = g^{\hat{\beta}^g} X^{\hat{\beta}^{\mathsf{X}}} \prod_{i \in [\text{sid}]} A_i^{\hat{\beta}^{\mathsf{A}_i}} C_i^{\hat{\beta}^{\mathsf{C}_i}} \;.$$

Then, $\mathcal{B}_{\text{wfros}}$ issues the query $(\vec{\alpha}, \vec{\beta})$ to H, where $\vec{\alpha}, \vec{\beta} \in \mathbb{Z}_p^{2Q_{\text{S}_1}+1}$ are such that

$$\alpha^{(i')} = \begin{cases} \hat{\alpha}^{\mathsf{X}} \,, & i' = 0 \\ \hat{\alpha}^{\mathsf{C}_i} \,, & i' = 2i-1 \,,\; i \in [\text{sid}] \\ -\hat{\alpha}^{\mathsf{A}_i} \,, & i' = 2i \,,\; i \in [\text{sid}] \\ 0 \,, & o.w. \end{cases}$$
$$\beta^{(i')} = \begin{cases} -\hat{\beta}^{\mathsf{X}} \,, & i' = 0 \\ -\hat{\beta}^{\mathsf{C}_i} \,, & i' = 2i-1 \,,\; i \in [\text{sid}] \\ \hat{\beta}^{\mathsf{A}_i} \,, & i' = 2i \,,\; i \in [\text{sid}] \\ 0 \,, & o.w. \end{cases} \;. \tag{59}$$

After receiving the output $(\delta_{\text{hid}}, \text{hid})$, $\mathcal{B}_{\text{wfros}}$ sets $T(A \,\|\, C \,\|\, m) \leftarrow \delta_{\text{hid}}$ and $\text{Hid}(A \,\|\, C \,\|\, m) \leftarrow \text{hid}$. Finally, $\mathcal{B}_{\text{wfros}}$ returns $T(A \,\|\, C \,\|\, m)$.

After $\mathcal{A}_{\text{alg}}$ outputs $\{(m^*_k, \sigma^*_k)\}_{k \in [Q_{\text{S}_1}+1]}$, $\mathcal{B}_{\text{wfros}}$ aborts if the conditions from the event WIN $\wedge$ E do not occur. Otherwise, $\mathcal{B}_{\text{wfros}}$ outputs $\mathcal{J} := \{\text{Hid}(\text{str}^*_k) \mid k \in [Q_{\text{S}_1}+1]\}$.

Following an analysis similar to $\mathcal{B}$ in the GGM (Section 4.2), we know $\mathcal{B}_{\text{wfros}}$ makes at most $Q_{\text{H}} + Q_{\text{S}_1} + 1$ queries to H and $\mathcal{B}_{\text{wfros}}$ simulates the $\text{OMUF}^{\mathcal{A}_{\text{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game statistically close to perfect with distance bounded by $\frac{Q_{\text{H}}+Q_{\text{S}_1}+1}{p}$. Therefore, the probability that WIN $\wedge$ E occurs when running $\mathcal{B}_{\text{wfros}}$ is at least $\mathsf{Pr}[\text{WIN} \wedge E] - \frac{Q_{\text{H}}+Q_{\text{S}_1}+1}{p}$.

It is left to show that if WIN $\wedge$ E occurs within the simulation, then $\mathcal{B}_{\text{wfros}}$ wins the WFROS game. We first show that $|\mathcal{J}| = Q_{\text{S}_1} + 1$. Suppose $|\mathcal{J}| \leqslant Q_{\text{S}_1}$. Then, we know there exists $k_1, k_2 \in [Q_{\text{S}_1} + 1]$ such that $k_1 \neq k_2$ and $\text{Hid}(\text{str}^*_{k_1}) = \text{Hid}(\text{str}^*_{k_2})$, which implies $\text{str}^*_{k_1} = \text{str}^*_{k_2}$. Therefore, we have

$$g^{s^*_{k_1}} X^{-c^*_{k_1} \cdot y^*_{k_1}} = g^{s^*_{k_2}} X^{-c^*_{k_2} \cdot y^*_{k_2}} \;,\; g^{t^*_{k_1}} X^{y^*_{k_1}} = g^{t^*_{k_2}} X^{y^*_{k_2}} \;,\; m^*_{k_1} = m^*_{k_2} \;. \tag{60}$$

Also, let $j = \text{Hid}(\text{str}^*_{k_1}) = \text{Hid}(\text{str}^*_{k_2})$. Since E occurs, by (56), we have

$$y^*_{k_1} = \hat{\beta}^{\mathsf{X}}_j + \sum_{i \in [Q_{\text{S}_1}]} y_i (\hat{\beta}^{\mathsf{C}_i}_j - c_i \cdot \hat{\beta}^{\mathsf{A}_i}_j) = y^*_{k_2} \;.$$

Since $y^*_{k_1} = y^*_{k_2}$ and $c^*_{k_1} = c^*_{k_2}$, by (60), we have

$$t^*_{k_1} = t^*_{k_2} \;,\; s^*_{k_1} = s^*_{k_2} \;.$$

However, since $(m^*_{k_1}, \sigma^*_{k_1})$ and $(m^*_{k_2}, \sigma^*_{k_2})$ are different message-signature pairs, we have

$$(m^*_{k_1}, c^*_{k_1}, s^*_{k_1}, y^*_{k_1}, t^*_{k_1}) \neq (m^*_{k_2}, c^*_{k_2}, s^*_{k_2}, y^*_{k_2}, t^*_{k_2}),$$

which yields a contradiction. Therefore, we have $|\mathcal{J}| = Q_{\mathrm{S}_1} + 1$.

Then, since E occurs in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game simulated by $\mathcal{B}_{\mathrm{wfros}}$, by (55) and (56), it holds that for any $j \in \mathcal{J}$

$$\alpha^{\mathsf{X}}_j + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i(\alpha^{\mathsf{C}_i}_j - c_i \cdot \alpha^{\mathsf{A}_i}_j) = -\delta_j \left(\hat{\beta}^{\mathsf{X}}_j + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i(\hat{\beta}^{\mathsf{C}_i}_j - c_i \cdot \hat{\beta}^{\mathsf{A}_i}_j) \right).$$

From the simulation of $\hat{\mathrm{H}}$, by (59), we have for any $j \in \mathcal{J}$

$$\alpha^{(0)}_j + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i(\alpha^{(2i-1)}_j + c_i \cdot \alpha^{(2i)}_j) = \delta_j \left(\beta^{(0)}_j + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i(\beta^{(2i-1)}_j + c_i \cdot \beta^{(2i)}_j) \right).$$

Therefore, $\mathcal{B}_{\mathrm{wfros}}$ wins the $\mathrm{WFROS}_{Q_{\mathrm{S}_1},p}$ game. □

C.2 Proof of Lemma 18

Proof. We first give a detailed description of $\mathcal{B}_{\mathrm{omdl}}$ playing the $\mathrm{OMDL}_{\mathbb{G}}$ game.

THE ADVERSARY $\mathcal{B}_{\mathrm{omdl}}$. To start with, $\mathcal{B}_{\mathrm{omdl}}$ initializes sid, $\mathcal{I}_{\mathrm{fin}}$, ℓ, T, hid, and Hid as described in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game.

After $\mathcal{B}_{\mathrm{omdl}}$ receives $(p, g, \mathbb{G}_\lambda)$ from the $\mathrm{OMDL}_{\mathbb{G}}$ game, $\mathcal{B}_{\mathrm{wfros}}$ sets $X \leftarrow\!\!\$\ \textsc{Chal}()$ and runs $\mathcal{A}_{\mathrm{alg}}$ on input $(p, g, \mathbb{G}_\lambda, X)$ and with access to the oracles $\hat{\mathrm{S}}_1$, $\hat{\mathrm{S}}_2$, and $\hat{\mathrm{H}}$. These oracles operate as follows:

Oracle $\hat{\mathbf{S}}_1$: After receving a query to $\hat{\mathrm{S}}_1$ from $\mathcal{A}_{\mathrm{alg}}$, $\mathcal{B}_{\mathrm{omdl}}$ increases sid by one and sets $A_{\mathrm{sid}} \leftarrow\!\!\$\ \textsc{Chal}()$ and $C_{\mathrm{sid}} \leftarrow\!\!\$\ \textsc{Chal}()$. Then, $\mathcal{B}_{\mathrm{omdl}}$ returns $(\mathrm{sid}, A_{\mathrm{sid}}, C_{\mathrm{sid}})$.

Oracle $\hat{\mathbf{S}}_2$: After receving a query (i, c_i) to $\hat{\mathrm{S}}_2$ from $\mathcal{A}_{\mathrm{alg}}$, if $i \notin [\mathrm{sid}] \backslash \mathcal{I}_{\mathrm{fin}}$, $\mathcal{B}_{\mathrm{omdl}}$ returns $\perp$. Otherwise, $\mathcal{B}_{\mathrm{omdl}}$ samples y_i uniformly from $\mathbb{Z}^*_p$ and sets $s_i \leftarrow \textsc{DLog}(AX^{c_i \cdot y_i})$ and $t_i \leftarrow \textsc{DLog}(CX^{-y_i})$. Then, $\mathcal{B}_{\mathrm{omdl}}$ returns (s_i, y_i, t_i).

Oracle $\hat{\mathbf{H}}$: Same as in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game.

After receiving the output $\{(m^*_k, \sigma^*_k)\}_{k \in [Q_{\mathrm{S}_1}+1]}$, $\mathcal{B}_{\mathrm{omdl}}$ aborts if the event WIN $\wedge$ $(\neg E)$ does not occur. Otherwise, we show in Claim 11 that $\mathcal{B}_{\mathrm{omdl}}$ can compute the discrete log of X.

Denote $x := \log_g(X)$. Then, for each $i \in [Q_{\mathrm{S}_1} + 1]$, $\mathcal{B}_{\mathrm{omdl}}$ computes the discrete log of A_i and C_i as $a_i \leftarrow s_i - c_i \cdot y_i \cdot x$ and $t'_i \leftarrow t_i + y_i \cdot x$. Finally, $\mathcal{B}_{\mathrm{omdl}}$ returns $(x, a_1, c_1, \ldots, a_{Q_{\mathrm{S}_1}}, c_{Q_{\mathrm{S}_1}})$.

ANALYSIS OF $\mathcal{B}_{\mathrm{omdl}}$. Note that $\mathcal{B}_{\mathrm{omdl}}$ makes one queries to $\textsc{Chal}$ to get X, two queries to $\textsc{Chal}$ when it receives a query to $\hat{\mathrm{S}}_1$, and two queries to $\textsc{Chal}$ when it receives a query to $\hat{\mathrm{S}}_2$. Therefore, $\mathcal{B}_{\mathrm{omdl}}$ makes $2Q_{\mathrm{S}_1} + 1$ queries to $\textsc{Chal}$ and $2Q_{\mathrm{S}_1}$ queries to $\textsc{DLog}$. Also, it is clear that $\mathcal{B}_{\mathrm{omdl}}$ simulates oracles S_1, S_2, H in the $\mathrm{OMUF}^{\mathcal{A}_{\mathrm{alg}}}_{\mathsf{BS}_2[\mathbb{G}]}$ game perfectly, and $\mathcal{B}_{\mathrm{omdl}}$ wins the OMDL game if it can compute the discrete log of X correctly. Therefore, we can conclude the lemma with the following claim.

Claim 11 *If* WIN $\wedge$ E *occurs when running* $\mathcal{B}_{\mathrm{omdl}}$*, then* $\mathcal{B}_{\mathrm{omdl}}$ *can compute the discrete log of* X*.*

□

Proof (of Claim 11). Suppose WIN $\wedge$ E occurs within the simulation. We know WIN occurs, but one of (55) and (56) does not hold.

Case 1: (55) does not hold. There exists $k \in [Q_{\mathrm{S}_1} + 1]$ and $j := \mathrm{Hid}(\mathrm{str}^*_k)$ such that $\hat{\alpha}^{\mathsf{X}}_j + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i(\hat{\alpha}^{\mathsf{C}_i}_j - c_i \cdot \hat{\alpha}^{\mathsf{A}_i}_j) \neq -\delta_j \cdot y^*_k$. Since WIN occurs, we know $c^*_k = \hat{\mathrm{H}}(\mathrm{str}^*_k) = \delta_j$. Then, since $\mathrm{Hid}(\mathrm{str}^*_k) = j$, we have

$$g^{s^*_k} X^{-\delta_j \cdot y^*_k} = g^{\hat{\alpha}^g_j} X^{\hat{\alpha}^{\mathsf{X}}_j} \prod_{i \in [\mathrm{sid}]} A_i^{\hat{\alpha}^{\mathsf{A}_i}_j} C_i^{\hat{\alpha}^{\mathsf{C}_i}_j}. \tag{61}$$

Similar to case 1, by substituting $A_i = g^{s_i} X^{-c_i \cdot y_i}$ and $C_i = g^{t_i} X^{y_i}$ into the equation (61), we have

$$g^{s_k^*} X^{-\delta_j \cdot y_k^*} = g^{\hat{\alpha}_j^g + \sum_{i \in [Q_{\mathrm{S}_1}]} (\hat{\alpha}_j^{\mathsf{A}_i} \cdot s_i + \hat{\alpha}_j^{\mathsf{C}_i} \cdot t_i)} X^{\hat{\alpha}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i (\hat{\alpha}_j^{\mathsf{C}_i} - c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i})} .$$

Therefore, $\mathcal{B}_{\mathrm{omdl}}$ can compute the discrete log of X as

$$x := \frac{s_k^* - \hat{\alpha}_j^g - \sum_{i \in [Q_{\mathrm{S}_1}]} (\hat{\alpha}_j^{\mathsf{A}_i} \cdot s_i + \hat{\alpha}_j^{\mathsf{C}_i} \cdot t_i)}{\hat{\alpha}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i (\hat{\alpha}_j^{\mathsf{C}_i} - c_i \cdot \hat{\alpha}_j^{\mathsf{A}_i}) + \delta_j \cdot y_k^*} .$$

Case 2: (56) does not hold. There exists $k \in [Q_{\mathrm{S}_1} + 1]$ and $j := \mathrm{Hid}(\mathrm{str}_k^*)$ such that $\hat{\beta}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i (\hat{\beta}_j^{\mathsf{C}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i}) \neq y_k^*$. Since $\mathrm{Hid}(\mathrm{str}_k^*) = j$, we have

$$g^{t_k^*} X^{y_k^*} = g^{\hat{\beta}_j^g} X^{\hat{\beta}_j^{\mathsf{X}}} \prod_{i \in [Q_{\mathrm{S}_1}]} A_i^{\hat{\beta}_j^{\mathsf{A}_i}} C_i^{\hat{\beta}_j^{\mathsf{C}_i}} . \tag{62}$$

From the simulation of $\hat{\mathrm{S}}_2$, for each $i \in [Q_{\mathrm{S}_1}]$, we have

$$g^{s_i} = A_i X^{c_i \cdot y_i} , \ g^{t_i} = C_i X^{-y_i} .$$

By substituting $A_i = g^{s_i} X^{-c_i \cdot y_i}$ and $C_i = g^{t_i} X^{y_i}$ into (62), we have

$$g^{t_k^*} X^{y_k^*} = g^{\hat{\beta}_j^g + \sum_{i \in [Q_{\mathrm{S}_1}]} (\hat{\beta}_j^{\mathsf{A}_i} \cdot s_i + \hat{\beta}_j^{\mathsf{C}_i} \cdot t_i)} X^{\hat{\beta}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i (\hat{\beta}_j^{\mathsf{C}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i})} .$$

Therefore, $\mathcal{B}_{\mathrm{omdl}}$ can compute the discrete log of X as

$$x := \frac{t_k^* - \hat{\beta}_j^g - \sum_{i \in [Q_{\mathrm{S}_1}]} (\hat{\beta}_j^{\mathsf{A}_i} \cdot s_i + \hat{\beta}_j^{\mathsf{C}_i} \cdot t_i)}{\hat{\beta}_j^{\mathsf{X}} + \sum_{i \in [Q_{\mathrm{S}_1}]} y_i (\hat{\beta}_j^{\mathsf{C}_i} - c_i \cdot \hat{\beta}_j^{\mathsf{A}_i}) - y_k^*} .$$

□

D Blindness Proofs

D.1 Blindness of BS_2

Proof. Let $\mathcal{A}$ be an adversary playing the $\mathrm{Blind}^{\mathcal{A}}_{\mathsf{BS}_2[\mathbb{G}]}$ game. Similar to the blindness proof of $\mathsf{BS}_1[\mathbb{G}]$, we can assume the randomness of $\mathcal{A}$ is fixed and $\mathcal{A}$ always finishes both signing sessions and receives valid signatures (σ_0, σ_1) without loss of generality.

Define the view of $\mathcal{A}$ after its execution as $\pi = (X, m_0, m_1, T_0, T_1, \sigma_0, \sigma_1)$, where $T_i := (A_i, C_i, c_i, s_i, y_i, t_i)$, denoting the transcripts learned from interactions with the i-th signing session and $\sigma_i = (c_i', s_i', y_i', t_i')$. Since the randomness of $\mathcal{A}$ is fixed, the only randomness left is the randomness in U_1 and U_2. Denote $\eta := (r_1^{(0)}, r_2^{(0)}, r_3^{(0)}, \gamma^{(0)}, r_1^{(1)}, r_2^{(1)}, r_3^{(1)}, \gamma^{(1)})$ as the total randomness. To prove the theorem, we need only show that the distribution of π is identical in both the case $b = 0$ and $b = 1$. We prove this by showing that for any fixed view Δ such that $\Pr[\pi = \Delta | b = 1] > 0$, there exists a unique value of the randomness η that makes $\pi = \Delta$ for the cases $b = 0$ and $b = 1$.

For both the cases $b = 0$ and $b = 1$, we now show that $\pi = \Delta$ if and only if for each $i \in \{0, 1\}$, it holds that

$$\begin{aligned} \gamma^{(i)} &= {y'_{b_i}}^{\Delta} / y_i^{\Delta} , \\ r_1^{(i)} &= {s'_{b_i}}^{\Delta} - \gamma^{(i)} (s_i^{\Delta} + r_3^{(i)} \cdot t_i^{\Delta}) , \\ r_2^{(i)} &= {t'_{b_i}}^{\Delta} - \gamma^{(i)} \cdot t_i^{\Delta} , \\ r_3^{(i)} &= c_i^{\Delta} - {c'_{b_i}}^{\Delta} . \end{aligned} \tag{63}$$

where the superscript $(\cdot)^{\Delta}$ represents the corresponding value in Δ. From the algorithms $\mathsf{BS}_2.\mathsf{U}_1$ and $\mathsf{BS}_2.\mathsf{U}_2$, it is clear that the "only if" part holds. For the "if" part, suppose (63) holds. Since the randomness of $\mathcal{A}$ is fixed, the view of $\mathcal{A}$ can differ only on the outputs c_0, c_1 from the oracle U_1 or the output (σ_0, σ_1) from the oracle U_2. Since both signatures in Δ are valid, we have

$$A_i^{\Delta} = g^{s_i^{\Delta}} X^{\Delta - c_i^{\Delta} \cdot y_i^{\Delta}} \ , \ C_i^{\Delta} = g^{t_i^{\Delta}} X^{\Delta y_i^{\Delta}} \ , \tag{64}$$

$$c'_{b_i}{}^{\Delta} = \mathrm{H}(g^{s'_{b_i}{}^{\Delta}} X^{\Delta - y'_{b_i}{}^{\Delta} \cdot c'_{b_i}{}^{\Delta}} \parallel g^{t'_{b_i}{}^{\Delta}} X^{\Delta y'_{b_i}{}^{\Delta}} \parallel m_{b_i}^{\Delta}) \ . \tag{65}$$

For c_i where $i \in \{0,1\}$, suppose the values in the view of $\mathcal{A}$ that have already determined when c_i is generated, which must include (X, m_i, A_i, C_i), is consistent with Δ. By (63), we have

$$\begin{aligned}
c_i &= r_3^{(i)} + \mathrm{H}(g^{r_1^{(i)}} A_i^{\gamma^{(i)}} C_i^{\gamma^{(i)} \cdot r_3^{(i)}} \parallel g^{r_2^{(i)}} C_i^{\gamma^{(i)}} \parallel m_{b_i}) \\
&= r_3^{(i)} + \mathrm{H}(g^{r_1^{(i)}} A_i^{\Delta \gamma^{(i)}} C_i^{\Delta \gamma^{(i)} \cdot r_3^{(i)}} \parallel g^{r_2^{(i)}} C_i^{\Delta \gamma^{(i)}} \parallel m_{b_i}^{\Delta}) \\
&= r_3^{(i)} + \mathrm{H}(g^{r_1^{(i)} + \gamma^{(i)}(s_i^{\Delta} + r_3^{(i)} \cdot t_i^{\Delta})} X^{\Delta - y_i^{\Delta} \cdot \gamma^{(i)} \cdot (c_i^{\Delta} - r_3^{(i)})} \parallel g^{r_2^{(i)} + \gamma^{(i)} \cdot t_i^{\Delta}} X^{\Delta y_i^{\Delta} \cdot \gamma^{(i)}} \parallel m_{b_i}^{\Delta}) \\
&= r_3^{(i)} + \mathrm{H}(g^{s'_{b_i}{}^{\Delta}} X^{\Delta - y'_{b_i}{}^{\Delta} \cdot c'_{b_i}{}^{\Delta}} \parallel g^{t'_{b_i}{}^{\Delta}} X^{\Delta y'_{b_i}{}^{\Delta}} \parallel m_{b_i}^{\Delta}) \\
&= r_3^{(i)} + c'_{b_i}{}^{\Delta} = c_i^{\Delta} \ .
\end{aligned}$$

where the third equality is due to (64), the fourth equality is due to (63), and the final equality is due to (65). Then, consider the step when (σ_0, σ_1) is output. Suppose the current view, which contains T_i, are consistent with Δ. By (63), we have

$$\begin{aligned}
y'_{b_i} &= \gamma^{(i)} \cdot y_i = \gamma^{(i)} \cdot y_i^{\Delta} = y'_{b_i}{}^{\Delta} \ , \\
s'_{b_i} &= r_1^{(i)} + \gamma^{(i)}(s_i + r_3^{(i)} \cdot t_i) = r_1^{(i)} + \gamma^{(i)}(s_i^{\Delta} + r_3^{(i) \cdot t_i^{\Delta}}) = s'_{b_i}{}^{\Delta} \ , \\
t'_{b_i} &= r_2^{(i)} + \gamma^{(i)} \cdot t_i = r_2^{(i)} + \gamma^{(i)} \cdot t_i^{\Delta} = t'_{b_i}{}^{\Delta} \ , \\
c'_{b_i} &= c_i - r_3^{(i)} = c_i^{\Delta} - r_3^{(i)} = c'_{b_i}{}^{\Delta} \ ,
\end{aligned}$$

which implies $(\sigma_0, \sigma_1) = (\sigma_0^{\Delta}, \sigma_1^{\Delta})$. Therefore, by induction, if (63) holds, we know $\pi = \Delta$. □

D.2 Blindness of BS_3

Proof. Let $\mathcal{A}$ be an adversary playing the $\mathrm{Blind}^{\mathcal{A}}_{\mathsf{BS}_3[\mathbb{G}]}$ game. Similar to the blindness proof of $\mathsf{BS}_1[\mathbb{G}]$ and $\mathsf{BS}_2[\mathbb{G}]$, we can assume the randomness of $\mathcal{A}$ is fixed and $\mathcal{A}$ always finishes both signing sessions and receives valid signatures (σ_0, σ_1) without loss of generality.

Define the view of $\mathcal{A}$ after its execution as $\pi = (X, Z, m_0, m_1, T_0, T_1, \sigma_0, \sigma_1)$, where $T_i := (A_i, C_i, c_i, s_i, y_i, t_i)$, denoting the transcripts learned from interactions with the i-th signing session and $\sigma_i = (c'_i, s'_i, y'_i, t'_i)$. Since the randomness of $\mathcal{A}$ is fixed, the only randomness left is the randomness in U_1 and U_2. Denote $\eta := (r_1^{(0)}, r_2^{(0)}, \gamma_1^{(0)}, \gamma_2^{(0)}, r_1^{(1)}, r_2^{(1)}, \gamma_1^{(1)}, \gamma_2^{(1)})$ as the total randomness. To prove the theorem, we need only show that the distribution of π is identical in both the case $b = 0$ and $b = 1$. We prove this by showing that for any fixed view Δ such that $\Pr[\pi = \Delta | b = 1] > 0$, there exists a unique value of the randomness η that makes $\pi = \Delta$ for the cases $b = 0$ and $b = 1$.

For both the cases $b = 0$ and $b = 1$, we now show that $\pi = \Delta$ if and only if for each $i \in \{0,1\}$, it holds that

$$\begin{aligned}
\gamma_1^{(i)} &= y'_{b_i}{}^{\Delta} / y_i^{\Delta} \ , \\
\gamma_2^{(i)} &= c_i^{\Delta} / c'_{b_i}{}^{\Delta} \ , \\
r_1^{(i)} &= s'_{b_i}{}^{\Delta} - s_i^{\Delta} \cdot (\gamma_1^{(i)} / \gamma_2^{(i)}) \ , \\
r_2^{(i)} &= t'_{b_i}{}^{\Delta} - \gamma_1^{(i)} \cdot t_i^{\Delta} \ ,
\end{aligned} \tag{66}$$

where the superscript $(\cdot)^{\Delta}$ represents the corresponding value in Δ. From the algorithms $\mathsf{BS}_3.\mathsf{U}_1$ and $\mathsf{BS}_3.\mathsf{U}_2$, it is clear that the "only if" part holds. For the "if" part, suppose (66) holds. Since the randomness of $\mathcal{A}$ is fixed, the view of $\mathcal{A}$ can differ only on the outputs c_0, c_1 from the oracle U_1 or the output (σ_0, σ_1) from the oracle U_2. Since both signatures in Δ are valid, we have

$$A_i^{\Delta} = g^{s_i^{\Delta}} X^{\Delta^{-c_i^{\Delta} \cdot y_i^{\Delta}}} \;,\; C_i^{\Delta} = g^{t_i^{\Delta}} Z^{\Delta^{y_i^{\Delta}}} \;. \tag{67}$$

$$c'_{b_i}{}^{\Delta} = \mathrm{H}(g^{s'_{b_i}{}^{\Delta}} X^{\Delta^{-y'_{b_i}{}^{\Delta} \cdot c'_{b_i}{}^{\Delta}}} \,\|\, g^{t'_{b_i}{}^{\Delta}} Z^{\Delta^{y'_{b_i}{}^{\Delta}}} \,\|\, m_{b_i}^{\Delta}) \;. \tag{68}$$

For c_i where $i \in \{0, 1\}$, suppose the values in the view of $\mathcal{A}$ that have already determined when c_i is generated, which must include (X, m_i, A_i, C_i), are consistent with Δ. By (63), we have

$$\begin{aligned}
c_i &= \gamma_2^{(i)} \cdot \mathrm{H}(g^{r_1^{(i)}} A_i^{\gamma_1^{(i)}/\gamma_2(i)} \,\|\, g^{r_2^{(i)}} C_i^{\gamma_1^{(i)}} \,\|\, m_{b_i}) \\
&= \gamma_2^{(i)} \cdot \mathrm{H}(g^{r_1^{(i)}} A_i^{\Delta^{\gamma_1^{(i)}/\gamma_2(i)}} \,\|\, g^{r_2^{(i)}} C_i^{\Delta^{\gamma_1^{(i)}}} \,\|\, m_{b_i}^{\Delta}) \\
&= \gamma_2^{(i)} \cdot \mathrm{H}(g^{r_1^{(i)} + s_i^{\Delta} \cdot (\gamma_1^{(i)}/\gamma_2^{(i)})} X^{\Delta^{-y_i^{\Delta} \cdot c_i^{\Delta} \cdot (\gamma_1^{(i)}/\gamma_2^{(i)})}} \,\|\, g^{r_2^{(i)} + \gamma^{(i)} \cdot t_i^{\Delta}} Z^{\Delta^{y_i^{\Delta} \cdot \gamma_1^{(i)}}} \,\|\, m_{b_i}^{\Delta}) \\
&= \gamma_2^{(i)} \cdot \mathrm{H}(g^{s'_{b_i}{}^{\Delta}} X^{\Delta^{-y'_{b_i}{}^{\Delta} \cdot c'_{b_i}{}^{\Delta}}} \,\|\, g^{t'_{b_i}{}^{\Delta}} Z^{\Delta^{y'_{b_i}{}^{\Delta}}} \,\|\, m_{b_i}^{\Delta}) \\
&= \gamma_2^{(i)} \cdot c'_{b_i}{}^{\Delta} = c_i^{\Delta} \;.
\end{aligned}$$

where the third equality is due to (67), the fourth equality is due to (66), and the final equality is due to (68). Then, consider the step when (σ_0, σ_1) are output. Suppose the current view, which contains T_i, is consistent with Δ. By (63), we have

$$\begin{aligned}
y'_{b_i} &= \gamma_1^{(i)} \cdot y_i = \gamma_1^{(i)} \cdot y_i^{\Delta} = y'_{b_i}{}^{\Delta} \;, \\
s'_{b_i} &= r_1^{(i)} + s_i(\gamma_1^{(i)}/\gamma_2^{(i)}) = r_1^{(i)} + s_i^{\Delta}(\gamma_1^{(i)}/\gamma_2^{(i)}) = s'_{b_i}{}^{\Delta} \;, \\
t'_{b_i} &= r_2^{(i)} + \gamma_1^{(i)} \cdot t_i = r_2^{(i)} + \gamma_1^{(i)} \cdot t_i^{\Delta} = t'_{b_i}{}^{\Delta} \;, \\
c'_{b_i} &= c_i/\gamma_2^{(i)} = c_i^{\Delta}/\gamma_2^{(i)} = c'_{b_i}{}^{\Delta} \;,
\end{aligned}$$

which implies $(\sigma_0, \sigma_1) = (\sigma_0^{\Delta}, \sigma_1^{\Delta})$. Therefore, by induction, if (66) holds, we know $\pi = \Delta$. $\square$