

Three-Move Blind Signatures from DL

Rutchathon Chairattana-Apirom¹, Michael Reichle^{2*}, and Stefano Tessaro¹

¹ Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle, US
`{rchairat,tessaro}@cs.washington.edu`

² INSAIT, Sofia University “St. Kliment Ohridski”, Bulgaria
`michael.reichle@insait.ai`

Abstract. This paper considers the problem of building blind signatures in pairing-free groups. We provide the first three-move blind signature which is provably one-more unforgeable, in the random-oracle model, under the minimal assumption that the discrete logarithm (DL) problem is hard. Our construction in fact achieves one-more *strong* unforgeability, and also supports partial blindness. Blindness is statistical, also in the ROM. Our construction makes black-box use of the underlying group and does not rely on non-black-box techniques. For most such constructions, three moves are necessary by a recent result of Dietz, Kastner, and Tessaro (CRYPTO ’26).

We build on recent work by Chairattana-Apirom, Reichle, and Tessaro (CRYPTO ’26), which achieved the same combination of round complexity and security guarantees, but under the stronger decisional Diffie–Hellman (DDH) assumption. In particular, we follow the same paradigm of boosting the security of a weakly secure scheme (namely, the Okamoto–Schnorr blind signature) to a fully secure one by authenticating the initial nonce. We provide, however, a substantially different instantiation of this paradigm that dispenses with the use of DDH and establishes security from DL alone.

* This work was done while the author was at ETH Zurich.

Table of Contents

1	Introduction	3
2	Technical Overview	5
3	Preliminaries	8
4	Blindable Σ -protocols	15
5	Blind Signature from DL	24
5.1	Straightline-extractable NIZK	25
5.2	Construction	27
5.3	Correctness and Security	30
6	Correctness Proof of $\mathbf{BS}_{\text{phf}}$ (Theorem 5.3)	31
7	OMSUF Proof of $\mathbf{BS}_{\text{phf}}$ (Theorem 5.4)	33
7.1	Useful Lemmata	33
7.2	Proof of Theorem 5.4	35
7.3	Proof of Lemma 7.3 (Ruling out Relaxed SLE Witness)	37
7.4	Proof of Lemma 7.4 (Ruling out H-Query Collision in Forgeries)	38
7.5	Proof of Lemma 7.5 (Reducing to AOMPR from Nonce Collision)	42
7.6	Proof of Lemma 7.6 (Extracting DL from Distinct Nonces)	45
7.7	Proof of Lemma 7.11 (Introducing PHF Aborts)	49
8	Blindness Proof of $\mathbf{BS}_{\text{phf}}$ (Theorem 5.5)	50
A	AI Usage	58
B	Proof of Theorem 3.11 ((2, poly)-Programmability)	58
C	Proof of Lemma 3.21 (Three-Fork Rewinding)	61

1 Introduction

A *blind signature* lets a user authenticate a message without disclosing it to the signer, who holds the signing key. Moreover, the resulting signature is unlinkable to the interaction in which it was issued. Introduced by Chaum [Cha82] in the context of electronic payments, blind signatures have become a fundamental tool for separating the issuance of an authorization from its subsequent use. Applications include electronic cash [Cha82, CFN90, OO92], e-voting [FOO93], and anonymous authentication. For example, they are widely adopted in industry [HIP⁺21, DGS⁺18, CDVW24, App, Goo] to implement anonymous authentication tokens, and underlie several proposals for fraud-detection in privacy-preserving ad-click measurement [PCM, HIJ⁺21, Tru].

PAIRING-FREE BLIND SIGNATURES. It is natural to ask for blind signatures built from the same algebraic ingredients as ordinary signatures. In particular, practical signatures such as Schnorr [Sch90], EdDSA [BDL⁺12], and ECDSA [Ame05] suggest that a prime-order pairing-free group (usually implemented by an elliptic curve) and a hash function should suffice, and it is natural to implement blind signatures from the same structure. In contrast to pairing-friendly curves, pairing-free curves are widely available in libraries and standards. This is also an appealing theoretical target, as building solutions that solely rely on the simple one-way homomorphic map helps toward post-quantum constructions. This was made evident by prior works adapting pairing-free blind signature schemes into trapdoor-free lattice-based constructions [HKLN20] and constructions from group actions [KLLQ23, HLM⁺25, HLPP26].

The main technical challenge here is not to blind the signing process, but rather to attain *one-more unforgeability* (OMUF) [JLO97], which requires a malicious user not to be able to sign more messages than the number of signing sessions it engages in, *even* when these sessions are concurrent and can be interleaved arbitrarily. Indeed, the classical three-move blind Schnorr protocol [CP93] is vulnerable to concurrent attacks based on the ROS problem [BLL⁺21], leaving it secure only for a limited number of concurrent sessions [FPS20]. The same difficulty affects other protocols based on linear identification schemes, such as Okamoto–Schnorr [Oka93].

A sequence of works [FPS20, KLX22, TZ22, CKM⁺23] developed compact three-move constructions which are proved to achieve OMUF in the random-oracle model (ROM) [BR93] and the *algebraic group model* (AGM) [FKL18]. The use of the AGM imposes a strong heuristic on these results, as it restricts the validity of the proof to algebraic adversaries, i.e., adversaries that are required to explain every group element they produce as a linear combination of input group elements. Therefore, understanding whether the AGM is necessary to solve a particular task is important. It is known [DHH⁺21] that the ROM is necessary, even only to achieve (non-blind) black-box pairing-free digital signatures, but no inherent barrier seems to force adoption of the AGM.

BEYOND THE AGM. A series of recent works confirmed this by providing several AGM-free constructions in the ROM. Chairattana-Apirom, Tessaro, and Zhu [CATZ24] gave several such constructions, including a four-move construction proved to achieve OMUF under the computational Diffie-Hellman (CDH) assumption. Subsequent works improved efficiency by relying on the decisional Diffie-Hellman (DDH) assumption [KRW24, KR25], obtained tight security [BHKR25], and developed threshold blind signatures relying on CDH [RR26]. Kloof, Lai, and Reichle [KLR26] then presented a four-move pairing-free blind signature with full provable one-more unforgeability in the ROM under the minimal assumption of the hardness of the discrete logarithm (DL) problem.

Unlike the state-of-the-art in the AGM, these constructions all require (at least) four moves. The gap was recently closed by two concurrent works [CART26, Che26] giving three-move blind signatures proved to achieve OMUF under DDH in the ROM. Dietz, Kastner, and Tessaro [DKT26] also recently proved that three moves are necessary for a large class of blind signatures in the ROM making black-box use of the group, and while a few works are not covered by this result, it still suggests an inherent barrier making it unlikely to achieve round optimal (i.e., two-move) constructions. Table 1 summarizes the relevant constructions.

Therefore, the recent results leave two different ways to avoid idealized group assumptions: use four moves and assume DL, or use three moves and assume DDH. We therefore ask whether the gap can be closed:

Scheme	Moves	$ \sigma $	$ \text{transcript} $	Assumption	OMSUF
Clause Schnorr [FPS20]	3	$\mathbb{G} + \mathbb{Z}_p$	$2\mathbb{G} + 3\mathbb{Z}_p$	OMDL, mROS	✓
Abe [Abe01, KLX22]	3	$2\mathbb{G} + 6\mathbb{Z}_p$	$\lambda + 3\mathbb{G} + 6\mathbb{Z}_p$	DL	✓
TZ [TZ22]	3	$4\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL	✓
Snowblind [CKM ⁺ 23]	3	$\mathbb{G} + 2\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL	✓
CTZ-1 [†] [CATZ24]	4	$\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH	
CTZ-2 [†] [CATZ24]	5	$\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH	✓ [†]
CTZ-3 [CATZ24]	4	L_{lin}	L_{lin}	CDH	
KRW [KRW24]	4	$2\mathbb{G} + 5\mathbb{Z}_p$	L_{lin}	DDH	
BHKR [BHKR25]	4	$10\mathbb{G} + 29\mathbb{Z}_p$	$37\mathbb{G} + 40\mathbb{Z}_p$	DDH	
KR-1 [KR25]	4	$\mathbb{G} + 5\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH	
KR-2 [KR25]	5	$\mathbb{G} + 6\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH	✓
RR [RR26]	4	$2\mathbb{G} + 8\mathbb{Z}_p$	L_{lin}	CDH	
KLR [KLR26]	4	$6\mathbb{Z}_p$	L_{lin}	DL	
CRT-c [CART26]	3	$15\mathbb{G} + 22\mathbb{Z}_p$	$17\mathbb{G} + 14\mathbb{Z}_p$	DDH	✓
CRT-s [CART26]	3	$21\mathbb{G} + 35\mathbb{Z}_p$	$17\mathbb{G} + 14\mathbb{Z}_p$	DDH	✓
Chen [Che26]	3	$21\mathbb{G} + 32\mathbb{Z}_p$	$27\mathbb{G} + 29\mathbb{Z}_p$	DDH	
This work	3	$L_{\text{alg}} + L_{\text{sle}}$	L_{alg}	DL	✓

Table 1. Pairing-free blind signatures with black-box use of the group. The upper block uses the ROM and AGM; the lower block uses only the ROM. Costs are in bits. Denote $\mathbb{G}$ as the length of a group-element encoding; $\mathbb{Z}_p$ as the length of one scalar encoding; λ as the security parameter; $L_{\text{lin}} = \Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$, $n = \lceil \log_2 p \rceil$, $L_{\text{alg}} = \Theta(n)(\mathbb{G} + \mathbb{Z}_p)$, and L_{sle} as the length of the straightline-extractable proof. Asymptotic entries may hide different constants. CRT-c and CRT-s denote the computationally and statistically blind variants of [CART26], respectively. †: weaker OMSUF definition.

Can a three-move pairing-free blind signature provably achieve full concurrent security in the ROM under the DL assumption alone?

The question of whether we can solely rely on DL is fundamental, as the assumption of hardness in recovering x from $X = xG$ is minimal in group-based cryptography, and thus a three-move construction under DL would be best possible.

OUR CONTRIBUTION. We answer this question affirmatively. We give the first three-move pairing-free blind signature for which OMUF is proved based solely on DL in the ROM, without assuming the AGM or the GGM. The construction uses the underlying group in a black-box way. In particular, it does not invoke a generic zero-knowledge proof for a circuit implementing elliptic-curve arithmetic.

We in fact prove the stronger notion of *one-more strong unforgeability (OMSUF)*: even producing an additional signature on an already signed message is forbidden, provided that it yields a new message–signature pair. Our construction also supports *partial blindness* [AF96]. The signer and user can agree on public information, such as an expiration date or a token class, while the remaining message stays hidden. With our instantiation, (partial) blindness is statistical in the ROM and holds also against a signer that chooses its public key maliciously.

Along the way, we also introduce novel blinding techniques applied to proofs for quadratic relations used in our construction. We give a detailed technical overview of our construction in the next section.

ROUND COMPLEXITY AND OTHER APPROACHES. Three-move blind signatures are a natural target in light of the lower bound of Dietz, Kastner, and Tessaro [DKT26], which rules out two-move constructions that can be framed in a combination of Maurer’s generic-group model [Mau05] and the ROM, under restrictions on oracle-query complexity and message length. In particular, the user and verifier make $O(\log \lambda)$ random-oracle queries, and messages have length $\omega(\log \lambda)$. We note that our scheme does not fit within this class, due to

the higher verification complexity, but the result is nonetheless an indication that three moves constitute a substantial barrier.

If non-black-box techniques are allowed, two moves are possible. Fischlin’s generic construction [Fis06] provides an early approach, and Ciampi, Della Monica, and Visconti [CDV26] obtain two-move GUC-secure blind signatures from one-way functions in a global non-programmable random-oracle model. Kastner, Tessaro, and Zaverucha [KTZ26] give a more concrete pairing-free approach using zero-knowledge proofs about a group-to-scalar conversion. Another line of work [FW24, DV26] asks for blind issuance of *ordinary Schnorr signatures*, rather than a new signature format. These constructions, however, rely on heavy zero-knowledge machinery. Our goal is a direct algebraic construction that achieves three moves and full concurrent security from DL alone, with only black-box use of the group.

PAPER OUTLINE. Section 2 gives the technical overview of our construction. Section 3 gives necessary preliminaries. Section 4 presents the blindable Σ -protocols used in our blind signature construction, including a detailed overview of our novel blinding techniques. Our main result is in Section 5. Formal security proofs are given in Section 6 (Correctness), Section 7 (OMSUF) and Section 8 (Blindness). We also detail our use of generative AI tools in Appendix A.

2 Technical Overview

Our work builds on recent works [CART26, Che26] that establish three-move blind signatures from DDH. We give a brief and simplified recap below, focusing on the approach by [CART26]. Let $\mathbb{G}$ be a group of prime order p with generators G and $H \leftarrow_s \mathbb{G}$. (We use additive notation.) Their blind signature builds on the Okamoto-Schnorr blind signature BS_{os} [PS00], where the signer with public key $W = \Phi_{\text{os}}(\mathbf{w}) := w_0G + w_1H$ runs the Okamoto-Schnorr identification scheme [Sch90, Oka93] with the user. The transcript $\tau_{\text{os}} = (A_{\text{os}} = \Phi_{\text{os}}(\mathbf{r}), c_{\text{os}}, z_{\text{os}} = c_{\text{os}}\mathbf{w} + \mathbf{r})$ is blinded into $\tilde{\tau}_{\text{os}} = (\tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}})$ and compiled into a signature via Fiat-Shamir [FS87] (i.e., $\tilde{c}_{\text{os}} = \text{H}(W, A_{\text{os}}, \text{msg})$ for random oracle H) on the user-side. Note that the signer-side of the protocol is a simple canonical Σ -protocol for linear relations, where blinding can be easily added [KRW24]. The scheme BS_{os} is OMUF-secure for limited concurrent signing sessions, and the prior work augments the protocol to achieve full OMUF security.

In particular, underlying their approach is the following property of BS_{os} : for an adversary with access to an unbounded number of signing sessions for a public key W , it cannot output two (unblinded) valid transcripts $\tau_{\text{os}} = (A_{\text{os}}, c_{\text{os}}, z_{\text{os}}), \tau'_{\text{os}} = (A'_{\text{os}}, c'_{\text{os}}, z'_{\text{os}})$ that share the same $A_{\text{os}} = A'_{\text{os}}$ and distinct $c_{\text{os}} \neq c'_{\text{os}}$. This property is satisfied assuming DL via (i) special-soundness of the underlying Σ -protocol allowing extraction of $\mathbf{w}'$ such that $W = \Phi_{\text{os}}(\mathbf{w}')$, and (ii) witness-indistinguishability argument showing that $\mathbf{w}'$ differs from the actual secret key $\mathbf{w}$ with high probability. This then allows extraction of the discrete logarithm of H base G . We now briefly recap how the prior work augments their protocol to utilize this property.

- **On the signer side:** (a) The signer tags the nonce A_{os} via an algebraic MAC and sends the tag T to the user. (b) It proves with an appropriate Σ -protocol that the tag T verifies on nonce A_{os} , denoting the transcript of this proof as τ_{mac} .
- **On the user side:** (a) The user commits to the blinding factors $\tilde{\mathbf{s}}_{\text{os}}$ in a commitment C_{rnd} , such that $\tilde{A}_{\text{os}} = \text{BlindNonce}(A_{\text{os}}; \tilde{\mathbf{s}}_{\text{os}})$, where BlindNonce denotes the function to derive the blinded nonce. (b) It also blinds and modifies the proof τ_{mac} into $\tilde{\tau}_{\text{mac}}$ which testifies that T (also committed to in C_{rnd}) is valid for the nonce A_{os} derived from $\tilde{A}_{\text{os}}$ by undoing blinding with $\tilde{\mathbf{s}}_{\text{os}}$, and this proof is compiled via Fiat-Shamir.

The final signature consists of the blinded Fiat-Shamir proofs $\tilde{\tau}_{\text{os}}$ and $\tilde{\tau}_{\text{mac}}$, and the commitment C_{rnd} . By unforgeability of the MAC and soundness properties of $\tilde{\tau}_{\text{mac}}$, the user is forced to reuse an unblinded nonce A_{os} in the signature. In this case, the property mentioned above ensures one-more unforgeability under the DL assumption if the commitment C_{rnd} is extractable.

RELIANCE ON DDH. The aforementioned protocol crucially relies on the DDH assumption, as discussed in [CART26]. The first reason is that the commitment C_{rnd} requires several properties (such as linear homomorphism and extractability) simultaneously.³ There is no suitable candidate from DL to the best of our knowledge. Secondly, the algebraic MAC relies on DDH and similarly, no construction from DL alone is known. (Existing works [DKPW12, CMZ14] show security under CDH or DDH, or relying on the GGM.) Alternatively, we may rely on the techniques from [Che26] instead. However, this does not change the picture as [Che26] relies on the blind signature from [BHKR25] which also relies on extractable, linearly-homomorphic commitments. While a blind signature from DL is known [KLR26], their construction design relies on hashing the signed message first which is incompatible with the aforementioned framework.

OUR APPROACH. We replace both ingredients that rely on DDH, namely the commitment scheme and the MAC, with DL-based alternatives. First, we employ straightline-extractable non-interactive zero-knowledge proofs (SLE NIZKs) that allow to extract the blinding factors $\tilde{\mathbf{st}}_{\text{os}}$ without rewinding, as required for the WI-based argument of BS_{os} . As before, if the nonce A_{os} is reused, we may conclude one-more unforgeability under the DL assumption.

Secondly, we craft a suitable replacement for the MAC to enforce nonce reuse. Our construction relies on programmable hash functions (PHFs) [Wat05, HK08]. These are keyed hash functions (with the key $\mathbf{K}$) that map messages $\mathbf{x} \in \{0, 1\}^n$ to a group element $\text{H}_{\mathbf{K}}^{MG}(\mathbf{x})$. Also, given a trapdoor td , we may compute a representation $\text{H}_{\mathbf{K}}^{MG}(\mathbf{x}) = a_{\mathbf{x}}G + b_{\mathbf{x}}V$ for fixed $G, V \in \mathbb{G}$. We say that it is (m, n) -programmable if for an adversarial choice of $(\mathbf{x}_1, \dots, \mathbf{x}_m)$ and $(\mathbf{y}_1, \dots, \mathbf{y}_n)$, we have that

$$\forall j \in [m] : a_{\mathbf{x}_j} = 0 \quad \wedge \quad \forall i \in [n] : a_{\mathbf{y}_i} \neq 0$$

with sufficient (i.e., at least inverse-polynomial) probability. The PHF from [Wat05, HK08] is known to be $(1, \text{poly})$ -programmable. We leverage this property as follows. The signer proves with an OR-compiled Σ -protocol Σ_{phf} that either (i) it knows the discrete logarithm of $X = xG$ or (ii)

$$\begin{aligned} A_{\text{os}} &= \sum_{i \in [n]} \mu_i (2^{i-1})G, \quad T = \rho V + \text{H}_{\mathbf{K}}^{MG}(\boldsymbol{\mu}), \quad \boldsymbol{\mu} \in \{0, 1\}^n, \\ \text{H}_{\mathbf{K}}^{MG}(\boldsymbol{\mu}) &= a_{\boldsymbol{\mu}}G + b_{\boldsymbol{\mu}}V \text{ with } a_{\boldsymbol{\mu}} \neq 0. \end{aligned} \tag{1}$$

Here, $\boldsymbol{\mu}$ is the bit-decomposition of the discrete logarithm r such that $A_{\text{os}} = rG$ with $n = \lceil \log p \rceil$ bits. Let us ignore blindness for now. Then, the Fiat-Shamir-compiled transcript $\tau_{\text{phf}} = (A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ and the randomized hash value T are included in the signature. Note that the signer employs the X branch in the real protocol whereas the PHF branch is employed in the security analysis to enforce integrity properties on A_{os} . As T is randomized with the additional factor ρ , the signer may simply issue a random group element $T \leftarrow \mathbb{G}$ in the real execution and simulate the PHF branch.

SHOWING ONE-MORE UNFORGEABILITY. To prove OMUF, we must simulate the signing oracle and argue that after only Q_S signing sessions, the adversary cannot provide $Q_S + 1$ signatures on distinct messages.

⁴ First, we extract the unblinded nonces $A_{\text{os}}^{(i)}$ from the SLE proofs in $\mathcal{A}$'s signatures. If some nonce $A_{\text{os}}^{(i)}$ is reused, this case remains similar to the prior work [CART26], as we employ a similar WI-based argument to argue that this cannot happen under the DL assumption. Else, there is a fresh nonce $A_{\text{os}}^* = r^*G$ which has never appeared in the signing sessions. This case is where the proof deviates significantly from the prior work.

In this second case, our approach is to simulate the signing protocol using the PHF branch of the OR-compiled Σ -protocol and rewind to extract the DL x of X on the other branch. To be more precise, the idea is to ensure that after setting up the PHF keys $\mathbf{K}$ with a trapdoor td :

³ While it is plausible that the argument in [CART26] works if C_{rnd} is statistically binding (instead of extractable), we do not know of an appropriate candidate scheme under DL, even under this relaxation.

⁴ We manage to prove one-more strong unforgeability for our blind signature, but postpone the discussion for exposition.

- (i) For each signing session with nonce $A_{\text{os}} = rG$ with bit-decomposition μ of its exponent r , the component $a_\mu \neq 0$ is non-zero over $\mathbb{Z}_p$ (computed using td and μ). This allows us to simulate signing using a witness for the PHF branch of Σ_{phf} and not the DL x of X .
- (ii) For the fresh nonce $A_{\text{os}}^* = r^*G$ and the bit-decomposition μ^* , the component $a_{\mu^*} = 0$ is zero (computed using td and μ^*). Rewinding will allow us to extract from a Σ_{phf} -transcript either the DL x of X or a witness satisfying Equation (1) containing $a_{\mu^*} \neq 0$. The latter case is a contradiction, so we may only extract x of X .

To this end, we set up $H = hG$ with known DL h so that (i) we may compute the DL r of $A_{\text{os}} = rG$ during the signing sessions and simulate, and (ii) using the extracted blinding factors $\tilde{\text{st}}_{\text{os}}$ and h , we may compute the discrete logarithm r^* of A_{os}^* and accordingly μ^* . Then, our hope is to employ (1, poly)-programmability of the PHF to generate an appropriate trapdoor td that ensures these two properties with sufficient probability.

However, there are two subtle issues with the above argument. First and more importantly, notice that a discrete logarithm $r^* \in \mathbb{Z}_p$ may have two possible decompositions of n bits, μ_0^* and μ_1^* such that $r^* = \sum_{i \in [n]} \mu_{0,i}^* 2^{i-1} = \sum_{i \in [n]} \mu_{1,i}^* 2^{i-1} \pmod{p}$.⁵ The adversary may use one of these decompositions to derive a valid Σ_{phf} transcript, so our rewinding extraction could extract an actual valid witness $a_{\mu^*} \neq 0$. Hence, (1, poly)-programmability of the PHF does not suffice for the proof strategy above as it guarantees $a_{\mu^*} = 0$ for only one choice of μ^* . To deal with this issue, we prove that the PHF from [Wat05, HK08] is (2, poly)-programmable. This ensures that for any two possible choices of bit-decompositions $\mu_0^* \neq \mu_1^*$ of r^* , we have $a_{\mu^*} = 0$ with sufficient probability.

Secondly, the sketched contradiction is only computational as the DL of V allows to compute a valid witness for any pair (A_{os}, T) . For the second point, we show that it is possible to compute the DL of V via rewinding if such a contradiction occurs. We refer to Section 5.3 for the more detailed proof sketch and Section 7 for the formal OMUF proof.

We note that the signature scheme in [DH26, Section 4.1] also is based on proving non-zero evaluation of the PHF with a Σ -protocol. However, our concrete instantiation of the associated Σ -protocol and the security analysis differ considerably. First, their scheme relies on dual-mode commitments, which significantly simplify the security analysis. More importantly, dual-mode commitments rely on DDH and there is no known construction from DL. Second, their construction signs messages $\mu \in \{0, 1\}^n$, whereas we can view our scheme as signing group elements $A \in \mathbb{G}$, which is vital for our application.⁶ In particular, our reliance on (2, poly)-programmability is a consequence of signing group elements, while their analysis follows from (1, poly)-programmability. Thirdly, we must ensure that the issuance protocol is blindable. This comes with additional technical challenges, as we discuss next.

ON BLINDNESS. To ensure blindness, the user commits to A_{os} , then blinds and adapts the Σ_{phf} transcript τ_{phf} into a Fiat-Shamir compiled proof $\tilde{\tau}_{\text{phf}}$ which shows that either the DL of X is known or Equation (1) holds, except that the nonce A_{os} is committed. The non-zero argument is performed following the technique in [KLR26] (as the technique is blinding-friendly).⁷ To prove that μ is a bit-vector, we show $\mu(1 - \mu) = \mathbf{0}$ following the technique in [YAZ⁺19, BLS19, CKLR21, CGKR22]. That is, we prove the quadratic constraint by turning it into a degree-two polynomial identity that is checked at a random input. We note that the resulting Σ -protocol is *not* canonical, as it proves a quadratic relation. Our novel blinding mechanism is technically involved due to the cross-terms that arise in the polynomial identity when blinding; we refer to Section 4 for details. In short, we make blinding work by switching to multiplicative blinding of the challenge (as additive blinding appears to fail) and extending the statement with additional commitments to μ , which facilitate the cancellation of the cross-terms. We believe that this may be of independent interest: the technique plausibly generalizes to blinding arbitrary quadratic statements. Finally, the SLE NIZK proves

⁵ For instance, $0 \in \mathbb{Z}_p$ has two decompositions, $(0, \dots, 0)$ and $(b_1, \dots, b_n)$, where the latter is the bit-decomposition of the integer p .

⁶ We note that our unforgeability argument relies on knowing the DL r of the signed message $A = rG$ as discussed above.

⁷ The linear map for the non-zero argument depends on the term T which must also be blinded, which must be dealt with. In contrast, the corresponding linear map in [KLR26] depends only on fixed generators. We refer to the main body for details.

that the committed nonce A_{os} indeed coincides with the nonce A_{os} derived by unblinding $\tilde{A}_{\text{os}}$ to link both components. We refer to Section 5.3 for a more detailed proof sketch of blindness.

EXTENSIONS. Our construction satisfies partial blindness and one-more strong unforgeability. In particular, proving one-more strong unforgeability presents non-trivial challenges. We refer to Section 5 for our full construction and more details on security proofs.

3 Preliminaries

NOTATION. Let $[n, m] = \{n, n+1, \dots, m\}$ for any $n, m \in \mathbb{Z}$ where $n \leq m$ and $[n] = [1, n]$ for any $n \geq 1$. We use λ as the security parameter. We employ representatives $[0, p-1]$ for $\mathbb{Z}_p$. For some random variable V with (finite) image $\mathcal{Y}$, we denote the *min-entropy* by $H_{\min}(V) := -\log \max_{y \in \mathcal{Y}} \Pr[V = y]$.

PAIRING-FREE GROUPS AND DL ASSUMPTION. Let GGen be an algorithm that on input 1^λ outputs a group description $\mathbb{G}$ of $\Theta(\lambda)$ -bit prime-order $p = p(\lambda)$ with generator G . We employ additive notation. Throughout, we assume that algorithms implicitly obtain as input $(\mathbb{G}, p, G) \leftarrow \text{GGen}(1^\lambda)$. Denote by $\text{dlog}_G X$ the discrete logarithm $x \in \mathbb{Z}_p$ of X with basis G , where $xG = X$, and write DL as a shorthand for discrete logarithm. We extend the notation to vectors of group elements $\mathbf{X} = (X_1, \dots, X_n)$ in the natural manner. For vectors $\mathbf{x} \in \mathbb{Z}_p^n$, $\mathbf{H} \in \mathbb{G}^n$ and $y \in \mathbb{Z}_p$, we write $\mathbf{x}G = (x_1G, \dots, x_nG)$, and $y\mathbf{H} = (yH_1, \dots, yH_n)$, $\mathbf{x}\mathbf{H} = (x_1H_1, \dots, x_nH_n)$. Similarly, we also write $\mathbf{x}^2 = (x_1^2, \dots, x_n^2)$.

We recall the DL and Rel-DL assumptions; the former tightly implies the latter. Let $(\mathbb{G}, p, G)$ be as above. We note that the assumptions depend on GGen employed to (implicitly) sample $(\mathbb{G}, p, G)$.

Definition 3.1 (DL Assumption). *The discrete logarithm (DL) assumption holds if for any PPT adversary $\mathcal{A}$, it holds that for $(\mathbb{G}, p, G) \leftarrow \text{GGen}(1^\lambda)$,*

$$\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{A}, \lambda) := \Pr[x = x' \mid x \leftarrow \mathbb{Z}_p, x' \leftarrow \mathcal{A}(G, xG)] = \text{negl}(\lambda) .$$

Definition 3.2 (Rel-DL Assumption). *The hard DL relation (Rel-DL) assumption holds if for any PPT adversary $\mathcal{A}$, it holds that*

$$\text{Adv}_{\text{GGen}, n}^{\text{rel-dl}}(\mathcal{A}, \lambda) := \Pr \left[0 = \beta_0 G + \sum_{i=1}^n \beta_i X_i \wedge \beta \neq \mathbf{0} \mid \beta \leftarrow \mathcal{A}(G, \mathbf{X}) \right] = \text{negl}(\lambda) ,$$

where $(\mathbb{G}, p, G) \leftarrow \text{GGen}(1^\lambda)$, $\mathbf{X} \leftarrow \mathbb{G}^n$, and $\beta = (\beta_i)_{i \in [0, n]}$.

Lemma 3.3 (DL implies Rel-DL.). *Let GGen be a group parameter generator outputting groups of prime-order $p = p(\lambda)$ and $n = n(\lambda)$ be a positive integer. For any adversary $\mathcal{A}$ against the rel-DL assumption running in time $t_{\mathcal{A}} = t_{\mathcal{A}}(\lambda)$, there exists an adversary $\mathcal{B}$ against DL assumption running in time $t_{\mathcal{B}} \approx t_{\mathcal{A}}$ such that $\text{Adv}_{\text{GGen}, n}^{\text{rel-dl}}(\mathcal{A}, \lambda) \leq \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}, \lambda) + 1/p$.*

Proof. The proof follows the proof of [JT20, Lemma 3] almost verbatim except that we allow for a fixed generator G to be within the linear relation checked at the end.

Consider the following adversary $\mathcal{B}$ against the DL assumption with input (G, X) . Then, it samples $\mathbf{a}, \mathbf{b} \leftarrow \mathbb{Z}_p^n$, sets $\mathbf{Y} = \mathbf{a}G + \mathbf{b}X$, and runs $\beta \leftarrow \mathcal{A}(G, \mathbf{Y})$. Then, $\mathcal{B}$ parses $\beta = (\beta_i)_{i \in [0, n]}$ checks that $\sum_{i=1}^n \beta_i b_i \neq 0$. If this doesn't hold, abort. Otherwise, return $-\frac{\beta_0 + \sum_{i=1}^n \beta_i a_i}{\sum_{i=1}^n \beta_i b_i}$.

If $\mathcal{A}$ succeeds, we have that $0_{\mathbb{G}} = \beta_0 G + \sum_{i=1}^n \beta_i Y_i = (\beta_0 + \sum_{i=1}^n \beta_i a_i)G + (\sum_{i=1}^n \beta_i b_i)X$. Then, if $\sum_{i=1}^n \beta_i b_i \neq 0$, the output of $\mathcal{B}$ is exactly the discrete logarithm $\text{dlog}_G X$.

Hence, we have that

$$\begin{aligned} \text{Adv}_{\text{GGen}, n}^{\text{rel-dl}}(\mathcal{A}, \lambda) &\leq \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}, \lambda) + \Pr_{\mathbf{a}, \mathbf{b}, \mathcal{A}} \left[\sum_{i=1}^n \beta_i b_i = 0 \right] \\ &\leq \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}, \lambda) + \frac{1}{p} . \end{aligned}$$

The last inequality follows from Schwarz-Zippel lemma and the fact that $\mathcal{A}$'s view is independent of $\mathbf{b}$ as $\mathbf{Y}$ hides $\mathbf{b}$. This is the same argument as in [JT20, Lemma 3]. $\square$

BLIND SIGNATURES. Now, we define blind signatures [Cha82] with security definitions from [JLO97]. We consider 3-move schemes with partial blindness [AF96]. A partially blind signature BS is a tuple of PPT algorithms (BS.Setup, BS.KeyGen, (BS.Sign₁, BS.U₁, BS.Sign₂, BS.U₂), BS.Ver) such that

BS.Setup(1^λ) $\rightarrow$ pp: Given security parameter 1^λ , outputs public parameters pp.

BS.KeyGen(pp) $\rightarrow$ (sk, pk): Given public parameters pp, outputs a secret key sk and public key pk.

BS.Sign₁(pp, sk, info) $\rightarrow$ (smsg₁, st_S): Given public parameters pp, a secret key sk and a common message info, outputs a signer message smsg₁ and signer state st_S.

BS.U₁(pp, pk, info, msg, smsg₁) $\rightarrow$ (umsg, st_U): Given public parameters pp, a public key pk, a common message info, a message msg and a signer message smsg₁, outputs a user message umsg and user state st_U.

BS.Sign₂(umsg, st_S) $\rightarrow$ smsg₂: Given public parameters pp, a secret key sk, a user message umsg and signer state st_S, outputs a signer message smsg₂.

BS.U₂(smsg₂, st_U) $\rightarrow$ σ : Given signer message smsg₂ and user state st_U, outputs a signature σ .

BS.Ver(pp, pk, info, msg, σ) $\rightarrow$ b: Given public parameters pp, a public key pk, a common message info, a message msg and a signature σ , deterministically outputs $b \in \{0, 1\}$.

If clear by context, we leave the user and signer states implicit for conciseness. We define correctness, partial blindness and one-more strong unforgeability below.

Definition 3.4 (Correctness). A partially blind signature BS is correct with error $\varepsilon = \varepsilon(\lambda)$ if for all public parameters $\text{pp} \in \text{BS.Setup}(1^\lambda)$, $(\text{sk}, \text{pk}) \in \text{BS.KeyGen}(\text{pp})$, messages msg and common messages info, the following probability is at least $1 - \varepsilon$

$$\Pr \left[\text{BS.Ver}(\text{pp}, \text{pk}, \text{info}, \text{msg}, \sigma) = 1 \mid \begin{array}{l} (\text{smsg}_1, \text{st}_S) \leftarrow \text{BS.Sign}_1(\text{pp}, \text{sk}, \text{info}) \\ (\text{umsg}, \text{st}_U) \leftarrow \text{BS.U}_1(\text{pp}, \text{pk}, \text{info}, \text{msg}, \text{smsg}_1) \\ \text{smsg}_2 \leftarrow \text{BS.Sign}_2(\text{umsg}, \text{st}_S) \\ \sigma \leftarrow \text{BS.U}_2(\text{smsg}_2, \text{st}_U) \end{array} \right].$$

One-more strong unforgeability ensures that it is hard to compute a fresh signature-message pair without interacting with the signer.

Definition 3.5 (One-More (Strong) Unforgeability). A partially blind signature scheme BS is one-more (strong) unforgeable if for any PPT $\mathcal{A}$ in the game OMUF (or OMSUF) as defined in Figure 1, the following advantage is negligible:

$$\text{Adv}_{\text{BS}}^{\text{omuf/omsuf}}(\mathcal{A}, \lambda) := \Pr[(\text{OMUF/OMSUF})_{\text{BS}}^{\mathcal{A}}(\lambda) = 1].$$

The following lemma also allows us to assume that any OM(S)UF adversary finishes all signing sessions. Note that similar assumption has been (implicitly) made in prior works on three-move blind signatures [FPS20, TZ22]. We give a proof of the lemma in ???. Moreover, the lemma extends naturally for adversaries in the ROM (with increased random oracle query count).

Lemma 3.6. Let $\mathcal{A}$ be an adversary against OMUF (resp. OMSUF) game of BS, running in time $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ and making $Q_{S_1} = Q_{S_1}(\lambda)$, $Q_{S_2} = Q_{S_2}(\lambda)$ to S_1, S_2 , respectively. Then, there exists an adversary $\mathcal{B}$ against OMUF (resp. OMSUF) game of BS, running in time $T_{\mathcal{B}} \approx T_{\mathcal{A}}$ and making $Q_S = Q_{S_1}$ queries to both S_1, S_2 , such that

$$\text{Adv}_{\text{BS}}^{\text{omuf}}(\mathcal{A}, \lambda) \leq \text{Adv}_{\text{BS}}^{\text{omuf}}(\mathcal{B}, \lambda), \quad \text{Adv}_{\text{BS}}^{\text{omsuf}}(\mathcal{A}, \lambda) \leq \text{Adv}_{\text{BS}}^{\text{omsuf}}(\mathcal{B}, \lambda).$$

Partial blindness ensures that the signer cannot link signing interactions to the issued signatures, even if the signing key is malicious.

Definition 3.7 (Partial Blindness). We say that BS is partially blind if for any adversary $\mathcal{A}$ in the game Blind as defined in Figure 2, the following advantage is negligible:

$$\text{Adv}_{\text{BS}}^{\text{blind}}(\mathcal{A}, \lambda) := \left| \Pr[\text{Blind}_{\text{BS}}^{\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|.$$

Game $\boxed{\text{OMUF}_{\text{BS}}^A(\lambda)}$: $\boxed{\text{OMSUF}_{\text{BS}}^A(\lambda)}$:	Oracle $S_1(\text{info})$:
1 : $\text{pp} \leftarrow \text{BS.Setup}(1^\lambda)$	1 : $\text{cnt} \leftarrow \text{cnt} + 1 ; \text{info}_{\text{cnt}} \leftarrow \text{info}$
2 : $(\text{sk}, \text{pk}) \leftarrow \text{BS.KeyGen}(\text{pp})$	2 : $(\text{smsg}_{1,\text{cnt}}, \text{st}_{\text{cnt}}) \leftarrow \text{BS.Sign}_1(\text{pp}, \text{sk}, \text{info})$
3 : $\text{cnt} \leftarrow 0 ; \mathcal{S} \leftarrow \emptyset$	3 : return $\text{smsg}_{1,\text{cnt}}$
4 : $(\text{info}^*, (\text{msg}^{(i)}, \sigma^{(i)})_{i \in [\ell]}) \leftarrow \mathcal{A}^{S_1, S_2}(\text{pp}, \text{pk})$	Oracle $S_2(\text{sid}, \text{umsg})$:
5 : return $(\ell > Q_{\text{S}, \text{info}^*}) \wedge$	1 : if $\text{sid} \notin [\text{cnt}] \vee \text{sid} \in \mathcal{S}$ then return $\perp$
6 : $(\forall i \in [\ell] : \text{BS.Ver}(\text{pp}, \text{pk}, \text{info}^*, \text{msg}^{(i)}, \sigma^{(i)}) = 1)$	2 : $\text{smsg}_{2,\text{sid}} \leftarrow \text{BS.Sign}_2(\text{umsg}, \text{st}_{\text{sid}})$
7 : $\boxed{\wedge \forall i \neq j \in [\ell] : \text{msg}^{(i)} \neq \text{msg}^{(j)}}$	3 : $Q_{\text{S}, \text{info}_{\text{sid}}} \leftarrow Q_{\text{S}, \text{info}_{\text{sid}}} + 1$
8 : $\boxed{\wedge \forall i \neq j \in [\ell] : (\text{msg}^{(i)}, \sigma^{(i)}) \neq (\text{msg}^{(j)}, \sigma^{(j)})}$	4 : $\text{// Assume } Q_{\text{S}, \text{info}_{\text{sid}}} \text{ initialized as } 0$
	5 : $\mathcal{S} \leftarrow \mathcal{S} \cup \{\text{sid}\}$
	6 : return $\text{smsg}_{2,\text{sid}}$

Fig. 1. One-More (Strong) Unforgeability security game of BS. The dashed box indicates the winning condition of the OMUF game, while the solid box indicates the winning condition of the OMSUF game.

Game $\text{Blind}_{\text{BS}}^A(\lambda)$:	Oracle $U_1(i, \text{smsg}_{i,1})$:
1 : $\text{pp} \leftarrow \text{BS.Setup}(1^\lambda)$	1 : if $i \notin \{0, 1\} \vee \text{sess}_i \neq \text{init}$ then return $\perp$
2 : $b \leftarrow \{0, 1\} ; b_0 \leftarrow b ; b_1 \leftarrow 1 - b$	2 : $\text{sess}_i \leftarrow \text{open}$
3 : $b' \leftarrow \mathcal{A}^{\text{init}, U_1, U_2}(\text{pp})$	3 : $(\text{umsg}_i, \text{st}_{U,i}) \leftarrow \text{BS.U}_1(\text{pp}, \text{pk}, \text{info}, \text{msg}_{b_i}, \text{smsg}_{i,1})$
4 : return $b' = b$	4 : return umsg_i
Init($\text{pk}', \text{info}', \text{msg}'_0, \text{msg}'_1$)	Oracle $U_2(i, \text{smsg}_{i,2})$:
1 : if $\text{sess}_0 = \perp \wedge \text{sess}_1 = \perp$ then	1 : if $i \notin \{0, 1\} \vee \text{sess}_i \neq \text{open}$ then return $\perp$
2 : $\text{sess}_0 \leftarrow \text{init} ; \text{sess}_1 \leftarrow \text{init}$	2 : $\text{sess}_i \leftarrow \text{completed}$
3 : $\text{pk} \leftarrow \text{pk}' ; \text{info} \leftarrow \text{info}'$	3 : $\sigma_{b_i} \leftarrow \text{BS.U}_2(\text{smsg}_{i,2}, \text{st}_{U,i})$
4 : $\text{msg}_0 \leftarrow \text{msg}'_0 ; \text{msg}_1 \leftarrow \text{msg}'_1$	4 : if $\text{sess}_0 = \text{sess}_1 = \text{completed}$ then
	5 : if $\sigma_0 = \perp \vee \sigma_1 = \perp$ then return $(\perp, \perp)$
	6 : return (σ_0, σ_1)
	7 : return $(i, \text{completed})$

Fig. 2. Blindness security game of BS.

COMMITMENT. A commitment scheme $\mathcal{C}$ with message space $\mathcal{C.M}$, key space $\mathcal{C.K}$, commitment space $\mathcal{C.C}$ and randomness space $\mathcal{C.R}$ is a tuple of PPT algorithms $(\mathcal{C.Setup}, \mathcal{C.Commit})$ such that

$\mathcal{C.Setup}(1^\lambda) \rightarrow \text{ck}$: Given security parameter 1^λ , outputs commitment key ck .

$\mathcal{C.Commit}(\text{ck}, m; r) \rightarrow C$: Given the commitment key ck , message $m \in \mathcal{C.M}$ and explicit randomness $r \in \mathcal{C.R}$, it outputs a commitment c to m .

We do not define an explicit verification algorithm; instead, we verify whether a commitment C opens to message m with randomness r via $C = \mathcal{C.Commit}(\text{ck}, m; r)$. Below, we recall desirable security properties of commitment schemes.

Definition 3.8 (Hiding). A commitment scheme $\mathcal{C}$ is (perfectly) hiding if for any adversary $\mathcal{A}$, we have that advantage $\text{Adv}_{\mathcal{C}}^{\text{hide}}(\mathcal{A}, \lambda)$ is zero, defined as

$$\text{Adv}_{\mathcal{C}}^{\text{hide}}(\mathcal{A}, \lambda) = \left| \Pr \left[b = b' \mid \begin{array}{l} \text{ck} \leftarrow \mathcal{C.Setup}(1^\lambda), (\text{m}_0, \text{m}_1, \text{st}_{\mathcal{A}}) \leftarrow \mathcal{A}(\text{pp}), \\ b \leftarrow \{0, 1\}, r \leftarrow \mathcal{C.R}, b' \leftarrow \mathcal{A}(c, \text{st}_{\mathcal{A}}) \end{array} \right] - \frac{1}{2} \right|,$$

where $c \leftarrow \mathcal{C.Commit}(\text{ck}, \text{m}_b; r)$.

Definition 3.9 ((Strong) Binding). A commitment scheme $\mathcal{C}$ is (computationally) binding if for any PPT adversary $\mathcal{A}$, we have

$$\text{Adv}_{\mathcal{C}}^{\text{bind}}(\mathcal{A}, \lambda) := \Pr \left[\text{m}_0 \neq \text{m}_1 \wedge c_0 = c_1 \mid \begin{array}{l} \text{ck} \leftarrow \mathcal{C.Setup}(1^\lambda), \\ (\text{m}_0, \text{m}_1, r_0, r_1) \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] = \text{negl}(\lambda),$$

where $c_i = \text{C.Commit}(\text{ck}, m_i; r_i)$ for $i \in \{0, 1\}$. We say that C is computationally “strongly” binding if the winning condition above is $(m_0 \neq m_1 \vee r_0 \neq r_1) \wedge c_0 = c_1$, and we denote the advantage with $\text{Adv}_{\text{C}}^{\text{bind}}(\mathcal{A}, \lambda)$ instead. We say the scheme is perfectly (strongly) binding if the advantage term above is 0 for all potentially inefficient adversaries $\mathcal{A}$.

Instantiation. We employ Pedersen commitments C_{ped} [Ped92] with message space $\mathbb{Z}_p^n$, defined as follows.

$\text{C}_{\text{ped}}.\text{Setup}(1^\lambda) \rightarrow \text{ck}$: Given security parameter 1^λ , samples elements $V, G_1, \dots, G_n \leftarrow_{\$} \mathbb{G}$, and outputs $\text{ck} = (G_1, \dots, G_n, V)$.
 $\text{C}_{\text{ped}}.\text{Commit}(\text{ck}, \mathbf{m}; r) \rightarrow C$: Given the commitment key ck , message $\mathbf{m} \in \mathbb{Z}_p^n$ and randomness $r \in \mathbb{Z}_p$, it outputs a commitment $C = rV + \sum_{i \in [n]} m_i G_i$.

For convenience, we sometimes employ Pedersen commitments inline without the above abstraction. Pedersen commitments C_{ped} satisfy correctness, perfect hiding, and strong binding under the DL assumption.

PROGRAMMABLE HASH FUNCTIONS. We recall that a *group hash function* $\text{PHF} = (\text{PHF.Gen}, \text{PHF.Eval}, \text{PHF.TrapGen}, \text{PHF.TrapEval})$ for GGen with input space $\text{PHF}.\mathcal{X} = \{0, 1\}^\ell$ is a tuple of PPT algorithms defined as follows.

$\text{PHF.Gen}(1^\lambda) \rightarrow K$: Given the security parameter λ , outputs a hash key K .
 $\text{PHF.Eval}(K, X) \rightarrow Y$: Given key K and input $X \in \text{PHF}.\mathcal{X}$, deterministically outputs an evaluation Y .
 $\text{PHF.TrapGen}(1^\lambda, G, H) \rightarrow (K', \text{td})$: Given the security parameter λ and generators $G, H \in \mathbb{G}$, outputs a hash key K' and trapdoor td .
 $\text{PHF.TrapEval}(\text{td}, X) \rightarrow (a_X, b_X)$: Given trapdoor td and input X , deterministically outputs $(a_X, b_X) \in \mathbb{Z} \times \mathbb{Z}_p$.

We may refer to Y or a_X as the PHF evaluation, if clear by context.

Definition 3.10 (Programmability). A group hash function PHF for GGen is (m, n, δ) -programmable if the following properties are satisfied.

Correctness. It holds that $\text{PHF.Eval}(K', X) = a_X G + b_X H$ for all $H, G \in \mathbb{G}$, $(K', \text{td}) \leftarrow \text{PHF.TrapGen}(1^\lambda, G, H)$, $X \in \text{PHF}.\mathcal{X}$ and $(a_X, b_X) \leftarrow \text{PHF.TrapEval}(\text{td}, X)$.

Well-distributed Keys. For all generators $G, H \in \mathbb{G}$, for all $K \leftarrow \text{PHF.Gen}(1^\lambda)$ and for all $(K', \text{td}) \leftarrow \text{PHF.TrapGen}(1^\lambda, G, H)$, the hash keys K and K' are identically distributed.

Well-distributed Logarithms. For all $\mathbb{G}$, all generators G, H , for all K' in the range of $(K', \text{td}) \leftarrow \text{PHF.TrapGen}(1^\lambda, G, H)$, for all $X_1, \dots, X_m, Z_1, \dots, Z_n \in \text{PHF}.\mathcal{X}$ such that $X_i \neq Z_j$ for $i, j \in [m] \times [n]$, and for all corresponding $(a_{X_i}, b_{X_i}) \leftarrow \text{PHF.TrapEval}(\text{td}, X_i)$ and $(a_{Z_i}, b_{Z_i}) \leftarrow \text{PHF.TrapEval}(\text{td}, Z_i)$,

$$\Pr[\forall i \in [m] : a_{X_i} = 0 \wedge \forall j \in [n] : a_{Z_j} \neq 0 \bmod p] \geq \delta,$$

where the probability is over the trapdoor td , that is distributed according to the output distribution of $\text{PHF.TrapGen}(1^\lambda, G, H)$, conditioned on K' .

We write (m, n) -programmable for short, if $\delta = 1$.⁸

We are going to consider the standard multi-generator PHF implicitly introduced by Waters [Wat05], with the trapdoor generation strategy by Hofheinz and Kiltz [HK08]. The input space is $\text{PHF}.\mathcal{X} = \{0, 1\}^\ell$.

$\text{PHF.Gen}(1^\lambda) \rightarrow \mathbf{K}$: Given the security parameter 1^λ , returns a uniformly sampled hash key $\mathbf{K} = (H_0, H_1, \dots, H_\ell) \leftarrow_{\$} \mathbb{G}^{\ell+1}$.

$\text{PHF.Eval}(\mathbf{K} = (H_0, H_1, \dots, H_\ell), \mathbf{x} = (x_1, \dots, x_\ell)) \rightarrow \text{H}_{\mathbf{K}}^{MG}(\mathbf{x})$: Given hash key $\mathbf{K}$ and input $\mathbf{x}$, returns

$$\text{H}_{\mathbf{K}}^{MG}(\mathbf{x}) = H_0 + \sum_{i=1}^{\ell} x_i H_i.$$

⁸ We sometimes also refer to this for inverse polynomial δ if clear from the context.

PHF.TrapGen($1^\lambda, G, H$) $\rightarrow (\mathbf{K}', \text{td})$: On input 1^λ and group elements $G, H \in \mathbb{G}$, picks $a_{i,j} \leftarrow \{-1, 0, 1\}$ for $i \in \{0, 1, \dots, \ell\}$ and $j \in [J]$, where J is to be chosen concretely as in the theorem below, $a_i = \sum_{j=1}^J a_{i,j}$ for $i \in [\ell]$ and $a_0 = -1 + \sum_{j=1}^J a_{0,j}$, as well as $b_0, b_1, \dots, b_\ell \leftarrow \mathbb{Z}_p$. Then, it returns $\mathbf{K}' = (H_0, H_1, \dots, H_\ell)$, where

$$H_0 = a_0 G + b_0 H, \quad H_i = a_i G + b_i H \text{ for } i \in [\ell],$$

as well as $\text{td} = (\mathbf{a}, \mathbf{b})$ with $\mathbf{a} = (a_0, a_1, \dots, a_\ell)$ and $\mathbf{b} = (b_0, b_1, \dots, b_\ell)$.

PHF.TrapEval($\text{td} = (\mathbf{a}, \mathbf{b}), \mathbf{x} = (x_1, \dots, x_\ell)$): On input of trapdoor td and input $\mathbf{x}$, returns $a_{\mathbf{x}} = a_0 + \sum_{i=1}^\ell a_i x_i \in \mathbb{Z}$ and $b_{\mathbf{x}} = b_0 + \sum_{i=1}^\ell b_i x_i \in \mathbb{Z}_p$.⁹

In Appendix B, we prove the following theorem, which (to the best of our knowledge) is novel, as prior results only established that the construction is a $(1, q, \delta)$ -PHF and a $(2, 1)$ -PHF.

Theorem 3.11. *There are constants c and d , such that for any q , the above construction with $J = \lceil q^2 c \rceil$ is a $(2, q, \delta)$ -PHF for $\delta = \frac{d}{q^2(\ell+1)}$, if $(\ell+1)J + 1 < p$.*

In addition, we introduce another setup algorithm **PHF.NonZeroGen**, with the same input-output specification as **PHF.TrapGen**, except that **PHF.TrapEval** will always output a non-zero value $a_{\mathbf{x}}$ in the first component, similar to the signature scheme in [DH26]. This property will be useful in our security analysis.

PHF.NonZeroGen($1^\lambda, G, H$) $\rightarrow (\mathbf{K}', \text{td})$: On input 1^λ and group elements $G, H \in \mathbb{G}$, picks $\mathbf{a} = (1, 0, \dots, 0) \in \mathbb{Z}_p^{\ell+1}$ and $\mathbf{b} = (b_0, \dots, b_\ell) \leftarrow \mathbb{Z}_p^{\ell+1}$. Then, it sets $H_i \leftarrow a_i G + b_i H$ for $i \in [0, \ell]$. It outputs $\mathbf{K}' = (H_0, \dots, H_\ell)$ and $\text{td} = (\mathbf{a}, \mathbf{b})$.

Lemma 3.12. *The PHF defined as above, except that **PHF.TrapGen** is replaced with **PHF.NonZeroGen**, is $(0, q, 1)$ -programmable for any $q = \text{poly}(\lambda)$.*

Proof. This is true, since it is *always* the case that $a_0 + \sum_{i \in [\ell]} \mu_i a_i = 1$.

Σ -PROTOCOLS. We establish Σ -protocol notation and security. Let $\mathfrak{R} = \{(\mathfrak{x}, \mathfrak{w})\}$ be an NP-relation and denote by $\mathfrak{L}_{\mathfrak{R}} = \{\mathfrak{x} \mid \exists \mathfrak{w} \text{ s.t. } (\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}\}$ the induced language. A parameterized Σ -protocol Σ for $\mathfrak{R}$ with challenge space $\Sigma.C$ is a tuple of PPT algorithms $(\Sigma.\text{Init}, \Sigma.\text{Resp}, \Sigma.\text{Ver})$, where $\Sigma.\text{Resp}$ and $\Sigma.\text{Ver}$ are deterministic, such that

$\Sigma.\text{Init}(\mathfrak{x}, \mathfrak{w}) \rightarrow (A, \text{st})$: Given a statement-witness pair $(\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}$, outputs a first flow message (i.e., nonce) A and a state st .
 $\Sigma.\text{Resp}(\text{st}, c) \rightarrow z$: Given a state st and a challenge $c \in \Sigma.C$, outputs a third flow message (i.e., response) z .
 $\Sigma.\text{Ver}(\mathfrak{x}, A, c, z) \rightarrow b$: Given a statement $\mathfrak{x}$, a nonce A , challenge $c \in \Sigma.C$, and a response z , outputs a bit $b \in \{0, 1\}$.

We refer to the tuple $\tau = (A, c, z)$ as a *transcript*. We say that τ is valid for $\mathfrak{x}$ if $\text{Ver}(\mathfrak{x}, A, c, z)$ outputs 1. We may omit $\mathfrak{x}$ if clear by context. We recall desired security notions for Σ -protocols below, following [KR25].

Definition 3.13 (Correctness). *Let $\mathfrak{R}$ be an NP-relation and Σ be a Σ -protocol for $\mathfrak{R}$ with challenge space $\Sigma.C$. We say Σ is correct, if for all $(\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}$, the following probability is 1:*

$$\Pr[\Sigma.\text{Ver}(\mathfrak{x}, A, c, z) = 1 \mid (A, \text{st}) \leftarrow \Sigma.\text{Init}(\mathfrak{x}, \mathfrak{w}); c \leftarrow \Sigma.C; z \leftarrow \Sigma.\text{Resp}(\text{st}, c)].$$

Definition 3.14 (Special Soundness). *Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ -protocol for $\mathfrak{R}$. We call Σ k -special sound, if there exists a deterministic polynomial-time extractor $\Sigma.\text{Ext}$ that given statement $\mathfrak{x}$ and k valid transcripts $\{(A, c_i, z_i)\}_{i \in [k]}$ with pairwise distinct $\{c_i\}_{i \in [k]}$, outputs a witness $\mathfrak{w}$ such that $(\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}$. We say that Σ is k -special sound for relation $\hat{\mathfrak{R}}$ if the extracted witness $\mathfrak{w}$ satisfies $(\mathfrak{x}, \mathfrak{w}) \in \hat{\mathfrak{R}}$.*

⁹ We parse $\mathbf{a} = (a_0, \dots, a_\ell), \mathbf{b} = (b_0, \dots, b_\ell)$ differently from prior algorithms.

Definition 3.15 (High Min-Entropy). Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ -protocol for $\mathfrak{R}$. We say that Σ has high min-entropy if for all $(\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}$, it holds for $(A, \text{st}) \leftarrow \Sigma.\text{Init}(\mathfrak{x}, \mathfrak{w})$ that

$$2^{-H_{\min}(A)} = \text{negl}(\lambda).$$

We denote $H_{\min}\Sigma := \min_{\mathfrak{x} \in \mathcal{L}_{\mathfrak{R}}} H_{\min}(A)$.

Definition 3.16 (Relaxed Strong 2-Special Soundness). Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ -protocol for $\mathfrak{R}$. We say that Σ is strong 2-special sound for NP-relation $\hat{\mathfrak{R}} \supseteq \mathfrak{R}$ if there exists a deterministic PT algorithm Ext such that given as input two valid transcripts $\tau = (A, c, z)$ and $\tau' = (A, c', z')$ for statement $\mathfrak{x}$ with $\tau \neq \tau'$, the extractor outputs a witness $\mathfrak{w} \leftarrow \text{Ext}(\mathfrak{x}, \tau, \tau')$ such that $(\mathfrak{x}, \mathfrak{w}) \in \hat{\mathfrak{R}}$.

Definition 3.17 (SHVZK). Let $\mathfrak{R}$ be an NP-relation and Σ be an Σ -protocol for $\mathfrak{R}$. We say that Σ is perfect special honest-verifier zero-knowledge (SHVZK), if there exists a PPT zero-knowledge simulator $\Sigma.\text{Sim}$ such that for any $(\mathfrak{x}, \mathfrak{w}) \in \mathfrak{R}$ and $c \in \Sigma.\mathcal{C}$ that $D_{\text{Real}} = D_{\text{SHVZK}}$ for

$$\begin{aligned} D_{\text{Real}} &:= \{(\mathfrak{x}, \mathfrak{w}, (A, c, z)) \mid A \leftarrow \Sigma.\text{Init}(\mathfrak{x}, \mathfrak{w}); z \leftarrow \Sigma.\text{Resp}(\text{st}, c)\}, \\ D_{\text{SHVZK}} &:= \{(\mathfrak{x}, \mathfrak{w}, (A, c, z)) \mid (A, z) \leftarrow \Sigma.\text{Sim}(\mathfrak{x}, c)\}. \end{aligned}$$

Definition 3.18 (Relaxed Unique Responses). Let Σ be a Σ -protocol for relation $\mathfrak{R}$. We say that Σ has unique responses if for any statement $\mathfrak{x}$ and (A, c) there exists at most one z such that (A, c, z) is valid for $\mathfrak{x}$. We say that Σ has relaxed unique responses for relation $\hat{\mathfrak{R}}$ if there exists a deterministic polynomial-time extractor $\Sigma.\text{URExt}$ that given statement $\mathfrak{x}$ and two valid transcripts $\{(A, c, z_b)\}_{b \in \{0,1\}}$ with $z_0 \neq z_1$, outputs a witness $\mathfrak{w}$ such that $(\mathfrak{x}, \mathfrak{w}) \in \hat{\mathfrak{R}}$.

We further recall the notion of a blindable Σ -protocol from [BHKR25]. We generalize the notation to fit our needs and follow [KR25] to generalize the security property.

Definition 3.19 (Blindable Σ -protocol). Let $\Sigma = (\Sigma.\text{Init}, \Sigma.\text{Resp}, \Sigma.\text{Ver})$ be a Σ -protocol for relation $\mathfrak{R}$ with challenge space $\Sigma.\mathcal{C}$, and suppose Σ is SHVZK.

Let $\Sigma.\text{Rand} = (\Sigma.\text{BlindNonce}, \Sigma.\text{BlindChal}, \Sigma.\text{BlindChal}^{-1}, \Sigma.\text{BlindResp})$ be a set of PPT randomization algorithms such that

$\Sigma.\text{BlindNonce}(\mathfrak{x}, A) \rightarrow (\tilde{A}, \tilde{\text{st}})$: Given a statement $\mathfrak{x}$ and nonce A , outputs a blinded nonce $\tilde{A}$ and a state $\tilde{\text{st}}$.
 $\Sigma.\text{BlindChal}(c, \tilde{\text{st}}) \rightarrow \tilde{c}$: Given a challenge $c \in \Sigma.\mathcal{C}$ and a state $\tilde{\text{st}}$, deterministically outputs a blinded challenge $\tilde{c}$.
 $\Sigma.\text{BlindChal}^{-1}(\tilde{c}, \tilde{\text{st}}) \rightarrow c$: Given a blinded challenge $\tilde{c} \in \Sigma.\mathcal{C}$ and a state $\tilde{\text{st}}$, deterministically outputs a challenge c .
 $\Sigma.\text{BlindResp}(z, c, \tilde{\text{st}}) \rightarrow \tilde{z}$: Given a response z , a challenge c , and a state $\tilde{\text{st}}$, deterministically outputs a blinded response $\tilde{z}$.

We say Σ is (strongly) $\Sigma.\text{Rand}$ -blindable if

1. for all $(\tilde{A}, \tilde{\text{st}}) \leftarrow \Sigma.\text{BlindNonce}(\mathfrak{x}, A)$, the function $\text{BlindChal}^{-1}(\cdot, \tilde{\text{st}})$ is the inverse of $\text{BlindChal}(\cdot, \tilde{\text{st}})$ and
2. for all $\mathfrak{x} \in \mathcal{L}_{\mathfrak{R}}$ (resp. all $\mathfrak{x}$) and all valid transcripts $\tau = (A, c, z)$, the distributions D_{SHVZK} and D_{Rand} are identical, where

$$\begin{aligned} D_{\text{SHVZK}} &:= \{(\mathfrak{x}, (\tilde{A}, \tilde{c}, \tilde{z})) \mid \tilde{c} \leftarrow \Sigma.\mathcal{C}; (\tilde{A}, \tilde{z}) \leftarrow \Sigma.\text{Sim}(\mathfrak{x}, \tilde{c})\}, \\ D_{\text{Rand}} &:= \left\{ (\mathfrak{x}, (\tilde{A}, \tilde{c}, \tilde{z})) \left| \begin{array}{l} (\tilde{A}, \tilde{\text{st}}) \leftarrow \Sigma.\text{BlindNonce}(\mathfrak{x}, A) \\ \tilde{c} \leftarrow \Sigma.\text{BlindChal}(c, \tilde{\text{st}}) \\ \tilde{z} \leftarrow \Sigma.\text{BlindResp}(z, c, \tilde{\text{st}}) \end{array} \right. \right\}. \end{aligned}$$

We also recall the notion of a translatable Σ -protocol [BHKR25] which allows to modify a transcript $\tau = (A, c, z)$ for statement $\mathfrak{x}$ into a transcript $\tau' = (A', c', z')$ for a modified statement $\mathfrak{x}'$. We simplify the notion and remove the requirement on the distribution of τ' , as this is taken care of by blindability.

Algorithm $\text{Fork}_3^A(\text{inp})$: <pre> 1 : Pick random coins ρ_A ; $\mathbf{h} \leftarrow \mathcal{C}^Q$ 2 : $(I, \text{aux}) \leftarrow \mathcal{A}(\text{inp}, \mathbf{h}; \rho_A)$ 3 : if $I = \perp$ then return $\perp$ 4 : $\mathbf{h}'_{[I:Q]} \leftarrow \mathcal{C}^{Q-I+1}$; $\mathbf{h}' = \mathbf{h}_{[1:I-1]} \parallel \mathbf{h}'_{[I:Q]}$ 5 : $(I', \text{aux}') \leftarrow \mathcal{A}(\text{inp}, \mathbf{h}'; \rho_A)$ 6 : $\mathbf{h}''_{[I':Q]} \leftarrow \mathcal{C}^{Q-I'+1}$; $\mathbf{h}'' = \mathbf{h}_{[1:I'-1]} \parallel \mathbf{h}''_{[I':Q]}$ 7 : $(I'', \text{aux}'') \leftarrow \mathcal{A}(\text{inp}, \mathbf{h}''; \rho_A)$ 8 : if $\{I, I', I''\} \neq 1 \vee \{h_I, h'_I, h''_I\} < 3$ then 9 : return $\perp$ 10 : return $(1, (\text{aux}, \text{aux}', \text{aux}''))$ </pre>

Fig. 3. Forking algorithm

Definition 3.20 (Translatable Σ -protocol). Let $\Sigma = (\Sigma.\text{Init}, \Sigma.\text{Resp}, \Sigma.\text{Ver})$ be a Σ -protocol for relation $\mathfrak{R}$ with challenge space $\Sigma.\mathcal{C}$.

Let $\Sigma.\mathcal{R}_{\text{Trans}}$ be a set and let $\Sigma.\text{Trans} = (\Sigma.\text{TransStmt}, \Sigma.\text{TransNonce}, \Sigma.\text{TransResp})$ be polynomial-time algorithms such that

$\Sigma.\text{TransStmt}(\mathbb{x}, r) \rightarrow \mathbb{x}'$: Given a statement $\mathbb{x}$ and coins $r \in \Sigma.\mathcal{R}_{\text{Trans}}$, outputs a transformed statement $\mathbb{x}'$.
 $\Sigma.\text{TransNonce}(\mathbb{x}, A, r) \rightarrow A'$: Given a statement $\mathbb{x}$, a nonce A and coins $r \in \Sigma.\mathcal{R}_{\text{Trans}}$, outputs a nonce A' .
 $\Sigma.\text{TransResp}(\mathbb{x}, A, c, z, r) \rightarrow z'$: Given a statement $\mathbb{x}$, a nonce A , a challenge c , a response z and coins $r \in \Sigma.\mathcal{R}_{\text{Trans}}$, outputs a response z' .

We say that Σ is (strongly) $\Sigma.\text{Trans}$ -translatable if for all $r \in \Sigma.\mathcal{R}_{\text{Trans}}$, all $\mathbb{x} \in \mathfrak{L}_{\mathfrak{R}}$ (resp. all $\mathbb{x}$) and all valid transcripts $\tau = (A, c, z)$ for $\mathbb{x}$, the transcript (A', c, z') is valid for $\mathbb{x}' = \Sigma.\text{TransStmt}(\mathbb{x}, r)$, where $A' = \Sigma.\text{TransNonce}(\mathbb{x}, A, r)$ and $z' = \Sigma.\text{TransResp}(\mathbb{x}, A, c, z, r)$.

FORKING LEMMA. The forking lemma allows to argue about the adversary's behavior upon rewinding [BN06]. We require a variant which allows to rewind twice. Our variant is, e.g., implied by [CART26, Lemma 2.19], we give the proof in Appendix C for completeness.

Lemma 3.21 (3-Fork Rewinding). Fix an integer $Q \geq 1$ and a set $\mathcal{C}$ of size $|\mathcal{C}| \geq 2$. Let $\mathcal{A}$ be a randomized algorithm that on input $\text{inp}, \mathbf{h} \in \mathcal{C}^Q$ returns a pair $(I \in [Q] \cup \{\perp\}, \text{aux})$. Let IG be a randomized algorithm for sampling inp . Let $\text{acc} := \Pr[I \neq \perp | \text{inp} \leftarrow \text{IG}; \mathbf{h} \leftarrow \mathcal{C}^Q; (I, \text{aux}) \leftarrow \mathcal{A}(\text{inp}, \mathbf{h})]$. Let $\text{frk}_3 = \Pr[b = 1 : \text{inp} \leftarrow \text{IG}; (b, \text{aux}, \text{aux}', \text{aux}'') \leftarrow \text{Fork}_3^A(\text{inp})]$ where Fork_3^A is defined in Figure 3. Then,

$$\text{frk}_3 \geq \frac{\text{acc}}{2} \left(\frac{\text{acc}}{4Q} - \frac{1}{|\mathcal{C}|} \right)^2 - \frac{1}{|\mathcal{C}|}, \text{ or alternatively, } \text{acc} \leq \sqrt[3]{32Q^2 \cdot \left(\text{frk} + \frac{1}{|\mathcal{C}|} \right)} + \frac{4Q}{|\mathcal{C}|}.$$

NON-INTERACTIVE PROOF SYSTEMS. A non-interactive proof system Π for relation $\mathfrak{R}$ using a random oracle $\mathbf{H}$ is a pair $\Pi = (\Pi.\text{Prove}, \Pi.\text{Ver})$ of PPT algorithms with access to a random oracle, where

$\Pi.\text{Prove}^{\mathbf{H}}(\mathbb{x}, \mathbf{w}) \rightarrow \pi$: Given statement $\mathbb{x}$ and witness $\mathbf{w}$, generates a proof π .
 $\Pi.\text{Ver}^{\mathbf{H}}(\mathbb{x}, \pi) \rightarrow b$: Given $\mathbb{x}$ and a proof π , outputs $b \in \{0, 1\}$.

We define correctness, zero knowledge and straightline extractability (SLE), following [KRW24]. We refer to a proof system that satisfies these properties as an SLE-NIZK.

Definition 3.22 (Correctness). Let $\Pi = (\Pi.\text{Prove}, \Pi.\text{Ver})$ be a non-interactive proof system for a relation $\mathfrak{R}$. It has correctness error $\varepsilon = \varepsilon(\lambda)$ if for all $(\mathbb{x}, \mathbf{w}) \in \mathfrak{R}$, it holds that

$$\Pr[\pi \leftarrow \Pi.\text{Prove}^{\mathbf{H}}(\mathbb{x}, \mathbf{w}) : \Pi.\text{Ver}^{\mathbf{H}}(\mathbb{x}, \pi) = 1] \geq 1 - \varepsilon,$$

where the probability is over the choice of $\mathbf{H}$ and the randomness of $\Pi.\text{Prove}, \Pi.\text{Ver}$. We say that Π is correct if $\varepsilon = \text{negl}(\lambda)$.

Definition 3.23 (Zero-Knowledge). Let $\Pi = (\Pi.\text{Prove}, \Pi.\text{Ver})$ be a non-interactive proof system for a relation $\mathfrak{R}$. We say that Π is zero-knowledge, if there exists a PPT simulator Sim such that for all PPT adversaries $\mathcal{A}$ the following advantage is negligible:

$$\text{Adv}_{\Pi, \text{Sim}}^{\text{zk}}(\mathcal{A}, \lambda) := \left| \Pr[b = 1 \mid b \leftarrow \mathcal{A}^{\text{H}, \mathcal{O}_{\text{Prove}}}(1^\lambda)] - \Pr[b = 1 \mid b \leftarrow \mathcal{A}^{\text{H}, \mathcal{O}_{\text{Sim}}}(1^\lambda)] \right|,$$

where $\mathcal{A}$ has (black-box) access to the random oracle H and to an oracle $\mathcal{O}_{\text{Prove}}$ or $\mathcal{O}_{\text{Sim}}$, which are as follows:

- $\mathcal{O}_{\text{Prove}}(\mathbb{x}, \mathbb{w})$: Return $\perp$ if $(\mathbb{x}, \mathbb{w}) \notin \mathfrak{R}$. Else, output $\pi \leftarrow \Pi.\text{Prove}^{\text{H}}(\mathbb{x}, \mathbb{w})$.
- $\mathcal{O}_{\text{Sim}}(\mathbb{x}, \mathbb{w})$: Return $\perp$ if $(\mathbb{x}, \mathbb{w}) \notin \mathfrak{R}$. Else, output $\pi \leftarrow \text{Sim}^{\text{H}}(\mathbb{x})$.

Furthermore, the simulator Sim is allowed to program the random oracle H on any fresh input to H , i.e. if $\text{H}(m)$ has not been queried before, then Sim is free to choose $\text{H}(m)$, else, programming fails.

We define (relaxed) straightline extractability, where it is possible to extract a witness for a relaxed relation $\hat{\mathfrak{R}} \supseteq \mathfrak{R}$ in a straightline manner. We refer to $\hat{\mathfrak{R}}$ as the knowledge relation.

Definition 3.24 (Relaxed Straightline Extractability). Let $\Pi = (\Pi.\text{Prove}, \Pi.\text{Ver})$ be a non-interactive proof system for a relation $\mathfrak{R}$ and let $\hat{\mathfrak{R}} \supseteq \mathfrak{R}$ be a relation. We say that Π is straightline extractable (SLE) for $\hat{\mathfrak{R}}$, if there exists a PPT extractor Ext such that for any $\mathcal{A}$ the following advantage is negligible:

$$\text{Adv}_{\Pi, \text{Ext}, \hat{\mathfrak{R}}}^{\text{sle}}(\mathcal{A}, \lambda) := \left| \Pr[b = 1 \mid b \leftarrow \mathcal{A}^{\text{H}, \mathcal{O}_{\text{Ver}}}(1^\lambda)] - \Pr[b = 1 \mid b \leftarrow \mathcal{A}^{\text{H}, \mathcal{O}_{\text{Ext}}}(1^\lambda)] \right|,$$

where $\mathcal{A}$ has access to the random oracle H and to an oracle $\mathcal{O}_{\text{Ver}}$ or $\mathcal{O}_{\text{Ext}}$, which are as follows:

- $\mathcal{O}_{\text{Ver}}(\mathbb{x}, \pi)$: Return $\Pi.\text{Ver}^{\text{H}}(\mathbb{x}, \pi)$.
- $\mathcal{O}_{\text{Ext}}(\mathbb{x}, \pi)$: If $\Pi.\text{Ver}^{\text{H}}(\mathbb{x}, \pi) = 0$, return 0. Otherwise, if $(\mathbb{x}, \mathbb{w}) \notin \hat{\mathfrak{R}}$ for $\mathbb{w} \leftarrow \text{Ext}(\mathcal{Q}, \mathbb{x}, \pi)$, return 0. Else, return 1. Here, $\mathcal{Q}$ denotes the set of $\mathcal{A}$'s H queries.

4 Blindable Σ -protocols

We present the Σ -protocols that we employ to construct our blind signature.

OKAMOTO-SCHNORR Σ -PROTOCOL. Denote by Σ_{os} the Σ -protocol underlying the Okamoto-Schnorr identification scheme [Sch90, Oka93]. The underlying relation, parameterized by $\text{par}_{\text{os}} = (G, H)$ is given by the linear map $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$ via

$$\mathfrak{R}_{\text{os}} := \{(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}}) \mid W = \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbb{w}_{\text{os}}) := w_0 G + w_1 H\}, \quad (2)$$

where $\mathbb{x}_{\text{os}} = (G, H, W)$ and $\mathbb{w}_{\text{os}} = (w_0, w_1)$. We may write Φ_{os} for short if par_{os} is clear by context. For completeness, we provide an explicit description of Σ_{os} in Figure 4 including blinding operations.

DISCRETE LOGARITHM Σ -PROTOCOL. Denote by Σ_{dl} the canonical Σ -protocol for proving knowledge of the discrete logarithm of a group element $X \in \mathbb{G}$ [Sch90]. The underlying relation, parameterized by $\text{par}_{\text{dl}} = G$, is given by the linear map $\Phi_{\text{dl}}^{\text{par}_{\text{dl}}}$ via

$$\mathfrak{R}_{\text{dl}} := \{(\mathbb{x}_{\text{dl}}, \mathbb{w}_{\text{dl}}) \mid X = \Phi_{\text{dl}}^{\text{par}_{\text{dl}}}(\mathbb{w}_{\text{dl}}) := xG\}, \quad (3)$$

where $\mathbb{x}_{\text{dl}} = (G, X)$ and $\mathbb{w}_{\text{dl}} = x$. We may write Φ_{dl} for short if par_{dl} is clear by context. Note that we employ challenge space $\mathcal{C}_{\text{dl}} = \mathbb{Z}_p^\times$ instead of $\mathbb{Z}_p$ and we blind as in [TZ22].¹⁰ For completeness, we provide an explicit description of Σ_{dl} in Figure 5 including blinding operations.

¹⁰ Looking ahead, this is because Σ_{phf} is OR-compiled with Σ_{dl} following [CDS94], and Σ_{phf} requires multiplicative blinding with challenges over $\mathbb{Z}_p^\times$. For OR-proof compatibility, we need Σ_{dl} to have a consistent choice (cf. Lemma 4.4).

Algorithm $\Sigma_{\text{os}}.\text{Init}(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}})$: 1 : parse $\mathbb{x}_{\text{os}} = (G, H, W)$, $\mathbb{w}_{\text{os}} = \mathbf{w}$ 2 : $\mathbf{r} \leftarrow \mathbb{Z}_p^2$ 3 : $A_{\text{os}} := \Phi_{\text{os}}(\mathbf{r})$ 4 : $\text{st}_{\text{os}} := (\mathbf{w}, \mathbf{r})$ 5 : return $(A_{\text{os}}, \text{st}_{\text{os}})$	Algorithm $\Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}})$: 1 : parse $\mathbb{x}_{\text{os}} = (G, H, W)$ 2 : $\alpha \leftarrow \mathbb{Z}_p^2$; $\gamma \leftarrow \mathbb{Z}_p$ 3 : $\tilde{A}_{\text{os}} := A_{\text{os}} + \Phi_{\text{os}}(\alpha) - \gamma W$ 4 : $\tilde{\text{st}}_{\text{os}} := (\alpha, \gamma)$ 5 : return $(\tilde{A}_{\text{os}}, \tilde{\text{st}}_{\text{os}})$
Algorithm $\Sigma_{\text{os}}.\text{Resp}(\text{st}_{\text{os}}, c_{\text{os}})$: 1 : assert $c_{\text{os}} \in \mathcal{C}_{\text{os}}$ 2 : parse $\text{st}_{\text{os}} = (\mathbf{w}, \mathbf{r})$ 3 : return $z_{\text{os}} := c_{\text{os}} \cdot \mathbf{w} + \mathbf{r}$	Algorithm $\Sigma_{\text{os}}.\text{BlindChal}(c_{\text{os}}, \tilde{\text{st}}_{\text{os}})$: 1 : parse $\tilde{\text{st}}_{\text{os}} = (\alpha, \gamma)$ 2 : return $\tilde{c}_{\text{os}} := c_{\text{os}} + \gamma$
Algorithm $\Sigma_{\text{os}}.\text{Ver}(\mathbb{x}_{\text{os}}, A_{\text{os}}, c_{\text{os}}, z_{\text{os}})$: 1 : parse $\mathbb{x}_{\text{os}} = (G, H, W)$ 2 : assert $A_{\text{os}} + c_{\text{os}}W = \Phi_{\text{os}}(z_{\text{os}})$ 3 : return 1	Algorithm $\Sigma_{\text{os}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{os}}, \tilde{\text{st}}_{\text{os}})$: 1 : parse $\tilde{\text{st}}_{\text{os}} = (\alpha, \gamma)$ 2 : return $c_{\text{os}} := \tilde{c}_{\text{os}} - \gamma$
	Algorithm $\Sigma_{\text{os}}.\text{BlindResp}(z_{\text{os}}, c_{\text{os}}, \tilde{\text{st}}_{\text{os}})$: 1 : parse $\tilde{\text{st}}_{\text{os}} = (\alpha, \gamma)$ 2 : return $\tilde{z}_{\text{os}} := z_{\text{os}} + \alpha$

Fig. 4. The Σ -protocol for relation $\mathfrak{R}_{\text{os}}$ with challenge space $\mathcal{C}_{\text{os}} = \mathbb{Z}_p$. We may write $\tilde{A}_{\text{os}} = \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}_{\text{os}})$ with explicit randomness $\tilde{\text{st}}_{\text{os}} = (\alpha, \gamma)$.

Algorithm $\Sigma_{\text{dl}}.\text{Init}(\mathbb{x}_{\text{dl}}, \mathbb{w}_{\text{dl}})$: 1 : parse $\mathbb{x}_{\text{dl}} = (G, X)$, $\mathbb{w}_{\text{dl}} = x$ 2 : $r \leftarrow \mathbb{Z}_p$ 3 : $A_{\text{dl}} := \Phi_{\text{dl}}(r)$ 4 : $\text{st}_{\text{dl}} := (x, r)$ 5 : return $(A_{\text{dl}}, \text{st}_{\text{dl}})$	Algorithm $\Sigma_{\text{dl}}.\text{BlindNonce}(\mathbb{x}_{\text{dl}}, A_{\text{dl}})$: 1 : parse $\mathbb{x}_{\text{dl}} = (G, X)$ 2 : $\alpha \leftarrow \mathbb{Z}_p$; $\gamma \leftarrow \mathbb{Z}_p^\times$ 3 : $\tilde{A}_{\text{dl}} := \gamma \cdot A_{\text{dl}} + \Phi_{\text{dl}}(\alpha)$ 4 : $\tilde{\text{st}}_{\text{dl}} := (\alpha, \gamma)$ 5 : return $(\tilde{A}_{\text{dl}}, \tilde{\text{st}}_{\text{dl}})$
Algorithm $\Sigma_{\text{dl}}.\text{Resp}(\text{st}_{\text{dl}}, c_{\text{dl}})$: 1 : assert $c_{\text{dl}} \in \mathcal{C}$ 2 : parse $\text{st}_{\text{dl}} = (x, r)$ 3 : return $z_{\text{dl}} := c_{\text{dl}} \cdot x + r$	Algorithm $\Sigma_{\text{dl}}.\text{BlindChal}(c_{\text{dl}}, \tilde{\text{st}}_{\text{dl}})$: 1 : parse $\tilde{\text{st}}_{\text{dl}} = (\alpha, \gamma)$ 2 : return $\tilde{c}_{\text{dl}} := c_{\text{dl}} \cdot \gamma$
Algorithm $\Sigma_{\text{dl}}.\text{Ver}(\mathbb{x}_{\text{dl}}, A_{\text{dl}}, c_{\text{dl}}, z_{\text{dl}})$: 1 : assert $A_{\text{dl}} + c_{\text{dl}}X = \Phi_{\text{dl}}(z_{\text{dl}})$ 2 : return 1	Algorithm $\Sigma_{\text{dl}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{dl}}, \tilde{\text{st}}_{\text{dl}})$: 1 : parse $\tilde{\text{st}}_{\text{dl}} = (\alpha, \gamma)$ 2 : return $c_{\text{dl}} := \tilde{c}_{\text{dl}}/\gamma$
	Algorithm $\Sigma_{\text{dl}}.\text{BlindResp}(z_{\text{dl}}, c_{\text{dl}}, \tilde{\text{st}}_{\text{dl}})$: 1 : parse $\tilde{\text{st}}_{\text{dl}} = (\alpha, \gamma)$ 2 : return $\tilde{z}_{\text{dl}} := \gamma z_{\text{dl}} + \alpha$

Fig. 5. The Σ -protocol for $\mathfrak{R}_{\text{dl}}$ with challenge space $\mathcal{C}_{\text{dl}} = \mathbb{Z}_p^\times$.

Σ -PROTOCOL FOR NON-ZERO PHF EVALUATION. Let $n = \lceil \log_2 p \rceil$. Let $\text{hpp} = (G, V, \mathbf{U} = (U_0, \dots, U_n)) \in \mathbb{G}^{n+3}$ with $V \neq 0$. Denote

$$\begin{aligned} \mathbb{x}_{\text{phf}} &= (\text{hpp}, C_{\S}, A_{\S}, T_{\S}, e_{\S}) \in \mathbb{G}^{n+6} \times \mathbb{Z}_p^\times, \\ \mathbb{w}_{\text{phf}} &= (\alpha, \beta, \nu, \rho, r_{\S}, \boldsymbol{\mu}) \in \mathbb{Z}_p^5 \times \{0, 1\}^n. \end{aligned} \quad (4)$$

We define the relation induced by $\Phi_{\text{phf}}^{\text{hpp}, T_{\S}}$ by

$$\mathfrak{R}_{\text{phf}} := \left\{ (\mathbb{x}_{\text{phf}}, \mathbb{w}_{\text{phf}}) \left| \begin{bmatrix} C_{\S} \\ A_{\S} \\ T_{\S} - U_0 \\ e_{\S} G \end{bmatrix}^{\top} = \Phi_{\text{phf}}^{\text{hpp}, T_{\S}}(\mathbb{w}_{\text{phf}}) := \begin{bmatrix} \nu G + r_{\S} V \\ \nu V + \sum_{i \in [n]} \mu_i (2^{i-1} G) \\ \rho V + \sum_{i \in [n]} \mu_i U_i \\ \alpha T_{\S} + \beta V \end{bmatrix}^{\top} \right. \right\} \quad (5)$$

We may write $\Phi_{\text{phf}}^{T_s}$ as a shorthand for $\Phi_{\text{phf}}^{\text{hpp}, T_s}$ if clear by context. This relation allows to enforce integrity properties on the group element $A = A_s - \nu V$, which is blinded with an additive term νV , if the exponent $\mu = \sum_{i \in [n]} \mu_i 2^{i-1} \bmod p$ of A is known.¹¹ Looking ahead, A will correspond to the nonce A_{os} from the Okamoto-Schnorr Σ -protocol in the security analysis of our blind signature. Let us explain the purpose of each row:

Commitment to blinding term for A . The element C_s is a Pedersen commitment to the blinding randomness ν for A_s .

Blinded Okamoto-Schnorr nonce. The element A_s corresponds to the value A , blinded additively by the term νV (i.e., $A_s = A + \nu V$), where $\mu = (\mu_1, \dots, \mu_n)$ is some bit-decomposition of the exponent $\mu = \sum_{i \in [n]} \mu_i 2^{i-1} \bmod p$ of A .

Blinded PHF evaluation. The element T_s is the evaluation $H_U^{MG}(\mu)$ of the PHF with key U on μ , but blinded with ρV .

Non-zero PHF evaluation. The element $e_s G$ corresponds to the non-zero proof of the PHF trapdoor evaluation, i.e., $e \neq 0 \bmod p$, where e is computed via PHF.TrapEval . Note that the G component in T_s contains the PHF evaluation e and non-zero evaluation is ensured by $e_s \neq 0$. If $e \neq 0$, the honest PHF evaluation yields a valid witness (assuming the trapdoor is known).

The Σ -protocol. We define the Σ -protocol Σ_{phf} for $\mathfrak{R}_{\text{phf}}$ in Figure 6. We discuss the design below. The Σ -protocol proves the linear constraints from the map Φ_{phf} with the usual generalization of Schnorr's Σ -protocol to linear relations (e.g., [Mau15]), with challenge space $\mathcal{C}_{\text{phf}} := \mathbb{Z}_p^\times$. Note that the full relation is non-linear due to the requirement that $\mu \in \{0, 1\}^n$. The latter is proven by showing that for all $i \in [n]$, we have that $\mu_i(\mu_i - 1) = 0$. To do so, we follow the technique in, e.g., [YAZ⁺19, BLS19, CKLR21, CGKR22], which we recap briefly below. Let $\mathbf{z}_\mu = c\mu + \mathbf{r}$ be the canonical Σ -protocol response to prove knowledge of μ with randomness $\mathbf{r} \in \mathbb{Z}_p^n$ and challenge c . We interpret

$$\mathbf{f}(c) := \mathbf{z}_\mu(\mathbf{z}_\mu - c \cdot \mathbf{1}) = \mu(\mu - 1)c^2 + \mathbf{f}_1 c + \mathbf{f}_0$$

as a vector of n polynomials with variable c , where $\mathbf{f}_1 = \mathbf{r}(2\mu - 1)$ and $\mathbf{f}_0 = \mathbf{r}^2$. Note that by design, the leading coefficient of f_i is 0 if and only if $\mu_i(\mu_i - 1) = 0$. Thus, f_i is a polynomial of degree at most 1 if and only if $\mu_i \in \{0, 1\}$. To convince the verifier of this claim, the prover commits (via Pedersen) to the non-zero coefficients $\mathbf{f}_1$ and $\mathbf{f}_0$ of $\mathbf{f}(X) = \mu(\mu - 1)X^2 + \mathbf{f}_1 X + \mathbf{f}_0$ in $\mathbf{D}_1 = \text{C}_{\text{ped}}.\text{Commit}(\mathbf{f}_1; \mathbf{t}_1)$ and $\mathbf{D}_0 = \text{C}_{\text{ped}}.\text{Commit}(\mathbf{f}_0; \mathbf{t}_0)$, respectively. Here, $\mathbf{t}_0, \mathbf{t}_1 \in \mathbb{Z}_p^n$ are the commitment randomness. These commitments are also sent in the first flow of the Σ -protocol. Upon reception of the challenge, the prover then reveals $\mathbf{t} = c \cdot \mathbf{t}_1 + \mathbf{t}_0$ which allows the verifier to check that

$$c \cdot \mathbf{D}_1 + \mathbf{D}_0 = \text{C}_{\text{ped}}.\text{Commit}(\mathbf{f}(c); \mathbf{t}) .$$

Note that the left-hand-side is the evaluation of the degree 1 polynomial $\mathbf{f}_1 X + \mathbf{f}_0$ on input c and the above check establishes equality with $\mathbf{f}(c)$. Thus, the degree-two polynomial $\mathbf{f}$ agrees with a degree-one polynomial on input c . Since the commitments $\mathbf{D}_1, \mathbf{D}_0$ were established before seeing the challenge c , the leading coefficient $\mu(\mu - 1)$ of $\mathbf{f}$ is zero with high probability under Schwarz-Zippel. It follows that $\mu_i \in \{0, 1\}$. Note that the formal argument establishes 3-special soundness for a relaxed relation (either binding of commitment is broken or $\mu_i(\mu_i - 1) = 0$ as desired).

Blinding the transcript. We note that blinding the proof for $\mu \in \{0, 1\}^n$ is less straightforward due to its non-linear nature. Let us give some intuition for the complications that arise. Following the blinding procedure in [TZ22], the challenge c and response $\mathbf{z}_\mu$ associated to the bit-decomposition μ are blinded via

$$\tilde{c} = c \cdot \gamma ; \quad \tilde{\mathbf{z}}_\mu = \gamma \mathbf{z}_\mu + \beta ,$$

¹¹ To be more precise, without knowing the PHF trapdoor behind hpp , it is hard to compute a valid transcript for Σ_{phf} ; it may help to think of a Fiat-Shamir compiled proof for relation $\mathfrak{R}_{\text{phf}}$ as a signature on A (though this view is not entirely accurate).

where $\beta \leftarrow \mathbb{Z}_p^n$ randomizes the response and $\gamma \leftarrow \mathbb{Z}_p^\times$ randomizes the challenge. A simple computation yields that the associated blinded polynomial $\tilde{f}(\tilde{c})$ in verification may be expressed as

$$\tilde{f}(\tilde{c}) = \tilde{z}_\mu(\tilde{z}_\mu - \tilde{c} \cdot \mathbf{1}) = \gamma^2 f(c) + 2\gamma z_\mu \beta - \tilde{c}\beta + \beta^2.$$

For blinding to succeed, we must therefore compute commitments $\tilde{D}_0, \tilde{D}_1$ and opening $\tilde{t}$ such that the following holds.

- The commitments $\tilde{D}_0, \tilde{D}_1$ are computable given the first flow A_{phf} of the to-be-blinded transcript. The opening $\tilde{t}$ may be computed given the full transcript $\tau_{\text{phf}} = (A_{\text{phf}}, c, z_{\text{phf}})$.
- We have $\tilde{c}\tilde{D}_1 + \tilde{D}_0 = C_{\text{ped}}.\text{Commit}(\tilde{f}(\tilde{c}); \tilde{t})$.
- The values $\tilde{D}_0, \tilde{D}_1$ and $\tilde{t}$ are well-distributed.

The latter point is easy to achieve by randomizing the commitments and adjusting the opening accordingly. Finding the appropriate commitments and openings in the first place turns out to be challenging. Observe that we may use the fact that we have appropriate commitments D_0, D_1 in τ_{phf} to express $f(c)$. (Thanks to homomorphic properties of C_{ped} , we may also adjust the commitments for the factor γ^2 .) Also, the terms $\tilde{c}\beta$ and β^2 are easy to deal with by accounting for β in $\tilde{D}_1$ (which is later multiplied by $\tilde{c}$ in verification) and β^2 in $\tilde{D}_0$.

Unfortunately, the response z_μ is not known yet when setting up the blinded commitments $\tilde{D}_0, \tilde{D}_1$. Without further modification, it seems that blinding fails. Before we proceed, let us remark that if we were to employ additive blinding for the challenge (i.e., set $\tilde{c} = c + \gamma$) as is more common in the literature, then the above term would contain even more cross-terms.¹² To summarize, we must find C_{ped} commitments D'_1, D'_0 given A_{phf} , in such a way that we may open $\tilde{c}D'_1 + D'_0$ to γz_μ after seeing the challenge c and response z_{phf} . Then, the above observations together with homomorphic properties of C_{ped} yield the desired blinding algorithms.

Our core insight is that we may extend the proven relation $\Phi_{\text{phf}}^{T_\S}$ with additional components to enable the computation of appropriate commitments D'_1, D'_0 . For this, let us inspect C_{ped} . As we use the generators G, V , commitments have the form $xG + yV$ with message $x \in \mathbb{Z}_p$ and randomness $y \in \mathbb{Z}_p$. To aid handling the cross-term z_μ , we let the prover instead prove the following modified relation induced by $\hat{\Phi}_{\text{phf}}^{T_\S}$ defined as follows. Let

$$\hat{\mathbb{X}}_{\text{phf}} = (C_\mu, \mathbb{X}_{\text{phf}}) \in \mathbb{G}^{2n+6} \times \mathbb{Z}_p^\times, \quad \hat{\mathbb{W}}_{\text{phf}} = (r_\mu, \mathbb{W}_{\text{phf}}) \in \mathbb{Z}_p^{n+5} \times \{0, 1\}^n.$$

Let $\mathbf{X} = \Phi_{\text{phf}}^{T_\S}(\mathbb{W}_{\text{phf}}) = (C_\$, A_\$, T_\$ - U_0, e_\$G)$ and let

$$(C_\mu, \mathbf{X}) = \hat{\Phi}_{\text{phf}}^{\text{hpp}, T_\S}(\hat{\mathbb{W}}_{\text{phf}}) := (\mu G + r_\mu V, \Phi_{\text{phf}}^{T_\S}(\mathbb{W}_{\text{phf}})). \quad (6)$$

In short, the commitments C_μ and their respective Σ -protocol nonces A_μ then allow to compute D'_1 and D'_0 , respectively, as desired. To see this, observe that C_μ and A_μ are now part of A_{phf} . Further, we have by verification that

$$cC_\mu + A_\mu = z_\mu G + z_r V,$$

where z_r is the response associated to r_μ . Multiplying the above by γ and using that $\tilde{c} = \gamma c$, we find that $D'_1 = C_\mu$ and $D'_0 = \gamma A_\mu$ suffice. Indeed, we have

$$\tilde{c}D'_1 + D'_0 = \gamma(cC_\mu) + \gamma A_\mu = \gamma z_\mu G + \gamma z_r V$$

which constitutes a commitment to γz_μ with opening $t' = \gamma z_r$, as desired. We refer to Figure 6 for the formal description.

¹² Notably, we employ multiplicative blinding as we did not manage to account for some of the arising cross-terms when blinding the challenge additively.

Translating the statement. Let $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$ and $\Sigma.\mathcal{R}_{\text{Trans}} = \mathbb{Z}_p^3 \times \mathbb{Z}_p^{\times}$. For our application, it is important that the statement $\mathbb{x}_{\text{phf}}$ may be randomized into the blinded statement $\tilde{\mathbb{x}}_{\text{phf}}$ as follows.

Denote $\text{coins}_{\text{phf}} = (\nu', r'_{\S}, \rho', e') \in \Sigma.\mathcal{R}_{\text{Trans}}$. The algorithm **TransStmt** receives $(\mathbb{x}_{\text{phf}}, \text{coins}_{\text{phf}})$ as input and outputs $\tilde{\mathbb{x}}_{\text{phf}} = (G, V, \mathbf{U}, \tilde{C}_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$, where

$$\begin{aligned} \tilde{C}_{\S} &= C_{\S} + \nu'G + r'_{\S}V, & \tilde{A}_{\S} &= A_{\S} + \nu'V, \\ \tilde{T}_{\S} &= T_{\S} + \rho'V, & \tilde{e}_{\S} &= e_{\S} \cdot e'. \end{aligned} \tag{7}$$

The above allows to randomize the non-persistent components of the statement $\mathbb{x}_{\text{phf}}$ by sampling $\text{coins}_{\text{phf}}$ at random.¹³ Our Σ -protocol Σ_{phf} allows to randomize the statement as above, while ensuring that an associated transcript $\tau_{\text{phf}} = (A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ remains valid (following the notion of translatability, cf. Definition 3.20). The mechanism is described in Figure 7.

Let us give some intuition. First, observe that the underlying witnesses ν in τ_{phf} is additively randomized into $\tilde{\nu} = \nu + \nu'$. Thus, the original Σ_{phf} response $z_{\nu} = c_{\text{phf}} \cdot \nu + r_{\nu}$ associated to ν may be shifted via

$$z'_{\nu} = c_{\text{phf}} \cdot \tilde{\nu} + r_{\nu} = z_{\nu} + c_{\text{phf}}\nu'.$$

The same mechanism works for $r_{\S}$ and ρ .

Since $e_{\S}$ is randomized multiplicatively, the mechanism differs. Here, we employ the blinding mechanism for the non-zero proof of [KLR26]. That is, the associated nonce and responses are multiplied by e' . To see why this works, denote by $A_e = r_{\alpha}T_{\S} + r_{\beta}V$ the nonce in τ_{phf} associated to the relation $e_{\S}G = \alpha T_{\S} + \beta V$. Similarly, denote by $z_{\alpha} = c_{\text{phf}} \cdot \alpha + r_{\alpha}$ and $z_{\beta} = c_{\text{phf}} \cdot \beta + r_{\beta}$ the responses associated to witnesses α and β , respectively. Observe that

$$\tilde{e}_{\S}G = e' \cdot (e_{\S}G) = e' \cdot (\alpha T_{\S} + \beta V) = (e' \cdot \alpha)T_{\S} + (e' \cdot \beta)V.$$

That is, the underlying witnesses (α, β) are multiplied by e' when $e_{\S}$ is randomized into $\tilde{e}_{\S}$. Thus, we multiply the responses z_{α} and z_{β} with e' to adjust the underlying witness. In turn, this also multiplies the randomness r_{α} and r_{β} , for which we must account in the nonce by setting $A'_e = e' A_e$.

Note that a subtle issue remains. Observe that the linear map $\Phi_{\text{phf}}^{T_{\S}}$ depends on $T_{\S}$. When randomizing $T_{\S}$ into $\tilde{T}_{\S}$, the underlying map also changes to $\Phi_{\text{phf}}^{\tilde{T}_{\S}}$. Fortunately, it suffices to adjust for the change by setting

$$z'_{\beta} = e'(z_{\beta} - \rho'z_{\alpha}).$$

This is shown in the formal analysis in Equation (10). Finally, we remark that these mechanisms work if τ_{phf} is valid, independent of whether τ_{phf} was computed honestly (as implicitly assumed in the discussion above).

SECURITY PROPERTIES OF THE Σ -PROTOCOLS. Below, we prove the desired properties of the above Σ -protocols.

Lemma 4.1. *The Σ -protocol Σ_{dl} is correct, SHVZK, 2-special sound, and has unique responses. Further, it is strongly Σ_{dl} .Rand-blindable with Σ_{dl} .Rand as given in Figure 5.*

Proof. This is almost the canonical Σ -protocol and randomization procedure for $\mathfrak{R}_{\text{dl}}$; the only exception is multiplicative blinding for the challenge. So the statement follows from [CART26, Lemma 2.17] as before, except for blindability. The latter is easy to verify.

We define $\tilde{\mathfrak{R}}_{\text{os}} = \{(\mathbb{x}, \mathbf{w}) \mid \phi_{\text{os}}(\mathbf{w}) = 0 \wedge \mathbf{w} \neq \mathbf{0}\}$ for the relaxed unique responses property of Σ_{os} .

¹³ Looking ahead, the group elements G, V are part of the public parameters and $\mathbf{U}$ is determined by the common message info. Therefore, these elements need not be randomized in our blind signature. In contrast, the elements $(C_{\S}, A_{\S}, T_{\S}, e_{\S})$ are specific to each signing session and must be randomized by the user, together with the τ_{phf} transcript issued by the signer.

Algorithm $\Sigma_{\text{phf}}.\text{Init}(\mathbb{x}_{\text{phf}}, \mathbb{w}_{\text{phf}})$: <pre> 1: parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\mathbb{S}}, A_{\mathbb{S}}, T_{\mathbb{S}}, e_{\mathbb{S}})$ 2: parse $\mathbb{w}_{\text{phf}} = (\alpha, \beta, \nu, \rho, r_{\mathbb{S}}, \mu)$ 3: $r_{\mu} \leftarrow \mathbb{Z}_p^n$; $C_{\mu} := \mu G + r_{\mu} V$ 4: $\mathbf{w} := (r_{\mu}, \alpha, \beta, \nu, \rho, r_{\mathbb{S}}, \mu) \in \mathbb{Z}_p^{5+2n}$ 5: $\mathbf{s} \leftarrow \mathbb{Z}_p^{5+2n}$; $\mathbf{A} := \hat{\Phi}_{\text{phf}}^{T_{\mathbb{S}}}(\mathbf{s})$ 6: $t_0, t_1 \leftarrow \mathbb{Z}_p^n$; $\mathbf{r} := (s_{n+5+1}, \dots, s_{n+5+n})$ 7: $\mathbf{f}_1 := \mathbf{r}(2\mu - 1)$; $\mathbf{f}_0 := \mathbf{r}^2$ 8: $D_1 := \mathbf{f}_1 G + t_1 V$; $D_0 := \mathbf{f}_0 G + t_0 V$ 9: $\text{st}_{\text{phf}} := (\mathbf{w}, \mathbf{s}, t_0, t_1)$ 10: return $(A_{\text{phf}} := (A, C_{\mu}, D_0, D_1), \text{st}_{\text{phf}})$ </pre> Algorithm $\Sigma_{\text{phf}}.\text{Resp}(\text{st}_{\text{phf}}, c_{\text{phf}})$: <pre> 1: assert $c_{\text{phf}} \in C_{\text{phf}}$ 2: parse $\text{st}_{\text{phf}} = (\mathbf{w}, \mathbf{s}, t_0, t_1)$ 3: $\mathbf{z} := c_{\text{phf}} \cdot \mathbf{w} + \mathbf{s}$ 4: $\mathbf{t} := c_{\text{phf}} \cdot t_1 + t_0$ 5: return $z_{\text{phf}} := (\mathbf{z}, \mathbf{t})$ </pre> Algorithm $\Sigma_{\text{phf}}.\text{Ver}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$: <pre> 1: parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\mathbb{S}}, A_{\mathbb{S}}, T_{\mathbb{S}}, e_{\mathbb{S}})$ 2: parse $A_{\text{phf}} = (A, C_{\mu}, D_0, D_1)$, $z_{\text{phf}} = (\mathbf{z}, \mathbf{t})$ 3: $\mathbf{Y} := (C_{\mu}, C_{\mathbb{S}}, A_{\mathbb{S}}, T_{\mathbb{S}} - U_0, e_{\mathbb{S}} G)$ 4: assert $e_{\mathbb{S}} \neq 0$ // explicit check for clarity 5: assert $\mathbf{A} + c_{\text{phf}} \mathbf{Y} = \hat{\Phi}_{\text{phf}}^{T_{\mathbb{S}}}(\mathbf{z})$ // response corresponding to μ 6: $\mathbf{z}_{\mu} := (z_{n+5+1}, \dots, z_{n+5+n})$ 7: $\mathbf{f} := \mathbf{z}_{\mu}(\mathbf{z}_{\mu} - c_{\text{phf}} \cdot \mathbf{1})$ 8: assert $c_{\text{phf}} D_1 + D_0 = \mathbf{f} G + \mathbf{t} V$ 9: return 1 </pre>	Algorithm $\Sigma_{\text{phf}}.\text{BlindNonce}(\mathbb{x}_{\text{phf}}, A_{\text{phf}})$: <pre> 1: parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\mathbb{S}}, A_{\mathbb{S}}, T_{\mathbb{S}}, e_{\mathbb{S}})$ 2: parse $A_{\text{phf}} = (A, C_{\mu}, D_0, D_1)$ 3: $\alpha \leftarrow \mathbb{Z}_p^{5+2n}$; $\gamma \leftarrow \mathbb{Z}_p^{\times}$; $\tau_0, \tau_1, \rho_{\mu} \leftarrow \mathbb{Z}_p^n$ 4: $\tilde{\mathbf{A}} := \gamma A + \hat{\Phi}_{\text{phf}}^{T_{\mathbb{S}}}(\alpha)$ // randomness corresponding to μ 5: $\tilde{\beta} := (\alpha_{n+5+1}, \dots, \alpha_{n+5+n})$ // parts in nonce A corresponding to C_{μ} 6: $\tilde{\mathbf{A}}_{\mu} := (A_1, \dots, A_n)$ 7: $\tilde{D}_1 := \gamma D_1 + 2\beta C_{\mu} - \beta G + \tau_1 V$ 8: $\tilde{D}_0 := \gamma^2 D_0 + 2\gamma \beta A_{\mu} + \beta^2 G + \tau_0 V$ 9: $\tilde{C}_{\mu} := C_{\mu} + \rho_{\mu} V$ 10: $\tilde{A}_{\text{phf}} := (\tilde{\mathbf{A}}, \tilde{C}_{\mu}, \tilde{D}_0, \tilde{D}_1)$ 11: $\tilde{\text{st}}_{\text{phf}} := (\alpha, \gamma, \rho_{\mu}, \tau_0, \tau_1)$ 12: return $(\tilde{A}_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$ </pre> Algorithm $\Sigma_{\text{phf}}.\text{BlindChal}(c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$: <pre> 1: parse $\tilde{\text{st}}_{\text{phf}} = (\alpha, \gamma, \rho_{\mu}, \tau_0, \tau_1)$ 2: return $\tilde{c}_{\text{phf}} := c_{\text{phf}} \cdot \gamma$ </pre> Algorithm $\Sigma_{\text{phf}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$: <pre> 1: parse $\tilde{\text{st}}_{\text{phf}} = (\alpha, \gamma, \rho_{\mu}, \tau_0, \tau_1)$ 2: return $c_{\text{phf}} := \tilde{c}_{\text{phf}} / \gamma$ </pre> Algorithm $\Sigma_{\text{phf}}.\text{BlindResp}(z_{\text{phf}}, c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$: <pre> 1: parse $z_{\text{phf}} = (\mathbf{z}, \mathbf{t}) \in \mathbb{Z}_p^{5+2n} \times \mathbb{Z}_p^n$ 2: parse $\tilde{\text{st}}_{\text{phf}} = (\alpha, \gamma, \rho_{\mu}, \tau_0, \tau_1)$ 3: $\tilde{c}_{\text{phf}} := \Sigma_{\text{phf}}.\text{BlindChal}(c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$ 4: $\mathbf{z}_r := (z_1, \dots, z_n)$ // response corr. to r_{μ} 5: $\tilde{\mathbf{t}} := \gamma^2 \mathbf{t} + 2\gamma \beta \mathbf{z}_r + \tilde{c}_{\text{phf}} \tau_1 + \tau_0$ 6: $\tilde{\mathbf{z}} := \gamma \mathbf{z} + \alpha + \tilde{c}_{\text{phf}} \cdot (\rho_{\mu}, \mathbf{0})$ 7: return $\tilde{z}_{\text{phf}} := (\tilde{\mathbf{z}}, \tilde{\mathbf{t}})$ </pre>
--	---

Fig. 6. Description of the Σ -protocol Σ_{phf} for relation $\mathfrak{R}_{\text{phf}}$ with challenge space $\mathcal{C} = \mathbb{Z}_p^{\times}$. Also describes randomization algorithms $\Sigma_{\text{phf}}.\text{Rand} = (\Sigma_{\text{phf}}.\text{BlindNonce}, \Sigma_{\text{phf}}.\text{BlindChal}, \Sigma_{\text{phf}}.\text{BlindChal}^{-1}, \Sigma_{\text{phf}}.\text{BlindResp})$. Above, we have $\mathbb{x}_{\text{phf}} \in \mathbb{G}^{n+6} \times \mathbb{Z}_p^{\times}$ and $\mathbb{w}_{\text{phf}} \in \mathbb{Z}_p^5 \times \{0, 1\}^n$.

Lemma 4.2. *The Σ -protocol Σ_{os} is correct, SHVZK, 2-special sound, and has relaxed unique responses for $\mathfrak{R}_{\text{os}}$. Further, it is strongly $\Sigma_{\text{os}}.\text{Rand}$ -blindable with $\Sigma_{\text{os}}.\text{Rand}$ as given in Figure 4.*

Proof. This is the canonical Σ -protocol and randomization procedure for $\mathfrak{R}_{\text{os}}$, so the statement follows, e.g., from [CART26, Lemma 2.17].

We define $\tilde{\mathfrak{R}}_{\text{phf}} = \{(\mathbb{x}, \mathbb{w}) \mid \Phi_{\text{phf}}^{T_{\mathbb{S}}}(\mathbb{w}) = \mathbf{0} \wedge \mathbb{w} \neq \mathbf{0}\} \cup \{(\mathbb{x} = (G, V), \mathbb{w}) \mid \mathbb{w}G = V\}$ for the relaxed unique responses property of Σ_{phf} . Furthermore, we define the knowledge relation $\hat{\mathfrak{R}}_{\text{phf}} = \{(\mathbb{x}, \mathbb{w}) \mid (\mathbb{x}, \mathbb{w}) \in \mathfrak{R}_{\text{phf}}\} \cup \{(\mathbb{x}, \mathbb{w}) \mid \mathbb{w}G = V\}$ for 3-special soundness of Σ_{phf} . The following lemma establishes the properties of Σ_{phf} required for our blind signature construction.

Lemma 4.3. *The Σ -protocol Σ_{phf} is correct, SHVZK, 3-special sound for $\hat{\mathfrak{R}}_{\text{phf}}$, and has relaxed unique responses for $\tilde{\mathfrak{R}}_{\text{phf}}$. Further, it is strongly $\Sigma_{\text{phf}}.\text{Rand}$ -blindable with $\Sigma_{\text{phf}}.\text{Rand}$ as given in Figure 6. Finally, it is strongly $\Sigma_{\text{phf}}.\text{Trans}$ -translatable with $\Sigma_{\text{phf}}.\text{Trans}$ as given in Figure 7.*

Proof. We show each claim below.

Algorithm $\Sigma_{\text{phf}}.\text{TransStmt}(\mathbb{x}_{\text{phf}}, \text{coins}_{\text{phf}})$:
1 : parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$; $\text{coins}_{\text{phf}} = (\nu', r'_{\S}, \rho', e')$
2 : $C'_{\S} := C_{\S} + \nu'G + r'_{\S}V$; $A'_{\S} := A_{\S} + \nu'V$
3 : $T'_{\S} := T_{\S} + \rho'V$; $e'_{\S} := e_{\S} \cdot e'$
4 : return $\mathbb{x}'_{\text{phf}} := (G, V, \mathbf{U}, C'_{\S}, A'_{\S}, T'_{\S}, e'_{\S})$
Algorithm $\Sigma_{\text{phf}}.\text{TransNonce}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, \text{coins}_{\text{phf}})$:
1 : parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$; $\text{coins}_{\text{phf}} = (\nu', r'_{\S}, \rho', e')$
2 : parse $A_{\text{phf}} = (\mathbf{A}, \mathbf{C}_{\mu}, \mathbf{D}_0, \mathbf{D}_1)$
3 : $A'_{n+4} := e' \cdot A_{n+4}$ // corresponds to nonce of non-zero PHF evaluation
4 : $\mathbf{A}' := (A_1, \dots, A_{n+3}, A'_{n+4})$
5 : return $A'_{\text{phf}} := (\mathbf{A}', \mathbf{C}_{\mu}, \mathbf{D}_0, \mathbf{D}_1)$
Algorithm $\Sigma_{\text{phf}}.\text{TransResp}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}, \text{coins}_{\text{phf}})$:
1 : parse $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$; $\text{coins}_{\text{phf}} = (\nu', r'_{\S}, \rho', e')$
2 : parse $z_{\text{phf}} = (\mathbf{z}, \mathbf{t})$
3 : parse $(z_1, \dots, z_n) = \mathbf{z}_{r_{\mu}}$; $(z_{n+5+1}, \dots, z_{n+5+n}) = \mathbf{z}_{\mu}$
4 : parse $(z_{n+1}, \dots, z_{n+5}) = (z_{\alpha}, z_{\beta}, z_{\nu}, z_{\rho}, z_{r_{\S}})$ // adapt response for (α, β) (i.e., non-zero PHF evaluation)
5 : $z'_{\alpha} := e' \cdot z_{\alpha}$; $z'_{\beta} := e' \cdot (z_{\beta} - \rho' z_{\alpha})$ // adapt response for ν, ρ and $r_{\S}$
6 : $z'_{\nu} := z_{\nu} + c_{\text{phf}} \cdot \nu'$; $z'_{\rho} := z_{\rho} + c_{\text{phf}} \cdot \rho'$; $z'_{r_{\S}} := z_{r_{\S}} + c_{\text{phf}} \cdot r'_{\S}$
7 : $\mathbf{z} := (\mathbf{z}_{r_{\mu}}, z'_{\alpha}, z'_{\beta}, z'_{\nu}, z'_{\rho}, z'_{r_{\S}}, \mathbf{z}_{\mu})$
8 : return $z'_{\text{phf}} := (\mathbf{z}', \mathbf{t})$

Fig. 7. Translation algorithms $\Sigma_{\text{phf}}.\text{Trans} = (\Sigma_{\text{phf}}.\text{TransStmt}, \Sigma_{\text{phf}}.\text{TransNonce}, \Sigma_{\text{phf}}.\text{TransResp})$ for Σ_{phf} given in Figure 6.

Correctness. We have $e_{\S} \neq 0$ by definition of $\mathfrak{R}_{\text{phf}}$. Let us show that the other assertions in verification (cf. Lines 5 and 8) pass. By linearity of $\hat{\Phi}_{\text{phf}}^{T_{\S}}$, we have

$$\mathbf{A} + c_{\text{phf}}\mathbf{Y} = \hat{\Phi}_{\text{phf}}^{T_{\S}}(\mathbf{s}) + c_{\text{phf}} \cdot \hat{\Phi}_{\text{phf}}^{T_{\S}}(\mathbf{w}) = \hat{\Phi}_{\text{phf}}^{T_{\S}}(\mathbf{s} + c_{\text{phf}}\mathbf{w}) = \hat{\Phi}_{\text{phf}}^{T_{\S}}(\mathbf{z}).$$

Further, we have

$$\begin{aligned} \mathbf{f} &= \mathbf{z}_{\mu}(\mathbf{z}_{\mu} - c_{\text{phf}} \cdot \mathbf{1}) \\ &= (c_{\text{phf}} \cdot \boldsymbol{\mu} + \mathbf{r})(c_{\text{phf}} \cdot \boldsymbol{\mu} + \mathbf{r} - c_{\text{phf}} \cdot \mathbf{1}) \\ &= c_{\text{phf}}^2(\boldsymbol{\mu}^2 - \boldsymbol{\mu}) + c_{\text{phf}}(2\boldsymbol{\mu}\mathbf{r} - \mathbf{r}) + \mathbf{r}^2 \end{aligned}$$

As $\boldsymbol{\mu}^2 - \boldsymbol{\mu} = 0$ due to $\boldsymbol{\mu} \in \{0, 1\}^n$, it follows that

$$c_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 = \mathbf{f}G + \mathbf{t}V$$

by construction of $\mathbf{D}_0, \mathbf{D}_1$ and $\mathbf{t}$.

SHVZK. On input $\mathbb{x}_{\text{phf}}$ and c_{phf} , the simulator samples $z_{\text{phf}} = (\mathbf{z}, \mathbf{t}) \leftarrow_{\S} \mathbb{Z}_p^{5+2n} \times \mathbb{Z}_p^n$ and $\mathbf{C}_{\mu}, \mathbf{D}_1 \leftarrow_{\S} \mathbb{G}^n$. Let $\mathbf{Y} = (\mathbf{C}_{\mu}, C_{\S}, A_{\S}, T_{\S} - U_0, e_{\S}G)$ as in verification. The simulator sets $\mathbf{A} = \hat{\Phi}_{\text{phf}}^{T_{\S}}(\mathbf{z}) - c_{\text{phf}}\mathbf{Y}$ and $\mathbf{D}_0 = \mathbf{f}G + \mathbf{t}V - c_{\text{phf}}\mathbf{D}_1$, where $\mathbf{f}$ is computed as in verification from $\mathbf{z}$. Then, it returns the nonce $A_{\text{phf}} = (\mathbf{A}, \mathbf{C}_{\mu}, \mathbf{D}_0, \mathbf{D}_1)$ and response z_{phf} . It is easy to see that the distribution is identical to an honestly computed transcript for statement $\mathbb{x}$ and challenge c_{phf} .

3-Special Soundness. Given three valid transcripts $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$, $(A_{\text{phf}}, c'_{\text{phf}}, z'_{\text{phf}})$ and $(A_{\text{phf}}, c^*_{\text{phf}}, z^*_{\text{phf}})$ for $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$ with pairwise distinct challenges, the extractor proceeds as follows. Let $\mathbf{w}' = (z - z')/(c_{\text{phf}} - c'_{\text{phf}})$ and $\mathbf{w}^* = (z - z^*)/(c_{\text{phf}} - c^*_{\text{phf}})$. Due to Line 5 in verification, it holds that $\hat{\Phi}^{T_{\S}}(\mathbf{w}') = \mathbf{Y} = \hat{\Phi}^{T_{\S}}(\mathbf{w}^*)$. Denote by $\boldsymbol{\mu}', \boldsymbol{\mu}^*, \mathbf{z}_{\mu}, \mathbf{z}'_{\mu}, \mathbf{z}^*_{\mu}$ the last n components of $\mathbf{w}', \mathbf{w}^*, \mathbf{z}, \mathbf{z}', \mathbf{z}^*$, respectively. If $\boldsymbol{\mu}' \neq \boldsymbol{\mu}^*$, then the extractor can compute the DL of V by definition of $\hat{\Phi}^{T_{\S}}$ (cf. Equation (6)), so the extractor succeeds. Otherwise, we have $\mathbb{x}_{\text{phf}} \in \mathfrak{L}_{\text{phf}}$ if $\boldsymbol{\mu} \in \{0, 1\}^n$. The latter follows by Line 8 in verification (unless the DL of V can be computed). To see this, denote $\boldsymbol{\beta} = \mathbf{z}_{\mu} - c_{\text{phf}}\boldsymbol{\mu}$, $\boldsymbol{\beta}' = \mathbf{z}'_{\mu} - c'_{\text{phf}}\boldsymbol{\mu}$ and $\boldsymbol{\beta}^* = \mathbf{z}^*_{\mu} - c^*_{\text{phf}}\boldsymbol{\mu}$. We have $\boldsymbol{\beta} = \boldsymbol{\beta}'$, since

$$\begin{aligned}\boldsymbol{\beta}' &= \mathbf{z}'_{\mu} - c'_{\text{phf}}\boldsymbol{\mu} = \mathbf{z}'_{\mu} - \mathbf{z}_{\mu} + \mathbf{z}_{\mu} - c'_{\text{phf}}\boldsymbol{\mu} \\ &= \boldsymbol{\mu}(c'_{\text{phf}} - c_{\text{phf}}) + \mathbf{z}_{\mu} - c'_{\text{phf}}\boldsymbol{\mu} = \mathbf{z}_{\mu} - c_{\text{phf}}\boldsymbol{\mu} = \boldsymbol{\beta},\end{aligned}$$

and a similar calculation shows $\boldsymbol{\beta} = \boldsymbol{\beta}^*$. Due to Line 8, we have

$$\begin{aligned}c_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 &= \mathbf{f}G + \mathbf{t}V, & \text{where } \mathbf{f} &= \mathbf{z}_{\mu}(\mathbf{z}_{\mu} - c_{\text{phf}} \cdot \mathbf{1}), \\ c'_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 &= \mathbf{f}'G + \mathbf{t}'V, & \text{where } \mathbf{f}' &= \mathbf{z}'_{\mu}(\mathbf{z}'_{\mu} - c'_{\text{phf}} \cdot \mathbf{1}), \\ c^*_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 &= \mathbf{f}^*G + \mathbf{t}^*V, & \text{where } \mathbf{f}^* &= \mathbf{z}^*_{\mu}(\mathbf{z}^*_{\mu} - c^*_{\text{phf}} \cdot \mathbf{1}),\end{aligned}$$

and thus

$$\begin{aligned}\mathbf{D}_1 &= (\mathbf{f} - \mathbf{f}')/(c_{\text{phf}} - c'_{\text{phf}})G + (\mathbf{t} - \mathbf{t}')/(c_{\text{phf}} - c'_{\text{phf}})V, \\ \mathbf{D}_1 &= (\mathbf{f} - \mathbf{f}^*)/(c_{\text{phf}} - c^*_{\text{phf}})G + (\mathbf{t} - \mathbf{t}^*)/(c_{\text{phf}} - c^*_{\text{phf}})V.\end{aligned}$$

Thus, we have $(\mathbf{f} - \mathbf{f}')/(c_{\text{phf}} - c'_{\text{phf}}) = (\mathbf{f} - \mathbf{f}^*)/(c_{\text{phf}} - c^*_{\text{phf}})$, else the above yields the DL of V (which the extractor outputs if so). Note that since $\boldsymbol{\beta} = \boldsymbol{\beta}' = \boldsymbol{\beta}^*$, we can write $\mathbf{f} = \mathbf{F}(c_{\text{phf}})$, $\mathbf{f}' = \mathbf{F}(c'_{\text{phf}})$, and $\mathbf{f}^* = \mathbf{F}(c^*_{\text{phf}})$, where $\mathbf{F}(X) = \varphi_2 X^2 + \varphi_1 X + \varphi_0$, where $\varphi_2 = \boldsymbol{\mu}(\boldsymbol{\mu} - 1)$, $\varphi_1 = 2\boldsymbol{\mu}\boldsymbol{\beta} - \boldsymbol{\beta}$ and $\varphi_0 = \boldsymbol{\beta}^2$.¹⁴ We obtain

$$\begin{aligned}(\mathbf{f} - \mathbf{f}')/(c_{\text{phf}} - c'_{\text{phf}}) &= (c_{\text{phf}} + c'_{\text{phf}})\varphi_2 + \varphi_1, \\ (\mathbf{f} - \mathbf{f}^*)/(c_{\text{phf}} - c^*_{\text{phf}}) &= (c_{\text{phf}} + c^*_{\text{phf}})\varphi_2 + \varphi_1 \\ \implies (c_{\text{phf}} + c'_{\text{phf}})\varphi_2 &= (c_{\text{phf}} + c^*_{\text{phf}})\varphi_2 \\ \implies (c'_{\text{phf}} - c^*_{\text{phf}})\varphi_2 &= \mathbf{0} \\ \implies \varphi_2 &= \mathbf{0}.\end{aligned}$$

In particular, we have $\boldsymbol{\mu}(\boldsymbol{\mu} - 1) = \mathbf{0}$ which means $\boldsymbol{\mu} \in \{0, 1\}^n$ as desired.

Relaxed-Unique Responses. Given nonce $A_{\text{phf}} = (\mathbf{A}, \mathbf{C}_{\mu}, \mathbf{D}_0, \mathbf{D}_1)$, challenge $c \in \mathbb{Z}_p^{\times}$ and responses $\mathbf{z}_{\text{phf},0} = (z_0, \mathbf{t}_0)$, $\mathbf{z}_{\text{phf},1} = (z_1, \mathbf{t}_1)$ such that $(A_{\text{phf}}, c_{\text{phf}}, \mathbf{z}_{\text{phf},b})$ is valid for $b \in \{0, 1\}$ and $\mathbf{z}_{\text{phf},0} \neq \mathbf{z}_{\text{phf},1}$, we can compute the DL v of V if $\mathbf{t}_0 \neq \mathbf{t}_1$ as $\mathbf{f}_0 G + \mathbf{t}_0 V = \mathbf{f}_1 G + \mathbf{t}_1 V$. Else, if $z_0 \neq z_1$, we obtain $\hat{\Phi}^{T_{\S}}_{\text{phf}}(z_0 - z_1) = 0$. Parse $\mathbf{z}_b = (\hat{z}_b, \mathbf{z}'_b)$ with $\hat{z}_b \in \mathbb{Z}_p^n$ and $\mathbf{z}'_b \in \mathbb{Z}_p^{5+n}$. If $\hat{z}_0 \neq \hat{z}_1$, then we can compute the DL v of V , similar to above. Else, we have $\Phi^{T_{\S}}_{\text{phf}}(z'_0 - z'_1) = 0$. The above considerations yield an extractor.

Blindability. Observe that $\Sigma_{\text{phf}}.\text{BlindChal}^{-1}(\cdot, \tilde{\text{st}})$ is the inverse of $\Sigma_{\text{phf}}.\text{BlindChal}(\cdot, \tilde{\text{st}})$. Next, let $(A_{\text{phf}}, c_{\text{phf}}, \mathbf{z}_{\text{phf}})$ be a valid transcript for $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, A_{\S}, T_{\S}, e_{\S})$, where $A_{\text{phf}} = (\mathbf{A}, \mathbf{C}_{\mu}, \mathbf{D}_0, \mathbf{D}_1)$ and $\mathbf{z}_{\text{phf}} = (\mathbf{z}, \mathbf{t})$. First, we show that the blinded transcript $(\tilde{A}_{\text{phf}}, \tilde{c}_{\text{phf}}, \tilde{\mathbf{z}}_{\text{phf}})$ remains valid, following the notation in Figure 6.

¹⁴ This follows as in the derivation of $\mathbf{f}$ in correctness.

Let $\tilde{Y} = (\tilde{C}_\mu, C_\$, A_\$, T_\$ - U_0, e_\$G)$. We show that the assertion in Line 5 passes; we have

$$\begin{aligned}\tilde{A} + \tilde{c}_{\text{phf}}\tilde{Y} &= \gamma\mathbf{A} + \hat{\Phi}_{\text{phf}}^{T_\$}(\alpha) + (\gamma \cdot c_{\text{phf}})\mathbf{Y} + \tilde{c}_{\text{phf}}(\rho_\mu V, \mathbf{0}) \\ &= \gamma\hat{\Phi}_{\text{phf}}^{T_\$}(\mathbf{z}) + \hat{\Phi}_{\text{phf}}^{T_\$}(\alpha) + \tilde{c}_{\text{phf}} \cdot \hat{\Phi}_{\text{phf}}^{T_\$}(\rho_\mu, \mathbf{0}) \\ &= \hat{\Phi}_{\text{phf}}^{T_\$}(\gamma\mathbf{z} + \alpha + \tilde{c}_{\text{phf}} \cdot (\rho_\mu, \mathbf{0})) \\ &= \hat{\Phi}_{\text{phf}}^{T_\$}(\tilde{\mathbf{z}}) .\end{aligned}$$

Next, we show that $\tilde{c}_{\text{phf}}\tilde{D}_1 + \tilde{D}_0 = \tilde{\mathbf{f}}G + \tilde{\mathbf{t}}V$ (cf. Line 8), where $\tilde{\mathbf{f}} = \tilde{z}_\mu(\tilde{z}_\mu - \tilde{c}_{\text{phf}} \cdot \mathbf{1})$. Note that $\tilde{z}_\mu = \gamma\mathbf{z}_\mu + \beta$. We have

$$\begin{aligned}\tilde{\mathbf{f}} &= (\gamma\mathbf{z}_\mu + \beta)(\gamma\mathbf{z}_\mu + \beta - \gamma c_{\text{phf}}\mathbf{1}) \\ &= \gamma^2\mathbf{z}_\mu(\mathbf{z}_\mu - c_{\text{phf}}\mathbf{1}) + 2\gamma\mathbf{z}_\mu\beta - \gamma c_{\text{phf}}\beta + \beta^2 \\ &= \gamma^2\mathbf{f} + 2\gamma\mathbf{z}_\mu\beta - \tilde{c}_{\text{phf}}\beta + \beta^2 .\end{aligned}$$

As it holds that $c_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 = \mathbf{f}G + \mathbf{t}V$ and $\tilde{c}_{\text{phf}} = c_{\text{phf}} \cdot \gamma$, we obtain

$$\tilde{\mathbf{f}}G + \tilde{\mathbf{t}}V = \tilde{c}_{\text{phf}} \cdot \gamma\mathbf{D}_1 + \gamma^2\mathbf{D}_0 - \gamma^2\mathbf{t}V + (2\gamma\mathbf{z}_\mu\beta - \tilde{c}_{\text{phf}}\beta + \beta^2)G + \tilde{\mathbf{t}}V . \quad (8)$$

At this point, the extension of Φ_{phf} to $\hat{\Phi}^{T_\$}$ is crucial: the response $\mathbf{z}_\mu$ is not yet known when $\tilde{D}_0$ and $\tilde{D}_1$ are setup in BlindNonce; we must express this term via values known in BlindNonce already.¹⁵ Due to the extension, we have

$$c_{\text{phf}}\mathbf{C}_\mu + \mathbf{A}_\mu = \mathbf{z}_\mu G + \mathbf{z}_r V ,$$

where $\mathbf{z}_\mu = (z_{n+6}, \dots, z_{2n+5})$ and $\mathbf{z}_r = (z_1, \dots, z_n)$. Therefore, we obtain

$$\tilde{c}_{\text{phf}} \cdot 2\beta\mathbf{C}_\mu + \gamma 2\beta\mathbf{A}_\mu = \gamma \cdot 2\beta(c_{\text{phf}}\mathbf{C}_\mu + \mathbf{A}_\mu) = \gamma \cdot 2\beta(\mathbf{z}_\mu G + \mathbf{z}_r V) ,$$

Plugging this into Equation (8), we obtain

$$\begin{aligned}\tilde{\mathbf{f}}G + \tilde{\mathbf{t}}V &= \tilde{c}_{\text{phf}} \cdot (\gamma\mathbf{D}_1 - \beta G + 2\beta\mathbf{C}_\mu) + \\ &\quad \gamma^2\mathbf{D}_0 + (\tilde{\mathbf{t}} - \gamma^2\mathbf{t})V + \beta^2 G + \gamma 2\beta\mathbf{A}_\mu - \gamma 2\beta\mathbf{z}_r V \\ &= \tilde{c}_{\text{phf}} \cdot (\tilde{D}_1 - \tau_1 V) + \\ &\quad \tilde{D}_0 - \tau_0 V + (\tilde{\mathbf{t}} - \gamma^2\mathbf{t})V - 2\gamma\beta\mathbf{z}_r V\end{aligned}$$

Since $\tilde{\mathbf{t}} = \gamma^2\mathbf{t} + 2\gamma\beta\mathbf{z}_r + \tilde{c}\tau_1 + \tau_0$, the assertion in Line 8 passes. Finally, observe that the blinded challenge $\tilde{c}_{\text{phf}}$ is uniform over $\mathbb{Z}_p^\times$ (randomized by $\gamma \leftarrow \mathbb{Z}_p^\times$), the response $z_{\text{phf}} = (\tilde{z}, \tilde{\mathbf{t}})$ is uniform over $\mathbb{Z}_p^{5+2n} \times \mathbb{Z}_p^n$ (randomized by (α, τ_0)), $\mathbf{D}_1$ is uniform over $\mathbb{G}^n$ (randomized by τ_1), and $\mathbf{C}_\mu$ is random over $\mathbb{G}^n$ (randomized by ρ_μ). The remaining values are determined by verification. This is exactly the simulated distribution which concludes the proof.

Translatability. Let $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ be a valid transcript for $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_\$, A_\$, T_\$, e_\$)$, where $A_{\text{phf}} = (\mathbf{A}, \mathbf{C}_\mu, \mathbf{D}_0, \mathbf{D}_1)$ and $z_{\text{phf}} = (\mathbf{z}, \mathbf{t})$. We show that the translated transcript $(A'_{\text{phf}}, c_{\text{phf}}, z'_{\text{phf}})$ still verifies for the translated statement $\tilde{\mathbb{x}}_{\text{phf}} = \Sigma_{\text{phf}}.\text{TransStmt}(\mathbb{x}_{\text{phf}}, \text{coins}_{\text{phf}})$ with $\text{coins}_{\text{phf}} = (\nu', r'_\$, \rho', e') \in \mathbb{Z}_p^3 \times \mathbb{Z}_p^\times$. Denote $\tilde{\mathbb{x}}_{\text{phf}} = (G, V, \mathbf{U}, \tilde{C}_\$, \tilde{A}_\$, \tilde{T}_\$, \tilde{e}_\$)$ and

$$\begin{aligned}A'_{\text{phf}} &= (\mathbf{A}', \mathbf{C}_\mu, \mathbf{D}_0, \mathbf{D}_1) , \\ z'_{\text{phf}} &= (\mathbf{z}', \mathbf{t}) , \\ \mathbf{z}' &= (z_{r_\mu}, z'_{\alpha}, z'_{\beta}, z'_{\nu}, z'_{\rho}, z'_{r_\$}, \mathbf{z}_\mu) ,\end{aligned}$$

¹⁵ Note that the evaluator of BlindNonce does not have access to the witness. Therefore, we cannot simply recompute the response $\mathbf{z}_r$ at this point.

as in Figure 7. Observe that the values $\mathbf{z}_\mu, c_{\text{phf}}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{t}$ remain unchanged. As $\mathbf{f}$ is computed from $\mathbf{z}_\mu$ and c_{phf} , we have $c_{\text{phf}}\mathbf{D}_1 + \mathbf{D}_0 = \mathbf{f}G + \mathbf{t}V$ as $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ is valid. Let $\mathbf{Y}' = (\mathbf{C}_\mu, \tilde{C}_\$, \tilde{A}_\$, \tilde{T}_\$ - U_0, \tilde{e}_\$G)$ with $\tilde{C}_\$ = C_\$ + \nu'G + r'_\$V$, $\tilde{A}_\$ = A_\$ + \nu'V$, $\tilde{T}_\$ = T_\$ + \rho'V$ and $\tilde{e}_\$ = e_\$ \cdot e'$. By verification and $e \neq 0$, we have $\tilde{e}_\$ \neq 0$. Further, we have

$$\mathbf{A}' + c_{\text{phf}}\mathbf{Y}' = \begin{bmatrix} (A_1, \dots, A_n)^\top \\ A_{n+1} \\ A_{n+2} \\ A_{n+3} \\ e'A_{n+4} \end{bmatrix}^\top + c_{\text{phf}} \left(\begin{bmatrix} \mathbf{C}_\mu^\top \\ C_\$ \\ A_\$ \\ T_\$ - U_0 \\ e' \cdot e_\$ \cdot G \end{bmatrix}^\top + \begin{bmatrix} \mathbf{0}^\top \\ \nu'G + r'_\$V \\ \nu'V \\ \rho'V \\ 0 \end{bmatrix}^\top \right). \quad (9)$$

Note that since $T_\$$ is adapted, the linear map $\Phi_{\text{phf}}^{T_\$}$ in verification also changes to $\Phi_{\text{phf}}^{\tilde{T}_\$}$. Let us first inspect the last component, since only this component is affected by this change. Using that $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ is valid, we obtain the desired

$$\begin{aligned} e'(A_{n+4} + c_{\text{phf}} \cdot e_\$G) &= e'(z_\alpha \cdot T_\$ + z_\beta \cdot V) \\ &= e' \cdot z_\alpha \cdot (T_\$ + \rho'V - \rho'V) + e' \cdot z_\beta \cdot V \\ &= e'z_\alpha \tilde{T}_\$ - e' \cdot z_\alpha \cdot \rho'V + e' \cdot z_\beta \cdot V \\ &= z'_\alpha \tilde{T}_\$ + z'_\beta V. \end{aligned} \quad (10)$$

Let us inspect the remaining components in Equation (9). For this, denote by $\bar{\Phi}_{\text{phf}}$ the map $\hat{\Phi}_{\text{phf}}^{\tilde{T}_\$}$ restricted to all but the last component. Note that $\bar{\Phi}_{\text{phf}}$ does not depend on $\tilde{T}_\$$. We have

$$\begin{aligned} &\begin{bmatrix} (A_1, \dots, A_n)^\top \\ A_{n+1} \\ A_{n+2} \\ A_{n+3} \end{bmatrix}^\top + c_{\text{phf}} \left(\begin{bmatrix} \mathbf{C}_\mu^\top \\ C_\$ \\ A_\$ \\ T_\$ - U_0 \end{bmatrix}^\top + \begin{bmatrix} \mathbf{0}^\top \\ \nu'G + r'_\$V \\ \nu'V \\ \rho'V \end{bmatrix}^\top \right) \\ &= \bar{\Phi}_{\text{phf}}(\mathbf{z}) + c_{\text{phf}} \cdot (\mathbf{0}, \nu'G + r'_\$V, \nu'V, \rho'V) \\ &= \bar{\Phi}_{\text{phf}}(\mathbf{z}) + c_{\text{phf}} \cdot \bar{\Phi}_{\text{phf}}(\mathbf{0}, 0, 0, \nu', \rho', r'_\$, \mathbf{0}) \\ &= \bar{\Phi}_{\text{phf}}(\mathbf{z}'), \end{aligned}$$

where the last equation follows by linearity of $\bar{\Phi}_{\text{phf}}$ and construction of $\mathbf{z}'$. Altogether, we obtain

$$\mathbf{A}' + c_{\text{phf}}\mathbf{Y}' = \hat{\Phi}_{\text{phf}}^{\tilde{T}_\$}(\mathbf{z}'). \quad \square$$

Finally, we need the following lemma to compile Σ_{dl} and Σ_{phf} into an OR-proof. It follows by construction (due to multiplicative blinding of Σ_{dl} and Σ_{phf}) and we omit the proof.

Lemma 4.4 (Σ_{dl} and Σ_{phf} compatibility). *Let $c, c_{\text{dl}}, c_{\text{phf}} \in \mathbb{Z}_p^\times$ such that $c = c_{\text{dl}} \cdot c_{\text{phf}}$. For all $\tilde{\mathbf{st}}_{\text{dl}}$ and $\tilde{\mathbf{st}}_{\text{phf}}$ in the range of (the second component of) $\Sigma_{\text{dl}}.\text{BlindNonce}$ and $\Sigma_{\text{phf}}.\text{BlindNonce}$, respectively, we have*

$$\Sigma_{\text{dl}}.\text{BlindChal}(\Sigma_{\text{phf}}.\text{BlindChal}(c, \tilde{\mathbf{st}}_{\text{phf}}), \tilde{\mathbf{st}}_{\text{dl}}) = \tilde{c}_{\text{dl}} \cdot \tilde{c}_{\text{phf}},$$

where $\tilde{c}_{\text{dl}} = \Sigma_{\text{dl}}.\text{BlindChal}(c_{\text{dl}}, \tilde{\mathbf{st}}_{\text{dl}})$ and $\tilde{c}_{\text{phf}} = \text{BlindChal}_{\text{phf}}(c_{\text{phf}}, \tilde{\mathbf{st}}_{\text{phf}})$.

5 Blind Signature from DL

We present our blind signature BS_{phf} and prove its security under the DL assumption. First, let us define the NIZK Π_{sle} which allows to prove knowledge of the blinding factors for Σ_{os} .

Algorithm $\Sigma_{\text{sle}}.\text{Init}(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}})$:	Algorithm $\Sigma_{\text{sle}}.\text{Resp}(\text{st}_{\text{sle}}, c_{\text{sle}})$:
1 : parse $\mathbb{x}_{\text{sle}} = (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}})$	1 : assert $c_{\text{sle}} \in \mathcal{C}_{\text{sle}}$
2 : parse $\mathbb{w}_{\text{sle}} = (\nu, r_{\S}, \alpha, \gamma, r_{\text{rnd}})$	2 : parse $\text{st}_{\text{sle}} = (\mathbf{w}, \mathbf{r})$
3 : $\mathbf{w} := \mathbb{w}_{\text{sle}} ; \mathbf{r} \leftarrow \mathbb{Z}_p^6$	3 : return $\mathbf{z}_{\text{sle}} := c_{\text{sle}} \cdot \mathbf{w} + \mathbf{r}$
4 : $A_{\text{sle}} := \Phi_{\text{sle}}(\mathbf{r})$	Algorithm $\Sigma_{\text{sle}}.\text{Ver}(\mathbb{x}_{\text{sle}}, A_{\text{sle}}, c_{\text{sle}}, \mathbf{z}_{\text{sle}})$:
5 : $\text{st}_{\text{sle}} := (\mathbf{w}, \mathbf{r})$	1 : parse $\mathbb{x}_{\text{sle}} = (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}})$
6 : return $(A_{\text{sle}}, \text{st}_{\text{sle}})$	2 : $\mathbf{T} := (C_{\S}, C_{\text{rnd}}, \tilde{A}_{\text{os}} - \tilde{A}_{\S})$
	3 : assert $A_{\text{sle}} + c_{\text{sle}} \cdot \mathbf{T} = \Phi_{\text{sle}}(\mathbf{z}_{\text{sle}})$
	4 : return 1

Fig. 8. The Σ -protocol for relation $\mathfrak{R}_{\text{sle}}$ with challenge space $\mathcal{C}_{\text{sle}} = \mathbb{Z}_p$.

5.1 Straightline-extractable NIZK

Let $\text{ck}_{\text{ped}} = (G_1, G_2, G_3, V) \in \mathbb{G}^4$ and $\mathbb{x}_{\text{os}} = (G, H, W) \in \mathbb{G}^3$. Further, let $(C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}}) \in \mathbb{G}^4$. We require a straightline-extractable SLE NIZK Π_{sle} for relation $\mathfrak{R}_{\text{sle}}$ defined as

$$\mathfrak{R}_{\text{sle}} := \left\{ (\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \left| \begin{array}{ll} C_{\S} = \nu G + r_{\S} V & \wedge \quad C_{\text{rnd}} = \text{C}_{\text{ped}}.\text{Commit}(\text{ck}_{\text{ped}}, \tilde{\text{st}}; r_{\text{rnd}}) \\ \tilde{A}_{\S} = A_{\text{os}} + \nu V & \wedge \quad \tilde{A}_{\text{os}} = \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}) \end{array} \right. \right\}, \quad (11)$$

where $\mathbb{x}_{\text{sle}} = (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}})$, $\mathbb{w}_{\text{sle}} = (\nu, r_{\S}, \tilde{\text{st}} = (\alpha, \gamma), r_{\text{rnd}}) \in \mathbb{Z}_p^6$.

Looking ahead, we will instantiate the scheme Π_{sle} with a Σ -protocol compiled via the randomized Fischlin transform [Fis05, Ks22, KRW24]. The NIZK Π_{sle} is correct, zero-knowledge, and straightline extractable for relation $\hat{\mathfrak{R}}_{\text{sle}} = \mathfrak{R}_{\text{sle}} \cup \mathfrak{R}_{\text{zero}}$, where the latter is defined according to Φ_{sle} (defined in Equation (13)) as:

$$\mathfrak{R}_{\text{zero}} := \{ (\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \mid \Phi_{\text{sle}}(\mathbb{w}_{\text{sle}}) = \mathbf{0} \wedge \mathbb{w}_{\text{sle}} \neq \mathbf{0} \}. \quad (12)$$

Instantiation. First, let us define the linear map $\Phi_{\text{sle}}^{\text{lepp}}$ parameterized by $\text{lepp} = (\text{ck}_{\text{ped}} = (G_1, G_2, G_3, V), \mathbb{x}_{\text{os}} = (G, H, W))$ as

$$(C_{\S}, C_{\text{rnd}}, \tilde{A}_{\text{os}} - \tilde{A}_{\S}) = \Phi_{\text{sle}}^{\text{lepp}}(\mathbb{w}_{\text{sle}}) := \begin{bmatrix} \nu G + r_{\S} V \\ r_{\text{rnd}} V + \alpha_1 G_1 + \alpha_2 G_2 + \gamma G_3 \\ \alpha_1 G + \alpha_2 H - \gamma W - \nu V \end{bmatrix}^{\text{T}}, \quad (13)$$

where $\mathbb{w}_{\text{sle}} = (\nu, r_{\S}, \tilde{\text{st}} = (\alpha, \gamma), r_{\text{rnd}}) \in \mathbb{Z}_p^6$. If clear by context, we may omit lepp and write Φ_{sle} for short. The Σ -protocol Σ_{sle} for $\mathfrak{R}_{\text{sle}}$ is given in Figure 8. Note that it is a canonical Σ -protocol for the linear map Φ_{sle} .

Lemma 5.1. *The Σ -protocol Σ_{sle} for relation $\mathfrak{R}_{\text{sle}}$ is correct, SHVZK, 2-special sound, has high min-entropy, and has relaxed strong 2-special soundness for $\mathfrak{R}_{\text{sle}}$.*

Proof. Note that if $(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \in \mathfrak{R}_{\text{sle}}$ (cf. Equation (11)), then

$$\begin{aligned} \alpha_1 G + \alpha_2 H - \gamma W - \nu V &= -(A_{\text{os}} + \nu V) + A_{\text{os}} + \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\alpha) - \gamma W \\ &= -\tilde{A}_{\S} + \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}) \\ &= \tilde{A}_{\text{os}} - \tilde{A}_{\S}. \end{aligned}$$

Thus, $\Phi_{\text{sle}}(\mathbb{w}_{\text{sle}}) = (C_{\S}, C_{\text{rnd}}, \tilde{A}_{\text{os}} - \tilde{A}_{\S})$ by construction. Similarly, if $\Phi_{\text{sle}}(\mathbb{w}_{\text{sle}}) = (C_{\S}, C_{\text{rnd}}, \tilde{A}_{\text{os}} - \tilde{A}_{\S})$, then let $A_{\text{os}} = \tilde{A}_{\S} - \nu V$. We have

$$\Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}) = \tilde{A}_{\S} - \nu V + \alpha_1 G + \alpha_2 H - \gamma W = \tilde{A}_{\text{os}},$$

and therefore $(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \in \mathfrak{R}_{\text{sle}}$. As Σ_{sle} is the canonical Σ -protocol for Φ_{sle} , the correctness, SHVZK, 2-special soundness properties follow, e.g., from [CART26, Lemma 2.17].

Let us show strong 2-special soundness. Assume we are given two valid Σ_{sle} transcripts $\tau_{\text{sle}} = (A_{\text{sle}}, c_{\text{sle}}, z_{\text{sle}})$ and $\tau'_{\text{sle}} = (A'_{\text{sle}}, c'_{\text{sle}}, z'_{\text{sle}})$ such that $A_{\text{sle}} = A'_{\text{sle}}$ and $\tau_{\text{sle}} \neq \tau'_{\text{sle}}$. If $c_{\text{sle}} \neq c'_{\text{sle}}$, then we can extract a witness $\mathbb{w}_{\text{sle}}$ such that $(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \in \mathfrak{R}_{\text{sle}}$ via the 2-special soundness extractor. Else, we have $c_{\text{sle}} = c'_{\text{sle}}$ and $z_{\text{sle}} \neq z'_{\text{sle}}$ and we obtain by the verification equation of Σ_{sle} that

$$\Phi_{\text{sle}}(z_{\text{sle}}) = \Phi_{\text{sle}}(z'_{\text{sle}}) \implies \Phi_{\text{sle}}(z_{\text{sle}} - z'_{\text{sle}}) = 0.$$

Since $z_{\text{sle}} - z'_{\text{sle}}$ is non-zero, the difference $z_{\text{sle}} - z'_{\text{sle}}$ is a witness for $\mathfrak{R}_{\text{zero}}$. The above considerations yield the desired extractor. $\square$

We instantiate Π_{sle} via the randomized Fischlin transform [Fis05, Ks22] applied to the Σ -protocol Σ_{sle} . We recall some additional preliminaries. For some Σ -protocol Σ , denote by $\text{RFisch}_{k,b,r,t}[\Sigma]$ the randomized Fischlin variant given in [KRW24, Appendix C], parameterized by $k \in \mathbb{R}$ and $b, r, t \in \mathbb{N}$. We recall the following lemma, which summarizes [KRW24, Lemmas 3, 4, 5].

Lemma 5.2. *Let Σ be a Σ -protocol for relation $\mathfrak{R}$ with challenge space $\mathcal{C}$ that is correct, SHVZK, has high min-entropy and relaxed strong 2-special soundness for relation $\hat{\mathfrak{R}} \supseteq \mathfrak{R}$. Let $k > 0$, and $b, r, t \in \mathbb{N}$. Suppose $t \leq k$ and $N = 2^k = |\mathcal{C}| \geq 2^4$. Let $\Pi = \text{RFisch}_{k,b,r,t}^H[\Sigma]$ be the randomized Fischlin transformation applied to Σ with random oracle $H : \{0, 1\}^* \rightarrow \{0, 1\}^b$. Then:*

- The NIZK Π is correct with error at most $\varepsilon \leq r \cdot e^{-2^{t-b}}$ and an honest prover runs in time $\text{poly}(2^t)$.
- There exists an extractor Ext such that for any (potentially unbounded) adversary $\mathcal{A}$ on relaxed straightline extractability for relation $\hat{\mathfrak{R}}$ inducing at most Q queries to H ,

$$\text{Adv}_{\Pi, \text{Ext}, \hat{\mathfrak{R}}}^{\text{sle}}(\mathcal{A}, \lambda) \leq Q \cdot 2^{-r \cdot b}.$$

In particular, if $Q = \text{poly}(\lambda)$ and $2^{-r \cdot b} = \text{negl}(\lambda)$, then $\text{Adv}_{\Pi, \hat{\mathfrak{R}}}^{\text{sle}}(\mathcal{A}, \lambda) = \text{negl}(\lambda)$.

- There exists a simulator Sim such that for any adversary $\mathcal{A}$ which makes at most Q_H and Q_P queries to H and $\mathcal{O}_{\text{Prove}}$, respectively, we have

$$\text{Adv}_{\Pi, \text{Sim}}^{\text{zk}}(\mathcal{A}, \lambda) \leq (Q_H + Q_P) \cdot Q_P \cdot 2^{-H_{\min}(\Sigma)} + 3 \cdot Q_P \cdot r \cdot 2^{-(k-b)/2}.$$

Proof. Correctness and SLE follow as in [Ks22]. Zero-knowledge follows as in [KRW24, Appendix C]. We note that [KRW24] omits a formal definition of its zero-knowledge property. We briefly explain why their proof implies our definition. In each call to the oracle $\mathcal{O}_{\text{Sim}}$ with $(\mathbb{x}, \mathbb{w}) \in \mathfrak{R}$, the simulator programs $H(\mathbb{x}, \mathbf{A}, i, c_i, z_i) := 0^b$ for $i \in [r]$, where (A_i, c_i, z_i) are simulated via the SHVZK property of Σ . This makes it possible to generate simulated proofs $\tilde{\pi}$ that follow (exactly) the distribution of honestly-generated proofs π , but biases the distribution of H outputs towards 0^b . To conclude, the authors analyze (i) the probability that the reprogramming is noticeable and (ii) the statistical distance between a random oracle and its programmed variant.

For (i), by the min-entropy property of Σ , any input of the form $s = (\mathbb{x}, \mathbf{A}, \cdot)$ is queried with probability at most $2^{-H_{\min}(\Sigma)}$. A union bound over (1) all hash queries and all prior $\mathcal{O}_{\text{Prove}}$ queries, and (2) all queries to $\mathcal{O}_{\text{Prove}}$ yields that programming is noticeable with probability at most $(Q_H + Q_P) \cdot Q_P \cdot 2^{-H_{\min}(\Sigma)}$.

For (ii), the authors show that the statistical distance between a random oracle $H' : \mathcal{C} \rightarrow \{0, 1\}^b$ and its programmed variant $H'[x \mapsto 0^b]$ on input $x = (\mathbb{x}, \mathbf{A}, i, c_i, z_i)$ for a random $c_i \leftarrow_s \mathcal{C}$ is at most $3 \cdot 2^{-(k-b)/2}$. Here, $H'[x \mapsto 0^b]$ denotes the random oracle obtained by reprogramming H' at input x with output 0^b . Again, a union bound over all repetitions r and all calls to $\mathcal{O}_{\text{Sim}}$ yields that the distance is at most $3 \cdot Q_P \cdot r \cdot 2^{-(k-b)/2}$.

The bounds from (i) and (ii) together imply the desired result. $\square$

We set $\Pi_{\text{sle}} = \text{RFisch}_{k,b,r,t}[\Sigma_{\text{sle}}]$ with a choice of parameters (k, b, r, t) that satisfies the requirements in Lemma 5.2. The desired properties follow from Lemmas 5.1 and 5.2.

Algorithm $\text{BS}_{\text{phf}}.\text{Setup}(1^\lambda)$: 1 : $G_1, G_2, G_3, H \leftarrow \mathbb{G}$; $V \leftarrow \mathbb{G} \setminus \{0\}$ 2 : $n := \lceil \log_2 p \rceil$ 3 : $\text{ck}_{\text{ped}} := (G_1, G_2, G_3, V)$ 4 : $\text{pp} \leftarrow (n, G, H, \text{ck}_{\text{ped}}, V)$ 5 : return pp Algorithm $\text{BS}_{\text{phf}}.\text{KeyGen}(\text{pp})$: 1 : $\mathbf{w} = (w_0, w_1) \leftarrow \mathbb{Z}_p^2$ 2 : $W \leftarrow w_0 G + w_1 H$ 3 : $x \leftarrow \mathbb{Z}_p$; $X \leftarrow xG$ 4 : $\text{pk} \leftarrow (W, X)$; $\text{sk} \leftarrow (\mathbf{w}, x, \text{pk})$ 5 : return (sk, pk)	Algorithm $\text{BS}_{\text{phf}}.\text{Sign}_1(\text{pp}, \text{sk}, \text{info})$: 1 : $\mathbf{U} := \text{H}_{\text{pb}}(\text{info})$ 2 : $\mathbb{x}_{\text{os}} = (G, H, W)$; $\mathbb{w}_{\text{os}} = \mathbf{w}$ 3 : $(A_{\text{os}}, \text{st}_{\text{os}}) \leftarrow \Sigma_{\text{os}}.\text{Init}(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}})$ 4 : $\mathbb{x}_{\text{dl}} = (G, X)$; $\mathbb{w}_{\text{dl}} = x$ 5 : $(A_{\text{dl}}, \text{st}_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{Init}(\mathbb{x}_{\text{dl}}, \mathbb{w}_{\text{dl}})$ 6 : $T_{\S} \leftarrow \mathbb{G}$; $e_{\S} \leftarrow \mathbb{Z}_p^\times$; $c_{\text{phf}} \leftarrow \mathbb{Z}_p^\times$ 7 : $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, 0, A_{\text{os}}, T_{\S}, e_{\S})$ 8 : $(A_{\text{phf}}, z_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{Sim}(\mathbb{x}_{\text{phf}}, c_{\text{phf}})$ 9 : return $\text{msg}_1 = (A_{\text{os}}, A_{\text{dl}}, A_{\text{phf}}, T_{\S}, e_{\S})$ Algorithm $\text{BS}_{\text{phf}}.\text{Sign}_2(\text{umsg} = (c_{\text{os}}, c_{\text{or}}))$: 1 : assert $c_{\text{or}} \neq 0$ 2 : $c_{\text{dl}} \leftarrow c_{\text{or}}/c_{\text{phf}}$ 3 : $z_{\text{os}} \leftarrow \Sigma_{\text{os}}.\text{Resp}(\text{st}_{\text{os}}, c_{\text{os}})$ 4 : $z_{\text{dl}} \leftarrow \Sigma_{\text{dl}}.\text{Resp}(\text{st}_{\text{dl}}, c_{\text{dl}})$ 5 : return $\text{msg}_2 = (c_{\text{phf}}, z_{\text{os}}, z_{\text{phf}}, z_{\text{dl}})$
Algorithm $\text{BS}_{\text{phf}}.\text{Ver}(\text{pp}, \text{pk}, \text{info}, \text{msg}, \sigma)$ 1 : parse $\sigma = (C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S}, C_{\text{rnd}}, \pi_{\text{sle}}, (\tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}})_{\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}})$ 2 : $\mathbf{U} := \text{H}_{\text{pb}}(\text{info})$; $\mathbb{x}_{\text{os}} = (G, H, W)$, $\mathbb{x}_{\text{dl}} = (G, X)$, $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$ 3 : $\mathbb{x}_{\text{sle}} := (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}})$ 4 : assert $\Sigma_{\text{xxx}}.\text{Ver}(\mathbb{x}_{\text{xxx}}, \tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}}) = 1$ for all $\text{xxx} \in \{\text{os}, \text{phf}, \text{dl}\}$ 5 : assert $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{dl}} \cdot \tilde{c}_{\text{phf}}) = \text{H}(\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \mathbb{x}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg})$ 6 : assert $\tilde{c}_{\text{dl}} \neq 0 \wedge \tilde{c}_{\text{phf}} \neq 0$ 7 : assert $\Pi_{\text{sle}}.\text{Ver}^{\text{H}_{\text{sle}}}(\mathbb{x}_{\text{sle}}, \pi_{\text{sle}}) = 1$ 8 : return 1	

Fig. 9. Setup, key generation, verification, and signer-side algorithms of the three-move blind signature scheme $\text{BS}_{\text{phf}} = \text{BS}_{\text{phf}}[\text{GGen}, \Pi_{\text{sle}}, \text{H}, \text{H}_{\text{pb}}, \text{H}_{\text{sle}}]$. $\text{H}_{\text{pb}} : \{0, 1\}^* \rightarrow \mathbb{G}^{n+1}$ generates parameters for PHF; $\text{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times$ generates challenges for Fiat-Shamir; H_{sle} is used in Π_{sle} . Recall that $(\mathbb{G}, p, G) \leftarrow \text{GGen}(1^\lambda)$ is an implicit input for all algorithms. We omitted the state for $\text{BS}_{\text{phf}}.\text{Sign}_2$, assuming that it can access values from the prior round.

5.2 Construction

We give our three-move blind signature construction $\text{BS}_{\text{phf}} = \text{BS}_{\text{phf}}[\text{GGen}, \Pi_{\text{sle}}, \text{H}, \text{H}_{\text{pb}}, \text{H}_{\text{sle}}]$ in Figures 9 and 10. The construction depends on the group generator GGen , the NIZK proof Π_{sle} described in the prior section, and hash functions H , H_{pb} , and H_{sle} which will be modeled as random oracles in our security proofs.

SETUP. The public parameters pp consist of $n = \lceil \log_2 p \rceil$, generators G and V , a random group element H and parameters $\text{ck}_{\text{ped}} = (G_1, G_2, G_3, V)$ for C_{ped} .

KEY GENERATION. The secret key sk in our construction consists of the Okamoto-Schnorr witness $\mathbf{w}$ for the statement $W = w_0 G + w_1 H$ and a discrete logarithm x of the group element $X = xG$. The public key pk contains W and X .

SIGNING. The goal of the signing protocol is for the signer to issue a signature σ to the user which consists of the following terms.

- The values $(C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$ which determine the Σ_{phf} statement $\tilde{\mathbb{x}}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$, where $\mathbf{U} = \text{H}_{\text{pb}}(\text{info})$ is derived from the common message info . Here, $(C_{\S} = \nu G + r_{\S} V, \tilde{A}_{\S} = A_{\text{os}} + \nu V)$ functions as a commitment to the unblinded nonce A_{os} and $(T_{\S}, \tilde{e}_{\S}) \in \mathbb{G} \times \mathbb{Z}_p^\times$.
- Σ -protocol transcripts $\tilde{\tau}_{\text{xxx}} = (\tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}})$ for Σ_{xxx} , compiled with Fiat-Shamir for $\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}$. The associated statements are $\mathbb{x}_{\text{dl}} = (G, X)$, $\mathbb{x}_{\text{os}} = (G, H, W)$, both determined by pk , and $\tilde{\mathbb{x}}_{\text{phf}}$ as described above. Notably, Σ_{dl} and Σ_{phf} are OR-compiled following [CDS94].

Algorithm $\text{BS}_{\text{phf}}.\text{U}_1(\text{pp}, \text{pk}, \text{info}, \text{msg}, \text{smsg}_1 = (A_{\text{os}}, A_{\text{dl}}, A_{\text{phf}}, T_{\mathbb{S}}, e_{\mathbb{S}}))$:

```

1 : assert  $e_{\mathbb{S}} \neq 0$ 
2 :  $\mathbb{x}_{\text{os}} = (G, H, W)$  ;  $\tilde{\text{st}}_{\text{os}} \leftarrow \mathbb{Z}_p^3$ 
3 :  $r_{\text{rnd}} \leftarrow \mathbb{Z}_p$  ;  $C_{\text{rnd}} = \text{C}_{\text{ped}}.\text{Commit}(\text{ck}_{\text{ped}}, \tilde{\text{st}}_{\text{os}}; r_{\text{rnd}})$ 
4 :  $\tilde{A}_{\text{os}} := \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}_{\text{os}})$ 
5 :  $\mathbb{x}_{\text{dl}} = (G, X)$ 
6 :  $(\tilde{A}_{\text{dl}}, \tilde{\text{st}}_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{BlindNonce}(\mathbb{x}_{\text{dl}}, A_{\text{dl}})$ 
7 :  $\mathbf{U} := \text{H}_{\text{pb}}(\text{info})$  ;  $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, 0, A_{\text{os}}, T_{\mathbb{S}}, e_{\mathbb{S}})$ 
8 :  $\nu, r_{\mathbb{S}}, \rho \leftarrow \mathbb{Z}_p$  ;  $e' \leftarrow \mathbb{Z}_p^\times$  ;  $\text{coins}_{\text{phf}} := (\nu, r_{\mathbb{S}}, \rho, e')$ 
9 :  $\tilde{\mathbb{x}}_{\text{phf}} = \Sigma_{\text{phf}}.\text{TransStmt}(\mathbb{x}_{\text{phf}}, \text{coins}_{\text{phf}})$ 
10 : parse  $\tilde{\mathbb{x}}_{\text{phf}} = (G, V, \mathbf{U}, C_{\mathbb{S}}, \tilde{A}_{\mathbb{S}}, \tilde{T}_{\mathbb{S}}, \tilde{e}_{\mathbb{S}})$ 
11 :  $\mathbf{A}'_{\text{phf}} := \Sigma_{\text{phf}}.\text{TransNonce}(\tilde{\mathbb{x}}_{\text{phf}}, A_{\text{phf}}, \text{coins}_{\text{phf}})$ 
12 :  $(\tilde{A}_{\text{phf}}, \tilde{\text{st}}_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{BlindNonce}(\tilde{\mathbb{x}}_{\text{phf}}, \mathbf{A}'_{\text{phf}})$ 
    // link blinded nonce  $\tilde{A}_{\text{os}}$  and  $(\tilde{A}_{\mathbb{S}}, C_{\mathbb{S}})$  via SLE NIZK  $\Pi_{\text{sle}}$ 
13 :  $\mathbb{x}_{\text{sle}} = (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\mathbb{S}}, C_{\text{rnd}}, \tilde{A}_{\mathbb{S}}, \tilde{A}_{\text{os}})$  ;  $\mathbf{w}_{\text{sle}} = (\nu, r_{\mathbb{S}}, \tilde{\text{st}}_{\text{os}}, r_{\text{rnd}})$ 
14 :  $\pi_{\text{sle}} \leftarrow \Pi_{\text{sle}}.\text{Prove}^{\text{H}_{\text{sle}}}(\mathbb{x}_{\text{sle}}, \mathbf{w}_{\text{sle}})$ 
15 :  $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}}) \leftarrow \text{H}(\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg})$ 
16 :  $c_{\text{os}} \leftarrow \Sigma_{\text{os}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{os}}, \tilde{\text{st}}_{\text{os}})$ 
17 :  $c' \leftarrow \Sigma_{\text{dl}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{or}}, \tilde{\text{st}}_{\text{dl}})$  ;  $c_{\text{or}} \leftarrow \Sigma_{\text{phf}}.\text{BlindChal}^{-1}(c', \tilde{\text{st}}_{\text{phf}})$ 
18 : return  $(c_{\text{os}}, c_{\text{or}})$ 

```

Algorithm $\text{BS}_{\text{phf}}.\text{U}_2(\text{smsg}_2 = (c_{\text{phf}}, z_{\text{os}}, z_{\text{phf}}, z_{\text{dl}}))$:

```

1 : assert  $c_{\text{phf}} \neq 0 \wedge c_{\text{or}} \neq 0$ 
2 :  $c_{\text{dl}} := c_{\text{or}}/c_{\text{phf}}$ 
3 : for  $\text{xxx} \in \{\text{os}, \text{dl}\}$  do
4 :   assert  $\Sigma_{\text{xxx}}.\text{Ver}(\mathbb{x}_{\text{xxx}}, A_{\text{xxx}}, c_{\text{xxx}}, z_{\text{xxx}}) = 1$ 
5 :    $\tilde{c}_{\text{xxx}} := \Sigma_{\text{xxx}}.\text{BlindChal}(c_{\text{xxx}}, \tilde{\text{st}}_{\text{xxx}})$ 
6 :    $\tilde{z}_{\text{xxx}} := \Sigma_{\text{xxx}}.\text{BlindResp}(z_{\text{xxx}}, c_{\text{xxx}}, \tilde{\text{st}}_{\text{xxx}})$ 
7 : assert  $\Sigma_{\text{phf}}.\text{Ver}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}) = 1$ 
8 :  $\tilde{c}_{\text{phf}} := \Sigma_{\text{phf}}.\text{BlindChal}(c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$ 
9 :  $z'_{\text{phf}} := \Sigma_{\text{phf}}.\text{TransResp}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}, \text{coins}_{\text{phf}})$ 
10 :  $\tilde{z}_{\text{phf}} := \Sigma_{\text{phf}}.\text{BlindResp}(z'_{\text{phf}}, c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$ 
11 :  $\tilde{\tau}_{\text{xxx}} := (\tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}})$  for  $\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}$ 
12 : return  $\sigma = (C_{\mathbb{S}}, \tilde{A}_{\mathbb{S}}, \tilde{T}_{\mathbb{S}}, \tilde{e}_{\mathbb{S}}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}_{\text{os}}, \tilde{\tau}_{\text{dl}}, \tilde{\tau}_{\text{phf}})$ 

```

Fig. 10. User-side algorithms of the blind signature scheme $\text{BS}_{\text{phf}} = \text{BS}_{\text{phf}}[\text{GGen}, \Pi_{\text{sle}}, \text{H}, \text{H}_{\text{pb}}, \text{H}_{\text{sle}}]$. $\text{H}_{\text{pb}} : \{0, 1\}^* \rightarrow \mathbb{G}^{n+1}$ generates parameters for PHF; $\text{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times$ generates challenges for Fiat-Shamir; H_{sle} is used in Π_{sle} . We omitted the state for $\text{BS}_{\text{phf}}.\text{U}_2$, assuming that it can access values from the prior round.

- Commitment C_{rnd} to the Okamoto-Schnorr blinding factors $\tilde{\text{st}}_{\text{os}}$ and a Π_{sle} proof π_{sle} computed by the user. The proof π_{sle} links the nonce A_{os} committed in $(C_{\text{s}}, \tilde{A}_{\text{s}})$ with the nonce derived by unblinding $\tilde{A}_{\text{os}}$ with $\tilde{\text{st}}_{\text{os}}$.

The signing protocol proceeds in three moves.

First move (S $\rightarrow$ U). First, the signer initializes the Σ_{os} protocol to obtain the nonce A_{os} . Then, it randomly samples $(T_{\text{s}}, e_{\text{s}}) \leftarrow \mathbb{G} \times \mathbb{Z}_p^\times$ and computes the nonces A_{phf} and A_{dl} for the OR-proof between Σ_{phf} (being simulated, with the statement $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, 0, A_{\text{os}}, T_{\text{s}}, e_{\text{s}})$ dependent on A_{os}) and Σ_{dl} (generated using the witness x). We note that although the signer does not “sign” the nonce A_{os} directly, this OR-proof will still bind the nonce A_{os} to the signature as outlined in Section 2. The signer sends $\text{msg}_1 = (A_{\text{os}}, A_{\text{dl}}, A_{\text{phf}}, T_{\text{s}}, e_{\text{s}})$ to the user.

Second move (U $\rightarrow$ S). Upon reception of msg_1 , the user blinds the nonce A_{os} into $\tilde{A}_{\text{os}}$. Next, it rerandomizes the statement $\mathbb{x}_{\text{phf}}$ for Σ_{phf} into $\tilde{\mathbb{x}}_{\text{phf}}$ following Equation (7). Then, the user adapts the nonce A_{phf} for $\tilde{\mathbb{x}}_{\text{phf}}$ via the translation algorithm Σ_{phf} and blinds it into $\tilde{A}_{\text{phf}}$. It also blinds the nonce A_{dl} into $\tilde{A}_{\text{dl}}$ and computes π_{sle} as described above. The statements, blinded nonces and the proof π_{sle} are then hashed along with the message to produce $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$ which are blinded into $(c_{\text{os}}, c_{\text{or}})$ and sent to the signer.

Third move (S $\rightarrow$ U). Given $(c_{\text{os}}, c_{\text{or}})$, the signer computes the responses $z_{\text{os}}, z_{\text{phf}}, z_{\text{dl}}$ for $\Sigma_{\text{os}}, \Sigma_{\text{phf}}, \Sigma_{\text{dl}}$, where c_{dl} and c_{phf} are chosen to satisfy $c_{\text{or}} = c_{\text{dl}} \cdot c_{\text{phf}}$. Then, the signer sends $\text{msg}_2 = (c_{\text{phf}}, z_{\text{os}}, z_{\text{phf}}, z_{\text{dl}})$ to the user.

Signature Derivation. To derive the signature σ as above, the user translates the response z_{phf} for consistency with $\tilde{\mathbb{x}}_{\text{phf}}$ and then blinds all transcripts τ_{xxx} for $\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}$.

VERIFICATION. Verification checks that (i) the transcripts for all three Σ -protocols are valid, (ii) the proof π_{sle} verifies, (iii) the challenges $\tilde{c}_{\text{dl}}$ and $\tilde{c}_{\text{phf}}$ are non-zero, and (iv) $\tilde{c}_{\text{os}}$ and $\tilde{c}_{\text{phf}} \cdot \tilde{c}_{\text{dl}}$ are derived from $\mathbf{H}$.

EFFICIENCY. The signature size of our scheme can be reduced by omitting some of the nonces in the $\Sigma_{\text{os}}, \Sigma_{\text{dl}}$ and Σ_{phf} transcripts. We detail the counted elements below according to the constructions given in Figures 4 to 6.

Σ_{os} transcript. The transcript contains $1\mathbb{G}$ and $3\mathbb{Z}_p$ elements. In the signature, only the $3\mathbb{Z}_p$ elements need to be included as the nonce can be derived from these values.

Σ_{dl} transcript. The transcript contains $1\mathbb{G}$ and $2\mathbb{Z}_p$ elements. In the signature, only the $2\mathbb{Z}_p$ elements need to be included as the nonce can be derived from these values.

Σ_{phf} transcript. The transcript contains $(4n+4)\mathbb{G}$ ($n+4$ for $\mathbf{A}$, n for $\mathbf{C}_\mu$, and $2n$ for $\mathbf{D}_0, \mathbf{D}_1$) and $(3n+6)\mathbb{Z}_p$ elements ($2n+5$ for $\mathbf{z}$, n for $\mathbf{t}$, 1 for c_{phf}). The signature can omit $\mathbf{A}, \mathbf{D}_0$ since they can be recomputed from the rest as seen in $\Sigma_{\text{phf}}.\text{Ver}$, resulting in $2n\mathbb{G} + (3n+6)\mathbb{Z}_p$ elements in the signature.

Additionally, the signer sends T_{s} and e_{s} during the protocol adding in $1\mathbb{G} + 1\mathbb{Z}_p$. The user also supplies the blinded nonce $\tilde{A}_{\text{s}}$, the commitments C_{s} and C_{rnd} , and the straightline-extractable proof π_{sle} in the signature.

The combined communication and signature size are as follows:

Signer communication. The first move consists of $(4n+7)\mathbb{G} + 1\mathbb{Z}_p$ elements. The third move consists of $(3n+9)\mathbb{Z}_p$ elements. In total, $(4n+7)\mathbb{G} + (3n+10)\mathbb{Z}_p$ elements.

User communication. The user communicates only $2\mathbb{Z}_p$ elements.

Signature size. The signature consists of $(2n+4)\mathbb{G} + (3n+12)\mathbb{Z}_p + |\pi_{\text{sle}}|$ elements. We note that the size of π_{sle} depends on the parameters set in Lemma 5.2; in particular, $|\pi_{\text{sle}}| = r \cdot |\tau_{\text{sle}}|$ where r is the parallel repetition parameter and $|\tau_{\text{sle}}| = 7\mathbb{Z}_p$ is the transcript size for the underlying Σ -protocol. Note also that in $\Sigma_{\text{sle}}, A_{\text{sle}}$ can be recovered from c_{sle} and $\mathbf{z}_{\text{sle}}$, meaning each transcript only needs to contain $(c_{\text{sle}}, \mathbf{z}_{\text{sle}})$, resulting in the claimed size.

For $\lambda = 128$ and $\lceil \log p \rceil = 256$, an example parameter set is $r = 8$ and $b = 16$ (aiming towards 128-bit security according to Lemma 5.2 and ignoring some of the losses). Therefore, the communication is $\approx 58\text{KB}$ and the signature size is $\approx 43\text{KB}$.

5.3 Correctness and Security

The following theorems establish the following properties for BS_{phf} : (1) the correctness guarantee in the random oracle model, proved in Section 6, (2) one-more strong unforgeability in the random oracle model, proved in Section 7, and (3) blindness in the random oracle model, proved in Section 8. We give proof sketches below and refer to the full proofs in the corresponding sections.

Theorem 5.3 (Correctness). *Let GGen be a group parameters generator outputting groups of prime order $p = p(\lambda)$, Π_{sle} be a NIZK proof as described in Section 5.1 with completeness error $\varepsilon = \varepsilon(\lambda)$, $H, H_{\text{pb}}, H_{\text{sle}}$ be hash functions modeled as random oracles, and $\text{BS}_{\text{phf}} = \text{BS}_{\text{phf}}[\text{GGen}, \Pi_{\text{sle}}, H, H_{\text{pb}}, H_{\text{sle}}]$. Then, BS_{phf} is correct with error ε .*

Theorem 5.4 (OMSUF). *Let $\text{GGen}, p, \Pi_{\text{sle}}, H, H_{\text{pb}}, H_{\text{sle}}$, and BS_{phf} be as in Theorem 5.3, and Ext be the extractor for the SLE property of Π_{sle} . Then, for any adversary $\mathcal{A}$ running in time $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ in the one-more strong unforgeability game of BS_{phf} inducing at most Q_S queries to signing oracle S_1 and $Q_{H_\star}$ queries to $H_\star \in \{H, H_{\text{pb}}, H_{\text{sle}}\}$, respectively, there exist adversaries $\mathcal{B}_{\text{sle}}$ and $\mathcal{B}_{\text{dl}}$ such that*

$$\begin{aligned} \text{Adv}_{\text{BS}_{\text{phf}}}^{\text{omsuf}}(\mathcal{A}, \lambda) &\leq \text{Adv}_{\Pi_{\text{sle}}, \text{Ext}, \hat{\mathfrak{R}}_{\text{sle}}}^{\text{sle}}(\mathcal{B}_{\text{sle}}, \lambda) + 6\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1 + Q_H^2}{p} \\ &\quad + (Q_{\text{pb}}/\delta + 1) \cdot \left(\sqrt[3]{32Q_H^2 \cdot \left(\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p-1} \right)} + \frac{4Q_H + 1}{p-1} \right), \end{aligned}$$

where $\mathcal{B}_{\text{sle}}$ runs in time roughly $T_{\mathcal{A}}$, $\mathcal{B}_{\text{dl}}$ runs in time roughly $3T_{\mathcal{A}}$, and $\delta = \frac{d}{Q_S^2(n+1)}$ with d as defined in Theorem 3.11 and $n = \lceil \log_2 p \rceil$.

Proof (sketch). The proof is rather involved. We give a brief overview here; we refer to Section 7 for details. Below, we assume that only info^* signing queries are made for simplicity.¹⁶ First, we let the game extract a witness $w_{\text{sle}}^{(i)}$ from $\pi_{\text{sle}}^{(i)}$ from every signature that $\mathcal{A}$ provides.¹⁷ While Π_{sle} is SLE for relation $\hat{\mathfrak{R}}_{\text{sle}} = \mathfrak{R}_{\text{sle}} \cup \mathfrak{R}_{\text{zero}}$, we may assume that $(x_{\text{sle}}^{(i)}, w_{\text{sle}}^{(i)}) \in \mathfrak{R}_{\text{sle}}$, as finding a witness for $\mathfrak{R}_{\text{zero}}$ is hard under the Rel-DL assumption. The witness $w_{\text{sle}}^{(i)} = (\nu^{(i)}, r_{\text{s}}^{(i)}, \alpha_{\text{os}}^{(i)}, \gamma_{\text{os}}^{(i)}, r_{\text{rnd}}^{(i)})$ determines the *unblinded* nonce

$$A_{\text{os}}^{(i)} := \tilde{A}_{\text{s}}^{(i)} - \nu^{(i)}V$$

associated to each signature. Further, we ensure that all the H queries associated to the signatures are distinct.¹⁸ Roughly, this is justified by the relaxed unique responses property of the Σ -protocols. However, note that the argument requires care. For instance, if there are two signatures that differ only in the Σ_{phf} responses, we may extract a witness $w \neq \mathbf{0}$ such that $\Phi_{\text{phf}}^{T_{\text{s}}}(\mathbf{w}) = \mathbf{0}$ by the relaxed unique responses property of Σ_{phf} . However, it is *not* computationally hard to find such w since T_{s} is under the control of the adversary.¹⁹ However, if this case occurs we can extract a representation T_{s} from the Σ_{phf} proof in the forgeries for which such a representation is hard to find (by setting up $\mathbf{U}$ via PHF.NonZeroGen). The other cases are simpler to analyze and may be argued by special soundness or relaxed unique responses of Σ_{os} .

Given the above changes, we distinguish two cases.

Case NonceColl. There are two identical nonces $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$. In this case, we may compute two valid transcripts $\tau_{\text{os}}^{(i_0)} = (A_{\text{os}}^{(i_0)}, c_{\text{os}}^{(i_0)}, z_{\text{os}}^{(i_0)})$ and $\tau_{\text{os}}^{(i_1)} = (A_{\text{os}}^{(i_1)}, c_{\text{os}}^{(i_1)}, z_{\text{os}}^{(i_1)})$ with shared nonce from the Σ_{os}

¹⁶ To generalize the argument to partial blindness, we may setup $\mathbf{U} = H_{\text{pb}}(\text{info})$ for $\text{info} \neq \text{info}^*$ via PHF.NonZeroGen which allows simulation as required in the argument sketched below.

¹⁷ Technically, the extraction happens at the time the H query associated to the i th signature is made. This ensures that the blinding factors are determined before H 's output is sampled.

¹⁸ Note that this step is required for strong unforgeability, as the challenges or responses of the Σ -protocols may differ. For conventional OMUF, this is already satisfied since the messages are distinct.

¹⁹ For instance, $T_{\text{s}} = -\beta V$ and $\alpha = 1$, then $w = (\alpha, \beta, 0, 0, 0, \mathbf{0})$ satisfies $\mathbf{0} = \Phi_{\text{phf}}^{T_{\text{s}}}(w)$.

transcripts provided in the signatures and the associated blinding factors extracted from the Π_{sle} proofs. Since the blinding factors are determined before the associated hash query is made, we have $c_{\text{os}}^{(i_0)} \neq c_{\text{os}}^{(i_1)}$ with sufficient probability. This allows us to extract a witness w_{os} for $\mathfrak{R}_{\text{os}}$ and under witness indistinguishability of Σ_{os} , we have $w_{\text{os}} \neq w$ which allows to compute the DL of H . This argument is similar to [CART26, Lemma 6.2], except that we require no rewinding and instead reduce to algebraic one-more preimage resistance [TZ23, Theorem 1] of Φ_{os} .

Case NonceDist. In this case, there is a fresh nonce $A_{\text{os}}^{(i^*)}$ extracted from $\mathcal{A}$'s signatures that never appeared in the signing queries (since $\ell > Q_S$). This allows for the following simulation strategy:

- We set up H with known DL, which allows to compute the discrete logarithms r_j of nonces $A_{\text{os},j} = r_j G$ sampled in signing sessions and r^* of the nonce $A_{\text{os}}^{(i^*)} = r^* G$.
- We set up $U = H_{\text{pb}}(\text{info}^*)$ via PHF.TrapGen.
- Denote by μ_j the bit-decomposition of $r_j \in [0, p-1]$. We ensure that PHF.TrapEval evaluates on input μ_j to a non-zero value $a_j \neq 0 \bmod p$ in the first component. Note that for r^* , there may be two possible bit-decompositions μ_0^* and μ_1^* of n bits for $r^* \in \mathbb{Z}_p$. For both, we ensure that PHF.TrapEval evaluates on input μ_b^* to a zero in the first component. This may be argued via Theorem 3.11.
- The game issues the Σ_{phf} transcripts with an honest witness, which may be computed due to $a_j \neq 0 \bmod p$. It simulates the Σ_{dl} transcripts.

At the end, the adversary provides a Σ_{phf} proof (or-compiled with Σ_{dl}) in $\tilde{\sigma}^{(i^*)}$ that proves that the PHF evaluates a bit-decomposition of the exponent of $A_{\text{os}}^{(i^*)}$ to a non-zero value. By rewinding, either we extract the DL of X or a witness contradicting the fact that both possible bit-decompositions of $A_{\text{os}}^{(i^*)}$ evaluate to zero, as ensured above. This contradiction is computational (more precisely, this allows us to extract the DL of V). As neither the DL of X nor V is required for simulation, this concludes the proof. $\square$

Theorem 5.5 (Blindness). *Let $G\text{Gen}, p, \Pi_{\text{sle}}, H, H_{\text{pb}}, H_{\text{sle}}$, and BS_{phf} be as in Theorem 5.3. Also, let Sim be the simulator for the zero-knowledge property for Π_{sle} . Then, for any adversary $\mathcal{A}$ in the blindness game of BS_{phf} inducing at most Q_{H^*} queries to $H^* \in \{H, H_{\text{pb}}, H_{\text{sle}}\}$, respectively, there exists an adversary $\mathcal{B}_{\text{zk}}$ inducing at most $Q_{H_{\text{sle}}}$ queries to H_{sle} such that*

$$\text{Adv}_{\text{BS}_{\text{phf}}}^{\text{blind}}(\mathcal{A}, \lambda) \leq \text{Adv}_{\Pi_{\text{sle}}, \text{Sim}}^{\text{zk}}(\mathcal{B}_{\text{zk}}, \lambda) + \frac{2Q_H + 4}{p}.$$

Proof (sketch). The blindness proof in Section 8 roughly follows these steps.

1. First, we make several game hops to a game, where instead of first computing $H(\cdot)$ and applying the blinding operations, the game samples the challenges $(c_{\text{os}}, c_{\text{or}})$ that the user-side outputs uniformly at random and reprograms H accordingly. This is possible due to the inputs to random oracle H having high min-entropy.
2. Then, simulate the NIZK proof π_{sle} in the signature with the simulator Sim . Also, via perfect hiding of C_{ped} , the commitment C_{rnd} does not reveal any information about the blinding factors.
3. Next, we rely on the blindability properties of $\Sigma_{\text{os}}, \Sigma_{\text{dl}}$ and Σ_{phf} along with translatability of Σ_{phf} to argue that the blinded transcripts of these Σ -protocols are indistinguishable from simulated transcripts. At this point, every part of the signature is independent of the signing sessions, except for the translated statement $\tilde{x}_{\text{phf}}$.
4. Finally, we observe that the tuple $(C_{\text{s}}, \tilde{A}_{\text{s}}, \tilde{T}_{\text{s}}, \tilde{e}_{\text{s}})$ in the translated statement $\tilde{x}_{\text{phf}}$ for Σ_{phf} is uniformly random in $\mathbb{G}^3 \times \mathbb{Z}_p^\times$. This means that the signatures are simply independent of the signing sessions, concluding the proof. $\square$

6 Correctness Proof of BS_{phf} (Theorem 5.3)

Recall that $\text{pp} = (n, G, H, \text{ck}_{\text{ped}}, V)$, $\text{sk} = (w, x, \text{pk})$ and $\text{pk} = (W, X)$. We denote by σ the final signature obtained from running the signing protocol honestly for message $\mu \in \{0, 1\}^*$ and common message info .

Recall that σ is of the form

$$\sigma = (C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S}, C_{\text{rnd}}, \pi_{\text{sle}}, (\tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}})_{\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}}).$$

For the correctness proof, we want to show that

- $\tilde{c}_{\text{dl}} \neq 0$ and $\tilde{c}_{\text{phf}} \neq 0$
- $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{dl}}, \tilde{c}_{\text{phf}}) = \text{H}(\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg})$ for the statements $\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \mathbb{x}_{\text{sle}}$ parsed as in $\text{BS}_{\text{phf}}.\text{Ver}$ algorithm in Figure 9.
- Each of the blinded transcripts $(\tilde{A}_{\text{xxx}}, \tilde{c}_{\text{xxx}}, \tilde{z}_{\text{xxx}})$ for $\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}$ is valid for the corresponding statement.
- The straightline-extractable proof π_{sle} is valid with ε error.

We now show the above in this order.

NON-ZERO CHALLENGES. Recall that the challenges $\tilde{c}_{\text{dl}}, \tilde{c}_{\text{phf}}$ are computed as $\tilde{c}_{\text{phf}} = \Sigma_{\text{phf}}.\text{BlindChal}(c_{\text{phf}}, \tilde{\text{st}}_{\text{phf}})$ and $\tilde{c}_{\text{dl}} = \Sigma_{\text{dl}}.\text{BlindChal}(c_{\text{dl}}, \tilde{\text{st}}_{\text{dl}})$. First, $c_{\text{phf}} \in \mathbb{Z}_p^\times$ by how it is sampled in $\text{BS}_{\text{phf}}.\text{Sign1}$. Then, we see that $c_{\text{dl}} = c_{\text{or}}/c_{\text{phf}}$ and $c_{\text{phf}} \neq 0$, so we only have to show that $c_{\text{or}} \neq 0$. This follows from the fact that $c_{\text{or}} = \Sigma_{\text{phf}}.\text{BlindChal}^{-1}(\Sigma_{\text{dl}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{or}}, \tilde{\text{st}}_{\text{dl}}), \tilde{\text{st}}_{\text{phf}})$, $\tilde{c}_{\text{or}} \neq 0$, and by $\Sigma_{\text{dl}}.\text{Rand}$ - and $\Sigma_{\text{phf}}.\text{Rand}$ -blindability, we have that $c_{\text{or}} \neq 0$.

COMBINED CHALLENGE IS THE HASH. This follows from how $\tilde{c}_{\text{os}}$ is defined as the hash, and Lemma 4.4.

VALID BLINDED Σ_{os} TRANSCRIPT. The signer sets $\mathbb{x}_{\text{os}} = (G, H, W)$ and $\mathbb{w}_{\text{os}} = \mathbf{w}$ where $W = w_0G + w_1H = \overline{\Phi}_{\text{os}}^{\text{par}_{\text{os}}}(\mathbb{w}_{\text{os}})$, so $(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}}) \in \mathfrak{R}_{\text{os}}$. Also, the transcript from the signer side $(A_{\text{os}}, c_{\text{os}}, z_{\text{os}})$ is computed as (1) $(A_{\text{os}}, \text{st}_{\text{os}}) \leftarrow \Sigma_{\text{os}}.\text{Init}(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}})$ in $\text{BS}_{\text{phf}}.\text{Sign1}$, (2) c_{os} picked by the user, and (3) $z_{\text{os}} \leftarrow \Sigma_{\text{os}}.\text{Resp}(\text{st}_{\text{os}}, c_{\text{os}})$. Hence, by perfect correctness of Σ_{os} , this passes the validity check by the user-side. Then, recall that in $\text{BS}_{\text{phf}}.\text{U}$, the blinded transcript $\tilde{\tau}_{\text{os}} := (\tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}})$ is derived via $\Sigma_{\text{os}}.\text{Rand}$, so by $\Sigma_{\text{os}}.\text{Rand}$ -blindability of Σ_{os} (Lemma 4.2), we have that $\tilde{\tau}_{\text{os}}$ is a valid Σ_{os} -transcript for $\mathbb{x}_{\text{os}}$.

VALID BLINDED Σ_{dl} TRANSCRIPT. The signer sets $\mathbb{x}_{\text{dl}} = (G, X)$ and $\mathbb{w}_{\text{dl}} = x$ where $X = xG = \Phi_{\text{dl}}^{\text{par}_{\text{dl}}}(\mathbb{w}_{\text{dl}})$, so $(\mathbb{x}_{\text{dl}}, \mathbb{w}_{\text{dl}}) \in \mathfrak{R}_{\text{dl}}$. Also, the transcript from the signer side $(A_{\text{dl}}, c_{\text{dl}}, z_{\text{dl}})$ is computed as (1) $(A_{\text{dl}}, \text{st}_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{Init}(\mathbb{x}_{\text{dl}}, \mathbb{w}_{\text{dl}})$ in $\text{BS}_{\text{phf}}.\text{Sign1}$, (2) c_{dl} picked by the user, and (3) $z_{\text{dl}} \leftarrow \Sigma_{\text{dl}}.\text{Resp}(\text{st}_{\text{dl}}, c_{\text{dl}})$. Hence, by perfect correctness of Σ_{dl} , this passes the validity check by the user-side. Then, recall that in $\text{BS}_{\text{phf}}.\text{U}$, the blinded transcript $\tilde{\tau}_{\text{dl}} := (\tilde{A}_{\text{dl}}, \tilde{c}_{\text{dl}}, \tilde{z}_{\text{dl}})$ is derived via $\Sigma_{\text{dl}}.\text{Rand}$, so by $\Sigma_{\text{dl}}.\text{Rand}$ -blindability of Σ_{dl} (Lemma 4.1), we have that $\tilde{\tau}_{\text{dl}}$ is a valid Σ_{dl} -transcript for $\mathbb{x}_{\text{dl}}$.

VALID BLINDED Σ_{phf} TRANSCRIPT. In addition to arguing that the signer-side transcript is valid, we have to show that the translation preserves validity of the transcripts with respect to the new statement. First, we argue that the signer-side transcript $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ is a valid transcript for $\mathbb{x}_{\text{phf}}$. Recall that $\mathbb{x}_{\text{phf}} = (G, V, \mathbf{U}, 0, A_{\text{os}}, T_{\S}, e_{\S})$ with $\mathbf{U} := \text{H}_{\text{pb}}(\text{info})$, and the transcript is computed by (1) sampling $c_{\text{phf}} \leftarrow \mathbb{Z}_p^\times$ and (2) $(A_{\text{phf}}, z_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{Sim}(\mathbb{x}_{\text{phf}}, c_{\text{phf}})$. Here, by perfect SHVZK and perfect correctness of Σ_{phf} , we have that $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ is always a valid transcript for $\mathbb{x}_{\text{phf}}$.

Next, we show that the translated transcript $(A'_{\text{phf}}, c_{\text{phf}}, z'_{\text{phf}})$ is valid for the translated statement $\tilde{\mathbb{x}}_{\text{phf}} = \Sigma_{\text{phf}}.\text{TransStmnt}(\mathbb{x}_{\text{phf}}, \text{coins}_{\text{phf}})$ where $\text{coins}_{\text{phf}} := (\nu, r_{\S}, \rho, e') \leftarrow \mathbb{Z}_p^3 \times \mathbb{Z}_p^\times$. Recall that in $\text{BS}_{\text{phf}}.\text{U}$, the user computes $A'_{\text{phf}} := \Sigma_{\text{phf}}.\text{TransNonce}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, \text{coins}_{\text{phf}})$ and $z'_{\text{phf}} := \Sigma_{\text{phf}}.\text{TransResp}(\mathbb{x}_{\text{phf}}, A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}, \text{coins}_{\text{phf}})$. By $\Sigma_{\text{phf}}.\text{Trans}$ -translatability of Σ_{phf} , proved in Lemma 4.3, we have that $(A'_{\text{phf}}, c_{\text{phf}}, z'_{\text{phf}})$ is a valid transcript for $\tilde{\mathbb{x}}_{\text{phf}}$.

Finally, by $\Sigma_{\text{phf}}.\text{Rand}$ -blindability, also proved in Lemma 4.3, we have that the blinded transcript $\tilde{\tau}_{\text{phf}} := (\tilde{A}_{\text{phf}}, \tilde{c}_{\text{phf}}, \tilde{z}_{\text{phf}})$ derived via $\Sigma_{\text{phf}}.\text{Rand}$, is always a valid transcript for the translated statement $\tilde{\mathbb{x}}_{\text{phf}}$.

VALID π_{sle} PROOFS. Recall that within $\text{BS}_{\text{phf}}.\text{U}_1$, the translated statement $\tilde{\mathbb{x}}_{\text{phf}}$ is of the form $\tilde{\mathbb{x}}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$. Also, $\mathbb{x}_{\text{sle}} = (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\S}, C_{\text{rnd}}, \tilde{A}_{\S}, \tilde{A}_{\text{os}})$ and $\mathbb{w}_{\text{sle}} = (\nu, r_{\S}, \tilde{\text{st}}_{\text{os}}, r_{\text{rnd}})$ where

$$\begin{aligned} C_{\S} &= \nu G + r_{\S} V \\ \tilde{A}_{\S} &= A_{\text{os}} + \nu V \\ C_{\text{rnd}} &= \text{C}_{\text{ped}}.\text{Commit}(\text{ck}_{\text{ped}}, \tilde{\text{st}}; r_{\text{rnd}}) \\ \tilde{A}_{\text{os}} &= \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{x}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}}) . \end{aligned}$$

The first two equalities follow from how $\Sigma_{\text{phf}}.\text{TransStmt}$ is defined. The above equations imply that $(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}}) \in \mathfrak{R}_{\text{sle}}$. Hence, by correctness of Π_{sle} , we have that the proof π_{sle} is a valid proof for $\mathbb{x}_{\text{sle}}$, except with probability ε . $\square$

7 OMSUF Proof of BS_{phf} (Theorem 5.4)

7.1 Useful Lemmata

Before we proceed with the proof, we introduce some useful lemmata. The first lemma aids in switching branches of the OR-proof.

Lemma 7.1 (Witness Indistinguishability of Non-zero Proof). *Let $X = xG \in \mathbb{G}$. Let $\text{hpp} = (G, V, \mathbf{U} = (U_0, \dots, U_n)) \in \mathbb{G}^{n+3}$, where $U_i = a_i G + b_i V$ for $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}^{n+1} \times \mathbb{Z}_p^{n+1}$ with $V \neq 0$. Let $\boldsymbol{\mu} \in \{0, 1\}^n$ such that $e \neq 0 \pmod p$, where $e = a_0 + \sum_{i \in [n]} \mu_i a_i$ or equivalently $(e, f) = \text{PHF}.\text{TrapEval}((\mathbf{a}, \mathbf{b}), \boldsymbol{\mu})$. Let $A = \sum_{i \in [n]} \mu_i (2^{i-1} G) \in \mathbb{G}$. Let $\mathbb{x}_{\text{dl}} = (G, X)$. Then, for any stateful adversary $\mathcal{A}$, we have*

$$\Pr[b \leftarrow \text{ExpDL}_{\mathbb{x}_{\text{dl}}, \text{hpp}, A, x}^{\mathcal{A}}(\lambda)] = \Pr[b \leftarrow \text{ExpPHF}_{\mathbb{x}_{\text{dl}}, \text{hpp}, A, \boldsymbol{\mu}, \mathbf{a}, \mathbf{b}}^{\mathcal{A}}(\lambda)] ,$$

where the experiments ExpDL and ExpPHF are formally defined in Figure 11.

Proof. Denote $C_{\S} = 0 \in \mathbb{G}$ and $\nu = r_{\S} = 0 \in \mathbb{Z}_p$. In ExpPHF , we have

$$C_{\S} = \nu G + r_{\S} V ; \quad A = \nu G + \sum_{i \in [n]} \mu_i (2^{i-1} G) ; \quad T_{\S} = \rho V + U_0 + \sum_{i \in [n]} \mu_i U_i$$

and since $U_i = a_i G + b_i V$, we also have

$$\begin{aligned} e_{\S} G &= (\alpha \cdot e) G = \alpha \cdot (a_0 + \sum_{i \in [n]} \mu_i a_i) G - \beta V + \beta V \\ &= \alpha \cdot (a_0 + \sum_{i \in [n]} \mu_i a_i G) + \alpha \cdot (\rho + b_0 + \sum_{i \in [n]} \mu_i b_i) V + \beta V \\ &= \alpha \cdot (\rho V + U_0 + \sum_{i \in [n]} \mu_i U_i) + \beta V \\ &= \alpha T_{\S} + \beta V . \end{aligned}$$

Further, observe that $e_{\S} \neq 0$ in ExpPHF since $e \neq 0 \pmod p$. Therefore, we have $(\mathbb{x}_{\text{phf}}, \mathbb{w}_{\text{phf}}) \in \mathfrak{R}_{\text{phf}}$. On the other hand, we trivially have in ExpDL that $(\mathbb{x}_{\text{dl}}, x) \in \mathfrak{R}_{\text{dl}}$. Also, observe that the values $T_{\S}$ and $e_{\S}$ are distributed identically in both games. Therefore, the statement follows from SHVZK of Σ_{phf} and Σ_{dl} .

The second lemma states that we can recompute the discrete logarithm of the nonce A_{os} given a valid blinded transcript $(\tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}})$ and its blinding factors.

Experiment $\text{ExpDL}_{\mathbb{X}_{\text{dl}}, \text{hpp}, A, x}^A(\lambda)$:	Experiment $\text{ExpPHF}_{\mathbb{X}_{\text{dl}}, \text{hpp}, A, \mu, \alpha, b}^A(\lambda)$:
1 : $(A_{\text{dl}}, \text{st}_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{Init}(\mathbb{X}_{\text{dl}}, \mathbb{W}_{\text{dl}})$	1 : $e := a_0 + \sum_{i \in [n]} \mu_i a_i ; f := b_0 + \sum_{i \in [n]} \mu_i b_i$
2 : $T_{\mathbb{S}} \leftarrow \mathbb{G} ; e_{\mathbb{S}} \leftarrow \mathbb{Z}_p^\times ; c_{\text{phf}} \leftarrow \mathbb{Z}_p^\times$	2 : $\alpha \leftarrow \mathbb{Z}_p^\times ; \rho \leftarrow \mathbb{Z}_p ; \beta := -\alpha \cdot (\rho + f)$
3 : $\mathbb{X}_{\text{phf}} := (G, V, \mathbf{U}, 0, A, T_{\mathbb{S}}, e_{\mathbb{S}})$	3 : $e_{\mathbb{S}} := \alpha \cdot e \bmod p ; T_{\mathbb{S}} := \rho V + U_0 + \sum_{i \in [n]} \mu_i U_i$
4 : $(A_{\text{phf}}, z_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{Sim}(\mathbb{X}_{\text{phf}}, c_{\text{phf}})$	4 : $\mathbb{W}_{\text{phf}} := (\alpha, \beta, 0, \rho, 0, \mu)$
5 : $c_{\text{or}} \leftarrow \mathcal{A}(A_{\text{dl}}, \text{hpp}, A, A_{\text{dl}}, A_{\text{phf}}, T_{\mathbb{S}}, e_{\mathbb{S}})$	5 : $\mathbb{X}_{\text{phf}} := (G, V, \mathbf{U}, 0, A, T_{\mathbb{S}}, e_{\mathbb{S}})$
6 : assert $c_{\text{or}} \neq 0$	6 : $(A_{\text{phf}}, \text{st}_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{Init}(\mathbb{X}_{\text{phf}}, \mathbb{W}_{\text{phf}})$
7 : $c_{\text{dl}} \leftarrow c_{\text{or}}/c_{\text{phf}}$	7 : $c_{\text{dl}} \leftarrow \mathbb{Z}_p^\times$
8 : $z_{\text{dl}} \leftarrow \Sigma_{\text{dl}}.\text{Resp}(\text{st}_{\text{dl}}, c_{\text{dl}})$	8 : $(A_{\text{dl}}, z_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{Sim}(\mathbb{X}_{\text{dl}}, c_{\text{dl}})$
9 : $b \leftarrow \mathcal{A}(c_{\text{phf}}, z_{\text{phf}}, z_{\text{dl}})$	9 : $c_{\text{or}} \leftarrow \mathcal{A}(A_{\text{dl}}, \text{hpp}, A, A_{\text{dl}}, A_{\text{phf}}, T_{\mathbb{S}}, e_{\mathbb{S}})$
10 : return b	10 : assert $c_{\text{or}} \neq 0$
	11 : $c_{\text{phf}} \leftarrow c_{\text{or}}/c_{\text{dl}}$
	12 : $z_{\text{phf}} \leftarrow \Sigma_{\text{phf}}.\text{Resp}(\text{st}_{\text{phf}}, c_{\text{phf}})$
	13 : $b \leftarrow \mathcal{A}(c_{\text{phf}}, z_{\text{phf}}, z_{\text{dl}})$
	14 : return b

Fig. 11. The experiments ExpDL and ExpPHF from Lemma 7.1.

Lemma 7.2. Let $A_{\text{os}} \in \mathbb{G}$. Let $\tilde{\tau}_{\text{os}} = (\tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}})$ be a valid transcript for $\mathbb{X}_{\text{os}} = (G, H, W = w_0G + w_1H)$. Assume we are given $\tilde{\text{st}} = (\alpha, \gamma)$ such that $\tilde{A}_{\text{os}} = \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{X}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}})$. There is an efficient PT algorithm that given the above as input, can recompute $(c_{\text{os}}, z_{\text{os}})$ such that $(A_{\text{os}}, c_{\text{os}}, z_{\text{os}})$ is a valid transcript for $\mathbb{X}_{\text{os}}$. Furthermore, if it is further given the witness $\mathbf{w} = (w_0, w_1)$ and the discrete logarithm h of H , then it can compute r such that $A_{\text{os}} = rG$.

Proof. Observe that

$$\Sigma_{\text{os}}.\text{Ver}(\mathbb{X}_{\text{os}}, \tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}}) = 1 \iff \tilde{A}_{\text{os}} + \tilde{c}_{\text{os}}W = \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\tilde{z}_{\text{os}}).$$

Furthermore, letting $c_{\text{os}} = \tilde{c}_{\text{os}} - \gamma$ and $z_{\text{os}} = \tilde{z}_{\text{os}} - \alpha$, we have that

$$\begin{aligned}
\tilde{A}_{\text{os}} &= \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{X}_{\text{os}}, A_{\text{os}}; \tilde{\text{st}} = (\alpha, \gamma)) \\
&= A_{\text{os}} + \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\alpha_{\text{os}}) - \gamma \cdot W \\
A_{\text{os}} &= \tilde{A}_{\text{os}} - \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\alpha) + \gamma \cdot W \\
&= \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\tilde{z}) - \tilde{c}_{\text{os}} \cdot W - \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\alpha_{\text{os}}) + \gamma \cdot W \\
&= \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\tilde{z}_{\text{os}} - \alpha) - (\tilde{c}_{\text{os}} - \gamma) \cdot W \\
&= \Phi_{\text{os}}^{\text{par}_{\text{os}}}(z_{\text{os}}) - c_{\text{os}} \cdot W \\
&\implies \Sigma_{\text{os}}.\text{Ver}(\mathbb{X}_{\text{os}}, A_{\text{os}}, c_{\text{os}}, z_{\text{os}}) = 1.
\end{aligned}$$

The second equality follows from opening how $\Sigma_{\text{os}}.\text{BlindNonce}$ is implemented. The third equality follows from changing sides of the equations. The fourth equality follows from $\tilde{\tau}_{\text{os}} = (\tilde{A}_{\text{os}}^{(i)}, \tilde{c}_{\text{os}}^{(i)}, \tilde{z}_{\text{os}}^{(i)})$ being a valid Σ_{os} transcript for the statement $\mathbb{X}_{\text{os}}$. The fifth equality follows from linearity of $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$. The last equality follows from how c_{os} and z_{os} are defined. The final implication is due to the definition of $\Sigma_{\text{os}}.\text{Ver}$. Thus, we have proved the first part of the lemma.

Then, given that $W = w_0G + w_1H$ and $H = hG$, we obtain

$$\begin{aligned}
A_{\text{os}} &= \Phi_{\text{os}}^{\text{par}_{\text{os}}}(z_{\text{os}}) - c_{\text{os}} \cdot W \\
&\implies A_{\text{os}} = (z_{\text{os},0} - c_{\text{os}}w_0)G + (z_{\text{os},1} - c_{\text{os}}w_1)H \\
&\implies A_{\text{os}} = (z_{\text{os},0} - c_{\text{os}}w_0 + h(z_{\text{os},1} - c_{\text{os}}w_1))G. \quad \square
\end{aligned}$$

7.2 Proof of Theorem 5.4

Let $\mathcal{A}$ be an adversary on the OMSUF game of BS_{phf} , running in time $T_{\mathcal{A}}$, inducing at most Q_S, Q_H, Q_{pb} queries to oracles S_1, H and H_{pb} , respectively. (Note that we count both $\mathcal{A}$'s direct random oracle queries and the queries made by the game during simulation.) Denote by $Q_{S, \text{info}}$ the number of S_1 queries with common message info . Note that $Q_S = \sum_{\text{info}} Q_{S, \text{info}}$. Without loss of generality, we may assume that $Q_S > 0$ (as $\mathcal{A}$ may query $\text{info} \neq \text{info}^*$ arbitrarily often). We may assume that $\mathcal{A}$ finishes all sessions, i.e., it also makes Q_S calls to S_2 , due to [CART26, Lemma 2.5]. Also, we assume that all H and H_{pb} queries that $\mathcal{A}$ makes are distinct for simplicity.²⁰

Let us establish some notation. We index all values being sent in the j -th signing session (i.e., with session identifier $\text{sid} = j$) with subscript j . At the end of the game, the adversary outputs ℓ message-signature pairs $\{\text{msg}^{(i)}, \tilde{\sigma}^{(i)}\}_{i \in [\ell]}$ for common message info^* . We assume that $\ell = Q_{S, \text{info}^*} + 1$ for simplicity. Also, we assume that the message-signature pairs verify correctly for info^* . Let $U = H_{\text{pb}}(\text{info}^*)$. We parse

$$\tilde{\sigma}^{(i)} = (C_{\text{s}}^{(i)}, \tilde{A}_{\text{s}}^{(i)}, \tilde{T}_{\text{s}}^{(i)}, \tilde{e}_{\text{s}}^{(i)}, C_{\text{rnd}}^{(i)}, \pi_{\text{sle}}^{(i)}, \tilde{\tau}_{\text{os}}^{(i)}, \tilde{\tau}_{\text{dl}}^{(i)}, \tilde{\tau}_{\text{phf}}^{(i)}),$$

where $\tilde{\tau}_{\text{xxx}}^{(i)} = (\tilde{A}_{\text{xxx}}^{(i)}, \tilde{c}_{\text{xxx}}^{(i)}, \tilde{z}_{\text{xxx}}^{(i)})$ for $\text{xxx} \in \{\text{os}, \text{dl}, \text{phf}\}$. We further parse

$$\begin{aligned} \text{hqry}^{(i)} &= (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i)}, \tilde{A}_{\text{os}}^{(i)}, \tilde{A}_{\text{dl}}^{(i)}, \tilde{A}_{\text{phf}}^{(i)}, \mathbb{x}_{\text{sle}}^{(i)}, \pi_{\text{sle}}^{(i)}, \text{msg}^{(i)}), \\ \mathbb{x}_{\text{os}} &= (G, H, W), & \mathbb{x}_{\text{dl}} &= (G, X), \\ \mathbb{x}_{\text{phf}}^{(i)} &= (G, V, U, C_{\text{s}}^{(i)}, \tilde{A}_{\text{s}}^{(i)}, \tilde{T}_{\text{s}}^{(i)}, \tilde{e}_{\text{s}}^{(i)}), & \mathbb{x}_{\text{sle}}^{(i)} &= (\text{ck}_{\text{ped}}, \mathbb{x}_{\text{os}}, C_{\text{s}}^{(i)}, \tilde{A}_{\text{s}}^{(i)}, C_{\text{rnd}}^{(i)}, \tilde{A}_{\text{os}}^{(i)}). \end{aligned}$$

We consider the following sequence of games, where $\Pr[\mathbf{G}^{\mathcal{A}} \Rightarrow 1]$ denotes the probability of $\mathcal{A}$ winning in game $\mathbf{G}$.

Game $\mathbf{G}_0$ (Original OMSUF Game): This game samples the public key $\text{pk} = (W, X)$ and secret key $\text{sk} = (w, x, \text{pk})$ as in the scheme, simulates the random oracles H and H_{pb} via lazy sampling, and runs the signing oracles honestly (cf. Definition 3.5). The adversary $\mathcal{A}$ wins if all forgeries $(\text{msg}^{(i)}, \sigma^{(i)})_{i \in [\ell]}$ verify for common message info^* and the pairs $(\text{msg}^{(i)}, \sigma^{(i)})$ are pairwise distinct. (Recall that we assume that $\ell = Q_{S, \text{info}^*} + 1$.) We have

$$\text{Adv}_{\text{BS}_{\text{phf}}}^{\text{omsuf}}(\mathcal{A}, \lambda) = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_1$ (Extract w_{sle} from π_{sle}): Denote by Ext an extractor for Π_{sle} (cf. definition 3.24). On the j th H query hqry_j , the game extracts a witness w_j from the proof $\pi_{\text{sle}, j}$ from hqry_j if the proof is valid and hqry_j follows the format in $\text{BS}_{\text{phf}}.\text{Ver}$. Also, we introduce additional abort conditions. In more detail, on the j th H query of the form

$$\text{hqry}_j = (\mathbb{x}_{\text{os}, j}, \mathbb{x}_{\text{dl}, j}, \tilde{\mathbb{x}}_{\text{phf}, j}, \tilde{A}_{\text{os}, j}, \tilde{A}_{\text{dl}, j}, \tilde{A}_{\text{phf}, j}, \mathbb{x}_{\text{sle}, j}, \pi_{\text{sle}, j}, \text{msg}_j),$$

the game runs $b = \Pi_{\text{sle}}.\text{Ver}^H(\mathbb{x}_{\text{sle}, j}, \pi_{\text{sle}, j})$. If $b = 1$, the game extracts a witness $w_{\text{sle}, j} \leftarrow \text{Ext}(Q_{H_{\text{sle}}}, \mathbb{x}_{\text{sle}, j}, \pi_{\text{sle}, j})$ from $\pi_{\text{sle}, j}$, where $Q_{H_{\text{sle}}}$ denotes the hash queries to H_{sle} . The game aborts if $(\mathbb{x}_{\text{sle}, j}, w_{\text{sle}, j}) \notin \mathfrak{R}_{\text{sle}}$. Furthermore, when $\mathcal{A}$ outputs its signatures for info^* and all signatures verify, the game recovers the hash query index $j^{(i)} \in [Q_H]$ which is associated to the i th forgery, i.e., such that $\text{hqry}_{j^{(i)}} = \text{hqry}^{(i)}$. Note that since we assume that all H queries are distinct and we count the game's queries towards Q_H , the index $j^{(i)}$ is well-defined. We parse

$$w_{\text{sle}}^{(i)} = (\nu^{(i)}, r_{\text{s}}^{(i)}, \alpha_{\text{os}}^{(i)}, \gamma_{\text{os}}^{(i)}, r_{\text{rnd}}^{(i)})$$

and let

$$A_{\text{os}}^{(i)} := \tilde{A}_{\text{s}}^{(i)} - \nu^{(i)} V.$$

The game further aborts if $(\mathbb{x}_{\text{sle}}^{(i)}, w_{\text{sle}}^{(i)}) \notin \mathfrak{R}_{\text{sle}}$. The following statement relates the advantage of $\mathcal{A}$ in $\mathbf{G}_0$ and $\mathbf{G}_1$; we refer to Section 7.3 for the formal proof.

²⁰ As we count the game's queries towards Q_{pb} , we also assume that $\mathcal{A}$ queries $H_{\text{pb}}(\text{info})$ before making a query to S_1 with common message info for convenience.

Lemma 7.3. *There are adversaries $\mathcal{B}_{\text{slc}}$ and $\mathcal{B}_{\text{dl}}$ with running time roughly that of $\mathcal{A}$ such that*

$$|\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\Pi_{\text{slc}}, \text{Ext}, \hat{\mathcal{R}}_{\text{slc}}}^{\text{slc}}(\mathcal{B}_{\text{slc}}, \lambda) + \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p}.$$

Game $\mathbf{G}_2$ (No signature with same H query): After $\mathcal{A}$ outputs its signatures, the game aborts if all signatures verify and there exist distinct indices $i_0, i_1 \in [\ell]$ such that their corresponding H query is identical. That is, the game aborts if $\text{hqry}^{(i_0)} = \text{hqry}^{(i_1)}$. Denote by **SameQuery** the event that an abort occurs in $\mathbf{G}_2$ but not in $\mathbf{G}_1$. We have

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] \leq \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] + \Pr[\text{SameQuery}].$$

The following lemma bounds $\Pr[\text{SameQuery}]$; the proof is given in Section 7.4.

Lemma 7.4. *There is an adversary $\mathcal{B}_{\text{dl}}$ with running time roughly that of $\mathcal{A}$ such that*

$$\begin{aligned} \Pr_{\mathbf{G}_1^{\mathcal{A}}}[\text{SameQuery}] &\leq 3 \cdot \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{4Q_{\text{H}} + 1}{p - 1} \\ &\quad + \sqrt[3]{32Q_{\text{H}}^2 \cdot \left(\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p - 1} \right)}. \end{aligned}$$

CASE DISTINCTION. We now consider the following events.

- **NonceColl:** The adversary $\mathcal{A}$ wins in $\mathbf{G}_2$ and there exists distinct $i_0, i_1 \in [\ell]$ such that $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$.
- **NonceDist:** The adversary $\mathcal{A}$ wins in $\mathbf{G}_2$ and for all distinct $i_0, i_1 \in [\ell]$ we have $A_{\text{os}}^{(i_0)} \neq A_{\text{os}}^{(i_1)}$.

Note that these events are efficiently checkable and disjoint. Also, $\text{NonceColl} \vee \text{NonceDist}$ is equivalent to $\mathcal{A}$ winning in $\mathbf{G}_2$, therefore

$$\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceColl}] + \Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceDist}].$$

Lemmas 7.5 and 7.6 bound the probability that **NonceColl** and **NonceDist** occur, respectively. We give proof sketches below and refer to Sections 7.5 and 7.6 for details. Combining the bounds above with the lemmata below, we conclude the theorem. $\square$

Lemma 7.5. *There exists an adversary $\mathcal{B}_{\text{dl}}$ with running time roughly that of $\mathcal{A}$ such that*

$$\Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceColl}] \leq 2 \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{Q_{\text{H}}^2}{p}.$$

Proof (sketch). This case is similar to the argument in [CART26, Lemma 6.2], except that our proof does not require rewinding due to the reliance on Π_{slc} . Here, there is a collision in the Okamoto-Schnorr nonces $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$. Since the blinding factors are extracted from π_{slc} , we may compute two valid transcripts $\tau_{\text{os}}^{(i_0)}$ and $\tau_{\text{os}}^{(i_1)}$ for $\mathbf{x}_{\text{os}}$ with identical nonce according to Lemma 7.2 from $\tilde{\tau}_{\text{os}}^{(i_0)}$ and $\tilde{\tau}_{\text{os}}^{(i_1)}$, respectively. Observe that the challenges $\tilde{c}_{\text{os}}^{(i_0)}$ and $\tilde{c}_{\text{os}}^{(i_1)}$ of $\tilde{\tau}_{\text{os}}^{(i_0)}$ and $\tilde{\tau}_{\text{os}}^{(i_1)}$, respectively, are distinct w.h.p. since the H queries $\text{hqry}^{(i_0)} \neq \text{hqry}^{(i_1)}$ in the signatures are different. Furthermore, the blinding factors are determined before $\tilde{c}_{\text{os}}^{(i_0)}$ and $\tilde{c}_{\text{os}}^{(i_1)}$ are sampled in H. This implies that $\tau_{\text{os}}^{(i_0)}$ and $\tau_{\text{os}}^{(i_1)}$ have distinct challenges w.h.p. and we may compute a witness $\mathbf{w}_{\text{os}}$ for $\mathbf{x}_{\text{os}}$. By witness indistinguishability, we have $\mathbf{w}_{\text{os}} \neq \mathbf{w}$, where $\mathbf{w}$ is the game's witness for Σ_{os} , which allows to compute the DL of H . We formalize the argument using algebraic one-more preimage resistance for ϕ_{os} (cf. Figure 12) to simulate signing, while $\mathcal{A}$'s signatures yield the additional preimage required to break the game.

Lemma 7.6. *Let $\delta = \frac{d}{Q_{\text{S}}^2(n+1)}$ with d as defined in Theorem 3.11. There is an adversary $\mathcal{B}_{\text{dl}}$ with running time roughly that of $\mathcal{A}$ such that*

$$\Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceDist}] \leq \frac{Q_{\text{pb}}}{\delta} \cdot \left(\sqrt[3]{32Q_{\text{H}}^2 \cdot \left(\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p - 1} \right)} + \frac{4Q_{\text{H}}}{p - 1} \right).$$

Proof (sketch). In this case, there is a fresh nonce $A_{\text{os}}^{(i*)}$ extracted from $\mathcal{A}$'s signatures that never appeared in the signing queries. Here, we rely on Lemma 7.1 and Theorem 3.11 to simulate signing while enforcing that the PHF evaluates the exponent of $A_{\text{os}}^{(i*)}$ to zero. Since the adversary provides a Σ_{phf} proof (or-compiled with Σ_{dl}) in $\tilde{\sigma}^{(i*)}$ that proves that the PHF evaluates to a non-zero value, either we may compute the DL of X or we get a contradiction. Note that the contradiction is only computational (more precisely, in this case we may extract the DL of V). As neither the DL of X nor V is required for simulation, this yields the desired reduction.

7.3 Proof of Lemma 7.3 (Ruling out Relaxed SLE Witness)

We consider the following intermediate game $\mathbf{G}_{0.1}$ between $\mathbf{G}_0$ and $\mathbf{G}_1$.

Game $\mathbf{G}_{0.1}$ (Extract arbitrary $\mathbb{w}_{\text{sle}}$ from π_{sle}): This game is identical to $\mathbf{G}_1$, except that the last abort condition $(\mathbb{x}_{\text{sle}}^{(i)}, \mathbb{w}_{\text{sle}}^{(i)}) \notin \mathfrak{R}_{\text{sle}}$ is omitted. That is, on the j th H query of the form

$$\text{hqry}_j = (\mathbb{x}_{\text{os},j}, \mathbb{x}_{\text{dl},j}, \tilde{\mathbb{x}}_{\text{phf},j}, \tilde{A}_{\text{os},j}, \tilde{A}_{\text{dl},j}, \tilde{A}_{\text{phf},j}, \mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j}, \text{msg}_j),$$

the game runs $b = \Pi_{\text{sle}}.\text{Ver}^H(\mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j})$, and if $b = 1$, the game extracts a witness $\mathbb{w}_{\text{sle},j} \leftarrow \text{Ext}(\mathcal{Q}_{\text{H}_{\text{sle}}}, \mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j})$ from $\pi_{\text{sle},j}$, where $\mathcal{Q}_{\text{H}_{\text{sle}}}$ denotes the hash queries to H_{sle} . The game aborts if $(\mathbb{x}_{\text{sle},j}, \mathbb{w}_{\text{sle},j}) \notin \mathfrak{R}_{\text{sle}}$.

We construct a reduction $\mathcal{B}_{\text{sle}}$ to straightline extractability of Π_{sle} for relation $\hat{\mathfrak{R}}_{\text{sle}}$ that bounds the abort probability. The reduction $\mathcal{B}_{\text{sle}}$ simulates the game to $\mathcal{A}$ as in $\mathbf{G}_0$, except for the following changes:

- It simulates the random oracle queries to H_{sle} via the random oracle of its own game.
- On the j th H query hqry_j of the above form, the reduction employs its verification oracle $\mathcal{O}$ (which is either equal to $\mathcal{O}_{\text{Ver}}$ or $\mathcal{O}_{\text{Ext}}$) to verify the proofs $\pi_{\text{sle},j}$. That is, it sets $b_j \leftarrow \mathcal{O}(\mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j})$. If $b_j \neq \Pi_{\text{sle}}.\text{Ver}_{\text{sle}}^H(\mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j})$, the game outputs 0.

After the interaction with $\mathcal{A}$ and the verification of $\mathcal{A}$'s signatures, the game outputs 1 if $\mathcal{A}$ wins (and $b_j = \Pi_{\text{sle}}.\text{Ver}_{\text{sle}}^H(\mathbb{x}_{\text{sle},j}, \pi_{\text{sle},j})$ for all $j \in [Q_H]$), else 0.

Observe that if $\mathcal{O} = \mathcal{O}_{\text{Ver}}$, then $\mathcal{B}_{\text{sle}}$ outputs 1 if and only if $\mathcal{A}$ wins $\mathbf{G}_0$. Else, if $\mathcal{O} = \mathcal{O}_{\text{Ext}}$, the reduction outputs 1 if and only if $\mathcal{A}$ wins $\mathbf{G}_0$ and does not trigger the newly introduced abort condition in $\mathbf{G}_{0.1}$. Thus, we have that

$$|\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_{0.1}^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\Pi_{\text{sle}}, \text{Ext}, \hat{\mathfrak{R}}_{\text{sle}}}^{\text{sle}}(\mathcal{B}_{\text{sle}}, \lambda).$$

Let us now relate the advantages of $\mathcal{A}$ in $\mathbf{G}_{0.1}$ and $\mathbf{G}_1$. Below, we follow the notation introduced in $\mathbf{G}_1$. Note that $\mathbf{G}_1$ is identical to $\mathbf{G}_{0.1}$, except that $\mathbf{G}_1$ aborts if $(\mathbb{x}_{\text{sle}}^{(i)}, \mathbb{w}_{\text{sle}}^{(i)}) \notin \mathfrak{R}_{\text{sle}}$ for some $i \in [\ell]$ and all of $\mathcal{A}$'s signatures verify. Since $\hat{\mathfrak{R}}_{\text{sle}} = \mathfrak{R}_{\text{sle}} \cup \mathfrak{R}_{\text{zero}}$, it suffices to bound the probability that $(\mathbb{x}_{\text{sle}}^{(i)}, \mathbb{w}_{\text{sle}}^{(i)}) \in \mathfrak{R}_{\text{zero}}$ in $\mathbf{G}_{0.1}$. Let $i^* \in [\ell]$ denote the smallest index such that $(\mathbb{x}_{\text{sle}}^{(i*)}, \mathbb{w}_{\text{sle}}^{(i*)}) \in \mathfrak{R}_{\text{zero}}$. Recall that $\mathbb{w}_{\text{sle}}^{(i*)} = (\nu^{(i*)}, r_{\text{s}}^{(i*)}, \alpha_{\text{os}}^{(i*)}, \gamma_{\text{os}}^{(i*)}, r_{\text{rnd}}^{(i*)})$. Furthermore, by definition of $\mathfrak{R}_{\text{zero}}$ (cf. Equation (12)), we have that $\mathbb{w}_{\text{sle}}^{(i*)} \neq 0$ and

$$\begin{aligned} 0 &= \nu^{(i*)}G + r_{\text{s}}^{(i*)}V, \\ 0 &= r_{\text{rnd}}^{(i*)}V + \alpha_1^{(i*)}G_1 + \alpha_2^{(i*)}G_2 + \gamma_{\text{os}}^{(i*)}G_3. \end{aligned}$$

Observe that $\mathbf{G}_{0.1}$ samples the group elements V, G_1, G_2, G_3 at random, conditioned on $V \neq 0$. Note that if $V = 0$, it is straightforward to compute a non-trivial Rel-DL solution (any non-zero coefficient of V suffices). Therefore, there is a reduction $\mathcal{B}_{\text{rel-dl}}$ to the Rel-DL assumption such that

$$|\Pr[\mathbf{G}_{0.1}^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\text{Gen}, 4}^{\text{rel-dl}}(\mathcal{B}_{\text{rel-dl}}, \lambda).$$

Collecting the bounds and applying Lemma 3.3 yields the desired bound.

7.4 Proof of Lemma 7.4 (Ruling out H-Query Collision in Forgeries)

We need to bound $\Pr[\text{SameQuery}]$. Denote by $(i_0, i_1) \in [\ell]^2$ the lexicographically smallest pair such that $\text{hqry}^{(i_0)} = \text{hqry}^{(i_1)}$. By definition, we have

$$\begin{aligned}\text{hqry}^{(i_0)} &= (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i_0)}, \tilde{A}_{\text{os}}^{(i_0)}, \tilde{A}_{\text{dl}}^{(i_0)}, \tilde{A}_{\text{phf}}^{(i_0)}, \mathbb{x}_{\text{sle}}^{(i_0)}, \pi_{\text{sle}}^{(i_0)}, \text{msg}^{(i_0)}) , \\ \text{hqry}^{(i_1)} &= (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i_1)}, \tilde{A}_{\text{os}}^{(i_1)}, \tilde{A}_{\text{dl}}^{(i_1)}, \tilde{A}_{\text{phf}}^{(i_1)}, \mathbb{x}_{\text{sle}}^{(i_1)}, \pi_{\text{sle}}^{(i_1)}, \text{msg}^{(i_1)}) .\end{aligned}$$

Let us define the following events:

- **SameChal**: The event **SameQuery** occurs and $\tilde{c}_{\text{dl}}^{(i_0)} = \tilde{c}_{\text{dl}}^{(i_1)}$.
- **DistChal**: The event **SameQuery** occurs and $\tilde{c}_{\text{dl}}^{(i_0)} \neq \tilde{c}_{\text{dl}}^{(i_1)}$.

Note that the events are disjoint and **SameChal** $\vee$ **DistChal** is equivalent to **SameQuery**. Thus, we have

$$\Pr[\text{SameQuery}] = \Pr[\text{SameChal}] + \Pr[\text{DistChal}] .$$

BOUNDING SameChal. To bound the probability that **SameChal** occurs, we introduce some intermediate games $\mathbf{G}_{X,i}$.

Game $\mathbf{G}_{X,0}$ ($\mathbf{G}_1$ with winning condition as SameChal): This game is identical to $\mathbf{G}_1$ except that $\mathcal{A}$ wins if and only if **SameChal** occurs. We have

$$\Pr[\text{SameChal}] = \Pr[\mathbf{G}_{X,0}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{X,1}$ (Abort if Σ_{os} transcripts distinct): When $\mathcal{A}$ outputs its signatures, the game aborts if $\tilde{\tau}_{\text{os}}^{(i_0)} \neq \tilde{\tau}_{\text{os}}^{(i_1)}$.

We give a reduction $\mathcal{B}_{\text{dl}}$ to the DL problem to upper bound the probability of an abort. On input (G, Y) , $\mathcal{B}_{\text{dl}}$ simulates $\mathbf{G}_{X,0}$ to $\mathcal{A}$, setting $H = Y$. When $\mathcal{A}$ outputs its forgeries, the reduction invokes the relaxed unique responses extractor $\Sigma_{\text{os}}.\text{URExt}$ on input of statement (G, H, W) and transcripts $\tilde{\tau}_{\text{os}}^{(i_0)}$ and $\tilde{\tau}_{\text{os}}^{(i_1)}$. If it obtains a witness $(y_0, y_1) \in \mathbb{Z}_p^2$ such that $y_0 G + y_1 H = 0$ and $(y_0, y_1) \neq \mathbf{0}$, then the reduction $\mathcal{B}_{\text{dl}}$ outputs $y = -y_0/y_1 \bmod p$. Note that $(\tilde{A}_{\text{os}}^{(i_0)}, \tilde{c}_{\text{os}}^{(i_0)}) = (\tilde{A}_{\text{os}}^{(i_1)}, \tilde{c}_{\text{os}}^{(i_1)})$ since $\text{hqry}^{(i_0)} = \text{hqry}^{(i_1)}$ (recall that the Σ_{os} challenge is determined by the H evaluation). Therefore, we have $\tilde{z}_{\text{os}}^{(i_0)} \neq \tilde{z}_{\text{os}}^{(i_1)}$ if an abort occurs. By the relaxed unique response property for relation $\tilde{\mathfrak{R}}_{\text{os}}$ we obtain

$$|\Pr[\mathbf{G}_{X,0}^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_{X,1}^{\mathcal{A}} \Rightarrow 1]| = \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) .$$

Remark 7.7. If $\mathcal{A}$ is successful in $\mathbf{G}_{X,1}$, we have $(\tilde{A}_{\text{phf}}^{(i_0)}, \tilde{c}_{\text{phf}}^{(i_0)}) = (\tilde{A}_{\text{phf}}^{(i_1)}, \tilde{c}_{\text{phf}}^{(i_1)})$ but $\tilde{z}_{\text{phf}}^{(i_0)} \neq \tilde{z}_{\text{phf}}^{(i_1)}$.

Proof. Recall that $\text{hqry}^{(i)} = (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i)}, \tilde{A}_{\text{os}}^{(i)}, \tilde{A}_{\text{dl}}^{(i)}, \tilde{A}_{\text{phf}}^{(i)}, \mathbb{x}_{\text{sle}}^{(i)}, \pi_{\text{sle}}^{(i)}, \text{msg}^{(i)})$ and since **SameQuery** occurs, we have $\text{hqry}^{(i_0)} = \text{hqry}^{(i_1)}$. As **SameChal** occurs, we have $\tilde{c}_{\text{dl}}^{(i_0)} = \tilde{c}_{\text{dl}}^{(i_1)}$. By unique responses of Σ_{dl} , it must hold that $\tilde{\tau}_{\text{dl}}^{(i_0)} = \tilde{\tau}_{\text{dl}}^{(i_1)}$. Further, we also have $\tilde{c}_{\text{phf}}^{(i_0)} = \tilde{c}_{\text{phf}}^{(i_1)}$ since

$$\tilde{c}_{\text{dl}}^{(i_0)} \cdot \tilde{c}_{\text{phf}}^{(i_0)} = \text{H}(\text{hqry}^{(i_0)}) = \text{H}(\text{hqry}^{(i_1)}) = \tilde{c}_{\text{dl}}^{(i_1)} \cdot \tilde{c}_{\text{phf}}^{(i_1)} .$$

Therefore, if $\mathcal{A}$ is successful in $\mathbf{G}_{X,1}$, it must hold that $\tilde{z}_{\text{phf}}^{(i_0)} \neq \tilde{z}_{\text{phf}}^{(i_1)}$ as the i_0 th and i_1 th signatures are distinct.

Game $\mathbf{G}_{X,2}$ (Set up $H = hG$ with known discrete logarithm): The game samples $h \leftarrow \mathbb{Z}_p$ and sets $H = hG$ instead of sampling $H \leftarrow \mathbb{G}$. We have

$$\Pr[\mathbf{G}_{X,1}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{X,2}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{X.3}$ (Abort if non-trivial DL relation found): When $\mathcal{A}$ outputs its signatures, the game runs Σ_{phf} 's unique responses extractor

$$\mathbb{w} \leftarrow \Sigma_{\text{phf}}.\text{URExt}(\tilde{\mathbf{x}}_{\text{phf}}^{(i_0)}, \tilde{\tau}^{(i_0)}, \tilde{\tau}^{(i_1)})$$

on input of statement $\tilde{\mathbf{x}}_{\text{phf}}^{(i_0)} = \tilde{\mathbf{x}}_{\text{phf}}^{(i_1)}$ and transcripts $\tilde{\tau}^{(i_0)}, \tilde{\tau}^{(i_1)}$. The game obtains $\mathbb{w}$ and aborts if $\mathbb{w}G = V$ or if $\mathbb{w} = (\alpha, \beta, \nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu})$ with $(\nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu}) \neq \mathbf{0}$.

By Remark 7.7 and since Σ_{phf} has relaxed unique responses for relation $\tilde{\mathfrak{R}}_{\text{phf}}$, we have $(\tilde{\mathbf{x}}_{\text{phf}}^{(i_0)}, \mathbb{w}) \in \tilde{\mathfrak{R}}_{\text{phf}}$.

By definition of $\tilde{\mathfrak{R}}_{\text{phf}}$, we have that $\Phi_{\text{phf}}^{\tilde{T}_{\mathbb{s}}^{(i_0)}}(\mathbb{w}) = \mathbf{0}$ and $\mathbb{w} \neq \mathbf{0}$ or otherwise, we obtain $\mathbb{w}G = V$. We give a reduction $\mathcal{B}_{\text{rel-dl}}$ to the Rel-DL assumption to upper bound the abort probability. The reduction obtains as input G and $N := n \cdot Q_{\text{pb}} + 1$ group elements $\mathbf{Y} \in \mathbb{G}^N$. It simulates the game $\mathbf{G}_{X.2}$ to $\mathcal{A}$, except for the following modifications:

- It sets $V = Y_1$ in the public parameters pp during setup. (If $V = 0$, it outputs $\beta = (1, \mathbf{0})$ as its non-trivial Rel-DL solution.)
- On the j th query to H_{pb} , it sets $\mathbf{U} = (U_0, Y_{1+(j-1) \cdot n+1}, \dots, Y_{1+j \cdot n})$ for random $U_0 \leftarrow_{\mathbb{S}} \mathbb{G}$.

Clearly, the simulation is perfect (except if $V = 0$ which is handled explicitly). If the abort event introduced in $\mathbf{G}_{X.3}$ occurs, we have that $\mathbb{w}G = V$ or $\Phi_{\text{phf}}^{\tilde{T}_{\mathbb{s}}^{(i_0)}}(\mathbb{w}) = \mathbf{0}$. Furthermore, by definition of $\Phi_{\text{phf}}^{\tilde{T}_{\mathbb{s}}^{(i_0)}}$, we have that

$$\mathbf{0} = \begin{bmatrix} \nu G + r_{\mathbb{s}} V \\ \nu V + \sum_{i \in [n]} \mu_i (2^{i-1} G) \\ \rho V + \sum_{i \in [n]} \mu_i U_i \\ \alpha \tilde{T}_{\mathbb{s}}^{(i_0)} + \beta V \end{bmatrix}^{\top}$$

By the abort conditions, we have $\mathbb{w}G = V$ or $(\nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu}) \neq \mathbf{0}$. In any case, adversary $\mathcal{B}_{\text{rel-dl}}$ may compute a non-zero β such that $\beta_0 G + \sum_{i \in [N]} \beta_i Y_i = 0$. We therefore obtain

$$|\Pr[\mathbf{G}_{X.2}^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_{X.3}^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\text{GGen}, N}^{\text{rel-dl}}(\mathcal{B}_{\text{rel-dl}}, \lambda).$$

Remark 7.8. If $\mathcal{A}$ is successful in $\mathbf{G}_{X.3}$, it must hold that $\mathbb{w} = (\alpha, \beta, \nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu})$ with $(\alpha, \beta) \neq 0$ and $\alpha T_{\mathbb{s}}^{(i_0)} + \beta V = 0$.

Proof. This is a consequence of the above discussions. In particular, the abort event introduced in $\mathbf{G}_{X.3}$ excludes that $\mathbb{w}G = V$. Thus, we have that $\Phi_{\text{phf}}^{\tilde{T}_{\mathbb{s}}^{(i_0)}}(\mathbb{w}) = \mathbf{0}$ since Σ_{phf} has relaxed unique responses for $\tilde{\mathfrak{R}}_{\text{phf}}$. Note that $\mathbb{w} = (\alpha, \beta, \nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu}) \neq \mathbf{0}$. We further exclude that $(\nu, \rho, r_{\mathbb{s}}, \boldsymbol{\mu}) \neq \mathbf{0}$. Therefore, it must hold that $(\alpha, \beta) \neq \mathbf{0}$.

Note that since $T_{\mathbb{s}}^{(i_0)}$ is adversarially-chosen, the representation (α, β) does not yet constitute a solution for a hard DL relation. We therefore aim to find a more controlled representation of $T_{\mathbb{s}}^{(i_0)}$ by rewinding. Since we may also extract a DL of X due to the OR-proof structure, we must first simulate Σ_{dl} while ensuring that the to-be-extracted representation of $T_{\mathbb{s}}^{(i_0)}$ remains useful.

Game $\mathbf{G}_{X.4}$ (Set up $\mathbf{U}$ to prepare Σ_{phf} simulation): The game sets up the outputs $\mathbf{U}$ in H_{pb} via PHF.NonZeroGen . That is, on every H_{pb} query info , the game samples $(\mathbf{U}, \text{td}_{\text{info}}) \leftarrow_{\mathbb{S}} \text{PHF.NonZeroGen}(1^{\lambda}, G, V)$ with $\text{td}_{\text{info}} = (\mathbf{a}_{\text{info}}, \mathbf{b}_{\text{info}})$. The game keeps track of the trapdoors td_{info} for every input info . Note that in $\mathbf{G}_{X.3}$, the H_{pb} outputs $\mathbf{U}$ are distributed at random which is exactly the distribution of $\mathbf{K} \leftarrow \text{PHF.Gen}(1^{\lambda})$. By the *well-distributed keys* property (cf. Lemma 3.12), we have that

$$\Pr[\mathbf{G}_{X.3}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{X.4}^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_{X.5}$ (Simulate Σ_{dl} and compute Σ_{phf} honestly): The game computes all nonces as $A_{\text{os},j} \leftarrow \Sigma_{\text{os}}.\text{Init}(\mathbb{x}_{\text{os}}, \mathbb{w}_{\text{os}})$ directly after sampling its secret key. Then, it employs the nonce $A_{\text{os},j}$ in the j th signing session. Also, it parses $A_{\text{os},j} = r_{j,0}G + r_{j,1}H$ and sets

$$r_j := r_{j,0} + h \cdot r_{j,1} \in \mathbb{Z}_p,$$

where $H = hG$. It decomposes $r_j = \sum_{i \in [n]} \mu_{j,i} 2^{i-1}$ in binary, where $\mu_{j,i} \in \{0,1\}$, and denotes $\boldsymbol{\mu}_j = (\mu_{j,1}, \dots, \mu_{j,n})$. Furthermore, in the j th signing session for $j \in [Q_S]$, the game computes the values $(T_{\$,j}, e_{\$,j})$ and the Σ_{dl} and Σ_{phf} transcripts $(A_{\text{dl},j}, c_{\text{dl},j}, z_{\text{dl},j})$ and $(A_{\text{phf},j}, c_{\text{phf},j}, z_{\text{phf},j})$ as in ExpPHF (cf. Figure 11).

Observe that by construction, we have $A_{\text{os},j} = \sum_{i \in [n]} \mu_{j,i} (2^{i-1}G)$. Also, due to the changes in $\mathbf{G}_{X.4}$ and by the *well-distributed logarithms* property (cf. Lemma 3.12), we have $e_j = a_0 + \sum_{i \in [n]} \mu_{j,i} a_{\text{info},i} \neq 0$.

Therefore, the conditions for Lemma 7.1 are satisfied and since in $\mathbf{G}_{X.4}$, the j th signing session is distributed as in ExpDL, we have

$$\Pr[\mathbf{G}_{X.4}^A \Rightarrow 1] = \Pr[\mathbf{G}_{X.5}^A \Rightarrow 1].$$

We are now ready to reduce $\mathcal{A}$ winning in $\mathbf{G}_{X.5}$ to the Rel-DL assumption. For this, observe that if we obtain a representation $T_{\$}^{(i_0)} = \rho^*V + U_0 + \sum_{i \in [n]} \mu_i^* U_i$ with $\boldsymbol{\mu}^* \in \{0,1\}^n$, then we have $\tilde{T}_{\$}^{(i_0)} = eG + (f + \rho^*)V$, where $e \neq 0$ and $(e, f) = \text{PHF}.\text{TrapEval}(\text{td}_{\text{info}*}, \boldsymbol{\mu}^*)$. By Remark 7.8, we also have $(\alpha, \beta) \neq 0$ with $\alpha \tilde{T}_{\$}^{(i_0)} + \beta V = 0$. Thus, $\alpha e \neq 0$ and we obtain

$$\alpha(eG + (f + \rho^*)V) + \beta V = 0,$$

and therefore $v = -\frac{\alpha e}{\beta + \alpha(f + \rho^*)}$ is the DL of V . Below, we describe how to derive the desired representation of $T_{\$}^{(i_0)}$ (or compute the DL of V directly) via rewinding.

REDUCTION TO DL. As described above, we exhibit an adversary $\mathcal{B}_{\text{dl}}$ on the DL assumption with advantage roughly that of $\mathcal{A}$ in $\mathbf{G}_{X.5}$ (albeit a polynomial loss due to rewinding). First, we describe a wrapper $\mathcal{W}_{\text{SameQuery}}$ that simulates $\mathbf{G}_{X.5}$ to $\mathcal{A}$, taking as input a DL challenge G, Y and challenge vector $\mathbf{c} = (c_1, \dots, c_{Q_H}) \in (\mathbb{Z}_p^\times)^{Q_H}$.

Wrapper $\mathcal{W}_{\text{SameQuery}}((G, Y), \mathbf{c})$: On input $G, Y \in \mathbb{G}$ and $\mathbf{c} = (c_1, \dots, c_{Q_H}) \in (\mathbb{Z}_p^\times)^{Q_H}$, the wrapper $\mathcal{W}_{\text{SameQuery}}$ simulates $\mathbf{G}_{X.5}$ to $\mathcal{A}$ as above, except for the following changes:

- The wrapper sets $\text{par} := (n, G, H, \text{ck}_{\text{ped}}, V)$, where $V := Y$; the other values are sampled as in $\mathbf{G}_{X.5}$.
- The wrapper sets $X := Y + x'G$ and $\text{sk} = (\mathbf{w}, \perp, \text{pk})$, where $x' \leftarrow \mathbb{Z}_p$.
- On the i th query to $\mathcal{H}$, the wrapper outputs $(\tilde{c}_{\text{os},i}, c_i)$ for $\tilde{c}_{\text{os},i} \leftarrow \mathbb{Z}_p$.

If $\mathcal{A}$ does *not* succeed, according to the winning conditions in $\mathbf{G}_{X.5}$, the wrapper outputs $(\perp, \perp)$. Else, denote by $I \in [Q_H]$ the index of the hash query

$$\text{hqry}^{(i_0)} = (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i_0)}, \tilde{A}_{\text{os}}^{(i_0)}, \tilde{A}_{\text{dl}}^{(i_0)}, \tilde{A}_{\text{phf}}^{(i_0)}, \mathbb{x}_{\text{sle}}^{(i_0)}, \pi_{\text{sle}}^{(i_0)}, \text{msg}^{(i_0)})$$

associated to the i_0 th message-signature pair $(\text{msg}^{(i_0)}, \tilde{\sigma}^{(i_0)})$. Further, let

$$\text{aux} := (\text{hqry}^{(i_0)}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i_0)}, \tilde{\sigma}^{(i_0)}, \text{td}_{\text{info}*}, \alpha, \beta),$$

where (α, β) is defined as in $\mathbf{G}_{X.3}$. The wrapper outputs (I, aux) .

Note that $\mathcal{W}_{\text{SameQuery}}$ simulates $\mathbf{G}_{X.5}$ perfectly in the view of $\mathcal{A}$, and we obtain

$$\text{acc} = \Pr[I \neq \perp \mid (I, \text{aux}) \leftarrow \mathcal{W}_{\text{SameQuery}}(G, Y, \mathbf{c})] = \Pr[\mathbf{G}_{X.5}^A \Rightarrow 1]. \quad (14)$$

The reduction $\mathcal{B}_{\text{dl}}$ proceeds as follows. Given challenge G, Y from the DL game, it outputs $y := 0$ if $Y = 0$. Otherwise, it runs $(b, (\text{aux}, \text{aux}', \text{aux}'')) \leftarrow \text{Fork}_3^{\mathcal{W}_{\text{SameQuery}}}(G, Y)$. If $b = 1$, $\text{aux} = (\text{hqry}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}, \text{td}, \alpha, \beta)$, $\text{aux}' = (\text{hqry}', \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}', \text{td}', \alpha', \beta')$, and $\text{aux}'' = (\text{hqry}'', \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}'', \text{td}'', \alpha'', \beta'')$. Further, $\mathcal{B}_{\text{dl}}$ parses

$$\begin{aligned} \tilde{\sigma} &= (C_{\$}, \tilde{A}_{\$}, \tilde{T}_{\$}, \tilde{e}_{\$}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}_{\text{os}}, \tilde{\tau}_{\text{dl}}, \tilde{\tau}_{\text{phf}}), \\ \tilde{\sigma}' &= (C_{\$}, \tilde{A}_{\$}, \tilde{T}_{\$}, \tilde{e}_{\$}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}'_{\text{os}}, \tilde{\tau}'_{\text{dl}}, \tilde{\tau}'_{\text{phf}}), \\ \tilde{\sigma}'' &= (C_{\$}, \tilde{A}_{\$}, \tilde{T}_{\$}, \tilde{e}_{\$}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}''_{\text{os}}, \tilde{\tau}''_{\text{dl}}, \tilde{\tau}''_{\text{phf}}). \end{aligned}$$

Let $\mathbb{x}_{\text{dl}} = (G, X)$ and $\mathbb{x}_{\text{phf}} = (G, V, U, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$. Next, $\mathcal{B}_{\text{dl}}$ parses $\tilde{\tau}_{\text{phf}} = (A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}), \tilde{\tau}'_{\text{phf}} = (A_{\text{phf}}, c'_{\text{phf}}, z'_{\text{phf}}), \tilde{\tau}''_{\text{phf}} = (A_{\text{phf}}, c''_{\text{phf}}, z''_{\text{phf}})$, and similarly $\tilde{\tau}_{\text{dl}} = (A_{\text{dl}}, c_{\text{dl}}, z_{\text{dl}}), \tilde{\tau}'_{\text{dl}} = (A_{\text{dl}}, c'_{\text{dl}}, z'_{\text{dl}})$, and $\tilde{\tau}''_{\text{dl}} = (A_{\text{dl}}, c''_{\text{dl}}, z''_{\text{dl}})$. Furthermore, parse $(c_{\text{os}}, c) = \text{H}(\text{hqry})$ with $c = c_{\text{phf}} \cdot c_{\text{dl}}, (c'_{\text{os}}, c') = \text{H}(\text{hqry}')$ with $c' = c'_{\text{phf}} \cdot c'_{\text{dl}}$, and $(c''_{\text{os}}, c'') = \text{H}(\text{hqry}'')$ with $c'' = c''_{\text{phf}} \cdot c''_{\text{dl}}$ in the first, second, and third run of $\mathcal{W}_{\text{SameQuery}}$, respectively. Note that if $b = 1$, then parsing as above succeeds and we have $|\{c, c', c''\}| = 3$ by the properties of the forking algorithm Fork_3 (cf. Figure 3). Therefore, we have two cases:

Same DL challenges: Assume we have $c_{\text{dl}} = c'_{\text{dl}} = c''_{\text{dl}}$, then $c_{\text{phf}}, c'_{\text{phf}}$ and c''_{phf} are pairwise distinct. In this case, the reduction $\mathcal{B}_{\text{dl}}$ employs the 3-special soundness extractor of Σ_{phf} to extract a witness $\mathbb{w}_{\text{phf}}$ such that $(\mathbb{x}_{\text{phf}}, \mathbb{w}_{\text{phf}}) \in \hat{\mathfrak{R}}_{\text{phf}} = \{(\mathbb{x}, \mathbb{w}) \mid (\mathbb{x}, \mathbb{w}) \in \mathfrak{R}_{\text{phf}}\} \cup \{(\mathbb{x} = (G, V), \mathbb{w}) \mid \mathbb{w}G = V\}$. If $\mathbb{w}_{\text{phf}}G = V$, then the reduction outputs $y := \mathbb{w}_{\text{phf}}$. Else, it parses $\mathbb{w}_{\text{phf}} = (\alpha^*, \beta^*, \nu^*, \rho^*, r_{\S}^*, \mu^*)$. By definition of $\mathfrak{R}_{\text{phf}}$ (cf. Equation (5)), we have that $\tilde{T}_{\S} = \rho^*V + U_0 + \sum_{i \in [n]} \mu_i U_i$. Since $(U, \text{td}) \in \text{PHF.NonZeroGen}(1^\lambda, G, V)$ and $\mu^* \in \{0, 1\}^n$, we have $\tilde{T}_{\S} = eG + (f + \rho^*)V$ with $e \neq 0$ and $(e, f) = \text{PHF.TrapEval}(\text{td}, \mu^*)$ (cf. Lemma 3.12). By Remark 7.8, we also have $(\alpha, \beta) \neq 0$ with $\alpha\tilde{T}_{\S} + \beta V = 0$. The reduction outputs $y := -\frac{\alpha e}{\beta + \alpha(f + \rho^*)}$. Since the above yields $\alpha(eG + (f + \rho^*)V) + \beta V = 0$ with $\alpha e \neq 0$, we have $yG = Y$.

Distinct DL challenges: If otherwise $|\{c_{\text{dl}}, c'_{\text{dl}}, c''_{\text{dl}}\}| \geq 2$, then $\mathcal{B}_{\text{dl}}$ employs the 2-special soundness extractor of Σ_{dl} to extract a witness x for (G, X) from the corresponding Σ_{dl} transcripts. Since $X = Y + x'G$, the reduction outputs $y := x - x'$.

By the above discussion, the output of $y \leftarrow^s \mathcal{B}_{\text{dl}}(G, Y)$ satisfies $yG = Y$ if $b = 1$. Thus, we have

$$\Pr[b = 1] \leq \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda).$$

Also, by the forking lemma (cf. Lemma 3.21) we have that

$$\text{acc} \leq \sqrt[3]{32Q_{\text{H}}^2 \cdot \left(\Pr[b = 1] + \frac{1}{p-1}\right)} + \frac{4Q_{\text{H}}}{p-1}.$$

Together with Equation (14) and the above, this yields

$$\Pr[\mathbf{G}_{X.5}^A \Rightarrow 1] \leq \sqrt[3]{32Q_{\text{H}}^2 \cdot \left(\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p-1}\right)} + \frac{4Q_{\text{H}}}{p-1}.$$

BOUNDING DistChal. To bound the probability that DistChal occurs, we introduce some intermediate games $\mathbf{G}_{Y.i}$.

Game $\mathbf{G}_{Y.0}$ ($\mathbf{G}_1$ with winning condition as DistChal): This game is identical to $\mathbf{G}_1$ except that $\mathcal{A}$ wins if and only if DistChal occurs. We have

$$\Pr[\text{DistChal}] = \Pr[\mathbf{G}_{Y.0}^A \Rightarrow 1].$$

Game $\mathbf{G}_{Y.1}$ (Setup $H = hG$ with known discrete logarithm): The game samples $h \leftarrow^s \mathbb{Z}_p$ and sets $H = hG$ instead of sampling $H \leftarrow^s \mathbb{G}$. We have

$$\Pr[\mathbf{G}_{Y.0}^A \Rightarrow 1] = \Pr[\mathbf{G}_{Y.1}^A \Rightarrow 1].$$

Game $\mathbf{G}_{Y.2}$ (Setup U to prepare Σ_{phf} simulation): The game sets up the outputs U in H_{pb} via PHF.NonZeroGen . That is, on every H_{pb} query info , the game samples $(U, \text{td}_{\text{info}}) \leftarrow^s \text{PHF.NonZeroGen}(1^\lambda, G, V)$ with $\text{td}_{\text{info}} = (\mathbf{a}_{\text{info}}, \mathbf{b}_{\text{info}})$. The game keeps track of the trapdoors td_{info} for every input info . Note that in $\mathbf{G}_{Y.1}$, the H_{pb} outputs U are distributed at random which is exactly the distribution of $K \leftarrow \text{PHF.Gen}(1^\lambda)$. By the *well-distributed keys* property (cf. Lemma 3.12), we have that

$$\Pr[\mathbf{G}_{Y.1}^A \Rightarrow 1] = \Pr[\mathbf{G}_{Y.2}^A \Rightarrow 1].$$

Game $\mathbf{G}_{Y,3}$ (Simulate Σ_{dl} and compute Σ_{phf} honestly): The game computes all nonces as $A_{\text{os},j} \leftarrow \Sigma_{\text{os}}.\text{Init}(\mathbf{x}_{\text{os}}, \mathbf{w}_{\text{os}})$ directly after sampling its secret key. Then, it employs the nonce $A_{\text{os},j}$ in the j th signing session. Also, it parses $A_{\text{os},j} = r_{j,0}G + r_{j,1}H$ and sets

$$r_j := r_{j,0} + h \cdot r_{j,1} \in \mathbb{Z}_p,$$

where $H = hG$. It decomposes $r_j = \sum_{i \in [n]} \mu_{j,i} 2^{i-1}$ in binary, where $\mu_{j,i} \in \{0,1\}$, and denotes $\boldsymbol{\mu}_j = (\mu_{j,1}, \dots, \mu_{j,n})$. Furthermore, in the j th signing session for $j \in [Q_S]$, the game computes the values $(T_{\$,j}, e_{\$,j})$ and the Σ_{dl} and Σ_{phf} transcripts $(A_{\text{dl},j}, c_{\text{dl},j}, z_{\text{dl},j})$ and $(A_{\text{phf},j}, c_{\text{phf},j}, z_{\text{phf},j})$ as in ExpPHF (cf. Figure 11).

Observe that by construction, we have $A_{\text{os},j} = \sum_{i \in [n]} \mu_{j,i} (2^{i-1}G)$. Also, due to the changes in $\mathbf{G}_{Y,2}$ and by the *well-distributed logarithms* property (cf. Lemma 3.12), we have $e_j = a_{\text{info},0} + \sum_{i \in [n]} \mu_{j,i} a_{\text{info},i} \neq 0$.

Therefore, the conditions for Lemma 7.1 are satisfied and since in $\mathbf{G}_{Y,2}$, the j th signing session is distributed as in ExpDL, we have

$$\Pr[\mathbf{G}_{Y,2}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{Y,3}^{\mathcal{A}} \Rightarrow 1].$$

We are now ready to give a reduction $\mathcal{B}_{\text{dl}}$ to upper bound the advantage of $\mathcal{A}$ in $\mathbf{G}_{Y,3}$ as the game never makes use of the DL x of X except when sampling X during setup. The reduction $\mathcal{B}_{\text{dl}}$ obtains a DL challenge Y and sets $\text{pk} = (W, X)$, where $W = w_0G + w_1H$ is sampled as in $\mathbf{G}_{Y,3}$ and $X = Y$. It then sends pk to $\mathcal{A}$ and simulates the random oracles $\mathbf{H}$, $\mathbf{H}_{\text{pb}}$ and the signing oracles as in $\mathbf{G}_{Y,3}$. Finally, when $\mathcal{A}$ outputs its signatures and if event DistChal occurs, the reduction runs the 2-special soundness extractor $\Sigma_{\text{dl}}.\text{Ext}$ on input the Σ_{dl} transcripts from the i_0 th and i_1 th signature. That is, it sets $\mathbf{w}_{\text{dl}} \leftarrow \Sigma_{\text{dl}}.\text{Ext}(X, \tau_{\text{dl}}^{(i_0)}, \tau_{\text{dl}}^{(i_1)})$. Finally, $\mathcal{B}_{\text{dl}}$ outputs $\mathbf{w}_{\text{dl}}$.

As the event DistChal occurs, we have $\tilde{c}^{(i_0)} \neq \tilde{c}^{(i_1)}$. Therefore, by 2-special soundness of Σ_{dl} , we have $\mathbf{w}_{\text{dl}}G = X$. Since $X = Y$, we have

$$\Pr[\mathbf{G}_{Y,3}^{\mathcal{A}} \Rightarrow 1] \leq \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda).$$

7.5 Proof of Lemma 7.5 (Reducing to AOMPR from Nonce Collision)

We consider an adversary $\mathcal{A}$ as in Section 7 on $\mathbf{G}_2$, except that the winning condition is changed to NonceColl occurring. Let $i_0 < i_1$ denote the lexicographically smallest pair of integers (i_0, i_1) that satisfy the conditions of NonceColl, i.e., with $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$. We stick to the same notations as in Section 7 and consider the following sequence of games $\mathbf{G}_{A,k}$ for $k \in [0, 2]$. Then, we will show that the probability that $\mathcal{A}$ wins in $\mathbf{G}_{A,2}$ is bounded by the probability that some reduction $\mathcal{B}$ wins in the DL game.

Game $\mathbf{G}_{A,0}$ ($\mathbf{G}_2$ with winning condition as NonceColl): This game is identical to $\mathbf{G}_2$ except that the adversary now only wins if NonceColl occurs at the end of the game. We can see that

$$\Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceColl}] = \Pr[\mathbf{G}_{A,0}^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_{A,1}$ (Abort if “extracted unblinded challenges” are equal): This game introduces an abort at the end of the game if for the two forgeries i_0 and i_1 and the extracted witnesses

$$\mathbf{w}_{\text{sle}}^{(i_0)} = (\nu^{(i_0)}, r_{\$}^{(i_0)}, \boldsymbol{\alpha}_{\text{os}}^{(i_0)}, \gamma_{\text{os}}^{(i_0)}, r_{\text{rnd}}^{(i_0)}), \mathbf{w}_{\text{sle}}^{(i_1)} = (\nu^{(i_1)}, r_{\$}^{(i_1)}, \boldsymbol{\alpha}_{\text{os}}^{(i_1)}, \gamma_{\text{os}}^{(i_1)}, r_{\text{rnd}}^{(i_1)}),$$

we have that

$$\tilde{c}_{\text{os}}^{(i_0)} - \gamma_{\text{os}}^{(i_0)} = \tilde{c}_{\text{os}}^{(i_1)} - \gamma_{\text{os}}^{(i_1)}.$$

Intuitively, this bad event implies that the unblinded challenges (according to $\Sigma_{\text{os}}.\text{BlindChal}$) for these forgeries are identical. This would make our final reduction fail to extract the underlying DL. Also, we note that the winning condition of $\mathcal{A}$ in this game is efficiently checkable, since we can extract the underlying blinding factors via the straightline extractor of Π_{sle} .

We will now argue the advantage change of $\mathcal{A}$ from $\mathbf{G}_{A,0}$. First, recall that in $\mathbf{G}_1$, the game runs the extractor for Π_{sle} on all $\mathbf{H}$ queries made by $\mathcal{A}$ such that the proof π_{sle} as part of the query can be verified.

Game $\text{AOMPR}_{\text{GGen}, \Phi_{\text{os}}}^{\mathcal{A}}(\lambda)$:	Oracle $\text{CHAL}()$:
1 : $(\mathbb{G}, p, G) \leftarrow \text{GGen}(1^\lambda); H \leftarrow \mathbb{G}$	1 : $\text{cid} \leftarrow \text{cid} + 1$
2 : $\text{cid} \leftarrow 0; Q \leftarrow 0$	2 : $\mathbf{x}_{\text{cid}} \leftarrow \mathbb{Z}_p^2$
3 : $\text{par}_{\text{os}} \leftarrow (\mathbb{G}, p, G, H)$	3 : return $\Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{x}_{\text{cid}})$
4 : $(\mathbf{y}_i)_{i \in [\text{cid}]} \leftarrow \mathcal{A}^{\text{CHAL}, \text{PI}}(\text{par}_{\text{os}})$	Oracle $\text{PI}(\mathbf{a} \in \mathbb{Z}_p^2, (b_i \in \mathbb{Z}_p)_{i \in [\text{cid}]})$:
5 : if $Q \geq \text{cid}$ then return 0	1 : $Q \leftarrow Q + 1$
6 : return $\forall i \in [\text{cid}] : \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{x}_i) = \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{y}_i)$	2 : return $\mathbf{a} + \sum_{i=1}^{\text{cid}} b_i \mathbf{x}_i$

Fig. 12. Algebraic One-More Preimage Resistance game for $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$.

Thus, for each query hqry containing a statement $\mathbf{x}_{\text{sle}}$ and π_{sle} , the game can extract a witness $\mathbf{w}_{\text{sle}}$ such that $(\mathbf{x}_{\text{sle}}, \mathbf{w}_{\text{sle}}) \in \mathfrak{R}_{\text{sle}}$.

Then, observe that the probability that the introduced abort occurs is bounded by the probability of the event **Bad** defined as: there exist two *distinct* H queries, denoted hqry and hqry' , of the form

$$\begin{aligned} \text{hqry} &= (\mathbf{x}_{\text{os}}, \mathbf{x}_{\text{dl}}, \tilde{\mathbf{x}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbf{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg}) , \\ \text{hqry}' &= (\mathbf{x}'_{\text{os}}, \mathbf{x}'_{\text{dl}}, \tilde{\mathbf{x}}'_{\text{phf}}, \tilde{A}'_{\text{os}}, \tilde{A}'_{\text{dl}}, \tilde{A}'_{\text{phf}}, \mathbf{x}'_{\text{sle}}, \pi'_{\text{sle}}, \text{msg}') , \end{aligned}$$

such that (a) both π_{sle} and π'_{sle} pass the verification, (b) for the outputs $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}}) \leftarrow \text{H}(\text{hqry})$ and $(\tilde{c}'_{\text{os}}, \tilde{c}'_{\text{or}}) \leftarrow \text{H}(\text{hqry}')$, and the extracted $\mathbf{w}_{\text{sle}} = (\nu, r_{\mathbb{S}}, \boldsymbol{\alpha}, \gamma_{\text{os}}, r_{\text{rnd}})$ and $\mathbf{w}'_{\text{sle}} = (\nu', r'_{\mathbb{S}}, \boldsymbol{\alpha}', \gamma'_{\text{os}}, r'_{\text{rnd}})$ are such that

$$\tilde{c}_{\text{os}} - \gamma_{\text{os}} = \tilde{c}'_{\text{os}} - \gamma'_{\text{os}} .$$

This upper bound follows because the corresponding H queries for the forgeries are also of this type and that the H queries corresponding to the i_0 -th and i_1 -th forgeries are distinct due to the change in $\mathbf{G}_2$, ruling out the **SameQuery** bad event.

We can then bound $\Pr[\text{Bad}]$ via the union bound over all pairs of H queries and for each of those pairs, we have that the probability of the event occurring is at most $1/p$. To see this, assume hqry was made before hqry' and observe that at the time of the query hqry' , $\tilde{c}_{\text{os}}, \gamma_{\text{os}}$ and γ'_{os} are already determined since the extractor is deterministic. Hence, with the randomness of sampling $\tilde{c}'_{\text{os}}$ that satisfies the equation, we have that this event only occurs with probability $1/p$. Finally, via the union bound,

$$\Pr[\mathbf{G}_{A,0}^{\mathcal{A}} \Rightarrow 1] \leq \Pr[\mathbf{G}_{A,1}^{\mathcal{A}} \Rightarrow 1] + \Pr[\text{Bad}] \leq \Pr[\mathbf{G}_{A,1}^{\mathcal{A}} \Rightarrow 1] + \frac{Q_{\text{H}}^2}{p} .$$

BOUNDING ADVANTAGE OF $\mathcal{A}$ IN $\mathbf{G}_{A,1}$. We will bound the advantage of $\mathcal{A}$ via a reduction to an intermediate assumption, named Algebraic One-More Preimage Resistance (AOMPR), introduced in [TZ23]. We first recall the AOMPR assumption with respect to the linear map $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$. The AOMPR game is defined in Figure 12 and its corresponding advantage for any adversary $\mathcal{A}$ is defined as

$$\text{Adv}_{\text{GGen}, \Phi_{\text{os}}}^{\text{aompr}}(\mathcal{A}, \lambda) := \Pr[\text{AOMPR}_{\text{GGen}, \Phi_{\text{os}}}^{\mathcal{A}}(\mathcal{A}, \lambda)] .$$

The following theorem showed that for the linear map $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$, the AOMPR assumption is secure assuming the DL assumption on GGen . Hence, we only need to show a reduction $\mathcal{B}$ to AOMPR instead of a direct reduction to DL.

Theorem 7.9 ([TZ23, Theorem 1] adapted for $\Phi_{\text{os}}^{\text{par}_{\text{os}}}$). *Let GGen be a group generator outputting groups of prime-order $p = p(\lambda)$. For any adversary $\mathcal{A}$ in AOMPR game, running in time $T_{\mathcal{A}} = T_{\mathcal{A}}(\lambda)$ and making at most $Q = Q(\lambda)$ queries to CHAL , there exists an adversary $\mathcal{B}$ running in time $T_{\mathcal{B}} \approx T_{\mathcal{A}}$ such that $\text{Adv}_{\text{GGen}, \Phi_{\text{os}}}^{\text{aompr}}(\mathcal{A}, \lambda) \leq 2 \cdot \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}, \lambda)$.*

Now, to complete the proof, we will rely on the following lemma.

Lemma 7.10. *There exists an adversary $\mathcal{B}$ in the AOMPR game, running in time roughly that of $\mathcal{A}$, making $Q_{S_1} + 1$ and Q_{S_1} queries to the oracles CHAL, PI, respectively, such that $\Pr[\mathbf{G}_{A.1}^A \Rightarrow 1] \leq \text{Adv}_{\text{GGen}, \Phi_{\text{os}}}^{\text{aompr}}(\mathcal{B}, \lambda)$.*

We prove the above lemma below. However, note that combining Theorem 7.9 and Lemma 7.10, concludes the proof as there exists an adversary $\mathcal{B}$ such that

$$\Pr[\mathbf{G}_{A.1}^A \Rightarrow 1] \leq 2 \cdot \text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}, \lambda) . \square$$

Proof (of Lemma 7.10). Now, we give the adversary $\mathcal{B}$ playing the AOMPR game and running $\mathcal{A}$ which takes as input $(\mathbb{G}, p, G, H)$, has access to the oracles CHAL, PI, and does the following:

Setting up pp and pk. Run the $\text{BS}_{\text{phf}}.\text{Setup}(1^\lambda)$ except that $(\mathbb{G}, p, G, H)$ are taken from $\mathcal{B}$'s input. Run $\text{BS}_{\text{phf}}.\text{KeyGen}(\text{pp})$ to generate (sk, pk) except that w is not sampled and $W \leftarrow \text{CHAL}()$ is generated via querying CHAL instead.

Answering oracle queries. For each of the oracles $S_1, S_2, H, H_{\text{pb}}, H_{\text{sle}}$, $\mathcal{B}$ simulates them as follows:

- **Random oracles $H_{\text{pb}}, H_{\text{sle}}$:** Simulate via lazy sampling and also keep track of the queries $\mathcal{Q}_{H_{\text{sle}}}$ of H_{sle} queries for the use of the extractor.
- **Random oracle H .** On the query hqry that can be parsed to the format

$$\text{hqry} = (\mathbb{X}_{\text{os}}, \mathbb{X}_{\text{dl}}, \tilde{\mathbb{X}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{X}_{\text{sle}}, \pi_{\text{sle}}, \text{msg}) ,$$

check if $\Pi_{\text{sle}}.\text{Ver}_{\text{sle}}^H(\mathbb{X}_{\text{sle}}, \pi_{\text{sle}}) = 1$. If so, run the extractor $w_{\text{sle}}^{(\text{hqry})} := \text{Ext}(\mathcal{Q}_{H_{\text{sle}}}, \mathbb{X}_{\text{sle}}, \pi_{\text{sle}})$. Otherwise, do nothing. Then, perform lazy sampling for the output of this hash query as normal.

- **First signing oracle S_1 .** Increment the session counter cnt . Keep track of info_{cnt} . Run $\text{BS}_{\text{phf}}.\text{Sign}_1$ as in Figure 9 except for the lines (2,3), where $\mathcal{B}$ instead makes a query $A_{\text{os}, \text{cnt}} \leftarrow \text{CHAL}()$ instead. Also, keep track of the state generated from the algorithm except for the state of Σ_{os} . Note that other operations can still be done since $\mathcal{B}$ sets up the rest of the public and secret keys (not including W) honestly.
- **Second signing oracle S_2 .** On query $(\text{sid} \in [\text{cnt}], \text{umsg}_{\text{sid}} = (c_{\text{os}, \text{sid}}, c_{\text{or}, \text{sid}}))$ and if sid has not been queried to S_2 before, run $\text{BS}_{\text{phf}}.\text{Sign}_2$ as in Figure 9 to compute the responses except for Line 3. To compute z_{os} , $\mathcal{B}$ makes a query $z_{\text{os}, \text{sid}} \leftarrow \text{PI}(\mathbf{0}, (\beta_i)_{i \in [0, \text{cnt}]})$ where $\beta_0 = c_{\text{os}, \text{sid}}$ and $\beta_i = 0$ for all $i \in [\text{cnt}]$ except for $\beta_{\text{sid}} = 1$. Return $(c_{\text{phf}}, z_{\text{os}, \text{sid}}, z_{\text{phf}}, z_{\text{dl}})$.

Breaking AOMPR using the forgeries. After the adversary outputs the forgeries $\{\text{msg}^{(i)}, \tilde{\sigma}^{(i)}\}_{i \in [\ell]}$, $\mathcal{B}$ checks first that $\mathcal{A}$ wins in $\mathbf{G}_{A.1}$, i.e., (1) NonceColl occurs and (2) the abort introduced in $\mathbf{G}_{A.1}$ does not occur. Then, $\mathcal{B}$ does the following step-by-step to extract the preimages for $(W, A_{\text{os}, 1}, \dots, A_{\text{os}, \text{cnt}})$:

- Preimage of W .** First, we will use the fact that $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$ and that $\tilde{c}_{\text{os}}^{(i_0)} - \gamma_{\text{os}}^{(i_0)} \neq \tilde{c}_{\text{os}}^{(i_1)} - \gamma_{\text{os}}^{(i_1)}$ (due to the abort introduced in $\mathbf{G}_{A.1}$). For $i \in \{i_0, i_1\}$, let $c_{\text{os}}^{(i)} := \tilde{c}_{\text{os}}^{(i)} - \gamma_{\text{os}}^{(i)}$ and $z_{\text{os}}^{(i)} := \tilde{z}_{\text{os}}^{(i)} - \alpha_{\text{os}}^{(i)}$. Then, compute

$$w = \frac{z_{\text{os}}^{(i_0)} - z_{\text{os}}^{(i_1)}}{c_{\text{os}}^{(i_0)} - c_{\text{os}}^{(i_1)}} .$$

- Preimage of $A_{\text{os}, \text{sid}}$ where sid is closed.** For $\text{sid} \in [\text{cnt}]$ that was queried to S_2 , using the extracted w from the above, compute $r_{\text{os}, \text{sid}} := z_{\text{os}, \text{sid}} - c_{\text{os}, \text{sid}} \cdot w$.
- Preimage of $A_{\text{os}, \text{sid}}$ where sid is not closed.** For $\text{sid} \in [\text{cnt}]$ that was not queried to S_2 , compute $r_{\text{os}, \text{sid}} \leftarrow \text{PI}(\mathbf{0}, e_{\text{sid}+1})$ where $e_i \in \mathbb{Z}_p^{\text{cnt}+1}$ is an all-zero vector except for the i -th position. Then, return $(w, r_{\text{os}, 1}, \dots, r_{\text{os}, \text{cnt}})$.

We will see that the view of $\mathcal{A}$ is exactly identical to its view in $\mathbf{G}_{A.1}$. First, all the public parameters, the public keys except for W , and the oracles except for S_1, S_2 are simulated identically to $\mathbf{G}_{A.1}$. For W , it is still generated in the same way as CHAL simply samples $x_0 \leftarrow \mathbb{Z}_p^2$ and returns $\Phi_{\text{os}}^{\text{par}}(x)$. Similarly, the nonces $A_{\text{os}, i}$ computed in S_1 have the same distribution with an underlying randomness x_i . Finally, the responses $z_{\text{os}, i}$ are also computed as $x_i + c_{\text{os}} \cdot x_0$. Therefore, the view of $\mathcal{A}$ remains identical to its view in $\mathbf{G}_{A.1}$.

Now, we argue that if $\mathcal{A}$ succeeds in $\mathbf{G}_{A.1}$, $\mathcal{B}$ also succeeds in the AOMPR game. First, we can see that the PI query count of $\mathcal{B}$ is always less than the number of CHAL queries as it makes one extra query to generate

W . Now, we will argue the correctness for each extraction. For preimages of type (c), they are guaranteed to be preimages of $A_{\text{os},\text{sid}}$ due to how PI is defined. For preimage of type (b), if $\mathbf{w}$ is a preimage of W , then by how PI is defined $\mathbf{z}_{\text{os},\text{sid}}$ is a preimage of $A_{\text{os},\text{sid}} + c_{\text{os},\text{sid}}W$, meaning the extracted $\mathbf{r}_{\text{os},\text{sid}} = \mathbf{z}_{\text{os},\text{sid}} - c_{\text{os},\text{sid}}\mathbf{w}$ is a preimage of $A_{\text{os},\text{sid}}$.

Finally, for (a) preimage $\mathbf{w}$ of W , recall that since the extracted witnesses $\mathbf{w}_{\text{sl}}^{(i_0)}$ and $\mathbf{w}_{\text{sl}}^{(i_1)}$ are valid witnesses for the relation $\mathfrak{R}_{\text{sl}}$ in Equation (11), we know that for $i \in \{i_0, i_1\}$ with $\mathbf{w}_{\text{sl}}^{(i)} = (\nu^{(i)}, r_{\text{s}}^{(i)}, \alpha_{\text{os}}^{(i)}, \gamma_{\text{os}}^{(i)}, r_{\text{rnd}}^{(i)})$:

$$\tilde{A}_{\text{os}}^{(i)} = \Sigma_{\text{os}}.\text{BlindNonce}(\mathbb{X}_{\text{os}}, A_{\text{os}}^{(i)}; \tilde{\mathbf{st}}^{(i)} = (\alpha_{\text{os}}^{(i)}, \gamma_{\text{os}}^{(i)})) .$$

Then, via Lemma 7.2, we can efficiently compute $(c_{\text{os}}^{(i)}, z_{\text{os}}^{(i)})$ from $\tilde{\tau}_{\text{os}}^{(i)}$ and $(\alpha_{\text{os}}^{(i)}, \gamma_{\text{os}}^{(i)})$, such that $(A_{\text{os}}^{(i)}, c_{\text{os}}^{(i)}, z_{\text{os}}^{(i)})$ is a valid transcript for $\mathbb{X}_{\text{os}} = (G, H, W)$.

Then, because $A_{\text{os}}^{(i_0)} = A_{\text{os}}^{(i_1)}$ and $(A_{\text{os}}^{(i)}, c_{\text{os}}^{(i)}, z_{\text{os}}^{(i)})$ for $i \in \{i_0, i_1\}$ are valid transcripts for $\mathbb{X}_{\text{os}}$, we have that

$$\begin{aligned} \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{z}_{\text{os}}^{(i_0)}) - c_{\text{os}}^{(i_0)} \cdot W &= \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{z}_{\text{os}}^{(i_1)}) - c_{\text{os}}^{(i_1)} \cdot W \\ \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{z}_{\text{os}}^{(i_0)} - \mathbf{z}_{\text{os}}^{(i_1)}) &= (c_{\text{os}}^{(i_0)} - c_{\text{os}}^{(i_1)}) \cdot W \\ W &= \Phi_{\text{os}}^{\text{par}_{\text{os}}} \left(\frac{\mathbf{z}_{\text{os}}^{(i_0)} - \mathbf{z}_{\text{os}}^{(i_1)}}{c_{\text{os}}^{(i_0)} - c_{\text{os}}^{(i_1)}} \right) = \Phi_{\text{os}}^{\text{par}_{\text{os}}}(\mathbf{w}) . \end{aligned}$$

This proves that $\mathbf{w}$ is a preimage of W . Therefore, $\mathcal{B}$ wins in AOMPR game if $\mathcal{A}$ wins in $\mathbf{G}_{A.1}$, and we can conclude that

$$\Pr[\mathbf{G}_{A.1}^{\mathcal{A}} \Rightarrow 1] \leq \text{Adv}_{\text{GGen}, \Phi_{\text{os}}}^{\text{aompr}}(\mathcal{B}, \lambda) . \square$$

7.6 Proof of Lemma 7.6 (Extracting DL from Distinct Nonces)

We consider an adversary $\mathcal{A}$ as in Section 7 on $\mathbf{G}_2$, except that the winning condition is changed to **NonceDist** occurring. Denote by $\mathcal{J}_{\text{info}^*}$ the indices of signing sessions whose common message is info^* . Note that by assumption, we have $|\mathcal{J}_{\text{info}^*}| = \ell - 1$. Let i^* denote the lexicographically smallest index $i^* \in [\ell]$ such that

$$A_{\text{os}}^{(i^*)} \notin \{(A_{\text{os},j}) \mid j \in \mathcal{J}_{\text{info}^*}\} .$$

That is, the nonce $A_{\text{os}}^{(i^*)}$ is distinct from all the nonces $A_{\text{os},j}$ sampled by the game in the signing oracles when the common message is info^* . Note that $A_{\text{os}}^{(i^*)}$ is well-defined by the pigeon-hole principle.

Game $\mathbf{G}_{B.0}$ ($\mathbf{G}_2$ with winning condition **NonceDist):** This game is identical to $\mathbf{G}_2$ except that the winning condition is now that **NonceDist** occurs. We have that

$$\Pr_{\mathbf{G}_2^{\mathcal{A}}}[\text{NonceDist}] = \Pr[\mathbf{G}_{B.0}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{B.1}$ (Setup $H = hG$ with known discrete logarithm): The game samples $h \leftarrow \mathbb{Z}_p$ and sets $H = hG$ instead of sampling $H \leftarrow \mathbb{G}$. We have

$$\Pr[\mathbf{G}_{B.0}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{B.1}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{B.2}$ (Guess H_{pb} query of info^*): Now, the game samples an index $i_{\text{pb}}^* \leftarrow [\ell]$ at random at the beginning of the game. Further, when $\mathcal{A}$ outputs its signatures at the end, the game aborts if $\text{info}^* \neq \text{info}_{i_{\text{pb}}^*}^*$, where $\text{info}_{i_{\text{pb}}^*}^*$ is the input to the i_{pb}^* th H_{pb} query. Note that such a query must exist since also the game's queries are counted towards Q_{pb} . Since the game's simulation is independent of i_{pb}^* , we obtain

$$\Pr[\mathbf{G}_{B.1}^{\mathcal{A}} \Rightarrow 1] \leq Q_{\text{pb}} \cdot \Pr[\mathbf{G}_{B.2}^{\mathcal{A}} \Rightarrow 1] .$$

Note that the game learns info^* at the latest in the first call to S_1 with common message info^* . In the following, we refer to such sessions as sessions with info^* .

Game $\mathbf{G}_{B.3}$ (Setup U to prepare Σ_{phf} simulation): The game sets up the values U when simulating H_{pb} depending on whether the input is equal to info^* . For input $\text{info} = \text{info}^*$, it samples U via PHF.TrapGen . For input $\text{info} \neq \text{info}^*$, it samples U with PHF.NonZeroGen instead. This setup enables the game to compute a valid Σ_{phf} witness in all signing sessions (with sufficient probability). In more detail, for every H_{pb} query with fresh input info , the game proceeds as follows.

Case $\text{info} = \text{info}^*$: It samples $(U, \text{td}_{\text{info}^*}) \leftarrow_s \text{PHF.TrapGen}(1^\lambda, G, V)$ with $\text{td}_{\text{info}^*} = (\mathbf{a}_{\text{info}^*}, \mathbf{b}_{\text{info}^*})$ and outputs U . For TrapGen , set the parameters $\ell = n$ (here, ℓ denotes the input length in the TrapGen definition), $q = Q_S$ (assuming that $Q_S \geq 1$), and $J = \lceil q^2 c \rceil$.

Case $\text{info} \neq \text{info}^*$: It samples $(U, \text{td}_{\text{info}}) \leftarrow_s \text{PHF.NonZeroGen}(1^\lambda, G, V)$ with $\text{td}_{\text{info}} = (\mathbf{a}_{\text{info}}, \mathbf{b}_{\text{info}})$ and outputs U .

Furthermore, the game stores $\text{td}_{\text{info}} = (\mathbf{a}_{\text{info}}, \mathbf{b}_{\text{info}})$ for later (in both cases).

Under the *well-distributed keys* property of PHF with both setups (cf. Theorem 3.11 and Lemma 3.12), we have

$$\Pr[\mathbf{G}_{B.2}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{B.3}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{B.4}$ (Abort if event BadEval occurs): The game samples the nonces $A_{\text{os},j}$ for Σ_{os} sampled in oracle S_1 at the beginning of the game and introduces additional abort events. That is, the game samples $A_{\text{os},j} \leftarrow_s \Sigma_{\text{os}}.\text{Init}(\mathbb{X}_{\text{os}}, \mathbb{W}_{\text{os}})$ directly after sampling its secret key. It employs the nonce $A_{\text{os},j}$ in the j th signing session. Also, it parses $A_{\text{os},j} = r_{j,0}G + r_{j,1}H$ and sets

$$r_j := r_{j,0} + h \cdot r_{j,1} \in \mathbb{Z}_p ,$$

where $H = hG$ and decomposes $r_j = \sum_{i \in [n]} \mu_{j,i} 2^{i-1}$ in binary, where $\mu_{j,i} \in \{0, 1\}$. Note that we use representatives $\mu \in [0, p-1]$, therefore this is possible by definition of $n = \lceil \log p \rceil$. We denote $\boldsymbol{\mu}_j = (\mu_{j,1}, \dots, \mu_{j,n})$. The game aborts in the j th invocation of S_1 if $a_j = 0 \bmod p$ for $(a_j, b_j) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}}, \boldsymbol{\mu}_j)$, where info is the corresponding common message.

When $\mathcal{A}$ outputs its signatures, the game computes the DL $r^{(i^*)}$ of $A_{\text{os}}^{(i^*)} = r^{(i^*)}G$ via the DL h of H , the Σ_{os} transcript $(\tilde{A}_{\text{os}}^{(i^*)}, \tilde{c}_{\text{os}}^{(i^*)}, \tilde{z}_{\text{os}}^{(i^*)})$ and the blinding factors $(\boldsymbol{\alpha}^{(i^*)}, \boldsymbol{\gamma}_{\text{os}}^{(i^*)})$ extracted from $\pi_{\text{sle}}^{(i^*)}$ as in Lemma 7.2. Note that $r^{(i^*)} \in \mathbb{Z}_p$ has at most two possible bit decompositions with n bits modulo p . To be precise, the representative $r^{(i^*)} \in [0, p-1]$ has bit decomposition $r^{(i^*)} = \sum_{i \in [n]} \mu_{0,i}^* 2^{i-1}$ and if $r^{(i^*)} + p < 2^n$, then also $r^{(i^*)} + p = \sum_{i \in [n]} \mu_{1,i}^* 2^{i-1}$. Note that since $2p > 2^{\lceil \log p \rceil} = 2^n$, these are all possible bit decompositions $\boldsymbol{\mu} \in \{0, 1\}^n$ such that $r^{(i^*)} = \sum_{i \in [n]} \mu_i 2^{i-1} \bmod p$. Denote $\boldsymbol{\mu}_0^* = (\mu_{0,1}, \dots, \mu_{0,n})$ and $\boldsymbol{\mu}_1^* = (\mu_{1,1}, \dots, \mu_{1,n})$ (if the second decomposition exists). The game aborts if $a_0^* \neq 0$ for $(a_0^*, b_0^*) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \boldsymbol{\mu}_0^*)$. If the second decomposition exists, the game also aborts if $a_1^* \neq 0$ for $(a_1^*, b_1^*) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \boldsymbol{\mu}_1^*)$.

Denote by BadEval the event that the aborts introduced in $\mathbf{G}_{B.4}$ occur. The following lemma then bounds the success probability of $\mathcal{A}$ in $\mathbf{G}_{B.4}$. Intuitively, the proof follows from $(0, Q_S, 1)$ -PHF property (Lemma 3.12) and $(2, Q_S, \delta)$ -PHF property (Theorem 3.11) of PHF, and we give the formal proof in Section 7.7.

Lemma 7.11. *Let $\delta = \frac{d}{Q_S^2(n+1)}$ and $p > (n+1)[Q_S^2 c] + 1$ with c, d as defined in Theorem 3.11. Then*

$$\Pr[\mathbf{G}_{B.3}^{\mathcal{A}} \Rightarrow 1] \leq 1/\delta \cdot \Pr[\mathbf{G}_{B.4}^{\mathcal{A}} \Rightarrow 1] .$$

Remark 7.12. Let $r^{(i^*)}G = A_{\text{os}}^{(i^*)}$ and $r_j G = A_{\text{os},j}$ for $j \in [Q_S]$ be as above. If $\mathcal{A}$ wins and BadEval does not occur, then the following properties hold.

- We have $a_j \neq 0 \bmod p$ in the j th signing session with common message info , where we denote $(a_j, b_j) = \text{PHF.TrapEval}(\text{td}_{\text{info}}, \boldsymbol{\mu}_j)$ and $r_j = \sum_{i \in [n]} \mu_{j,i} 2^{i-1}$. Note that this holds for any session with arbitrary common message info .
- We have $a_0^* = 0$, where $(a_0^*, b_0^*) = \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \boldsymbol{\mu}_0^*)$ and $r^{(i^*)} = \sum_{i \in [n]} \mu_{0,i}^* 2^{i-1}$.
- If $r^{(i^*)} + p < 2^n$, then also $a_1^* = 0$ for $(a_1^*, b_1^*) = \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \boldsymbol{\mu}_1^*)$ and $r^{(i^*)} + p = \sum_{i \in [n]} \mu_{1,i}^* 2^{i-1}$.

Game $\mathbf{G}_{B.5}$ (Simulate Σ_{dl} and compute Σ_{phf} honestly): The game simulates the values $(T_{\S,j}, e_{\S,j})$ and the Σ_{dl} and Σ_{phf} transcripts $(A_{\text{dl},j}, c_{\text{dl},j}, z_{\text{dl},j})$ and $(A_{\text{phf},j}, c_{\text{phf},j}, z_{\text{phf},j})$ in the j th signing session for $j \in [Q_S]$ as in **ExpPHF** using an appropriate witness for Σ_{phf} .

Note that by Remark 7.12, the conditions for Lemma 7.1 are met and since in $\mathbf{G}_{B.4}$, the j th signing session is distributed as in **ExpDL**, we have

$$\Pr[\mathbf{G}_{B.4}^A \Rightarrow 1] = \Pr[\mathbf{G}_{B.5}^A \Rightarrow 1] .$$

Note that as a consequence, the game never makes use of the DL x of X . Below, we show how to use an adversary on $\mathbf{G}_{B.5}$ to break the DL assumption. The intuition is that a successful adversary $\mathcal{A}$ on $\mathbf{G}_{B.5}$ must provide a proof for non-zero evaluation of the PHF (in the exponent) in the form of a Σ_{phf} transcript $\tilde{\tau}_{\text{phf}}^{(i^*)} = (\tilde{A}_{\text{phf}}^{(i^*)}, \tilde{c}_{\text{phf}}^{(i^*)}, \tilde{z}_{\text{phf}}^{(i^*)})$, while the abort event **BadEval** ensures that the PHF evaluation of the adversary's forgery must evaluate to zero. Note that this is only computationally guaranteed under the DL assumption, as we formalize below.

REDUCTION TO DL. We show that a successful adversary on $\mathbf{G}_{B.5}$ can be used to break the DL assumption via a forking argument. First, let us define a wrapper $\mathcal{W}_{\text{NonceDist}}$ that simulates the game $\mathbf{G}_{B.5}$ to $\mathcal{A}$, taking as input a DL challenge (G, Y) and challenges $\mathbf{c}$.

Wrapper $\mathcal{W}_{\text{NonceDist}}((G, Y), \mathbf{c})$: On input $G, Y \in \mathbb{G}$ and $\mathbf{c} = (c_1, \dots, c_{Q_H}) \in (\mathbb{Z}_p^\times)^{Q_H}$, the wrapper $\mathcal{W}_{\text{NonceDist}}$ simulates $\mathbf{G}_{B.5}$ to $\mathcal{A}$ as above, except for the following changes:

- The wrapper sets $\text{par} := (n, G, H, \text{ck}_{\text{ped}}, V)$, where $V := Y$, $\text{ck}_{\text{ped}} = (G_1, G_2, G_3, V)$ and $n = \lceil \log_2 p \rceil$. Note that $H = hG$ with $h \leftarrow \mathbb{Z}_p$.
- The wrapper sets $\text{pk} = (W = w_0G + w_1H, X)$ with $X := Y + x'G$, and $\text{sk} = (\mathbf{w} = (w_0, w_1), \perp, \text{pk})$. Here, $x' \leftarrow \mathbb{Z}_p$.
- On the i th query to $\mathbf{H}$, the wrapper outputs $(\tilde{c}_{\text{os},i}, c_i)$ for $\tilde{c}_{\text{os},i} \leftarrow \mathbb{Z}_p$.

If $\mathcal{A}$ does *not* succeed, according to the winning conditions in $\mathbf{G}_{B.5}$, the wrapper outputs $(\perp, \perp)$. Else, denote by $I \in [Q_H]$ the index of the hash query

$$\text{hqry}^{(i^*)} = (\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i^*)}, \tilde{A}_{\text{os}}^{(i^*)}, \tilde{A}_{\text{dl}}^{(i^*)}, \tilde{A}_{\text{phf}}^{(i^*)}, \mathbb{x}_{\text{sle}}^{(i^*)}, \pi_{\text{sle}}^{(i^*)}, \text{msg}^{(i^*)})$$

associated to the i^* th message-signature pair $(\text{msg}^{(i^*)}, \tilde{\sigma}^{(i^*)})$. Further, let

$$\text{aux} := (\mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}^{(i^*)}, \tilde{\sigma}^{(i^*)}, \text{td}_{\text{info}}, A_{\text{os}}^{(i^*)}, r^{(i^*)}, \nu^{(i^*)}, r_{\S}^{(i^*)}) .$$

The wrapper outputs (I, aux) .

Note that the wrapper simulates $\mathbf{G}_{B.5}$ perfectly in the view of $\mathcal{A}$. Therefore, we have

$$\text{acc} := \Pr[I \neq \perp \mid (I, \text{aux}) \leftarrow \mathcal{W}_{\text{NonceDist}}^A((G, Y), \mathbf{c})] = \Pr[\mathbf{G}_{B.5}^A \Rightarrow 1] ,$$

where $Y \leftarrow \mathbb{G}$ and $\mathbf{c} \leftarrow (\mathbb{Z}_p^\times)^{Q_H}$.

Remark 7.13. Also, we note that the wrapper introduces additional running time due to the simulation of the PHF using the trapdoor generated keys. This also affects the following reduction's running time. In particular, running **TrapEval** takes time linear in $J = O(nQ_S^2)$. For simplicity, we ignore this overhead in the running time calculation in the theorem statement as it is polynomial in Q_S which is upper bounded by T_A .

The reduction $\mathcal{B}_{\text{dl}}$ proceeds as follows. Given challenge (G, Y) , it outputs $y := 0$ if $Y = 0$. Else, it runs

$$(b, (\text{aux}, \text{aux}', \text{aux}'')) \leftarrow \mathcal{Fork}_3^{\mathcal{W}_{\text{NonceDist}}}(G, Y) .$$

If $b = 1$, then $\mathcal{B}_{\text{dl}}$ parses $\text{aux} = (\mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}, \text{td}, A_{\text{os}}, r, \nu, r_{\S})$, $\text{aux}' = (\mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}', \text{td}', A_{\text{os}}, r, \nu, r_{\S})$, and $\text{aux}'' = (\mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{\sigma}'', \text{td}'', A_{\text{os}}, r, \nu, r_{\S})$. Further, $\mathcal{B}_{\text{dl}}$ parses

$$\begin{aligned} \tilde{\sigma} &= (C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}_{\text{os}}, \tilde{\tau}_{\text{dl}}, \tilde{\tau}_{\text{phf}}) , \\ \tilde{\sigma}' &= (C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}'_{\text{os}}, \tilde{\tau}'_{\text{dl}}, \tilde{\tau}'_{\text{phf}}) , \\ \tilde{\sigma}'' &= (C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S}, C_{\text{rnd}}, \pi_{\text{sle}}, \tilde{\tau}''_{\text{os}}, \tilde{\tau}''_{\text{dl}}, \tilde{\tau}''_{\text{phf}}) . \end{aligned}$$

Let $\mathbb{x}_{\text{dl}} = (G, X)$ and $\mathbb{x}_{\text{phf}} = (G, V, U, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$. Next, $\mathcal{B}_{\text{dl}}$ parses $\tilde{\tau}_{\text{phf}} = (A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}}), \tilde{\tau}'_{\text{phf}} = (A_{\text{phf}}, c'_{\text{phf}}, z'_{\text{phf}}), \tilde{\tau}''_{\text{phf}} = (A_{\text{phf}}, c''_{\text{phf}}, z''_{\text{phf}})$, and similarly $\tilde{\tau}_{\text{dl}} = (A_{\text{dl}}, c_{\text{dl}}, z_{\text{dl}}), \tilde{\tau}'_{\text{dl}} = (A_{\text{dl}}, c'_{\text{dl}}, z'_{\text{dl}}), \tilde{\tau}''_{\text{dl}} = (A_{\text{dl}}, c''_{\text{dl}}, z''_{\text{dl}})$. Note that by the properties of the forking algorithm Fork_3 , the parsing steps succeed if $b = 1$.

If $|\{c_{\text{dl}}, c'_{\text{dl}}, c''_{\text{dl}}\}| \geq 2$, then $\mathcal{B}_{\text{dl}}$ employs the 2-special soundness extractor of Σ_{dl} to extract a witness x for (G, X) from the corresponding Σ_{dl} transcripts. The reduction $\mathcal{B}_{\text{dl}}$ outputs $y := x - x'$ which constitutes a DL response since $X = Y + x'G$.

Else, it must hold that $c_{\text{phf}}, c'_{\text{phf}}$ and c''_{phf} are pairwise distinct.²¹ Thus, in this case $\mathcal{B}_{\text{dl}}$ invokes the 3-special soundness extractor of Σ_{phf} on input $(\tilde{\mathbb{x}}_{\text{phf}}, \tilde{\tau}_{\text{phf}}, \tilde{\tau}'_{\text{phf}}, \tilde{\tau}''_{\text{phf}})$ to extract a witness $\mathbb{w}_{\text{phf}}$ such that $(\mathbb{x}_{\text{phf}}, \mathbb{w}_{\text{phf}}) \in \{(\mathbb{x}, \mathbb{w}) \mid (\mathbb{x}, \mathbb{w}) \in \mathfrak{R}_{\text{phf}}\} \cup \{(\mathbb{x} = (G, V), \mathbb{w}) \mid \mathbb{w}G = V\}$. If $\mathbb{w}_{\text{phf}}G = V$, then the reduction outputs $y := \mathbb{w}_{\text{phf}}$. Else, it parses $\mathbb{w}_{\text{phf}} = (\tilde{\alpha}, \tilde{\beta}, \tilde{\nu}, \tilde{\rho}, \tilde{r}_{\S}, \tilde{\mu})$. We have by definition of $\mathfrak{R}_{\text{phf}}$ that

$$\begin{bmatrix} C_{\S} \\ \tilde{A}_{\S} \\ \tilde{T}_{\S} - U_0 \\ e_{\S}G \end{bmatrix}^{\top} = \Phi_{\text{phf}}^{\tilde{T}_{\S}}(\mathbb{w}_{\text{phf}}) = \begin{bmatrix} \tilde{\nu}G + \tilde{r}_{\S}V \\ \tilde{\nu}V + \sum_{i \in [n]} \tilde{\mu}_i(2^{i-1}G) \\ \tilde{\rho}V + \sum_{i \in [n]} \tilde{\mu}_i U_i \\ \tilde{\alpha}\tilde{T}_{\S} + \tilde{\beta}V \end{bmatrix}^{\top}$$

Note that $rG = A_{\text{os}} = \tilde{A}_{\S} - \nu V$ and $C_{\S} = \nu G + r_{\S}V$ by definition of $\mathbf{G}_{B.5}$. If $\tilde{\nu} \neq \nu$ or $\tilde{r}_{\S} \neq r_{\S}$, then $\mathcal{B}_{\text{dl}}$ outputs $y := (\nu - \tilde{\nu})/(\tilde{r}_{\S} - r_{\S})$. Here, we have $yG = Y$ as desired since $\nu G + r_{\S}V = C_{\S} = \tilde{\nu}G + \tilde{r}_{\S}V$.

Else, the reduction parses $\text{td} = (\mathbf{a}, \mathbf{b})$. By definition of $A_{\text{os}} = A_{\S} - \nu V$ in $\mathbf{G}_{B.5}$ and since $A_{\S} = \tilde{\nu}V + \sum_{i \in [n]} \tilde{\mu}_i(2^{i-1}G)$ with $\tilde{\nu} = \nu$, we have $A_{\text{os}} = \sum_{i \in [n]} \tilde{\mu}_i(2^{i-1}G) = rG$. Therefore, $\tilde{\mu}$ forms a bit-decomposition with n bits of r modulo p . Note that by Remark 7.12, both possible bit-decompositions μ' with n bits of r modulo p evaluate to zero via the PHF, since BadEval does not occur. That is, we have $a^* = 0$ with $(a^*, b^*) \leftarrow \text{PHF.TrapEval}(\text{td}, \tilde{\mu})$. Thus, we have

$$\begin{aligned} \tilde{T}_{\S} &= \tilde{\rho}V + U_0 + \sum_{i \in [n]} \tilde{\mu}_i U_i \\ &= \tilde{\rho}V + a^*G + b^*V \\ &= (\tilde{\rho} + b^*)V. \end{aligned}$$

At the same time, we have

$$e_{\S}G = \tilde{\alpha}\tilde{T}_{\S} + \tilde{\beta}V,$$

where $e_{\S} \neq 0$. Together, this allows to compute the DL of V since

$$e_{\S}G = \tilde{\alpha}\tilde{T}_{\S} + \tilde{\beta}V = (\tilde{\alpha}(\tilde{\rho} + b^*) + \tilde{\beta})V \implies \frac{\tilde{\alpha}(\tilde{\rho} + b^*) + \tilde{\beta}}{e_{\S}}V = G$$

Finally, the reduction outputs $y := e_{\S}/(\tilde{\alpha}(\tilde{\rho} + b^*) + \tilde{\beta})$.

By the above discussion, the output of $y \leftarrow \mathcal{B}_{\text{dl}}(G, Y)$ satisfies $yG = Y$ if $b = 1$. By the forking lemma (cf. Lemma 3.21), we obtain

$$\Pr[\mathbf{G}_{B.5}^A \Rightarrow 1] \leq \sqrt[3]{32Q_{\text{H}}^2 \cdot \left(\text{Adv}_{\text{GGen}}^{\text{dl}}(\mathcal{B}_{\text{dl}}, \lambda) + \frac{1}{p-1} \right)} + \frac{4Q_{\text{H}}}{p-1}.$$

Formally, this may be shown as in the rewinding-based argument in the proof of Lemma 7.4, where $\Pr[\mathbf{G}_{X.5}^A \Rightarrow 1]$ is upper bounded; we omit details. The statement follows by collecting the above bounds. $\square$

²¹ Formally, this can be shown as in the rewinding-based argument in the proof of Lemma 7.4, where $\Pr[\mathbf{G}_{X.5}^A \Rightarrow 1]$ is upper bounded.

7.7 Proof of Lemma 7.11 (Introducing PHF Aborts)

We will prove this lemma via the following sequence of games $\mathbf{G}_{B.3.0}, \dots, \mathbf{G}_{B.3.4}$ where $\mathbf{G}_{B.3.0}$ is equivalent to $\mathbf{G}_{B.3}$ and $\mathbf{G}_{B.3.4}$ is equivalent to $\mathbf{G}_{B.4}$. Note that the bound on p is to ensure that we may apply Theorem 3.11; it ensures that no wrap-around over $\mathbb{Z}_p$ occurs when evaluating PHF.TrapEval .

Game $\mathbf{G}_{B.3.1}$ (Introduce bad flag for info $\neq \text{info}^*$ at the end of the game): In this game, the game introduces a boolean flag **bad** initialized to $\text{bad} \leftarrow 0$. At the end of the game when $\mathcal{A}$ outputs its forgeries, the game checks for signing sessions $j \notin \mathcal{J}_{\text{info}^*}$ if $a_j = 0$ computed via $(a_j, b_j) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}}, \mu_j)$, where **info** is the corresponding common message of the session j . If so, set a flag $\text{bad} = 1$. Otherwise, do nothing. Then, the game also changes the winning condition of $\mathcal{A}$ to be such that it wins in $\mathbf{G}_{B.3}$ and $\text{bad} = 0$.

We note that due to $(0, Q_S, 1)$ -PHF property proved in Lemma 3.12, regardless of any choices of μ_j and U generated via $\text{PHF.NonZeroGen}(1^\lambda, G, V)$, this check will not be triggered. Hence, we have that

$$\Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1] = \Pr[\mathbf{G}_{B.3}^A \Rightarrow 1 \wedge \text{bad} = 0] = \Pr[\mathbf{G}_{B.3}^A \Rightarrow 1].$$

Game $\mathbf{G}_{B.3.2}$ (Introduce bad' flag for info = info* at the end of the game): In this game, the game introduces a boolean flag **bad'** initialized to $\text{bad}' \leftarrow 0$. At the end of the game when $\mathcal{A}$ outputs its forgeries, the game checks for two conditions:

- For signing sessions $j \in \mathcal{J}_{\text{info}^*}$, check if $a_j = 0$ computed via $(a_j, b_j) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \mu_j)$. If so, set a flag $\text{bad}' = 1$. Otherwise, do nothing.
- For the i^* -th forgery, the game computes two bit-decomposition representatives μ_0^*, μ_1^* as in $\mathbf{G}_{B.4}$ and checks whether there exists a bit $b \in \{0, 1\}$ such that $a_b^* \neq 0 \pmod p$, with a_b^* computed as $(a_b^*, b_b^*) \leftarrow \text{PHF.TrapEval}(\text{td}_{\text{info}^*}, \mu_b^*)$. If so, set $\text{bad}' = 1$.

Then, the game also changes the winning condition of $\mathcal{A}$ to be such that it wins in $\mathbf{G}_{B.3.1}$ and $\text{bad}' = 0 \pmod p$. Note that the event $\text{bad}' = 0$ is equivalent to the event that for all $j \in \mathcal{J}_{\text{info}^*}$, $a_j \neq 0$ and for both $b \in \{0, 1\}$, $a_b^* = 0$.

Then, we have that

$$\begin{aligned} & \Pr[\mathbf{G}_{B.3.2}^A \Rightarrow 1] \\ &= \Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1 \wedge \text{bad}' = 0] \\ &= \Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1 \wedge (\forall j \in \mathcal{J}_{\text{info}^*} : a_j \neq 0 \pmod p) \wedge (\forall b \in \{0, 1\} : a_b^* = 0)]. \end{aligned}$$

We will now use $(2, Q_S, \delta)$ -PHF (proved in Theorem 3.11) to now show that this probability is at least $\delta \cdot \Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1]$. Note that $\mathcal{A}$'s view within this game is independent of the trapdoor $\text{td}_{\text{info}^*}$. Hence, we can consider $\text{td}_{\text{info}^*}$ being sampled at the end conditioning on the value U being sampled. For simplicity, we let ρ denote the random coins sampled in the game that are independent of the generation of $(U, \text{td}_{\text{info}^*}) \leftarrow \text{PHF.TrapGen}(1^\lambda, G, V)$. Then, fixing the PHF key $U = \hat{U}$ sampled for $H_{\text{pb}}(\text{info}^*)$ and the random coins $\rho = \hat{\rho}$, we have that the choices of μ_j for $j \in \mathcal{J}_{\text{info}^*}$ and μ_b^* for $b \in \{0, 1\}$ are fixed, since these only depend on U and the rest of the randomness. Then, $(2, Q_S, \delta)$ -PHF implies the following bound. For any $\hat{U}, \hat{\rho}$, we have

$$\begin{aligned} & \Pr \left[\begin{array}{c} \mathbf{G}_{B.3.1}^A \Rightarrow 1 \wedge \forall j \in \mathcal{J}_{\text{info}^*} : a_j \neq 0 \pmod p \\ \wedge \forall b \in \{0, 1\} : a_b^* = 0 \end{array} \middle| U = \hat{U} \wedge \rho = \hat{\rho} \right] \\ &= \Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1 \mid U = \hat{U} \wedge \rho = \hat{\rho}] \\ & \quad \cdot \Pr[(\forall j \in \mathcal{J}_{\text{info}^*} : a_j \neq 0 \pmod p) \wedge (\forall b \in \{0, 1\} : a_b^* = 0) \mid U = \hat{U} \wedge \rho = \hat{\rho}] \\ &\geq \delta \cdot \Pr[\mathbf{G}_{B.3.1}^A \Rightarrow 1 \mid U = \hat{U} \wedge \rho = \hat{\rho}] \end{aligned}$$

The first equality is due to the probability being only dependent on the trapdoor td , which is independent of the view of $\mathcal{A}$. The second inequality follows from $(2, Q_S, \delta)$ -PHF property (substituting in Theorem 3.11 $q = Q_S$ being the number of queries and $\ell = n$ being the input length).

Then, via the law of total probability (summing over the choices of $\mathbf{U}$ and ρ), we obtain

$$\Pr[\mathbf{G}_{B.3.2}^{\mathcal{A}} \Rightarrow 1] \geq \delta \cdot \Pr[\mathbf{G}_{B.3.1}^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_{B.3.3}$ (Move the setting of bad and bad' flags earlier): In this game, the game moves the setting of the flags **bad** and **bad'** to be the same as in $\mathbf{G}_{B.4}$. Note that the game does not introduce the abort yet at this point, so the view of the adversary remains identical to $\mathbf{G}_{B.3.2}$. Hence,

$$\Pr[\mathbf{G}_{B.3.3}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{B.3.2}^{\mathcal{A}} \Rightarrow 1] .$$

Finally, we observe that $\mathbf{G}_{B.3.3}$ and $\mathbf{G}_{B.3.4} = \mathbf{G}_{B.4}$ only differ when **bad** = 1 or **bad'** = 1 is being set in $\mathbf{G}_{B.3.3}$, which corresponds to the aborts in $\mathbf{G}_{B.4}$. Then, since the winning events for both games consider only the case that **bad** = 0 and **bad'** = 0, we have that

$$\Pr[\mathbf{G}_{B.4}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{B.3.4}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{B.3.3}^{\mathcal{A}} \Rightarrow 1] .$$

Combining all the inequalities concludes the proof. $\square$

8 Blindness Proof of $\mathbf{BS}_{\text{phf}}$ (Theorem 5.5)

Let $\mathcal{A}$ be an adversary in the blindness game of $\mathbf{BS}_{\text{phf}}$ that makes at most Q_{H} and Q_{pb} queries to H and H_{pb} , respectively. Note that $\mathcal{A}$ can be unbounded in its running time. Let us consider the following sequence of games where we denote $\Pr[\mathbf{G}^{\mathcal{A}} \Rightarrow 1]$ as the probability (as a function of λ , omitted) of the game $\mathbf{G}$ outputting 1 while running $\mathcal{A}$.

Game $\mathbf{G}_0$ (Original blindness game): This game is identical to the original blindness game, i.e., the game runs $\text{pp} \leftarrow \mathbf{BS}_{\text{phf}}.\text{Setup}(1^\lambda)$, samples $b \leftarrow \{0, 1\}$, and interacts with $\mathcal{A}$ on input pp while answering the oracles INIT , U_1 , U_2 honestly. When $\mathcal{A}$ outputs b' , the game outputs 1 iff $b' = b$. By definition, we have that

$$\text{Adv}_{\mathbf{BS}_{\text{phf}}}^{\text{blind}}(\mathcal{A}, \lambda) := \left| \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2} \right| .$$

Game $\mathbf{G}_1$ (Abort if H was queried before $\mathcal{A}$ gets (σ_0, σ_1)): In this game, we introduce an additional abort into both signing sessions if the following occurs. Before the H -query (which is of the form $(\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg})$) in Line 15 in $\mathbf{BS}_{\text{phf}}.\text{U}_1$ is to be queried, the game checks if this particular query has been made beforehand. If so, the game aborts. Moreover, if before (σ_0, σ_1) are returned to $\mathcal{A}$ one of these two queries was made by $\mathcal{A}$, the game also aborts.

To argue the advantage change, observe first that C_{rnd} which is kept hidden until (σ_0, σ_1) is uniformly random in $\mathbb{G}$ due to how it is computed as $C_{\text{rnd}} = \text{C}_{\text{ped}}.\text{Commit}_{\text{ck}_{\text{ped}}}(\tilde{\text{st}}_{\text{os}}; r_{\text{rnd}})$ with $r_{\text{rnd}} \leftarrow \mathbb{Z}_p$. Then, the probability that a prior query to H contains C_{rnd} is only $1/p$. Therefore, via a union bound over all H queries (Q_{H} by $\mathcal{A}$ and 2 by the game during U_1) and over two signings sessions, we have that

$$|\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]| \leq \frac{2Q_{\text{H}} + 4}{p} .$$

Game $\mathbf{G}_2$ (Program H with random $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$): Instead of computing $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$ in each signing session via a query to H , this game first samples $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}}) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times$ and programs

$$\text{H}(\mathbb{x}_{\text{os}}, \mathbb{x}_{\text{dl}}, \tilde{\mathbb{x}}_{\text{phf}}, \tilde{A}_{\text{os}}, \tilde{A}_{\text{dl}}, \tilde{A}_{\text{phf}}, \mathbb{x}_{\text{sle}}, \pi_{\text{sle}}, \text{msg}) := (\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}}) ,$$

for each session.

Due to the abort introduced in $\mathbf{G}_1$ and that $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$ still have the same distribution, this game is perfectly indistinguishable from $\mathbf{G}_1$, i.e.,

$$\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] .$$

Game $\mathbf{G}_3$ (Sample challenges $(c_{\text{os}}, c_{\text{or}})$ in the *protocol* uniformly): In this game, instead of sampling $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$ and computing the challenges $(c_{\text{os}}, c_{\text{or}})$ returned in $\mathbf{U}_1$ as

$$\begin{aligned} (\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}}) &\leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times \\ c_{\text{os}} &= \Sigma_{\text{os}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{os}}, \tilde{\text{st}}_{\text{os}}), \\ c_{\text{or}} &= \Sigma_{\text{phf}}.\text{BlindChal}^{-1}(\Sigma_{\text{dl}}.\text{BlindChal}^{-1}(\tilde{c}_{\text{or}}, \tilde{\text{st}}_{\text{dl}}), \tilde{\text{st}}_{\text{phf}}) = \tilde{c}_{\text{or}} \cdot \gamma_{\text{dl}}^{-1} \cdot \gamma_{\text{phf}}^{-1}, \end{aligned}$$

the game now does the following:

$$\begin{aligned} (c_{\text{os}}, c_{\text{or}}) &\leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times \\ \tilde{c}_{\text{os}} &= \Sigma_{\text{os}}.\text{BlindChal}(c_{\text{os}}, \tilde{\text{st}}_{\text{os}}), \\ \tilde{c}_{\text{or}} &= \Sigma_{\text{dl}}.\text{BlindChal}(\Sigma_{\text{phf}}.\text{BlindChal}(c_{\text{or}}, \tilde{\text{st}}_{\text{phf}}), \tilde{\text{st}}_{\text{dl}}) = c_{\text{or}} \cdot \gamma_{\text{dl}} \cdot \gamma_{\text{phf}}. \end{aligned} \tag{15}$$

We note that this does not change the view of $\mathcal{A}$ due to the $\Sigma_{\text{os}}.\text{Rand}$, $\Sigma_{\text{dl}}.\text{Rand}$, and $\Sigma_{\text{phf}}.\text{Rand}$ –blindability of Σ_{os} , Σ_{dl} , and Σ_{phf} , respectively. This is because the property guarantees that the BlindChal algorithms are invertible (and hence are bijections). Hence, this change is simply reversing the sampling of $(\tilde{c}_{\text{os}}, \tilde{c}_{\text{or}})$. Thus,

$$\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1].$$

Note that from this point onward, the simulation of the user sessions except for the final signatures does not depend on the messages $\text{msg}_0, \text{msg}_1$ picked by the adversary. In particular, this is because all the aborts (both in $\text{BS}_{\text{phf}}.\mathbf{U}_1$ and $\text{BS}_{\text{phf}}.\mathbf{U}_2$) only depend on the transcript at hand which only consists of public information known to $\mathcal{A}$ (the challenges c_{os} and c_{or} are uniformly random). Therefore, in the following games, we assume that the transcripts generated for both sessions via the interaction with $\mathcal{A}$ are always valid, i.e., they pass the checks performed in $\text{BS}_{\text{phf}}.\mathbf{U}_2$. Otherwise, the game will return $(\perp, \perp)$. Note here that this implies that $c_{\text{phf}}, c_{\text{dl}}, e_{\text{s}} \neq 0$ due to the non-zero checks in $\text{BS}_{\text{phf}}.\mathbf{U}_1$ (Line 1) and $\text{BS}_{\text{phf}}.\mathbf{U}_2$ (Line 1).

Game $\mathbf{G}_4$ (Delay all computation to before (σ_0, σ_1) are determined): This game now delays all user-side computation for both signing sessions, except for the check that $e_{\text{s}} \neq 0$, to the oracle $\mathbf{U}_2$. In particular, the two oracles now perform the following procedures: (We omitted session numbering for readability.)

Oracle $\mathbf{U}_1$. Check that the input $(\text{info}, \text{msg}, \text{smsg}_1)$ is of the right format and that $e_{\text{s}} \neq 0$. If the checks pass, return $(c_{\text{os}}, c_{\text{or}}) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p^\times$.

Oracle $\mathbf{U}_2$. On input smsg_2 and the input from the prior round $(\text{info}, \text{msg}, \text{smsg}_1)$, run the simulation algorithm SimBS in Figure 13.

We note that from this point onwards, we will only modify the simulation algorithm in Figure 13 with the denoted changes attributed to the games. (The final simulation algorithm SimBS' is given in the same figure.)

Also, this change is only syntactic since the output of $\mathbf{U}_1$ is independent of everything else by the change in $\mathbf{G}_3$. Therefore,

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_5$ (Simulate π_{sle}): In this game, at the end of the second call to $\mathbf{U}_2$ when running SimBS , we compute the NIZK proof π_{sle} via $\pi_{\text{sle}} \leftarrow \Pi_{\text{sle}}.\text{Sim}^{\text{H}_{\text{sle}}}(\mathbb{x}_{\text{sle}})$ where $\mathbb{x}_{\text{sle}}$ is computed as in game $\mathbf{G}_4$. This change is depicted in Line 24 of SimBS and Line 23 of SimBS' .

We bound the advantage change from $\mathbf{G}_4$ via a reduction $\mathcal{B}_{\text{zk}}$ that simulates the blindness game as in $\mathbf{G}_4$ and runs $\mathcal{A}$ as a subroutine. Here, $\mathcal{B}_{\text{zk}}$ would set up $\text{pp} \leftarrow \text{BS}_{\text{phf}}.\text{Setup}(1^\lambda)$, samples $b \leftarrow \{0, 1\}$, interacts with $\mathcal{A}$ on input pp while answering the oracles $\text{INIT}, \mathbf{U}_1, \mathbf{U}_2$ honestly, forwards $\mathcal{A}$'s queries to the random oracle H_{sle} to its own game, and simulates H and H_{pb} to $\mathcal{A}$ via lazy sampling. However, the only difference from game $\mathbf{G}_4$ is that when computing the proof π_{sle} , $\mathcal{B}_{\text{zk}}$ makes a query $\pi_{\text{sle}} \leftarrow \mathcal{O}(\mathbb{x}_{\text{sle}}, \mathbb{w}_{\text{sle}})$ to its own oracle (which could be $\mathcal{O}_{\text{Prove}}$ or $\mathcal{O}_{\text{Sim}}$). Then, $\mathcal{B}_{\text{zk}}$ outputs 1 if the guess from $\mathcal{A}$ $b' = b$ and 0 otherwise. The query complexities of $\mathcal{B}_{\text{zk}}$ and $\mathcal{A}$ are the same, and it is then easy to see that

$$|\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1]| \leq \text{Adv}_{\Pi_{\text{sle}}, \text{Sim}}^{\text{zk}}(\mathcal{B}_{\text{zk}}, \lambda).$$

Alg. SimBS(info, msg, msg ₁ , (c _{os} , c _{or}), msg ₂) : // G ₄	Alg. SimBS'(info, msg, msg ₁ , (c _{os} , c _{or}), msg ₂) : // G ₁₁
1 : parse msg ₁ = (A _{os} , A _{dl} , A _{phf} , T _s , e _s) 2 : parse msg ₂ = (c _{phf} , z _{os} , z _{phf} , z _{dl}) 3 : $\mathbb{x}_{os} = (G, H, W)$; $\mathbb{x}_{dl} = (G, X)$ 4 : $\mathbb{x}_{phf} = (G, V, \mathbf{U}, 0, A_{os}, T_s, e_s)$ 5 : $c_{dl} := c_{or}/c_{phf}$ 6 : $\tilde{\mathbf{st}}_{os} \leftarrow \mathbb{Z}_p^3$; $r_{rnd} \leftarrow \mathbb{Z}_p$ 7 : $C_{rnd} := C_{ped} \cdot \text{Commit}_{ck_{ped}}(\tilde{\mathbf{st}}_{os}; r_{rnd})$ // to G ₆ 8 : $\tilde{A}_{os} := \Sigma_{os} \cdot \text{BlindNonce}(\mathbb{x}_{os}, A_{os}; \tilde{\mathbf{st}}_{os})$ // to G ₇ 9 : $\tilde{c}_{os} := \Sigma_{os} \cdot \text{BlindChal}(c_{os}, \tilde{\mathbf{st}}_{os})$ // to G ₇ 10 : $\tilde{z}_{os} := \Sigma_{os} \cdot \text{BlindResp}(z_{os}, c_{os}, \tilde{\mathbf{st}}_{os})$ // to G ₇ 11 : $(\tilde{A}_{dl}, \tilde{\mathbf{st}}_{dl}) \leftarrow \Sigma_{dl} \cdot \text{BlindNonce}(\mathbb{x}_{dl}, A_{dl})$ // to G ₈ 12 : $\tilde{c}_{dl} := \Sigma_{dl} \cdot \text{BlindChal}(c_{dl}, \tilde{\mathbf{st}}_{dl})$ // to G ₈ 13 : $\tilde{z}_{dl} := \Sigma_{dl} \cdot \text{BlindResp}(z_{dl}, c_{dl}, \tilde{\mathbf{st}}_{dl})$ // to G ₈ 14 : $\nu, r_s, \rho \leftarrow \mathbb{Z}_p$; $e' \leftarrow \mathbb{Z}_p^\times$; $\text{coins}_{phf} := (\nu, r_s, \rho, e')$ 15 : $\tilde{\mathbb{x}}_{phf} := \Sigma_{phf} \cdot \text{TransStmt}(\mathbb{x}_{phf}, \text{coins}_{phf})$ // to G ₁₀ 16 : parse $\tilde{\mathbb{x}}_{phf} = (G, V, \mathbf{U}, C_s, \tilde{A}_s, \tilde{T}_s, \tilde{e}_s)$ 17 : $A'_{phf} := \Sigma_{phf} \cdot \text{TransNonce}(\mathbb{x}_{phf}, A_{phf}, \text{coins}_{phf})$ 18 : $(\tilde{A}_{phf}, \tilde{\mathbf{st}}_{phf}) := \Sigma_{phf} \cdot \text{BlindNonce}(\tilde{\mathbb{x}}_{phf}, A'_{phf})$ // to G ₉ 19 : $z'_{phf} := \Sigma_{phf} \cdot \text{TransResp}(\mathbb{x}_{phf}, A_{phf}, c_{phf}, z_{phf}, \text{coins}_{phf})$ 20 : $\tilde{c}_{phf} := \Sigma_{phf} \cdot \text{BlindChal}(c_{phf}, \tilde{\mathbf{st}}_{phf})$ // to G ₉ 21 : $\tilde{z}_{phf} := \Sigma_{phf} \cdot \text{BlindResp}(z'_{phf}, c_{phf}, \tilde{\mathbf{st}}_{phf})$ // to G ₉ 22 : $\mathbb{x}_{sle} = (ck_{ped}, \mathbb{x}_{os}, C_s, C_{rnd}, \tilde{A}_s, \tilde{A}_{os})$ 23 : $\mathbf{w}_{sle} = (\nu, r_s, \tilde{\mathbf{st}}_{os}, r_{rnd})$ 24 : $\pi_{sle} \leftarrow \Pi_{sle} \cdot \text{Prove}^{H_{sle}}(\mathbb{x}_{sle}, \mathbf{w}_{sle})$ // to G ₄ 25 : $\tilde{c}_{or} := \Sigma_{dl} \cdot \text{BlindChal}(\Sigma_{phf} \cdot \text{BlindChal}(c_{or}, \tilde{\mathbf{st}}_{phf}), \tilde{\mathbf{st}}_{dl})$ // to G ₅ 26 : $\tilde{\tau}_{xxx} := (\tilde{A}_{xxx}, \tilde{c}_{xxx}, \tilde{z}_{xxx})$ for $xxx \in \{\text{os}, \text{dl}, \text{phf}\}$ 27 : $H(\mathbb{x}_{os}, \mathbb{x}_{dl}, \tilde{\mathbb{x}}_{phf}, \tilde{A}_{os}, \tilde{A}_{dl}, \tilde{A}_{phf}, \mathbb{x}_{sle}, \pi_{sle}, \text{msg}) := (\tilde{c}_{os}, \tilde{c}_{or})$ 28 : return $\sigma = (C_s, \tilde{A}_s, \tilde{T}_s, \tilde{e}_s, C_{rnd}, \pi_{sle}, \tilde{\tau}_{os}, \tilde{\tau}_{dl}, \tilde{\tau}_{phf})$	1 : parse msg ₁ = (A _{os} , A _{dl} , A _{phf} , T _s , e _s) 2 : parse msg ₂ = (c _{phf} , z _{os} , z _{phf} , z _{dl}) 3 : $\mathbb{x}_{os} = (G, H, W)$; $\mathbb{x}_{dl} = (G, X)$ 4 : $\mathbb{x}_{phf} = (G, V, \mathbf{U}, 0, A_{os}, T_s, e_s)$ 5 : $c_{dl} := c_{or}/c_{phf}$ 6 : $r_{rnd} \leftarrow \mathbb{Z}_p$ 7 : $C_{rnd} := C_{ped} \cdot \text{Commit}_{ck_{ped}}((0, 0, 0); r_{rnd})$ // since G ₇ 8 : $\tilde{c}_{os} \leftarrow \mathbb{Z}_p$ // since G ₈ 9 : $(\tilde{A}_{os}, \tilde{z}_{os}) := \Sigma_{os} \cdot \text{Sim}(\mathbb{x}_{os}, \tilde{c}_{os})$ // since G ₈ 10 : 11 : $\tilde{c}_{dl} \leftarrow \mathbb{Z}_p^\times$ // since G ₉ 12 : $(\tilde{A}_{dl}, \tilde{z}_{dl}) \leftarrow \Sigma_{dl} \cdot \text{Sim}(\mathbb{x}_{dl}, \tilde{c}_{dl})$ // since G ₉ 13 : 14 : 15 : $C_s, \tilde{A}_s, \tilde{T}_s \leftarrow \mathbb{G}$; $\tilde{e}_s \leftarrow \mathbb{Z}_p^\times$ 16 : $\tilde{\mathbb{x}}_{phf} := (G, V, \mathbf{U}, C_s, \tilde{A}_s, \tilde{T}_s, \tilde{e}_s)$ // G ₁₁ 17 : $\tilde{c}_{phf} \leftarrow \mathbb{Z}_p^\times$ // since G ₁₀ 18 : $(\tilde{A}_{phf}, \tilde{z}_{phf}) \leftarrow \Sigma_{phf} \cdot \text{Sim}(\tilde{\mathbb{x}}_{phf}, \tilde{c}_{phf})$ // since G ₁₀ 19 : 20 : 21 : 22 : $\mathbb{x}_{sle} = (ck_{ped}, \mathbb{x}_{os}, C_s, C_{rnd}, \tilde{A}_s, \tilde{A}_{os})$ 23 : $\pi_{sle} \leftarrow \Pi_{sle} \cdot \text{Sim}^{H_{sle}}(\mathbb{x}_{sle})$ // since G ₅ 24 : 25 : $\tilde{c}_{or} := \tilde{c}_{phf} \cdot \tilde{c}_{dl}$ // since G ₆ 26 : $\tilde{\tau}_{xxx} := (\tilde{A}_{xxx}, \tilde{c}_{xxx}, \tilde{z}_{xxx})$ for $xxx \in \{\text{os}, \text{dl}, \text{phf}\}$ 27 : $H(\mathbb{x}_{os}, \mathbb{x}_{dl}, \tilde{\mathbb{x}}_{phf}, \tilde{A}_{os}, \tilde{A}_{dl}, \tilde{A}_{phf}, \mathbb{x}_{sle}, \pi_{sle}, \text{msg}) := (\tilde{c}_{os}, \tilde{c}_{or})$ 28 : return $\sigma = (C_s, \tilde{A}_s, \tilde{T}_s, \tilde{e}_s, C_{rnd}, \pi_{sle}, \tilde{\tau}_{os}, \tilde{\tau}_{dl}, \tilde{\tau}_{phf})$

Fig. 13. Simulation algorithm of U_2 starting from **G**₄ to **G**₁₁. We assume that the assertions made in $\text{BS}_{phf}.U_2$ have already been checked before running this algorithm. We denote the two versions (**G**₄ and **G**₁₁) and detail the game transition with comments.

Game G_6 (Set $\tilde{c}_{\text{or}} = \tilde{c}_{\text{phf}} \cdot \tilde{c}_{\text{dl}}$): In this game, we change how $\tilde{c}_{\text{or}}$ is computed from Equation (15) to

$$\tilde{c}_{\text{or}} = \tilde{c}_{\text{phf}} \cdot \tilde{c}_{\text{dl}} . \quad (16)$$

This change is depicted in Line 25 of SimBS and SimBS' in Figure 13.

Note that this does not change the view of $\mathcal{A}$ because from Equation (15) and how $c_{\text{or}}, c_{\text{dl}}, c_{\text{phf}} \neq 0$ (due to the sessions being non-aborting) are defined ($c_{\text{or}} = c_{\text{dl}} \cdot c_{\text{phf}}$), we have that Equation (16) is satisfied via Lemma 4.4. Therefore,

$$\Pr[G_5^{\mathcal{A}} \Rightarrow 1] = \Pr[G_6^{\mathcal{A}} \Rightarrow 1] .$$

Game G_7 (Sample C_{rnd} as commitment of $(0, 0, 0)$): This game now computes the commitment C_{rnd} as $C_{\text{rnd}} := C_{\text{ped}}.\text{Commit}_{\text{ck}_{\text{ped}}}((0, 0, 0); r_{\text{rnd}})$ for $r_{\text{rnd}} \leftarrow \mathbb{Z}_p$ instead of committing to the blinding randomness $\tilde{\text{st}}_{\text{os}} \leftarrow \mathbb{Z}_p^3$. This change is depicted in Line 7 of both SimBS and SimBS' in Figure 13.

This does not change the view of $\mathcal{A}$ due to the perfect hiding property of C_{ped} . Therefore,

$$\Pr[G_6^{\mathcal{A}} \Rightarrow 1] = \Pr[G_7^{\mathcal{A}} \Rightarrow 1] .$$

Game G_8 (Simulate Σ_{os} transcripts): In this game, we simulate the transcript $\tilde{\tau}_{\text{os}} = (\tilde{A}_{\text{os}}, \tilde{c}_{\text{os}}, \tilde{z}_{\text{os}})$ as follows: (a) Sample $\tilde{c}_{\text{os}} \leftarrow \mathbb{Z}_p$, (b) Simulate $(\tilde{A}_{\text{os}}, \tilde{z}_{\text{os}}) \leftarrow \Sigma_{\text{os}}.\text{Sim}(\mathbb{x}_{\text{os}}, \tilde{c}_{\text{os}})$. This change is depicted in Line 9 of SimBS' removing the pseudocode from Lines 8 to 10 of SimBS in Figure 13.

Due to the strong $\Sigma_{\text{os}}.\text{Rand-blindability}$ of Σ_{os} (proved in Lemma 4.2) and the fact that the transcript $(A_{\text{os}}, c_{\text{os}}, z_{\text{os}})$ is valid (via the validity check in $\text{BS}_{\text{phf}}.\text{U}_2$), we have that the transcript still has the same distribution as in the prior game. Thus,

$$\Pr[G_7^{\mathcal{A}} \Rightarrow 1] = \Pr[G_8^{\mathcal{A}} \Rightarrow 1] .$$

Game G_9 (Simulate Σ_{dl} transcripts): In this game, we simulate the transcript $\tilde{\tau}_{\text{dl}} = (\tilde{A}_{\text{dl}}, \tilde{c}_{\text{dl}}, \tilde{z}_{\text{dl}})$ as follows: (a) Sample $\tilde{c}_{\text{dl}} \leftarrow \mathbb{Z}_p^\times$, (b) Simulate $(\tilde{A}_{\text{dl}}, \tilde{z}_{\text{dl}}) \leftarrow \Sigma_{\text{dl}}.\text{Sim}(\mathbb{x}_{\text{dl}}, \tilde{c}_{\text{dl}})$. This change is depicted in Line 12 of SimBS' removing the pseudocode from Lines 11 to 13 of SimBS in Figure 13.

Due to the strong $\Sigma_{\text{dl}}.\text{Rand-blindability}$ of Σ_{dl} (proved in Lemma 4.1) and the fact that the transcript $(A_{\text{dl}}, c_{\text{dl}}, z_{\text{dl}})$ is valid (via the validity check in $\text{BS}_{\text{phf}}.\text{U}_2$), we have that the transcript still has the same distribution as in the prior game. Thus,

$$\Pr[G_8^{\mathcal{A}} \Rightarrow 1] = \Pr[G_9^{\mathcal{A}} \Rightarrow 1] .$$

Game G_{10} (Simulate Σ_{phf} transcripts): In this game, we simulate the transcript $\tilde{\tau}_{\text{phf}} = (\tilde{A}_{\text{phf}}, \tilde{c}_{\text{phf}}, \tilde{z}_{\text{phf}})$ as follows: (a) Sample $\tilde{c}_{\text{phf}} \leftarrow \mathbb{Z}_p^\times$, (b) Simulate $(\tilde{A}_{\text{phf}}, \tilde{z}_{\text{phf}}) \leftarrow \Sigma_{\text{phf}}.\text{Sim}(\tilde{\mathbb{x}}_{\text{phf}}, \tilde{c}_{\text{phf}})$. Note that the simulator is run on the *translated statement*. This change is depicted in Line 18 of SimBS' removing the pseudocode from Lines 17 to 21 of SimBS in Figure 13.

To argue that this change is indistinguishable, we will rely on the strong $\Sigma_{\text{phf}}.\text{Trans-translatability}$ and the strong $\Sigma_{\text{phf}}.\text{Rand-blindability}$, proved in Lemma 4.3. First, due to the strong $\Sigma_{\text{phf}}.\text{Trans-translatability}$ and the fact that the transcript $(A_{\text{phf}}, c_{\text{phf}}, z_{\text{phf}})$ is valid for the statement $\mathbb{x}_{\text{phf}}$ (w.r.t $\Sigma_{\text{phf}}.\text{Ver}$), we have that the translated transcript $(A'_{\text{phf}}, c_{\text{phf}}, z'_{\text{phf}})$ (translated via the coins $\text{coins}_{\text{phf}}$) is also valid for the translated statement $\tilde{\mathbb{x}}_{\text{phf}}$ (w.r.t. $\Sigma_{\text{phf}}.\text{Ver}$). Next, due to the $\Sigma_{\text{phf}}.\text{Rand-blindability}$ of Σ_{phf} , we have that the simulated transcript using the translated statement $\tilde{\mathbb{x}}_{\text{phf}}$ still has the same distribution as in the blinded transcript. Thus,

$$\Pr[G_9^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{10}^{\mathcal{A}} \Rightarrow 1] .$$

At this point, none of the Σ -protocol transcripts directly rely on the Σ -protocol transcript from $\mathcal{A}$, except for the translated statement $\tilde{\mathbb{x}}_{\text{phf}}$.

Game G_{11} (Sample $(C_{\text{s}}, \tilde{A}_{\text{s}}, \tilde{T}_{\text{s}}, \tilde{e}_{\text{s}})$ uniformly): This game now samples $C_{\text{s}}, \tilde{A}_{\text{s}}, \tilde{T}_{\text{s}} \leftarrow \mathbb{G}$ and $\tilde{e}_{\text{s}} \leftarrow \mathbb{Z}_p^\times$ uniformly at random from their corresponding domains. This change is depicted in Line 15 of both SimBS and SimBS' in Figure 13. Note that this change is possible since we do not rely on the coins for translating the transcripts anymore due to the change in G_{10} .

To argue the indistinguishability from the prior game, we observe first how the algorithm $\Sigma_{\text{phf}}.\text{TransStmt}$ is defined. On input $\mathfrak{x}_{\text{phf}} = (G, V, \mathbf{U}, 0_{\mathbb{G}}, A_{\text{os}}, T_{\S}, e_{\S})$ as in $\mathbf{G}_{10}$ and coins $\text{coins}_{\text{phf}} = (\nu, r_{\S}, \rho, e') \leftarrow_{\S} \mathbb{Z}_p^3 \times \mathbb{Z}_p^{\times}$, the algorithm maps $\mathfrak{x}_{\text{phf}} \mapsto \tilde{\mathfrak{x}}_{\text{phf}} = (G, V, \mathbf{U}, C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$ where

$$\begin{aligned} C_{\S} &= \nu G + r_{\S} V, \\ \tilde{A}_{\S} &= A_{\text{os}} + \nu V, \\ \tilde{T}_{\S} &= T_{\S} + \rho V, \\ \tilde{e}_{\S} &= e' \cdot e_{\S}. \end{aligned}$$

Since $V \neq 0_{\mathbb{G}}$ by how it is sampled and fixing the statement $\mathfrak{x}_{\text{phf}}$, one can easily give a bijection mapping the tuple $(C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$ to the randomness $(\nu, r_{\S}, \rho, e')$. Hence, $(C_{\S}, \tilde{A}_{\S}, \tilde{T}_{\S}, \tilde{e}_{\S})$ is uniformly random in $\mathbb{G}^3 \times \mathbb{Z}_p^{\times}$. Therefore, we have that

$$\Pr[\mathbf{G}_{10}^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_{11}^{\mathcal{A}} \Rightarrow 1].$$

ADVANTAGE OF $\mathcal{A}$ IN $\mathbf{G}_{11}$. Finally, we observe that the view of $\mathcal{A}$ in $\mathbf{G}_{11}$ does not depend on bit b as the simulation of the signatures does not rely at all on the signing transcript. Hence,

$$\Pr[\mathbf{G}_{11}^{\mathcal{A}} \Rightarrow 1] = \frac{1}{2}.$$

The theorem then follows from combining all the advantage bounds. $\square$

Acknowledgements

Chairattana-Apirom and Tessaro’s research was partially supported by NSF grants CNS-2026774, CNS-2154174, CNS-2426905, a gift from Microsoft, and a Stellar Development Foundation Academic Research 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, Berlin, 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, Berlin, Heidelberg, November 1996.
- Ame05. American National Standards Institute, Inc. ANSI X9.62 public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA), November 16, 2005.
- App. iCloud private relay overview. https://www.apple.com/privacy/docs/iCloud_Private_Relay_Overview_Dec2021.PDF.
- 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.
- BHKR25. Nicholas Brandt, Dennis Hofheinz, Michael Klooß, and Michael Reichle. Tightly-secure blind signatures in pairing-free groups. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 337–369. Springer, Singapore, December 2025.
- BLL⁺21. Fabrice Benhamouda, Tancrede 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, Cham, October 2021.
- BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Cham, August 2019.
- BN06. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.

- 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.
- CART26. Rutchathon Chairattana-Apirom, Michael Reichle, and Stefano Tessaro. Playing tag with okamoto-schnorr: Three-move pairing-free blind signatures from ddh. In *Annual International Cryptology Conference*, pages 393–425. Springer, 2026.
- CATZ24. Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Pairing-free blind signatures from CDH assumptions. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part I*, volume 14920 of *LNCS*, pages 174–209. Springer, Cham, August 2024.
- CDS94. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO’94*, volume 839 of *LNCS*, pages 174–187. Springer, Berlin, Heidelberg, August 1994.
- CDV26. Michele Ciampi, Pierpaolo Della Monica, and Ivan Visconti. Round-optimal GUC-secure blind signatures from minimal computational and setup assumptions. In *Advances in Cryptology – CRYPTO 2026*, volume 16806 of *Lecture Notes in Computer Science*, pages 522–554. Springer, 2026.
- CDVW24. Sofia Celi, Alex Davidson, Steven Valdez, and Christopher A. Wood. Privacy Pass Issuance Protocols. RFC 9578, June 2024.
- 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, New York, August 1990.
- CGKR22. Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, and Michael Reichle. Sharp: Short relaxed range proofs. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 609–622. ACM Press, November 2022.
- Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO’82*, pages 199–203. Plenum Press, New York, USA, 1982.
- Che26. Yanbo Chen. Three-move blind signatures in pairing-free groups. In *Annual International Cryptology Conference*, pages 588–617. Springer, 2026.
- CKLR21. Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. Efficient range proofs with transparent setup from bounded integer commitments. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 247–277. Springer, Cham, October 2021.
- CKM⁺23. Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Snowblind: A threshold blind signature in pairing-free groups. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 710–742. Springer, Cham, August 2023.
- CMZ14. Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic MACs and keyed-verification anonymous credentials. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 2014*, pages 1205–1216. ACM Press, November 2014.
- 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, Berlin, Heidelberg, August 1993.
- DGS⁺18. Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. Privacy pass: Bypassing internet challenges anonymously. *PoPETs*, 2018(3):164–180, July 2018.
- DH26. Marian Dietz and Dennis Hofheinz. Schnorr-like signatures in the non-observable random oracle model. Cryptology ePrint Archive, Paper 2026/1097, 2026.
- DHH⁺21. Nico Döttling, Dominik Hartmann, Dennis Hofheinz, Eike Kiltz, Sven Schäge, and Bogdan Ursu. On the impossibility of purely algebraic signatures. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 317–349. Springer, Cham, November 2021.
- DKPW12. Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message authentication, revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 355–374. Springer, Berlin, Heidelberg, April 2012.
- DKT26. Marian Dietz, Julia Kastner, and Stefano Tessaro. On the impossibility of round-optimal pairing-free blind signatures in the ROM. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026*, volume 16806, pages 490–521. Springer, August 2026.
- DV26. Pierpaolo Della Monica and Ivan Visconti. Improved concurrent-secure blind Schnorr signatures. In *Advances in Cryptology – CRYPTO 2026*, volume 16806 of *Lecture Notes in Computer Science*, pages 457–489. Springer, 2026.
- Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Berlin, Heidelberg, August 2005.

- Fis06. Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, Berlin, Heidelberg, August 2006.
- 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, Cham, August 2018.
- FOO93. Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT’92*, volume 718 of *LNCS*, pages 244–251. Springer, Berlin, Heidelberg, December 1993.
- 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, Cham, May 2020.
- FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO’86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987.
- FW24. Georg Fuchsbauer and Mathias Wolf. Concurrently secure blind Schnorr signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 124–160. Springer, Cham, May 2024.
- Goo. VPN by Google One, explained. <https://one.google.com/about/vpn/howitworks>.
- HIJ⁺21. Sharon Huang, Subodh Iyengar, Sundar Jeyaraman, Shiv Kushwah, Chen-Kuei Lee, Zutian Luo, Payman Mohassel, Ananth Raghunathan, Shaahid Shaikh, Yen-Chieh Sung, and Albert Zhang. DIT: De-Identified Authenticated Telemetry at Scale. Technical report, April 2021.
- 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.
- HK08. Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38. Springer, Berlin, Heidelberg, August 2008.
- HK11. Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. Cryptology ePrint Archive, Report 2011/270, 2011.
- HKLN20. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 500–529. Springer, Cham, August 2020.
- HLM⁺25. Lucjan Hanzlik, Yi-Fu Lai, Marzio Mula, Eugenio Paracucchi, Daniel Slamanig, and Gang Tang. Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum group actions. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part IV*, volume 16248 of *LNCS*, pages 35–69. Springer, Singapore, December 2025.
- HLPP26. Lucjan Hanzlik, Yi-Fu Lai, Eugenio Paracucchi, and Edoardo Persichetti. Post-quantum blind signature from standard group action assumptions and more. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part I*, volume 16541 of *LNCS*, pages 277–307. Springer, Cham, May 2026.
- JLO97. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski, Jr., editor, *CRYPTO’97*, volume 1294 of *LNCS*, pages 150–164. Springer, Berlin, Heidelberg, August 1997.
- 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, Cham, November 2020.
- KLLQ23. Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 729–761. Springer, Cham, August 2023.
- KLR26. Michael Klooß, Russell W. F. Lai, and Michael Reichle. Blind signatures from arguments of inequality. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part VII*, volume 16806 of *Lecture Notes in Computer Science*, pages 426–456. Springer, 2026.
- KLX22. Julia Kastner, Julian Loss, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 468–497. Springer, Cham, March 2022.
- KR25. Michael Klooß and Michael Reichle. Blind signatures from proofs of inequality. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 157–189. Springer, Cham, August 2025.

- KRW24. Michael Klooß, Michael Reichle, and Benedikt Wagner. Practical blind signatures in pairing-free groups. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 363–395. Springer, Singapore, December 2024.
- Ks22. Yashvanth Kondi and abhi shelat. Improved straight-line extraction in the random oracle model with applications to signature aggregation. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part II*, volume 13792 of *LNCS*, pages 279–309. Springer, Cham, December 2022.
- KTZ26. Julia Kastner, Stefano Tessaro, and Greg Zaverucha. Round-optimal pairing-free blind signatures. Cryptology ePrint Archive, Paper 2026/091, 2026.
- 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, Berlin, Heidelberg, December 2005.
- Mau15. Ueli Maurer. Zero-knowledge proofs of knowledge for group homomorphisms. *DCC*, 77(2-3):663–676, 2015.
- Oka93. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO’92*, volume 740 of *LNCS*, pages 31–53. Springer, Berlin, Heidelberg, August 1993.
- OO92. Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In Joan Feigenbaum, editor, *CRYPTO’91*, volume 576 of *LNCS*, pages 324–337. Springer, Berlin, Heidelberg, August 1992.
- 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.
- Ped92. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO’91*, volume 576 of *LNCS*, pages 129–140. Springer, Berlin, Heidelberg, August 1992.
- PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.
- RR26. Michael Reichle and Zoé Reinke. Threshold blind signatures from CDH. In Shi Bai and Edoardo Persichetti, editors, *PKC 2026, Part I*, volume 16551 of *LNCS*, pages 269–301. Springer, Cham, May 2026.
- 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, New York, August 1990.
- Tru. Trust tokens. <https://developer.chrome.com/docs/privacy-sandbox/trust-tokens/>.
- TZ22. Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 782–811. Springer, Cham, May / June 2022.
- TZ23. Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Cham, April 2023.
- Wat05. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Berlin, Heidelberg, May 2005.
- YAZ⁺19. Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 147–175. Springer, Cham, August 2019.

A AI Usage

Except as described below, all results, proofs, constructions, and technical ideas in this paper were developed entirely by the human authors and were not generated using AI tools.

The sole exception is Theorem 3.11, which does not appear to follow from the statements proved in prior work. We initially identified a possible proof strategy and used GPT-5.6 Sol to check whether this approach could be made rigorous. In doing so, the model produced a substantially cleaner proof, using only the probabilistic and combinatorial tools already present in the analysis of Hofheinz and Kiltz [HK08]. We subsequently verified the argument in full and take responsibility for its correctness.

We emphasize that Theorem 3.11 is not the core conceptual and technical contribution on which the blind-signature construction rests: the protocol design and its security analysis are independent of the novelty of this particular PHF bound. If Theorem 3.11 were instead treated as a known auxiliary lemma, or replaced by an equivalent suitable programmability statement, the main contribution of the paper would remain unchanged.

We additionally used GPT-6 Astra as a manuscript-reviewing and editing tool. In particular, it was used to inspect drafts for minor presentation issues and small technical inconsistencies, several of which we corrected after independent verification, and to assist with polishing and improving the exposition. It was not used to generate any other theorem, proof, construction, attack, or cryptographic argument appearing in the paper.

B Proof of Theorem 3.11 ((2, poly)-Programmability)

We note that in this section, we stick to using capital letters to denote bit-vectors, e.g., $X = (x_1, \dots, x_\ell)$. This is to make the notation consistent with prior works on PHFs. Now, we let $a_{ij} \leftarrow_{\$} \{-1, 0, 1\}$ for $i = 0, \dots, \ell$ and $j = 1, \dots, J$. Let $a_i = \sum_{j=1}^J a_{ij}$, and define

$$a_X = -1 + a_0 + \sum_{i=1}^{\ell} x_i a_i \in \mathbb{Z}.$$

We need to prove that for all fixed choices of the key K' and $X, Y, Z_1, \dots, Z_q \in \{0, 1\}^\ell$ such that $Z_j \notin \{X, Y\}$ for all $j \in [q]$, we have

$$\Pr[a_X = 0 \wedge a_Y = 0 \wedge \forall j \in [q] : a_{Z_j} \neq 0] \geq \delta.$$

Note that this statement holds conditioned on the choice of the key K' , because the key K' is statistically independent from the a_{ij} 's. Observe that $a_{Z_j} \neq 0$ over the integers implies $a_{Z_j} \neq 0 \bmod p$ if $(\ell + 1)J + 1 < p$ by construction.

RANDOM WALK BACKGROUND. Let $(\xi_i)_{i \geq 1}$ be independent random variables, each uniform in $\{-1, 0, 1\}$. For any $T \subseteq \mathbb{N}$, define $W(T) = \sum_{i \in T} \xi_i$. We also define $p_n(t) = \Pr[W([n]) = t]$. We need the following lemma, which collects a few facts from Theorem 18 in [HK11].

Lemma B.1. *There exists a constant $\Lambda > 0$, as well as $\lambda_c > 0$ for every $c > 0$, such that for every $t \in \mathbb{Z}$ and $n \in \mathbb{N}$,*

$$p_n(t) \leq \frac{\Lambda}{\sqrt{n}},$$

and if additionally $|t| \leq c\sqrt{n}$ (for $|t| \leq n$),

$$p_n(t) \geq \frac{\lambda_c}{\sqrt{n}}.$$

Furthermore, for $t \geq 0$, $p_n(t+1) \leq p_n(t)$, and $p_n(t) = p_n(-t)$.

We need the following lemma, which strengthens Lemma 19 from [HK11]. We use $\lambda = \lambda_1$ in the following.

Lemma B.2. Let $R \subsetneq S \subseteq \mathbb{N}$ such that $|R| \geq J$ and $|S \setminus R| \geq J$. Then, for any $t \in \mathbb{Z}$ with $|t| \leq \sqrt{|S|}$,

$$\max_{u \in \mathbb{Z}} \Pr[W(R) = u \mid W(S) = t] \leq \frac{\sqrt{2}A^2}{\lambda\sqrt{J}}.$$

Proof. Let $r = |R|$ and $s = |S \setminus R|$. Then, because the ξ_i 's are independent,

$$\begin{aligned} \Pr[W(R) = u \mid W(S) = t] &= \frac{p_r(u) \cdot p_s(t-u)}{p_{r+s}(t)} \\ &\leq \frac{A^2}{\lambda} \sqrt{\frac{1}{r} + \frac{1}{s}} \leq \frac{A^2}{\lambda} \sqrt{\frac{2}{J}}, \end{aligned}$$

where we repeatedly used Lemma B.1. □

CORE LEMMA. The rest of the proof relies on the following core lemma.

Lemma B.3. For every $X, Y \in \{0, 1\}^\ell$, there exists an event $\text{Bal}_{X,Y} \subseteq \{a_X = a_Y = 0\}$ such that:

1. $\Pr[\text{Bal}_{X,Y}] \geq \frac{c_0}{(\ell+1)J}$.
2. for every $Z \notin \{X, Y\}$,

$$\Pr[a_Z = 0 \mid \text{Bal}_{X,Y}] \leq \frac{C_0}{\sqrt{J}}.$$

Here, in particular, $c_0 = \min\{\lambda, 2\lambda^2, 2\lambda^3\}$ and $C_0 = \max\left\{A, \frac{\sqrt{2}A^2}{\lambda}, \frac{A^3}{\lambda^3}\right\}$.

We can now first see how this concludes the proof of the theorem. Note that for any X, Y and $Z_1, \dots, Z_q$ with $Z_i \notin \{X, Y\}$ for all $i \in [q]$, we have

$$\Pr[a_X = a_Y = 0 \wedge (\forall i \in [q] : a_{Z_i} \neq 0)] \geq \Pr[\text{Bal}_{X,Y} \wedge (\forall i \in [q] : a_{Z_i} \neq 0)].$$

Furthermore, with $\text{Bal} = \text{Bal}_{X,Y}$,

$$\begin{aligned} \Pr[\text{Bal} \wedge (\forall i \in [q] : a_{Z_i} \neq 0)] &= \Pr[\text{Bal}] \cdot \Pr[\forall i \in [q] : a_{Z_i} \neq 0 \mid \text{Bal}] \\ &\geq \frac{c_0}{(\ell+1)J} \cdot (1 - \Pr[\exists i \in [q] : a_{Z_i} = 0 \mid \text{Bal}]) \\ &\geq \frac{c_0}{(\ell+1)J} \left(1 - \sum_{i=1}^q \Pr[a_{Z_i} = 0 \mid \text{Bal}]\right) \\ &\geq \frac{c_0}{(\ell+1)J} \left(1 - \frac{qC_0}{\sqrt{J}}\right) \geq \frac{c_0}{2(\ell+1)J}, \end{aligned}$$

if we set $J = q^2 \lceil 4C_0^2 \rceil$, then the theorem follows with $c = \lceil 4C_0^2 \rceil$ and $d = \frac{c_0}{2(4C_0^2+1)}$.

Proof (Lemma B.3). For convenience, we identify the integers between 1 and $\mu := (\ell+1)J$ with the universe $\mathcal{U} = \{0, 1, \dots, \ell\} \times [J]$, and we partition this into blocks $B_i = \{i\} \times [J]$. For $X = (x_1, \dots, x_\ell) \in \{0, 1\}^\ell$ also let

$$S_X = B_0 \cup \bigcup_{i: x_i=1} B_i.$$

Then, $W(S_X) = a_X + 1$, and thus $a_X = 0$ is equivalent to $W(S_X) = 1$.

We now consider three different cases.

Case 1: $X = Y$. Here we define $\text{Bal}_{X,Y} = \{W(S_X) = 1\}$. Then, by Lemma B.1,

$$\Pr[\text{Bal}_{X,X}] = p_{|S_X|}(1) \geq \frac{\lambda}{\sqrt{|S_X|}} \geq \frac{\lambda}{\sqrt{(\ell+1)J}} \geq \frac{c_0}{(\ell+1)J}.$$

Now, fix $Z \neq X$. If $S_Z \setminus S_X \neq \emptyset$, then S_X leaves at least J values in S_Z unset. And thus

$$\Pr[W(S_Z) = 1 \mid \text{Bal}_{X,Y}] \leq \max_u p_{|S_Z \setminus S_X|}(u) \leq p_{|S_Z \setminus S_X|}(0) \leq \frac{\Lambda}{\sqrt{J}} \leq \frac{C_0}{\sqrt{J}}.$$

The first inequality follows from considering the probability that $W(S_Z \setminus S_X) = 1 - W(S_Z \cap S_X)$.

Otherwise, $S_Z \subseteq S_X$. In this case, we have that $|S_Z| \geq J$ and $|S_X \setminus S_Z| \geq J$, and thus by Lemma B.2 with $R = S_Z$ and $S = S_X$,

$$\Pr[W(S_Z) = 1 \mid \text{Bal}_{X,Y}] = \Pr[W(S_Z) = 1 \mid W(S_X) = 1] \leq \frac{\sqrt{2}\Lambda^2}{\lambda\sqrt{J}} \leq \frac{C_0}{\sqrt{J}}.$$

Case 2: $X \neq Y$ and $S_X \subsetneq S_Y$. Define $A = S_X$ and $B = S_Y \setminus S_X$, and define $C = \mathcal{U} \setminus S_Y$. Also, $|A| \geq J$ and $|B| \geq J$. Then, we have $\text{Bal}_{X,Y} = \{W(A) = 1, W(B) = 0\}$, and, by Lemma B.1 and independence,

$$\Pr[\text{Bal}_{X,Y}] = p_{|A|}(1) \cdot p_{|B|}(0) \geq \frac{\lambda^2}{\sqrt{|A| \cdot |B|}} \geq \frac{2\lambda^2}{|A| + |B|} \geq \frac{c_0}{(\ell+1)J}.$$

Now, for any Z , we define $A_Z = S_Z \cap A$, $B_Z = S_Z \cap B$, and $C_Z = S_Z \cap C$. We have four cases: (2.1) $C_Z \neq \emptyset$ (2.2) $C_Z = \emptyset$, $\emptyset \subsetneq B_Z \subsetneq B$ (2.3) $C_Z = \emptyset$, $B_Z \in \{\emptyset, B\}$, $\emptyset \subsetneq A_Z \subsetneq A$, and (2.4) $C_Z = \emptyset$, $B_Z \in \{\emptyset, B\}$, $A_Z \in \{\emptyset, A\}$.

For Case 2.1, we can use Lemma B.1 and the fact that $|C_Z| \geq J$ to upper bound

$$\begin{aligned} \Pr[W(S_Z) = 1 \mid \text{Bal}, (\xi_i)_{i \notin C}] \\ \leq \Pr[W(C_Z) = 1 - W(A_Z) - W(B_Z) \mid \text{Bal}, (\xi_i)_{i \notin C}] \leq \frac{\Lambda}{\sqrt{|C_Z|}} \leq \frac{C_0}{\sqrt{J}}, \end{aligned}$$

where $\text{Bal} = \text{Bal}_{X,Y}$, and we conditioned on all values $(\xi_i)_{i \notin C}$. The statement without the conditioning then follows by averaging.

For Case 2.2, we observe that, after fixing all $(\xi_i)_{i \notin B}$,

$$\Pr[W(S_Z) = 1 \mid \text{Bal}, (\xi_i)_{i \notin B}] \leq \Pr[W(B_Z) = 1 - W(A_Z) \mid \text{Bal}, (\xi_i)_{i \notin B}].$$

and then invoke Lemma B.2 only based on B and $W(B) = 0$, thus upper bounding this as

$$\Pr[W(S_Z) = 1 \mid \text{Bal}, (\xi_i)_{i \notin B}] \leq \frac{\sqrt{2}\Lambda^2}{\lambda\sqrt{J}} \leq \frac{C_0}{\sqrt{J}},$$

where we have used that $|B \setminus B_Z| \geq J$ and $|B_Z| \geq J$ by our assumption that B_Z is strictly between $\emptyset$ and B . The full statement follows by averaging over the $(\xi_i)_{i \notin B}$.

Case 2.3 is very similar to Case 2.2, and yields the same upper bound.

Now, we finally observe that Case 2.4 cannot occur. This is because we cannot have $A_Z = \emptyset$, since $A = S_X$, and some terms are necessarily shared. Therefore, either $S_Z = S_X$, in which case $X = Z$, or $S_Z = S_Y$, in which case $Y = Z$.

Case 3: $S_X \not\subseteq S_Y$ and $S_Y \not\subseteq S_X$. We let $A = S_X \setminus S_Y$, $B = S_Y \setminus S_X$ and $C = S_X \cap S_Y$. Let $m = \min\{|A|, |B|, |C|\}$, $T_m = \{0, 1, \dots, \lfloor \sqrt{m} \rfloor\}$ ($|T_m| = \lfloor \sqrt{m} \rfloor + 1$), and define

$$\text{Bal}_{X,Y} = \bigcup_{\tau \in T_m} \{W(A) = W(B) = \tau, W(C) = 1 - \tau\}.$$

Note that $\text{Bal}_{X,Y}$ implies $W(S_X) = W(S_Y) = 1$. Furthermore, $\tau \in T_m$ implies that $|\tau| \leq \sqrt{m}$ and $|1 - \tau| \leq \sqrt{m}$, and thus, by Lemma B.1, we have

$$\begin{aligned} \Pr[\text{Bal}_{X,Y}] &= \sum_{\tau \in T_m} p_{|A|}(\tau) p_{|B|}(\tau) p_{|C|}(1 - \tau) \\ &\geq |T_m| \frac{\lambda^3}{\sqrt{|A||B||C|}} \geq \frac{\sqrt{m}\lambda^3}{\sqrt{|A||B||C|}}. \end{aligned}$$

Note that because $m = \min\{|A|, |B|, |C|\}$, and for any two distinct $r, s \in \{|A|, |B|, |C|\}$, we have $\sqrt{rs} \leq \frac{r+s}{2} \leq \frac{(\ell+1)J}{2}$, we immediately get

$$\Pr[\text{Bal}_{X,Y}] \geq \frac{2\lambda^3}{(\ell+1)J} \geq \frac{c_0}{(\ell+1)J}.$$

We now introduce $Z \notin \{X, Y\}$. With $D = \mathcal{U} \setminus (A \cup B \cup C)$, we denote $A_Z = S_Z \cap A$, $B_Z = S_Z \cap B$, $C_Z = S_Z \cap C$, and $D_Z = S_Z \cap D$.

The case $D_Z \neq \emptyset$ leads to a bound of $\Pr[W(S_Z) = 1 \mid \text{Bal}_{X,Y}] \leq \frac{A}{\sqrt{|D_Z|}} \leq \frac{C_0}{\sqrt{J}}$ similar to Case 2.1 above via Lemma B.1. Hence, let us assume that $D_Z = \emptyset$. Here, similar to Cases 2.2 and 2.3 we can use Lemma B.2 to prove a bound for the case where there exists $R \in \{A, B, C\}$ such that $\emptyset \subsetneq (S_Z \cap R) \subsetneq R$, and get $\Pr[W(S_Z) = 1 \mid \text{Bal}_{X,Y}] \leq \frac{\sqrt{2}\Lambda^2}{\lambda\sqrt{J}} \leq \frac{C_0}{\sqrt{J}}$.

Therefore, we are left in the situation where $D_Z = \emptyset$, $A_Z \in \{\emptyset, A\}$, $B_Z \in \{\emptyset, B\}$, and $C_Z \in \{\emptyset, C\}$. Note that $C_Z = C$, since S_X, S_Y , and S_Z all contain B_0 . It can also not be that $S_Z = A \cup C = S_X$ and $S_Z = B \cup C = S_Y$. Therefore, S_Z is either C or $A \cup B \cup C$. Conditioned on $\text{Bal}_{X,Y}$, we know that for some $\tau \in T_m$, we have $W(C) = 1 - \tau$ and $W(A) = W(B) = \tau$; this means that $W(S_Z) \in \{1 - \tau, 1 + \tau\}$. For $W(S_Z) = 1$ to additionally happen, we need thus $\tau = 0$. However, by Lemma B.1,

$$\begin{aligned} \Pr[W(A) = W(B) = 0, W(C) = 1 \mid \text{Bal}_{X,Y}] &= \frac{p_{|A|}(0)p_{|B|}(0)p_{|C|}(1)}{\Pr[\text{Bal}_{X,Y}]} \\ &\leq \frac{\Lambda^3}{\sqrt{|A||B||C|}} \cdot \frac{\sqrt{|A||B||C|}}{\sqrt{m}\lambda^3} \leq \frac{C_0}{\sqrt{m}}, \end{aligned}$$

and the claim follows because $m \geq J$. □

C Proof of Lemma 3.21 (Three-Fork Rewinding)

The proof simply follows from the two-index forking lemma of [CART26, Lemma 2.19, ePrint version] and ruling out an additional collision event that $\mathbf{h}'_I = \mathbf{h}''_I$.

For completeness, we recall their forking algorithm Fork' and their lemma below, with the algorithm $\mathcal{A}'(\text{inp}, \mathbf{h}'; \rho_{\mathcal{A}'})$ outputting two indices (I_0, I_1) that are *not necessarily* distinct. The forking algorithm also forks on two indices, once for each.

Algorithm $\text{Fork}'^{\mathcal{A}'}(\text{inp})$: <pre> 1 : Pick random coins $\rho_{\mathcal{A}'}$; $\mathbf{h} \leftarrow \mathcal{C}^Q$ 2 : $((I_0, I_1), (\text{aux}_{I_k})_{k \in \{0,1\}}) \leftarrow \mathcal{A}'(\text{inp}, \mathbf{h}; \rho_{\mathcal{A}'})$ 3 : if $(I_0, I_1) = \perp$ then return $\perp$ 4 : $\mathbf{h}'_{[I_0:Q]} \leftarrow \mathcal{C}^{Q-I_0+1}$; $\mathbf{h}' = \mathbf{h}_{[1:I_0-1]} \ \mathbf{h}'_{[I_0:Q]}$ 5 : $((I'_0, I'_1), (\text{aux}'_{I'_k})_{k \in \{0,1\}}) \leftarrow \mathcal{A}'(\text{inp}, \mathbf{h}'; \rho_{\mathcal{A}'})$ 6 : $\mathbf{h}''_{[I_1:Q]} \leftarrow \mathcal{C}^{Q-I_1+1}$; $\mathbf{h}'' = \mathbf{h}_{[1:I_1-1]} \ \mathbf{h}''_{[I_1:Q]}$ 7 : $((I''_0, I''_1), (\text{aux}''_{I''_k})_{k \in \{0,1\}}) \leftarrow \mathcal{A}'(\text{inp}, \mathbf{h}''; \rho_{\mathcal{A}'})$ 8 : if $I_0 \notin \{I'_0, I'_1\} \vee I_1 \notin \{I''_0, I''_1\}$ then return $\perp$ 9 : $h_{I_0} = h'_{I_0} \vee h_{I_1} = h''_{I_1}$ then return $\perp$ 10 : return $(1, (\text{aux}_{I_k})_{k \in \{0,1\}}, (\text{aux}'_{I'_0}, \text{aux}''_{I''_1}))$ </pre>

Lemma C.1 ([CART26, Lemma 2.19]). Fix $Q \geq 1$, $\mathcal{C}$, and IG as in Lemma 3.21. Let $\mathcal{A}'$ be a randomized algorithm that on input $\text{inp}, \mathbf{h} \in \mathcal{C}^Q$ returns a pair $(I = (I_0, I_1) \in [Q]^2 \cup \{\perp\}, (\text{aux}_{I_k})_{k \in \{0,1\}})$. Let acc' also be defined as the probability that $\mathcal{A}'$ outputs $I \neq \perp$. Let $\text{frk}' = \Pr[b = 1 : \text{inp} \leftarrow \text{IG}; (b, \text{aux}, \text{aux}') \leftarrow \text{Fork}'^{\mathcal{A}'}(\text{inp})]$ where $\text{Fork}'^{\mathcal{A}'}$ is defined above. Then,

$$\text{frk}' \geq \frac{\text{acc}'}{2} \left(\frac{\text{acc}'}{4Q} - \frac{1}{|\mathcal{C}|} \right)^2, \text{ or alternatively, } \text{acc}' \leq \sqrt[3]{32Q^2 \cdot \text{frk}'} + \frac{4Q}{|\mathcal{C}|}.$$

One can use the above lemma to prove our results by constructing a wrapper adversary $\mathcal{A}'$ with the syntax $(I_0, I_1, \text{aux}) \leftarrow \mathcal{A}'(\text{inp}, \mathbf{h}; \rho_{\mathcal{A}'})$ that does the following:

- Run $(I, \text{aux}) \leftarrow \mathcal{A}(\text{inp}, \mathbf{h}; \rho_{\mathcal{A}'})$
- Return (I, I, aux)

Observe that $\text{acc} = \text{acc}'$.

Then, we have that

$$\begin{aligned}
\text{frk}_3 &= \Pr[b = 1 : \text{inp} \leftarrow \text{IG}; (b, \text{aux}, \text{aux}', \text{aux}'') \leftarrow \text{Fork}_3^{\mathcal{A}}(\text{inp})] \\
&= \Pr[b' = 1 \wedge h'_{I_0} \neq h''_{I_1} : \text{inp} \leftarrow \text{IG}; (b', \text{aux}, \text{aux}', \text{aux}'') \leftarrow \text{Fork}'^{\mathcal{A}}(\text{inp})] \\
&\geq \Pr[b' = 1 : \text{inp} \leftarrow \text{IG}; (b', \text{aux}, \text{aux}', \text{aux}'') \leftarrow \text{Fork}'^{\mathcal{A}}(\text{inp})] \\
&\quad - \Pr[h'_{I_0} = h''_{I_1} : \text{inp} \leftarrow \text{IG}; (b', \text{aux}, \text{aux}', \text{aux}'') \leftarrow \text{Fork}'^{\mathcal{A}}(\text{inp})] \\
&\geq \text{frk}' - \frac{1}{|\mathcal{C}|} \\
&\geq \frac{\text{acc}'}{2} \left(\frac{\text{acc}'}{4Q} - \frac{1}{|\mathcal{C}|} \right)^2 - \frac{1}{|\mathcal{C}|}.
\end{aligned}$$

Here, $\mathbf{h}', \mathbf{h}''$ are defined within Fork' . The second equality follows from the fact that $\mathcal{A}'$ outputs the same indices $I_0 = I_1$ which is the output of $\mathcal{A}$. The third inequality follows from $\Pr[A|B] \geq \Pr[A] - \Pr[B]$ for any events A, B . The fourth inequality follows from the definition of frk' and a simple bound on the collision probability for the uniform distribution on $\mathcal{C}$. The last inequality follows from the above lemma. $\square$