Blind Signatures from Arguments of Inequality

Michael Klooß[1] and Russell W. F. Lai[2] and Michael Reichle[3]

[1] Karlsruhe Institute of Technology, Karlsruhe, Germany
KASTEL Security Research Labs, Karlsruhe, Germany
klooss@mail.informatik.kit.edu
[2] Aalto University, Espoo, Finland
russell.lai@aalto.fi
[3] Department of Computer Science, ETH Zurich, Zurich, Switzerland
michael.reichle@inf.ethz.ch

Abstract. Blind signatures are an important tool for privacy-preserving applications with a long history dating back to Chaum's seminal work in Crypto'82. In this work, we present the first pairing-free blind signature in the random oracle model that is concurrently-secure under the discrete logarithm assumption (without the algebraic group model).
On a technical level, our work extends the recent *proofs of inequality* technique (Klooß and Reichle, Crypto'25) to *arguments* of inequality. The original technique relies on puncturing the verification key to make forgeries *statistically* impossible. We show how *computational* puncturing can be used, which opens the possibilities of applying the technique under weaker assumptions, at the cost of requiring a *rewinding-based* security reduction. In particular, this demonstrates that (notoriously difficult) rewinding-based techniques are still a viable path to Fiat–Shamir-based blind signatures.

1 Introduction

Blind signatures were introduced by David Chaum in 1982 [15] and allow a signer to interactively issue a signature σ on some message μ to a user. The distinctive feature, coined *blindness*, is that when the user presents the signature-message pair (σ, m) later, the signer cannot link the pair to the signing session in which σ was issued. At the same time, it must remain hard to forge signatures, formalized by the notion of *one-more unforgeability*: even if ℓ signing sessions are completed in a concurrent manner, it remains difficult to output $\ell+1$ valid signatures for distinct messages. These security properties give rise to many applications in privacy-preserving technologies, such as e-cash [15, 17, 44], anonymous credentials [10, 12], e-voting [16, 27], or privacy-preserving authentication [20, 32].

Blind Signatures in Pairing-free Groups Blind signatures in pairing-free groups are mostly constructed via the Fiat–Shamir paradigm, following the template of Blind Schnorr signatures [46, 48]. It is based on Σ-protocols that are compiled via Fiat–Shamir into a signature scheme [23]. Here, the verification

key is some statement $\mathbb{x} \in \mathcal{L}_{\mathsf{R}}$, where R is a hard relation and $\mathcal{L}_{\mathsf{R}}$ its induced language.[4] For interactive signing, the *user* chooses the challenge, i.e., the protocol proceeds in three moves: (1) The signer sends the first flow a^* of the Σ-protocol for a statement $\mathbb{x} \in \mathcal{L}_{\mathsf{R}}$. (2) The user derives a randomized a from a^*. Then it hashes a and message μ to compute a challenge $\gamma = \mathsf{H}(\mathbb{x}, \mu, a)$ (following the Fiat–Shamir transformation). The challenge γ is *blinded* as γ^* and then sent. (3) The signer outputs the Σ-protocol response z^* for challenge γ^*, and the user *unblinds* z. Finally, the Σ-protocol transcript $\sigma = (a, \gamma, z)$ forms the signature.[5] The randomization steps for blindness usually require that the witness relation R is a linear, e.g. a discrete logarithm (DL) relation. We note that there are other ways to construct blind signatures in pairing-free groups [24, 35], but these generally require non-black-box use of the group. Here, we focus on constructions that employ the group in a black-box manner.

Proving unforgeability The core challenge to instantiate the Fiat–Shamir paradigm for blind signatures is proving one-more unforgeability. In ordinary signatures, the challenge γ is derived from H as above, and unforgeability follows from a rewinding-based argument [46]. However, recall that in the context of blind signatures, the transcript is randomized by the user which includes the challenge γ. The latter is typically randomized to a uniform challenge γ' over the challenge space $\mathcal{C}$. The signer only sees the randomized challenge γ'. In the unforgeability game, the choice of γ' is now in the control of adversary $\mathcal{A}$ which means that signing must be simulated *without* programming H. This is not merely an issue related to provable security: the freedom over the choice of γ' can lead to attacks when the adversary is allowed to open many *concurrent* signing sessions at once [6, 21, 36, 50]. Therefore, many schemes only satisfy a weaker unforgeability notion: at most polylog-many sessions can be opened (concurrently) by the adversary [1, 3, 31, 46].

Schemes with concurrent security: On the positive side, there is much progress in recent years on instantiating the Fiat–Shamir paradigm securely (*i.e.*, for more than polylog concurrent sessions). For instance, there are boosting techniques [13, 29, 38, 45], allowing for more concurrent sessions. However, the total number of sessions is still limited (and fixed a priori). Also, the communication and computational complexity grow logarithmically and linearly, respectively, in the number of signing sessions in the state-of-the-art boosting transform [13].

For *unlimited* concurrent security, there are schemes [1, 19, 26, 33, 49] proven secure under standard assumptions (*e.g.*, the discrete logarithm (DL) assumption) but under the additional restriction that the adversary behaves in an *algebraic manner* (as formalized in the algebraic group model [25]). This approach also extends to blind signatures from group actions [28], where security is proven in the algebraic group action model [22].

[4] Hard relations allow sampling statements so that finding a witness is hard, e.g. DL.

[5] Note that often the first flow a may be omitted from the final signature σ. This is possible if a can be derived from (γ, z) which is true for most canonical Σ-protocols.

Until recently, a secure instantiation of the Fiat–Shamir paradigm *without* algebraic models remained elusive: [14] give the first secure instantiation, proven secure under the CDH assumption in the random oracle model alone. Since then, there have been several works building on these techniques: [39, 40] improve efficiency and [11] achieves tight security, under DDH. Recently, a more efficient instantiation is also given by [47] under CDH.

Open Problems A central goal of cryptography is to rely on weak assumptions, e.g. the DL assumption, to build advanced primitives. Indeed, there have been several attempts to construct blind signatures from DL alone [3, 13, 38, 45, 46], yet all of these fall short of the goal of unlimited concurrency. To date, all blind signatures over pairing-free groups either rely on stronger assumptions [11, 14, 39, 40, 47] (namely, CDH[6] or DDH) or the algebraic group model (AGM) [1, 19, 33, 49]. Blind signatures from the DL assumption alone remains arguably one of the most elusive and challenging goals in the realm of pairing-free blind signatures.

1.1 Our contributions

We present the first blind signature from DL in pairing-free groups in the random oracle model with unlimited concurrent security. We compare the blind signature scheme's communication and signature size to the state of the art in Table 1. Our DL-based scheme is quite competitive, despite — or perhaps because — relying only on the DL assumption.

On a technical level, we instantiate the proof of inequality paradigm from [39] in the computational regime, hence *arguments* of inequality. That is, the strategy of [39] relies on statistically binding commitments, instantiated via ElGamal encryption. Instead, we observe that the paradigm is quite generic and instantiate it via computationally binding commitments, namely Pedersen commitments. As Pedersen commitments are binding under the DL assumption, this enables us to prove security under DL with a *rewinding-based* proof technique.

Our core idea — designing a puncturable protocol so that any "fork" yields a useful witness — may be applicable in other contexts, where rewinding has been avoided due to the technical difficulties it introduces.

2 Technical overview

Our starting point is the technique from [39] based on proofs of inequality. It follows the Fiat–Shamir template. The idea of [39] is to use a Σ-protocol that allows to prove that two ciphertexts encrypt distinct messages, based on ElGamal encryption. We recall the structure below:

[6] There exist groups in which CDH and DL are equivalent [7, 8, 9, 42, 43]. However, the security loss of the DL-to-CDH reduction is high. Hence even there, direct DL-based security is of interest.

Table 1: Pairing-free blind signature schemes with unlimited concurrency.

Scheme	Moves	Signature	Communication	Assumptions; Remark
Clause Schnorr [26]	3	$1\mathbb{G} + 1\mathbb{Z}_p$	$2\mathbb{G} + 3\mathbb{Z}_p$	OMDL, mROS
Abe [1, 33]	3	$2\mathbb{G} + 6\mathbb{Z}_p$	$\lambda + 3\mathbb{G} + 6\mathbb{Z}_p$	DL
TZ [49]	3	$4\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL
Snowblind [19]	3	$1\mathbb{G} + 2\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	DL
CTZ-1 [14]	4	$1\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH; OMUF-1
CTZ-2 [14]	5	$1\mathbb{G} + 4\mathbb{Z}_p$	$5\mathbb{G} + 5\mathbb{Z}_p$	CT-OMCDH; OMUF-1
CTZ-3 [14]	4	$\Theta(\lambda) \cdot (\lambda + \mathbb{G} + \mathbb{Z}_p)$	$\Theta(\lambda) \cdot (\lambda + \mathbb{G} + \mathbb{Z}_p)$	CDH
KRW [40]	4	$2\mathbb{G} + 5\mathbb{Z}_p$	$\Theta(\lambda) \cdot (\lambda + \mathbb{G} + \mathbb{Z}_p)$	DDH
BHKR [11]	4	$10\mathbb{G} + 29\mathbb{Z}_p$	$37\mathbb{G} + 40\mathbb{Z}_p$	DDH
KR-1 [39]	4	$1\mathbb{G} + 5\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH
KR-2 [39]	5	$1\mathbb{G} + 6\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	DDH
RR [47]	4	$2\mathbb{G} + 8\mathbb{Z}_p$	$\Theta(\lambda) \cdot (\lambda + \mathbb{G} + \mathbb{Z}_p)$	CDH
Our work	4	$6\mathbb{Z}_p$	$\Theta(\lambda) \cdot (\lambda + \mathbb{G} + \mathbb{Z}_p)$	DL

The schemes above the line are proven to be secure in the ROM and the AGM, while those below the line only rely on the ROM. The remark "OMUF-1" means that the scheme satisfies a weaker one-more unforgeability notion.

- *Public Parameters.* The public parameters crs contain the public key pk and a random ciphertext $\boldsymbol{c}$. Multiple random oracles are used, in particular $\mathsf{H}_\mu \colon \{0,1\}^* \to \mathbb{G}$ for the message and H_{ch} for Fiat–Shamir.
- *Verification key.* The verification key vk is an element in a hard subset membership language (namely, DDH tuples $\mathbf{D}$). The secret key is the witness (namely, a discrete logarithm d_1 of the tuple $\mathbf{D}$).
- *Signing.* A signature for (hashed) message $\mu = \mathsf{H}_\mu(\mathsf{msg})$ is a Fiat–Shamir-compiled OR-proof π for the statement:
  - vk is in the hard subset (and one knows the membership witness); **or**
  - the (public) ciphertext $\boldsymbol{c}$ and a given ciphertext $\boldsymbol{c}_\mu$ (which in the protocol will encrypt μ) contain distinct messages. This is proven by showing that $\boldsymbol{c} - \boldsymbol{c}_\mu$ does *not* encrypt 0.

  The honest signer uses the membership witness d_1 for vk in the first branch and simulates the second branch. The user leverages homomorphic properties to transform the transcript for the second branch into a proof for a ciphertext with randomness 0.
- *Verification.* Construct $\boldsymbol{c}_\mu$ as an encryption of μ with randomness 0, and verify the proof π.

Looking ahead, we implement the OR-proof following [3, 18]. This construction provides the security reduction an *all-but-one* trapdoor: If $\boldsymbol{c}$ is an encryption of μ^*, then the reduction can sign every message (hash) $\mu \neq \mu^*$ using the inequality branch. Following the proof technique from [37, 40], the adversary is first forced to provide a forgery for message μ^* embedded in the random oracle H_μ, via a guessing argument. After a few game hops, both statements of the OR-proof are false for message μ^* (that is, $\mathbf{D}$ is not a valid DDH-tuple and μ^* is encrypted in

$\boldsymbol{c}$). Therefore, it is *impossible* for the adversary to succeed (except with negligible probability due to the soundness error of π).

Our DL-based instantiation The above template is quite generic, and the construction does not rely on $\boldsymbol{c}$ being a ciphertext. Thus, it is natural to explore the setting where $\boldsymbol{c}$ is a computationally binding commitment, instead of a perfectly binding ciphertext. This would enable an instantiation with Pedersen commitments. As Pedersen commitments are secure under the DL assumption alone, this allows us to avoid DDH in our security analysis. Below, we elaborate on the necessary changes in construction and proof strategy.

Our approach: Recall that our goal is to obtain a blind signature with security based on the DL assumption. Following the above approach, we replace the ElGamal ciphertext $\boldsymbol{c}$ in the public parameters (resp. $\boldsymbol{c}_\mu$ in the signature) with Pedersen commitments $C = \mu^* G + r^* H$ (resp. $C_\mu = \mu G + rH$), where $G, H \in \mathbb{G}$ and $r, r^* \in \mathbb{Z}_p$, using additive notation for the group $\mathbb{G}$. Furthermore, we replace the DDH-language with a proof of knowledge for discrete logarithms of some element $X = xG$, as in [14, 47]. Since Pedersen is binding under DL, this intuitively yields a blind signature under DL. To complete this idea, we must resolve two challenges:

(1) *Blindness:* We need a blinding mechanism for the Pedersen-based inequality argument. For ElGamal encryption, the decryption key served as a witness for the reduction. Importantly, that witness was *independent* of the ciphertext $\boldsymbol{c}$. For Pedersen commitments, no such witness exists and this not only affects the security reduction, but it also requires a non-trivial blinding procedure.
(2) *Soundness:* Unlike for ElGamal commitments in [39], in our setting the language of inequality of messages in Pedersen openings is trivial. An unbounded adversary can find an opening $C = \mu G + rH$ such that $C - C_\mu$ does not open to 0. Hence, we speak of *arguments* and our security proof must rely on an *efficient* way to extract a DL witness. For this, in the final step of the reduction, we rely on rewinding to extract a DL. This is in contrast to prior works, where either the statement itself contains a solution to a hard problem (*e.g.*, for CDH [14, 40, 47]) or it is statistically hard to compute a forgery, even for an unbounded adversary [11, 39]. While some prior works have employed rewinding-based arguments to extract a solution to an underlying DL problem, this step is very delicate. Indeed, several prior works contain errors in exactly this final step [1, 3, 4].

Let us now outline our solutions.

From ElGamal to Pedersen: First, we must design an appropriate Σ-protocol for the inequality argument. For ElGamal, the Σ-protocol in [39] roughly proves in zero-knowledge the correct decryption of $\boldsymbol{c}$ with the secret key x, scaled by a random factor $y \leftarrow \mathbb{Z}_p^\times$. That is, $y \cdot (1, -x) \cdot \boldsymbol{c}_\mu = y \cdot \mu$. Thus, in the protocol, this OR-branch proves that (1) the witness is a scaled secret key $(y, -xy)$ and (2) the decryption yields $\mu_\$ = y \cdot \mu \neq 0$, where $\mu \in \mathbb{Z}_p^\times$ is the non-zero message

encrypted in $\boldsymbol{c}$. As $\mu_\$$ is uniform over $\mathbb{Z}_p^\times$ (for uniform $y \leftarrow \mathbb{Z}_p^\times$), it can safely be revealed as part of the statement. This implements a non-zero proof for encrypted messages, which establishes the inequality of the encrypted messages as desired. The protocol is blindable by randomizing y and removing the user's randomness from the ciphertext $\boldsymbol{c}_\mu$, leveraging (1) linearity of the Σ-protocol and (2) the secret key x.

We now turn to our construction where we use Pedersen commitments. It is not hard to see that the "proof of non-zero committed message" from ElGamal ciphertext can be adapted to Pedersen commitments. Given a commitment $C = \mu G + rH$, the "decryption" equation is $\mu G = C - rH = (1, -r) \cdot \binom{C}{H}$. To randomize the message (and witness $(1, -r)$), the user chooses a random unit $y \leftarrow \mathbb{Z}_p^\times$ and sets $\mu_\$ G = y \cdot \mu G = (y, -yr) \cdot \binom{C}{H}$. A crucial difference is that there is no fixed decryption key and thus no fixed witness for this Σ-protocol. Instead, the witness depends on the commitment randomness r (which the reduction will obtain along with μ through a NIZK proof from the user). This lack of fixed witness also shows in blinding: for proving (non-zero) ElGamal encryption, blinding is rather straightforward; for Pedersen commitments, blinding becomes a little more complex. We refer to our protocol for details.

Proving OMUF: We turn towards one-more unforgeability (OMUF). Let us first discuss the proof from [39] adapted to our setting and see where it fails in the computational regime. Recall that in the OMUF game $\mathsf{G}_{\mathsf{omuf}}$, the adversary $\mathcal{A}$ (in the role of the user) obtains a verification key vk and access to oracle $\mathcal{O}_{\mathsf{Sign}}$ for the signing protocol. The adversary's goal is to provide $\ell + 1$ forgeries $(\widehat{\sigma}_i, \widehat{\mathsf{msg}}_i)$ with distinct messages $\widehat{\mathsf{msg}}_i$ such that only ℓ signing sessions via $\mathcal{O}_{\mathsf{Sign}}$ were finished. Roughly, [39] modify the OMUF game $\mathsf{G}_{\mathsf{omuf}}$ via indistinguishable game hops such that $\mathcal{A}$'s success in $\mathsf{G}_{\mathsf{omuf}}$ translates to success in the modified game G', and the following properties hold.

(1) *Forcing μ^*:* The game G' samples μ^* in advance and embeds μ^* into H_μ later. Importantly, it forces the adversary to (i) provide a forgery for μ^*, *i.e.*, $\mu^* = \mathsf{H}_\mu(\widehat{\mathsf{msg}}_i)$ for some $i \in [\ell + 1]$, and (ii) the hash μ^* is never signed in $\mathcal{O}_{\mathsf{Sign}}$. Otherwise, the adversary loses G'.

(2) *Puncturing at μ^*:* The game G' sets up the commitment C in the public parameters as a commitment to μ^*.

(3) *Signing for $\mu \neq \mu^*$:* The game G' implements the signing oracle $\mathcal{O}_{\mathsf{Sign}}$ by using the inequality-branch of the OR-proof while the DL-branch is simulated. To provide the required witness, the user's first protocol message includes a straightline extractable proof of knowledge of opening π_μ for C_μ. When the adversary $\mathcal{A}$ starts a session, G' extracts from π_μ a witness for $C - C_\mu = (\mu^* - \mu)G + (r^* - r)H$ and can prove $\mu^* - \mu \neq 0$; if $\mu = \mu^*$ is extracted, G' responds randomly (as $\mathcal{A}$ must not finish this session by Item (1)).

(4) *Forgery and DL extraction:* By Item (1), a successful adversary in G' provides a forgery for $\mu^* = \mathsf{H}_\mu(\widehat{\mathsf{msg}}_i)$. By Item (2), the game *cannot derive* a witness for this statement: $C - C_\mu = 0G + (r^* - r)H$ is a commitment to zero. This will make it possible to extract an embedded DL through a rewinding-based reduction we describe below.

We note that the above setting in G' is analogous to [39], except for the last step: In [39], the puncturing was statistical and the only way for the adversary to win is to break the soundness of the Fiat–Shamir transformed proof system (which is statistically hard in the ROM). In more detail, the OR-branch ciphertext $\boldsymbol{c} = \boldsymbol{c} - \boldsymbol{c}_\mu$ is an ElGamal encryption of zero in [39], hence the statement is false, and the verification key vk (*i.e.*, the DDH-branch) was switched to a non-DDH tuple. Hence the statement has no witness and statistical soundness applies.

In our setting, both the DL-branch and the inequality-branch do have a witness.[7] However, we can set things up in such a way, that the game/reduction does not know these witnesses. The intuition is then: if, by rewinding, we find two accepting transcripts for a fixed statement, we can appeal to special soundness and extract a witness for at least one branch. Thus, if we can embed the DL challenge in both branches simultaneously, we succeed in computing the challenge DL. This is exactly what the reduction will do, to be elaborated below.

Given a DL challenge $Y = yG$, the reduction simulates game G' and embeds Y into the public parameters and verification key. Upon success of $\mathcal{A}$ (in G'), the reduction rewinds $\mathcal{A}$ prior to the Fiat–Shamir challenge query (associated to the forgery for $\mu^* = \mathsf{H}_\mu(\widehat{\mathsf{msg}}_i)$) and reprograms the random oracle H_{ch} of the Fiat–Shamir proof. By the forking lemma, we obtain a second accepting transcript for the same fixed statement (vk, C_{μ^*}) with reasonable probability. It remains to explain the embedding of the DL challenge Y.

In its setup, the reduction sets the verification key to $\mathsf{vk} = X = Y$ and it sets the Pedersen commitment key $H = sY$ for $s \leftarrow \mathbb{Z}_p$. Observe that the distribution of vk and H is unchanged. Also observe that game G' never made use of the DL of vk or H. (In particular, the reduction can still implement $\mathcal{O}_{\mathsf{Sign}.}$) Thus, the simulation of G' is perfect.

Finally, let us look at the two accepting transcripts with statements (vk, C_{μ^*}). If the challenge in the DL-branch differs, then clearly we obtain y by special soundness. The other case, that the challenge in inequality-branch differs, needs closer inspection. In this case, the claim is that the Pedersen commitment $C - C_{\mu^*}$ is not a commitment to 0. We make use of following facts:

- By definition of our signature scheme, $C_{\mu^*} = \mu^* G$ (*i.e.*, the commitment in the signature is derandomized[8]).
- The reduction knows $C - C_{\mu^*} = r^* H$ (where by construction $C = \mu^* G + r^* H$).
- By special soundness, we obtain a non-zero opening $C - C_{\mu^*} = mG + rH$ (*i.e.*, $m \neq 0$).

Putting things together, the reduction knows

$$r^* H = C - C_{\mu^*} = mG + rH$$

and since $m \neq 0$, $G = \frac{(r^* - r)}{m} H$. Thus, the reduction also succeeds in the inequality/non-zero branch.

[7] Although an unbounded adversary can compute the DLs and equivocate Pedersen commitments, no signature scheme can be secure against unbounded adversaries anyway.

[8] The argument would go through given the opening randomness r to C_{μ^*} in the signature. But ensuring $r = 0$ is possible, and it reduces signature size.

Comparison to rewinding in prior works: It is well-known that rewind-based arguments in concurrent settings are subtle and very intricate [30, 34]. Thus, it is natural to wonder where our approach differs. The difference between our approach and prior works is captured by the fact that upon successful rewinding, our reduction *always* succeeds, *i.e.*, it will extract a DL from *either branch*. The difficulties in prior works arose, because their reductions know the witness in *one* branch, but not the other (where the challenge is embedded). As a consequence, they must argue that, with high probability, they can obtain related transcripts for the *other* branch. For this, a first idea is to reduce to witness indistinguishability, but that is highly non-trivial, as the witness indistinguishability experiment itself cannot be rewound. To reiterate: Due to puncturing of the statements on μ^* and forcing the adversary's forgery to be on μ^*, we need not appeal to any kind of indistinguishability of branches. The DL challenge is embedded in both branches, and hence any pair of related transcripts allows us to extract it.

3 Preliminaries

When applicable, a family of objects, in particular algorithms, is uniform unless otherwise specified. Throughout, $\lambda \in \mathbb{N}$ denotes the security parameter. We write log for the base 2 logarithm. We write $y \leftarrow A(x)$ to run (probabilistic) algorithm A with fresh randomness on input x; we write $A \rightleftarrows B$ for interactive protocols; and we write $y \leftarrow S$ to sample y uniformly from a set S. We may also write $\mathcal{U}_S$ for the uniform distribution over S. For two distributions D_1 and D_2, we write $D_1 \sim D_2$ if D_1 and D_2 are identically distributed. Vectors are columns. Finally, write $y \coloneqq x$ for algorithmic assignment and $\mathsf{H}(x) \coloneqq y$ to program a random oracle at query x to output y. Throughout the paper, we assume that algorithms check their inputs are in the right space (*e.g.*, encode a group element), and return $\perp$ otherwise. In security proofs, we say that a game *aborts its entire execution* if the game aborts its interaction with the adversary and outputs 0. We write **req** C (resp. **parse** $y \leftarrow x$) to denote that an algorithm outputs $\perp$ if condition C is false (resp. parsing x as y fails).

Forking Lemma We recall the general forking lemma by Bellare and Neven [5].

Lemma 3.1 (General Forking Lemma [5, Lemma 1]). *Fix an integer $q \geq 1$ and a set $\mathcal{C}$ of size $h \geq 2$. Let $\mathcal{A}$ be a randomized algorithm that on input $\mathsf{par}, \boldsymbol{h} \coloneqq (h_1, \ldots, h_q)$ returns $I \in [0, \ldots, q]$ and an arbitrary string Δ. Let IG be a randomized algorithm called the input generator. The accepting probability acc of $\mathcal{A}$ is defined as*

$$\mathsf{acc} = \Pr\left[\mathsf{par} \leftarrow \mathsf{IG},\ \boldsymbol{h} \leftarrow \mathcal{C}^q,\ (I, \Delta) \leftarrow \mathcal{A}(\mathsf{par}, \boldsymbol{h}) : I \geq 1\right].$$

The forking algorithm $\mathsf{Fork}_{\mathcal{A}}$ associated to $\mathcal{A}$ is the randomized algorithm that takes input par and proceeds as in Fig. 1. Let

$$\mathsf{frk} = \Pr\left[\mathsf{par} \leftarrow \mathsf{IG};\ (b, (\Delta_1, \Delta_2)) \leftarrow \mathsf{Fork}_{\mathcal{A}}(\mathsf{par}) \ :\ b = 1\right].$$

Then, it holds that $\mathsf{acc} \leq q/h + \sqrt{q \cdot \mathsf{frk}}$.

Algorithm $\mathsf{Fork}_{\mathcal{A}}(\mathsf{par})$

$\rho \leftarrow \{0,1\}^{\ell_{\mathcal{A}}}$ // $\ell_{\mathcal{A}}$-bit randomness used by $\mathcal{A}$
$\boldsymbol{h} \coloneqq (h_1, \cdots, h_q) \leftarrow \mathcal{C}^q$
$(I, \Delta) \coloneqq \mathcal{A}(\mathsf{par}, \boldsymbol{h}; \rho)$
if $I = 0$ **then**
  return $(0, (\bot, \bot))$
$(h'_I, \cdots, h'_q) \leftarrow \mathcal{C}^{q-I+1}$
$\boldsymbol{h}' \coloneqq (h_1, \cdots, h_{I-1}, h'_I, \cdots, h'_q)$
$(I', \Delta') \coloneqq \mathcal{A}(\mathsf{par}, \boldsymbol{h}'; \rho)$
if $I = I' \ \wedge \ h_I \neq h'_I$ **then**
  return $(1, (\Delta, \Delta'))$
else
  return $(0, (\bot, \bot))$

Fig. 1: Description of the forking algorithm $\mathsf{Fork}_{\mathcal{A}}$.

Prime-order Groups Let GenGrp be an algorithm that on input 1^λ outputs a group description $\mathbb{G}$, the prime-order order p of $\mathbb{G}$, and a generator G of $\mathbb{G}$. We employ additive notation for $\mathbb{G}$ and lower-case (resp. upper-case) letters for $\mathbb{Z}_p$ elements (resp. group elements). Bold letters denote vectors. Throughout, we assume that $(\mathbb{G}, p, G)$ is setup via $(\mathbb{G}, p, G) \leftarrow \mathsf{GenGrp}(1^\lambda)$ and is passed to other algorithms as input. We leave this implicit.

Assumptions We rely on the DL assumption over prime-order groups.

Definition 3.2 (DL). *The advantage of an adversary $\mathcal{A}$ against the DL problem is defined as*

$$\mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{A}}(\lambda) = \Pr[x'G = X \mid x \leftarrow \mathbb{Z}_p, X = xG, x' \leftarrow \mathcal{A}(1^\lambda, X)].$$

The discrete logarithm (DL) assumption for $(\mathbb{G}, G)$ holds, if for every PPT adversary $\mathcal{A}$, the advantage $\mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{A}}$ is negligible.

Four-move Partially Blind Signatures We define 4-move blind signatures [15]. For brevity, we directly define their extension to partial blindness [2]. We follow closely the definitions from [40].

Definition 3.3 (Partially Blind Signature Scheme). *A 4-move partially blind signature scheme with message space $\mathcal{M}$ and common message space $\mathcal{I}$ is a tuple of PPT algorithms $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{BSign}, \mathsf{BUser}, \mathsf{Verify})$ with the following syntax:*

– $\mathsf{KeyGen}(1^\lambda)$: *outputs a pair of keys* $(\mathsf{vk}, \mathsf{sk})$. *We assume* vk *can be efficiently computed from* sk.
– $\mathsf{BSign}(\mathsf{sk}, \tau) \rightleftarrows \mathsf{BUser}(\mathsf{vk}, m, \tau)$*: The signer protocol* $\mathsf{BSign} = (\mathsf{BSign}_1, \mathsf{BSign}_2)$ *takes as input a secret key* sk *and common message* $\tau \in \mathcal{I}$. *The user protocol* $\mathsf{BUser} = (\mathsf{BUser}_1, \mathsf{BUser}_2, \mathsf{BUser}_3)$ *takes as input a key* vk*, a message* $m \in \mathcal{M}$ *and common message* $\tau \in \mathcal{I}$. *After the execution,* BUser *returns a signature* σ *and we write* $\sigma \leftarrow \langle \mathsf{BSign}(\mathsf{sk}, \tau), \mathsf{BUser}(\mathsf{vk}, m, \tau) \rangle$.
– $\mathsf{Verify}(\mathsf{vk}, m, \tau, \sigma)$ *is deterministic and takes as input public key* vk*, message* $m \in \mathcal{M}$*, a common message* τ*, and a signature* σ*, and outputs* $b \in \{0, 1\}$.

We define correctness, partial blindness and one-more unforgeability below. In short, correctness demands that except with negligible probability, an honest signing protocol execution yields a valid signature. One-more unforgeability affirms that one cannot output more valid signatures σ_i for distinct messages m_i than the number of successfully completed signing sessions. Partial blindness asserts that a (malicious) signer cannot link a concrete signing session with shared common message τ with the obtained signature-message pair (σ_i, m_i). Note that partial blindness is a *stronger* security property than conventional blindness. We follow the definitions from [39, 40] (almost) verbatim.

Definition 3.4 (Correctness). *A partially blind signature* BS *is correct with error* γ *if for all* $(\mathsf{vk}, \mathsf{sk}) \in \mathsf{KeyGen}(1^\lambda)$ *and all* $m \in \mathcal{M}, \tau \in \mathcal{I}$*, it holds that*

$$\Pr[\sigma \leftarrow \langle \mathsf{BSign}(\mathsf{sk}, \tau), \mathsf{BUser}(\mathsf{vk}, m, \tau) \rangle \colon \mathsf{Verify}(\mathsf{vk}, m, \tau, \sigma) = 1] \geq 1 - \gamma(\lambda)$$

If γ *is negligible, we say* BS *is correct; if* $\gamma = 0$*, it is perfectly correct.*

Definition 3.5 (One-More Unforgeability). *Let* $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{BSign}, \mathsf{BUser}, \mathsf{Verify})$ *be a blind signature scheme. Let* $\mathcal{A}$ *be an algorithm playing the following game:*

(1) Run $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and let* $\mathcal{O} = (\mathcal{O}_{\mathsf{BSign}_1}, \mathcal{O}_{\mathsf{BSign}_2})$ *be an interactive oracle running* $\mathsf{BSign}(\mathsf{sk}, \cdot)$. *We identify the* i*-th interaction with* $\mathcal{O}$ *with session identifier* $\mathsf{sid} = i$.
(2) Run $(\tau, ((m_1, \sigma_1), \ldots, (m_{Q_{\mathsf{frg}}}, \sigma_{Q_{\mathsf{frg}}}))) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{vk})$ *where* $\mathcal{A}$ *can query* $\mathcal{O}$ *in an arbitrarily interleaved way.*
(3) Output 1 *if and only if* $\mathcal{A}$ *completed at most* $Q_{\mathsf{frg}} - 1$ *interactions with* $\mathcal{O}$ *on input* τ*, and for each* $i \in [Q_{\mathsf{frg}}]$ *it holds that* $\mathsf{Verify}(\mathsf{vk}, m_i, \tau, \sigma_i) = 1$*, and all* m_i *for* $i \in [Q_{\mathsf{frg}}]$ *are pairwise distinct.*

We denote by $\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *the probability that the above game outputs* 1. *We say that* BS *is one-more unforgeable (OMUF), if for every PPT algorithm* $\mathcal{A}$*, it holds that* $\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$. *One-more strong unforgeability is defined in a similar manner, except that the distinctness requirement on the messages is relaxed: The adversary wins also if the pairs* (σ_i, m_i) *are pairwise distinct.*

Definition 3.6 (Partial Blindness). *Let* $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{BSign}, \mathsf{BUser}, \mathsf{Verify})$ *be a blind signature scheme. For an algorithm* $\mathcal{A}$ *and bit* $b \in \{0, 1\}$*, consider the following game:*

(1) Run $(\mathsf{vk}, m_0, m_1, \tau, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda)$.
(2) Let $\mathcal{O}_0$ *be an interactive oracle simulating* $\mathsf{BUser}(\mathsf{vk}, m_b, \tau)$ *and* $\mathcal{O}_1$ *be an interactive oracle simulating* $\mathsf{BUser}(\mathsf{vk}, m_{1-b}, \tau)$.
(3) Run $\mathsf{st}' \leftarrow \mathcal{A}^{\mathcal{O}_0, \mathcal{O}_1}(\mathsf{st})$, *where* $\mathcal{A}$ *has arbitrary interleaved one-time access to* $\mathcal{O}_0$ *and* $\mathcal{O}_1$. *Let* σ_b, σ_{1-b} *be the local outputs of* $\mathcal{O}_0, \mathcal{O}_1$, *respectively. We say that* $\mathcal{A}$ *finalized the interaction when it outputs* st'.
(4) If $\sigma_0 = \perp$ *or* $\sigma_1 = \perp$, *run* $b' \leftarrow \mathcal{A}(\mathsf{st}', \perp, \perp)$. *Else, run* $b' \leftarrow \mathcal{A}(\mathsf{st}', \sigma_b, \sigma_{1-b})$.
(5) Output b'.

We denote by $\mathsf{AdvBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ *difference between the probability that the above game with* $b = 0$ *outputs* 1 *and the probability that the game with* $b = 1$ *outputs* 1. *We say that* BS *satisfies partial blindness if* $\mathsf{AdvBlind}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$.

Σ-protocols We introduce (standard) notation and definitions regarding Σ-protocols for (linear) NP-relations.

Definition 3.7. *A Σ-protocol with efficiently sampleable challenge space $\mathcal{C}$ is a tuple Σ of PPT algorithms $\Sigma = (\mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ such that*

- $\mathsf{Init}(\mathbb{x}, \mathbb{w})$*: given a statement-witness pair $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$, outputs a first flow message a (a.k.a. commitment) and a state* st*, where we assume* st *includes* $(\mathbb{x}, \mathbb{w})$*;*
- $\mathsf{Resp}(\mathsf{st}, \gamma)$*: given a state* st *and a challenge $\gamma \in \mathcal{C}$, outputs a third flow message (*i.e.*, response) z,*
- $\mathsf{Verify}(\mathbb{x}, a, \gamma, z)$*: given a (purported) statement $\mathbb{x}$, a first flow message a, challenge $\gamma \in \mathcal{C}$, and a response z, outputs a bit $b \in \{0, 1\}$. The output b of* Verify *is deterministic.*

We call the tuple (a, γ, z) the transcript. *We say that a transcript is* accepting *or* valid *for $\mathbb{x}$ if it passes verification. When the context is clear, we may omit $\mathbb{x}$.*

We require the standard notions of correctness, special honest-verifier zero-knowledge (SHVZK), and relaxed (2-)special soundness. Correctness asserts that honestly-computed transcripts verify, SHVZK asserts that transcripts can be simulated, and relaxed (2-)special soundness asserts that given two *related* transcripts (*i.e.*, transcripts (a, γ, z) and (a, γ', z') with $\gamma \neq \gamma'$), it is possible to extract a witness for some (possibly relaxed) relation $\widetilde{\mathsf{R}}$. We recall the properties formally below.

Definition 3.8 (Correctness). *Let* R *be an NP-relation and* $\Sigma = (\mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ *be a Σ-protocol for* R *with challenge space $\mathcal{C}$. We say* Σ *is* correct*, if for all* $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$*, it holds that*

$$\Pr[\mathsf{Verify}(\mathbb{x}, a, \gamma, z) = 1 \mid (a, \mathsf{st}) \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w});\ \gamma \leftarrow \mathcal{C};\ z \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma)] = 1.$$

Definition 3.9 (Special Soundness). *Let* R *be an NP-relation and* $\Sigma = (\mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ *be an Σ-protocol for* R. *We call* Σ (2-)special sound*, if there exists a* deterministic *polynomial-time extractor* Ext *such that given statement $\mathbb{x}$ and two valid transcripts $\{(a, \gamma_b, z_b)\}_{b \in \{0,1\}}$ with $\gamma_0 \neq \gamma_1$, outputs a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$.*

Definition 3.10 (Special SHVZK). *Let* R *be an NP-relation and* $\Sigma = (\mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$ *be an* Σ*-protocol for* R*. We say that* Σ *is* (perfect) special honest-verifier zero-knowledge *(*SHVZK*), if there exists a PPT zero-knowledge simulator* Sim *such that for any* $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$ *and* $\gamma \in \mathcal{C}$*, it holds that* $\mathsf{D}_{\mathsf{Real}} = \mathsf{D}_{\mathsf{SHVZK}}$ *for*

$$D_{\mathsf{Real}} \coloneqq \{(\mathbb{x}, \mathbb{w}, (a, \gamma, z)) \mid a \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w});\ z \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma)\},$$
$$D_{\mathsf{SHVZK}} \coloneqq \{(\mathbb{x}, \mathbb{w}, (a, \gamma, z)) \mid (a, z) \leftarrow \mathsf{Sim}(\mathbb{x}, \gamma)\}.$$

We further recall the notion of randomizable transcripts from [39].

Definition 3.11 (Randomizable Transcripts). *Let* Σ *be a* Σ*-protocol for relation* R *with challenge space* $\mathcal{C}$*, and suppose* Σ *is* SHVZK*. Let* Rand *be an efficient randomization algorithm, such that* $\mathsf{Rand}(\mathbb{x}, (a, \gamma, z))$*, given a valid transcript* (a, γ, z) *for* $\mathbb{x}$ *outputs a new valid transcript for* $\mathbb{x}$*. We say* Σ *has* Rand-randomizable transcripts *(resp.* strongly Rand-randomizable transcripts*) if for all* $\mathbb{x} \in \mathcal{L}_{\mathsf{R}}$ *(resp. all* $\mathbb{x}$*) and all accepting* $\pi^* = (a^*, \gamma^*, z^*)$*, the distributions*

$$D_{\mathsf{SHVZK}} \coloneqq \{(\mathbb{x}, (a, \gamma, z)) \mid \gamma \leftarrow \mathcal{C};\ (a, z) \leftarrow \mathsf{Sim}(\mathbb{x}, \gamma)\}$$
$$D_{\mathsf{Rand}} \coloneqq \{(\mathbb{x}, (a, \gamma, z)) \mid (a, \gamma, z) \leftarrow \mathsf{Rand}(\mathbb{x}, \pi^*)\}$$

are identical. We may omit Rand *if clear by context.*

Canonical Σ-protocols We instantiate the Σ-protocols in this work with canonical Σ-protocols for linear maps [39, 41]. Looking ahead, we sometimes break the abstraction and leverage the concrete structure of canonical Σ-protocols; in particular, for the blinding operations of the user (cf. Section 4.2).

Let $\phi\colon \mathcal{W} \to \mathcal{X}$ be a $\mathbb{Z}_p$-linear map. Define the *canonical* Σ*-protocol* Σ_ϕ for the preimage relation $\mathsf{R}_\phi \coloneqq \{(\boldsymbol{x}, \boldsymbol{w}) \mid \phi(\boldsymbol{w}) = \boldsymbol{x}\}$ with challenge space $\mathcal{C} = \mathbb{Z}_p$ as follows:

- $\mathsf{Init}(\boldsymbol{x}, \boldsymbol{w})$: Sample $\boldsymbol{r} \leftarrow \mathcal{W}$. Output $(\mathsf{st}, \boldsymbol{a})$ where $\mathsf{st} = (\boldsymbol{w}, \boldsymbol{r})$ and $\boldsymbol{a} = \phi(\boldsymbol{r})$.
- $\mathsf{Resp}(\mathsf{st}, \gamma)$: Output $\boldsymbol{z} = \boldsymbol{r} + \gamma \boldsymbol{w}$
- $\mathsf{Verify}(\boldsymbol{x}, \boldsymbol{a}, \gamma, \boldsymbol{z})$: Return 1 if $\phi(\boldsymbol{z}) = \boldsymbol{a} + \gamma \boldsymbol{x}$, else return 0. (It implicitly checks that all elements are in their respective spaces of definition, i.e., in $\mathcal{X}, \mathcal{C}, \mathcal{W}$ respectively).

We recall some useful facts about Σ_ϕ from [39, Lemma 2.9].

Lemma 3.12. *Let* Σ_ϕ *the above canonical* Σ*-protocol for* R_ϕ*. Then* Σ_ϕ *is 2-special sound,* SHVZK*, and has strongly* Rand*-randomizable transcripts, where the* SHVZK *simulator* Sim *and* Rand *are as follows:*

- $\mathsf{Sim}(\boldsymbol{x}, \gamma)$ *samples* $\boldsymbol{z} \leftarrow \mathcal{W}$*, sets* $a = \gamma \cdot \boldsymbol{x} - \phi(\boldsymbol{z})$ *and outputs* $(a, \boldsymbol{z})$*.*
- $\mathsf{Rand}(\boldsymbol{x}, (a^*, \gamma^*, \boldsymbol{z}^*))$ *samples* $\gamma' \leftarrow \mathcal{C}$ *and* $\boldsymbol{z}' \leftarrow \mathcal{W}$ *and outputs*

$$(a, \gamma, \boldsymbol{z}) = (a^* - \gamma' \boldsymbol{x} + \phi(\boldsymbol{z}'), \gamma^* + \gamma', \boldsymbol{z}^* + \boldsymbol{z}'). \tag{3.1}$$

Non-Interactive Proof Systems We recall straightline-extractable non-interactive zero-knowledge proofs as defined in [39, 40], and follow their definitions almost verbatim. As in [37, 39], we additionally consider a common random string crs as input in our definitions. The crs can be derived in the random oracle.

Definition 3.13 (Non-Interactive Proof System). *A* non-interactive proof system Π *for NP-relation* R *in the ROM is a pair* $\Pi = (\mathsf{Prove}, \mathsf{Verify})$ *of PPT algorithms with access to a random oracle* H *and a CRS* $\mathsf{crs} \in \{0,1\}^{\ell(\lambda)}$*, where*

- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w})$*: generates a proof* π *given* $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$*.*
- $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \pi)$*: verifies a proof* π *for statement* $\mathbb{x}$ *and outputs* 0 *or* 1*.*

We require the usual notions of correctness, zero-knowledge and straightline $\widetilde{\mathsf{R}}$-extractability. Here, $\widetilde{\mathsf{R}}$ is a (possibly relaxed) knowledge relation. The notions are formally defined below, following the definitions from [39, 40] (almost) verbatim. Note that we define straightline notions of zero-knowledge and knowledge soundness, *i.e.*, the simulators/extractors have no (black-box) access to the adversary.

We call Π *correct* with error $\varepsilon_{\mathsf{cor}}$ if for any $\mathsf{crs} \in \{0,1\}^{\ell}$ and $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$ every generated proof is valid (*i.e.*, Verify outputs 1) except with probability $\varepsilon_{\mathsf{cor}} = \varepsilon_{\mathsf{cor}}(\lambda, \mathbb{x})$.

Definition 3.14 ((Statistical) Correctness). *Let* $\Pi = (\mathsf{Prove}, \mathsf{Verify})$ *be a non-interactive proof system for NP-relation* R*. It has* correctness error $\varepsilon_{\mathsf{cor}}$ *if for all* $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}_\lambda$*, it holds that*

$$\Pr[\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \pi) = 1 \mid \pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w})] \geq 1 - \varepsilon_{\mathsf{cor}}(\lambda, \mathbb{x})$$

where the probability is over the choice of H*,* crs*, and the randomness of* $\mathsf{Prove}, \mathsf{Verify}$*. We say* Π *is* (perfectly) correct *if* $\max_{\mathbb{x} \in \mathcal{L}_{\mathsf{R},\lambda}} \varepsilon_{\mathsf{cor}}(\lambda, \mathbb{x})$ *is negligible (resp.* 0*).*

We call Π *zero-knowledge*, if there is a simulator (which can choose crs and program H), such that no PPT distinguisher can distinguish between an honest setup with Prove oracle from a simulated setup with Sim oracle.

Definition 3.15 ((Straightline) Zero-Knowledge). *Let* $\Pi = (\mathsf{Prove}, \mathsf{Verify})$ *be a non-interactive proof system for NP-relation* R*. Let* $(\mathsf{SimSetup}, \mathsf{Sim})$ *be PPT algorithms; we usually omit* $\mathsf{SimSetup}$ *and only speak of* Sim*. Let* $\mathcal{A}$ *be an algorithm and let*

$$\mathsf{Real}^{\Pi}_{\mathcal{A}}(\lambda) := \Pr[b = 1 \mid \mathsf{crs} \leftarrow \{0,1\}^{\ell};\ b \leftarrow \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{Prove}}}(1^\lambda, \mathsf{crs})]$$
$$\mathsf{Ideal}^{\Pi}_{\mathcal{A},\mathsf{Sim}}(\lambda) := \Pr[b = 1 \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SimSetup}(1^\lambda);\ b \leftarrow \mathcal{A}^{\mathsf{H}, \mathcal{O}_{\mathsf{Sim}}}(1^\lambda, \mathsf{crs})]$$

Here, $\mathcal{A}$ *has (black-box) access to oracles* H *and* $\mathcal{O}_{\mathsf{Prove}}$ *or* $\mathcal{O}_{\mathsf{Sim}}$*, defined as:*

- H *is a random oracle. The simulator learns all queries to* H *and programs* H *on fresh inputs. (If* $\mathsf{H}(m)$ *is still undefined,* Sim *chooses* $\mathsf{H}(m)$ *given* m*.)*
- $\mathcal{O}_{\mathsf{Prove}}(\mathbb{x}, \mathbb{w})$*: Return* $\perp$ *if* $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}$*. Else, output* $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, \mathbb{x}, \mathbb{w})$*.*
- $\mathcal{O}_{\mathsf{Sim}}(\mathbb{x}, \mathbb{w})$*: Return* $\perp$ *if* $(\mathbb{x}, \mathbb{w}) \notin \mathsf{R}$*. Else, output* $\pi \leftarrow \mathsf{Sim}^{\mathsf{H}}(\mathsf{td}, \mathbb{x})$*.*

For any distinguisher $\mathcal{A}$, we define the following (where $Q_{\mathsf{H}}, Q_{\mathsf{Prove}}$ bounds the number of allowed queries).

- Setup indistinguishability*: For any PPT $\mathcal{A}$, the standard distinguishing advantage $\mathsf{AdvCRS}_{\mathcal{A}}^{\Pi,\mathsf{SimSetup}}(\lambda)$ between real and trapdoored crs is negligible.*
- Zero-knowledge simulation indistinguishability*: For any PPT $\mathcal{A}$ which makes a most Q_{H} (resp. Q_{Sim}) queries to H (resp. Prove or Sim), the advantage $\mathsf{AdvZK}_{\mathcal{A}}^{\Pi,\mathsf{Sim}}(\lambda, Q_{\mathsf{H}}, Q_{\mathsf{Sim}}) = |\mathsf{Real}_{\mathcal{A}}^{\Pi}(\lambda) - \mathsf{Ideal}_{\mathcal{A},\mathsf{Sim}}^{\Pi}(\lambda)|$ is negligible.*

We call Π zero-knowledge *if there is a zero-knowledge simulator for Π with setup and zero-knowledge indistinguishability.*

Finally, we call Π *straightline $\widetilde{\mathsf{R}}$-extractable* for a knowledge relation $\widetilde{\mathsf{R}}$ (where usually $\mathsf{R} \subseteq \widetilde{\mathsf{R}}$), if there exists an extractor $(\mathsf{ExtSetup}, \mathsf{Ext})$ which is allowed to choose crs via $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda)$, such that: (1) $\mathsf{ExtSetup}$ is indistinguishable from uniform; (2) Given $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda)$ as the crs, for any accepting proof $(\mathbb{x}, \pi)$ an adversary submits to a verification oracle, the extractor can provide a witness $\widetilde{\mathbb{w}}$ such that $(\mathbb{x}, \widetilde{\mathbb{w}}) \in \widetilde{\mathsf{R}}$, given only the trapdoor td and a list of all random oracle queries.

Definition 3.16 (Relaxed Knowledge Soundness). *Let $\Pi = (\mathsf{Prove}, \mathsf{Verify})$ be a non-interactive proof system for a relation R and let $\widetilde{\mathsf{R}}$ be an NP-relation. Let $(\mathsf{ExtSetup}, \mathsf{Ext})$ be PPT algorithms. Let $\mathcal{A}$ be an oracle algorithm and let*

$$\mathsf{Real}_{\mathcal{A}}(\lambda) := \Pr[b = 1 \mid \mathsf{crs} \leftarrow \{0,1\}^{\ell(\lambda)};\ b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_{\mathsf{Verify}}}(1^\lambda, \mathsf{crs})]$$
$$\mathsf{Ideal}_{\mathcal{A}}(\lambda) := \Pr[b = 1 \mid (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{ExtSetup}(1^\lambda);\ b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_{\mathsf{Ext}}}(1^\lambda, \mathsf{crs})]$$

Here, $\mathcal{A}$ has (black-box) access to oracles H and $\mathcal{O}_{\mathsf{Verify}}$ or $\mathcal{O}_{\mathsf{Ext}}$, defined as:

- *H is a random oracle. The extractor knows the H-query-response-list $\mathcal{Q}$.*
- *$\mathcal{O}_{\mathsf{Verify}}(\mathbb{x}, \pi)$: Return $\mathsf{Verify}(\mathbb{x}, \pi)$.*
- *$\mathcal{O}_{\mathsf{Ext}}(\mathbb{x}, \pi)$: If $\mathsf{Verify}(\mathbb{x}, \pi) = 1$ and $(\mathbb{x}, \mathbb{w}) \in \widetilde{\mathsf{R}}$ for $\mathbb{w} \leftarrow \mathsf{Ext}(\mathsf{td}, \mathcal{Q}, \mathbb{x}, \pi)$, return 1. Else, return 0. Here, $\mathcal{Q}$ denotes current H-query-response-list.*

For any adversary $\mathcal{A}$, we define the following.

- Setup indistinguishability*: For any PPT $\mathcal{A}$, the standard distinguishing advantage $\mathsf{AdvCRS}_{\mathcal{A}}^{\Pi,\mathsf{ExtSetup}}(\lambda)$ between real and trapdoored crs is negligible.*
- Knowledge soundness for knowledge relation $\widetilde{\mathsf{R}}$*: For any PPT $\mathcal{A}$ which makes at most Q_{H} (resp. Q_{Ext}) queries to H (resp. $\mathcal{O}_{\mathsf{Verify}}$ or $\mathcal{O}_{\mathsf{Ext}}$) the advantage $\mathsf{AdvExt}_{\mathcal{A}}^{\Pi,\widetilde{\mathsf{R}}}(\lambda, Q_{\mathsf{H}}, Q_{\mathsf{Ext}}) \coloneqq |\mathsf{Real}_{\mathcal{A}}(\lambda) - \mathsf{Ideal}_{\mathcal{A}}(\lambda)|$ is negligible*

We call Π straightline $\widetilde{\mathsf{R}}$-extractable *if there is a knowledge extractor for Π with setup indistinguishability and knowledge soundness.*

4 Our Blind Signatures from DL

We present our blind signature based on inequality arguments for Pedersen commitments. The construction is a natural adaptation of the ElGamal-based scheme in [39] to Pedersen commitments. Our notation is kept in line with [39]. With the technique sketched in Section 2, we prove security under the DL assumption in the random oracle model.

4.1 Underlying Σ-protocol

Our blind signature issues a Fiat–Shamir compiled Σ-protocol in an oblivious manner. We establish some notation for the Σ-protocols we employ.

Pedersen Openings. Let Φ_{ped}, parameterized by $H \in \mathbb{G}$, be defined as

$$\Phi^H_{\mathsf{ped}}(C, (x, y)) = \big(xH + yC\big) = (x, y) \cdot \begin{pmatrix} H \\ C \end{pmatrix}. \tag{4.1}$$

We omit parameter H if clear by context. Clearly, the function Φ_{ped} is linear for fixed C. We define the relation $\mathsf{R}_{\mathsf{ped}}$ with induced language $\mathcal{L}_{\mathsf{ped}}$ as

$$\mathsf{R}_{\mathsf{ped}} \coloneqq \big\{(\mathbb{x}, \mathbb{w}) \mid \mu_{\$} G = \Phi^H_{\mathsf{ped}}(C, (x, y))\big\}, \tag{4.2}$$

where $\mathbb{x} = (G, H, C, \mu_{\$}) \in \mathbb{G}^3 \times \mathbb{Z}_p$ and $\mathbb{w} = (x, y) \in \mathbb{Z}_p^2$. Note that if $C = \mu G + rH$ is a Pedersen commitment with opening (μ, r), then $\mathbb{w} = (-\alpha r, \alpha)$ is a valid witness under $\mathsf{R}_{\mathsf{ped}}$ with $\mu_{\$} = \alpha\mu$ for any $\alpha \in \mathbb{Z}_p$.

Notice that for $\mu \neq 0$ the value $\mu_{\$} = \alpha\mu$ is uniform over $\mathbb{Z}_p^{\times}$ for $\alpha \leftarrow \mathbb{Z}_p^{\times}$. Therefore, we can reveal $\mu_{\$}$ together with a proof of *knowledge* of $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}_{\mathsf{ped}}$ to prove knowledge of a non-zero opening, and equivalently inequality of committed messages.

DL. We define the linear function Φ_{dl}, implicitly parameterized by $G \in \mathbb{G}$, as

$$\Phi_{\mathsf{dl}}(x) = xG \tag{4.3}$$

and the corresponding relation

$$\mathsf{R}_{\mathsf{dl}} \coloneqq \big\{(\mathbb{x}, \mathbb{w}) \mid X = \Phi_{\mathsf{dl}}(x)\big\}, \tag{4.4}$$

where $\mathbb{x} = (G, X) \in \mathbb{G}^2$, $\mathbb{w} = x \in \mathbb{Z}_p$.

4.2 Blind Signature

We are almost ready to present our blind signature. Before, let us define the NIZK we rely on.

Non-interactive Proof System Let Π_μ be a NIZK for the following relation

$$\mathsf{R}_\mu := \{(\mathbb{x}, \mathbb{w}) \mid C = \mu G + tH\} \tag{4.5}$$

using the random oracle H_Π with common reference string crs_μ, where $\mathbb{x} = (C, G, H) \in \mathbb{G}^3$ and $\mathbb{w} = (\mu, t) \in \mathbb{Z}_p^2$. To instantiate Π_μ, we employ the proof system from [40, Section 4.1]. This proof system is zero-knowledge and straightline $\widetilde{\mathsf{R}}_\mu$-extractable for the knowledge relation

$$\widetilde{\mathsf{R}}_\mu := \{(\mathbb{x} = (C, G, H),\ \mathbb{w} = (x, t)) \mid xG = H \vee (\mathbb{x}, \mathbb{w}) \in \mathsf{R}_\mu\}. \tag{4.6}$$

While their proof system does not require a common reference string, we define our blind signature with explicit crs_μ for consistency (with our definitions).

Random Oracles and CRS Let Σ_{dl} and Σ_{ped} be canonical Σ-protocols for relation $\mathsf{R}_{\mathsf{ped}}$ and R_{dl} (cf. Section 3). We note that for blinding, we sometimes exploit the specific algebraic structure of the transcript. Let

$$\mathsf{crs}_{\mathsf{ped}} := (G, \mathsf{crs}_\mu).$$

We assume that $\mathsf{crs}_{\mathsf{ped}}$ is passed to all other algorithms implicitly. We assume several random oracles (which can be obtained from a single random oracle by standard techniques). In our construction we use the random oracles $\mathsf{H}_\Pi, \mathsf{H}_{\mathsf{par}}, \mathsf{H}_\mu, \mathsf{H}_{\mathsf{ch}}$:

- We denote by H_Π the random oracle for the NIZK Π_μ.
- We always set $(H_\tau, C_\tau) = \mathsf{H}_{\mathsf{par}}(\tau)$ for $\mathsf{H}_{\mathsf{par}} \colon \{0,1\}^* \to \mathbb{G}^2$ to generate a fresh commitment key and Pedersen commitment for each common message τ.
- We hash-then-sign via $\mu = \mathsf{H}_\mu(\mathsf{msg})$, where $\mathsf{H}_\mu \colon \{0,1\}^* \to \mathbb{Z}_p$.
- We let $\mathsf{H}_{\mathsf{ch}} \colon \{0,1\}^* \to \mathbb{Z}_p$ be the random oracle used for Fiat–Shamir.

Denote by $\mathsf{Sim}_{\mathsf{ped}}$ and $\mathsf{Sim}_{\mathsf{dl}}$ the SHVZK simulator of Σ_{ped} and Σ_{dl}, respectively. Note that $\mathsf{Sim}_{\mathsf{ped}}$ and $\mathsf{Sim}_{\mathsf{dl}}$ obtain a statement-challenge pair $(\mathbb{x}, \gamma)$ as input and generate valid Σ-protocol transcripts with challenge γ.

Construction We present our construction $\mathsf{BS}_{\mathsf{dl}}$ in Figs. 2 and 3. On a high-level structure, the blind signature is as follows.

- *CRS.* The common reference string is $\mathsf{crs}_{\mathsf{ped}} = (G, \mathsf{crs}_\mu)$, where G generates group $\mathbb{G}$ of prime-order p and crs_μ is the CRS for Π_μ.
- *Verification key.* The verification key vk is $X = xG$, where $x \leftarrow \mathbb{Z}_p$ is the secret key. In other words, x is the witness for statement $\Phi_{\mathsf{dl}}(x) = X$.
- *Signing.* In the protocol, the user commits to $\mu = \mathsf{H}_\mu(\mathsf{msg})$ in C_μ. A signature for hashed message μ with common message τ is a Fiat-Shamir-compiled OR-proof π for
  - knowledge of x such that $\Phi_{\mathsf{dl}}(x) = X$; **or**
  - knowledge of $\mathbb{w}_{\mathsf{ped}} = (x, y)$ such that $(\mathbb{x}_{\mathsf{ped}}, \mathbb{w}_{\mathsf{ped}}) \in \mathsf{R}_{\mathsf{ped}}$ with $\mathbb{x}_{\mathsf{ped}} = (G, H_\tau, C_\tau - C_\mu, \mu_{\$})$, where $\mu_{\$} \neq 0$ (*i.e.*, the commitments C_τ and C_μ contain distinct messages). Recall that $(H_\tau, C_\tau) = \mathsf{H}_{\mathsf{par}}(\tau)$.

$\mathsf{BS}_{\mathsf{dl}}.\mathsf{KeyGen}(1^\lambda)$

$x \leftarrow \mathbb{Z}_p; \quad X := xG$

return $(\mathsf{vk}, \mathsf{sk}) := (X, x)$

$\mathsf{BS}_{\mathsf{dl}}.\mathsf{BSign}(\mathsf{sk}, \tau) \rightleftarrows \mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}(\mathsf{vk}, m, \tau)$

Proceeds in 4 moves and is given in Fig. 3

$\mathsf{BS}_{\mathsf{dl}}.\mathsf{Verify}(\mathsf{vk}, \tau, \mathsf{msg}, \sigma)$

parse $(\mu_\$, \pi) := \sigma$

parse $(A_{\mathsf{ped}}, A_{\mathsf{dl}}, \gamma_{\mathsf{ped}}, \gamma_{\mathsf{dl}}, \boldsymbol{z}_{\mathsf{ped}}, z_{\mathsf{dl}}) := \pi$

$\mu := \mathsf{H}_\mu(\mathsf{msg}); \quad (H_\tau, C_\tau) := \mathsf{H}_{\mathsf{par}}(\tau)$

$C_\mu := \mu G; \quad C := C_\tau - C_\mu$

$\mathbb{x}_{\mathsf{ped}} := (G, H_\tau, C, \mu_\$); \quad \mathbb{x}_{\mathsf{dl}} := (G, X)$

$\gamma := \mathsf{H}_{\mathsf{ch}}(\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, A_{\mathsf{dl}})$

if $\mu_\$ = 0$ **then return** 0

if $\mathsf{Verify}_{\mathsf{ped}}(\mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}}) = 0$ **then return** 0

if $\mathsf{Verify}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}}, A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}}) = 0$ **then return** 0

if $\gamma \neq \gamma_{\mathsf{ped}} + \gamma_{\mathsf{dl}}$ **then return** 0

return 1

Fig. 2: Key generation and verification of our DL-based blind signature $\mathsf{BS}_{\mathsf{dl}}$.

The honest signer uses the witness x for X in the DL-branch to issue transcripts and simulates the inequality-branch.

– *Verification.* Construct $C_\mu = \mu G$ as commitment to μ with randomness 0, and verify the proof π.

The OR-proof is instantiated following [3, 18]. That is, the challenge $\gamma = \gamma_{\mathsf{dl}} + \gamma_{\mathsf{ped}}$ for the OR-proof is secret shared between both Σ-protocols. In the protocol, the DL-branch is executed via the witness, whereas the inequality-branch is simulated via $\mathsf{Sim}_{\mathsf{ped}}$.

Blinding Operations: We remark that the user must randomize the Σ-protocol transcripts, as well as the commitment C^*_μ and the non-zero message $\mu^*_\$$ sent by the signer. (We follow the convention that non-blinded values X are denoted by X^*, whereas the blinded value is denoted by X.)

This is done as follows. During signing, the user employs a commitment $C^*_\mu = \mu G + tH_\tau$ with randomness $t \leftarrow \mathbb{Z}_p$. The transcript for the inequality-branch is then adapted by the user to the derandomized commitment $C_\mu = \mu G$ before blinding (highlighted via). Note that C_μ is independently-distributed from the signing session. Furthermore, the user blinds the randomized non-zero message $\mu_\$$ (highlighted via). Note that the above modifications also require changes to the Σ-protocols as the statement and witness is modified as consequence. After these adaptations, the user blinds the transcript following the randomization Rand from Lemma 3.12. The steps towards randomizing the challenge (resp. response) are highlighted via (resp.). These modifications rely on the canonical structure of the underlying Σ-protocols.

Remark 4.1 (Optimization). We can omit $(A_{\mathsf{dl}}, A_{\mathsf{ped}})$ from the signature σ, as these can be recomputed from the statements and $(\gamma_{\mathsf{dl}}, \gamma_{\mathsf{ped}}, z_{\mathsf{dl}}, \boldsymbol{z}_{\mathsf{ped}})$.

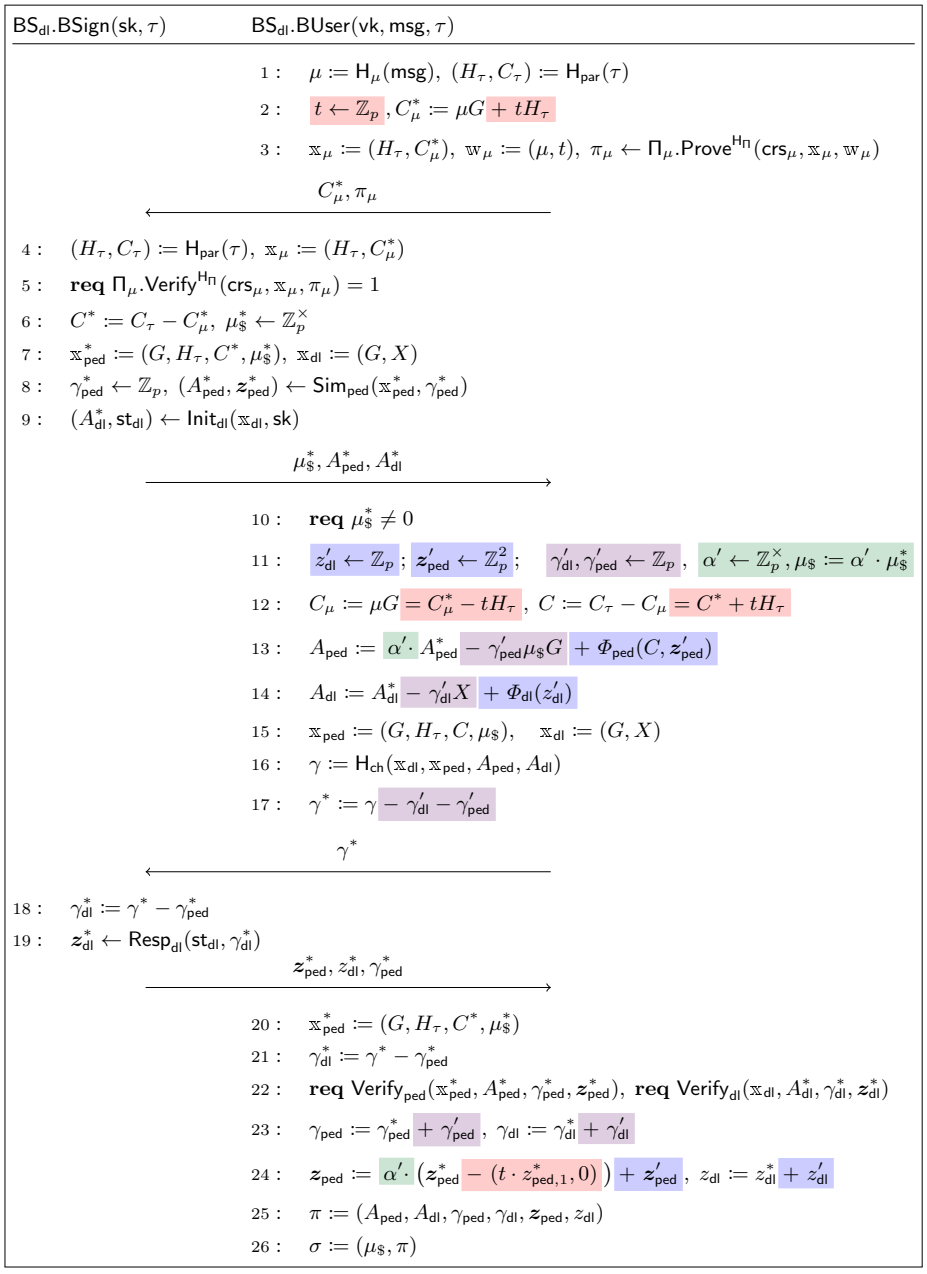

Fig. 3: The signing session for $\mathsf{BS}_{\mathsf{dl}}$ for message $\mathsf{msg} \in \{0,1\}^*$ and common message $\tau \in \{0,1\}^*$. The signer and user abort (*i.e.*, output $\perp$) if $\mathbf{req}\ C$ is evaluated for a false condition C. Recall that $\mathsf{sk} = x$ is a witness for $\mathcal{L}_{\mathsf{dl}}$ membership of $\mathsf{vk} = X$. The colors highlight terms for masking challenges γ, responses z, and statements $\mu_\$$ and C_μ.

5 Analysis of our Blind Signature from DL

Let us prove correctness, one-more unforgeability and partial blindness of $\mathsf{BS}_{\mathsf{dl}}$.

5.1 Correctness

Correctness follows from a straightforward calculation.

Theorem 5.1 (Correctness). $\mathsf{BS}_{\mathsf{dl}}$ *is correct with correctness error* γ_{err}*, where* γ_{err} *is the correctness error of* Π_μ*.*

Proof. Denote by $\sigma = (\mu_\$, \pi)$ the output of a honest signing session. Assume that π_μ verifies (which holds except with probability γ_{err}, where γ_{err} is the correctness error of Π_μ). We need to show that $\pi = (A_{\mathsf{ped}}, A_{\mathsf{dl}}, \gamma_{\mathsf{ped}}, \gamma_{\mathsf{dl}}, \boldsymbol{z}_{\mathsf{ped}}, z_{\mathsf{dl}})$ is a valid OR-proof for statements $\mathbb{x}_{\mathsf{dl}} := (G, X)$ and $\mathbb{x}_{\mathsf{ped}} := (G, H_\tau, C, \mu_\$)$. Let $\gamma' := \mathsf{H}_{\mathsf{ch}}(\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, A_{\mathsf{dl}})$ and $\gamma_{\mathsf{ped}} := \gamma' - \gamma_{\mathsf{dl}}$. By construction, we have $\gamma' = \gamma$, where γ is the challenge. Similarly, we have that $\mu_\$ = \mu^*_\$ \alpha' \neq 0$ as $\mu^*_\$, \alpha' \neq 0$. It remains to show that

$$\mathsf{Verify}_{\mathsf{ped}}(\mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}}) = 1 \ \wedge \ \mathsf{Verify}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}}, A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}}) = 1.$$

We show that the Σ_{ped} transcript verifies below. It can be shown analogously that the Σ_{dl} transcript verifies, and we omit details. Below, for some Σ_{ped} transcript $\tau_k = (A_k, \gamma_k, \boldsymbol{z}_k)$, we denote $\tau_k[A] = A_k$, $\tau_k[\gamma] = \gamma_k$ and $\tau_k[\boldsymbol{z}] = \boldsymbol{z}_k$ for convenience. We also denote the i-th component z_i of $\boldsymbol{z}_k = (z_0, z_1)$ via $\tau_k[\boldsymbol{z}]_i$.

Let us show that $\mathsf{Verify}_{\mathsf{ped}}(\mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}}) = 1$. Note that by Lemma 3.12, the simulator $\mathsf{Sim}_{\mathsf{ped}}$ outputs a pair $(A_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}})$ such that

$$\Phi_{\mathsf{ped}}(C^*, \boldsymbol{z}^*_{\mathsf{ped}}) = A^*_{\mathsf{ped}} + \gamma^*_{\mathsf{ped}} \mu^*_\$ G,$$

that is, we have $\mathsf{Verify}_{\mathsf{ped}}(\mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}}) = 1$ for $\mathbb{x}^*_{\mathsf{ped}} = (G, H_\tau, C^*, \mu^*_\$)$ and $\mathbb{x}_{\mathsf{ped}} = (G, H_\tau, C, \mu_\$)$. Denote by $\tau^* := (A^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}})$ the accepting transcript.

First, we show that after the transformation highlighted with , the modified transcript $\tau_0 := \tau^* - (0, 0, (t \cdot \tau^*[\boldsymbol{z}]_1, 0))$ verifies with respect to $\mathbb{x}'_{\mathsf{ped}} = (G, H_\tau, C, \mu^*_\$)$, where $C = C_\tau - C_\mu = C^* + t \cdot H_\tau$:

$$\begin{aligned}
\Phi_{\mathsf{ped}}(C, \tau_0[\boldsymbol{z}]) &= \Phi_{\mathsf{ped}}(C, \tau^*[\boldsymbol{z}] - (t \cdot \tau^*[\boldsymbol{z}]_1, 0)) \\
&= \Phi_{\mathsf{ped}}(C, \tau^*[\boldsymbol{z}]) - \Phi_{\mathsf{ped}}(C, (t \cdot \tau^*[\boldsymbol{z}]_1, 0)) \\
&= \Phi_{\mathsf{ped}}(C^* + tH_\tau, \tau^*[\boldsymbol{z}]) - \Phi_{\mathsf{ped}}(C, (t \cdot \tau^*[\boldsymbol{z}]_1, 0)) \\
&= \tau^*[\boldsymbol{z}] \cdot \begin{pmatrix} H_\tau \\ C^* + tH_\tau \end{pmatrix} - \Phi_{\mathsf{ped}}(C, (t \cdot \tau^*[\boldsymbol{z}]_1, 0)) \\
&= \tau^*[\boldsymbol{z}] \cdot \begin{pmatrix} H_\tau \\ C^* \end{pmatrix} + \tau^*[\boldsymbol{z}] \begin{pmatrix} 0 \\ tH_\tau \end{pmatrix} - (t \cdot \tau^*[\boldsymbol{z}]_1, 0) \begin{pmatrix} H_\tau \\ C \end{pmatrix} \\
&= \Phi_{\mathsf{ped}}(C^*, \tau^*[\boldsymbol{z}]) \\
&= \tau_0[A] + \tau_0[\gamma] \mu^*_\$ G.
\end{aligned}$$

Next, we show that after the transformation highlighted with , the modified transcript $\tau_1 \coloneqq (\alpha' \cdot \tau_0[A], \tau_0[\gamma], \alpha' \cdot \tau_0[\boldsymbol{z}])$ still verifies with respect to $\mathbb{x}_{\mathsf{ped}} = (G, H_\tau, C, \mu_\$)$, where $\mu_\$ = \alpha' \cdot \mu_\$^*$:

$$\begin{aligned}
\Phi_{\mathsf{ped}}(C, \tau_1[\boldsymbol{z}]) &= \Phi_{\mathsf{ped}}(C, \alpha' \cdot \tau_0[z]) \\
&= \alpha' \cdot \Phi_{\mathsf{ped}}(C, \tau_0[z]) \\
&= \alpha' \cdot (\tau_0[A] + \tau_0[\gamma]\mu_\$^* G) \\
&= \alpha' \cdot \tau_0[A] + \tau_0[\gamma]\, \alpha' \cdot \mu_\$^* G \\
&= \tau_1[A] + \tau_1[\gamma]\mu_\$ G.
\end{aligned}$$

Next, we show that after the transformation highlighted with , the modified transcript $\tau_2 \coloneqq \tau_1 + (\Phi_{\mathsf{ped}}(C, \boldsymbol{z}'_{\mathsf{ped}}), 0, \boldsymbol{z}'_{\mathsf{ped}})$ still verifies with respect to $\mathbb{x}_{\mathsf{ped}}$:

$$\begin{aligned}
\Phi_{\mathsf{ped}}(C, \tau_2[\boldsymbol{z}]) &= \Phi_{\mathsf{ped}}(C, \tau_1[\boldsymbol{z}] + \boldsymbol{z}'_{\mathsf{ped}}) \\
&= \Phi_{\mathsf{ped}}(C, \tau_1[\boldsymbol{z}]) + \Phi_{\mathsf{ped}}(C, \boldsymbol{z}'_{\mathsf{ped}}) \\
&= \tau_1[A] + \tau_1[\gamma]\mu_\$ G + \Phi_{\mathsf{ped}}(C, \boldsymbol{z}'_{\mathsf{ped}}) \\
&= \tau_2[A] + \tau_2[\gamma]\mu_\$ G.
\end{aligned}$$

Finally, we show that after the transformation highlighted with , the modified transcript $\tau_3 \coloneqq \tau_2 + (-\gamma'_{\mathsf{ped}}\mu_\$ G, \gamma'_{\mathsf{ped}}, 0)$ still verifies with respect to $\mathbb{x}_{\mathsf{ped}}$:

$$\begin{aligned}
\Phi_{\mathsf{ped}}(C, \tau_3[\boldsymbol{z}]) &= \Phi_{\mathsf{ped}}(C, \tau_2[\boldsymbol{z}]) \\
&= \tau_2[A] + \tau_2[\gamma]\mu_\$ G \\
&= \tau_2[A] + \gamma^*_{\mathsf{ped}}\mu_\$ G + \gamma'_{\mathsf{ped}}\mu_\$ G - \gamma'_{\mathsf{ped}}\mu_\$ G \\
&= \tau_2[A] + (\gamma^*_{\mathsf{ped}} + \gamma'_{\mathsf{ped}})\mu_\$ G - \gamma'_{\mathsf{ped}}\mu_\$ G \\
&= \tau_3[A] + \tau_3[\gamma]\mu_\$ G.
\end{aligned}$$

As the transcripts remain accepting after each transformation, the final transcript accepts. Notice that τ_3 is identical to the Σ_{ped} transcript in the issued signature σ. The acceptance of the Σ_{dl} transcript in σ follows similarly. □

5.2 Blindness

Since our protocol is an adaptation of [39], it is unsurprising that the blindness proof requires little changes. We present the proof with the changes below. Let us give a brief sketch for completeness. The proof proceeds via a hybrid argument; the following modifications are made to both signing oracles. First, the NIZK π_μ is simulated, then the Σ-protocol transcripts $(A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}})$ and $(A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, z_{\mathsf{ped}})$ are

simulated via SHVZK and the random oracle is programmed accordingly.[9] Also, $\mu^*_{\$}$ sampled at random. Finally, the commitment C^*_μ is switched to a commitment to zero. Again, the distribution is identical due to the perfect hiding property of Pedersen commitments.

Theorem 5.2 (Blindness). *Let $\mathbb{G}$ be a group of prime order p. For any PPT adversary $\mathcal{A}$ that causes at most Q_{ch} random oracle queries to H_{ch}, there are reductions $\mathcal{B}_{\mathsf{stp}}$ and $\mathcal{B}_{\mathsf{zk}}$ whose running time is roughly that of the blindness game, such that*

$$\mathsf{AdvBlind}^{\mathsf{BS}_{\mathsf{dl}}}_{\mathcal{A}}(\lambda) \leq 2 \cdot (\mathsf{AdvCRS}^{\Pi,\mathsf{SimSetup}}_{\mathcal{B}_{\mathsf{stp}}}(\lambda) + \mathsf{AdvZK}^{\Pi_\mu}_{\mathcal{B}_{\mathsf{zk}}}(\lambda)) + \frac{4Q_{\mathsf{ch}}}{p}.$$

Proof. This proof is taken almost verbatim from [39], with only the changes applied that are required to switch from DDH and ElGamal encryption claims $\mathbb{x}_{\mathsf{DDH}}, \mathbb{x}_{\mathsf{elg}}$ to DLog and Pedersen commitment claims $\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}$. We argue by game hops, gradually modifying the game, primarily the oracles $\mathcal{O}_0, \mathcal{O}_1$, until the adversary has no information about the secret bit anymore. For each game Game i, we denote by ε_i the advantage of $\mathcal{A}$ in Game i.

Game 0 (Honest). This is the real blindness game, where $\mathcal{A}$ has access to the honest oracles $\mathcal{O}_0, \mathcal{O}_1$ and $b \leftarrow \{0,1\}$ is the challenge bit. The adversary chooses a common message τ and two messages $(\mathsf{msg}_0, \mathsf{msg}_1)$ and then interacts with $(\mathcal{O}_0, \mathcal{O}_1)$, which run the protocol $\mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}(\tau, \mathsf{msg}_b)$ and $\mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}(\tau, \mathsf{msg}_{1-b})$, respectively. On finalization, $\mathcal{A}$ learns (σ_b, σ_{1-b}) outputs its guess b' for b. (If any $\sigma_i = \bot$, then $\mathcal{A}$ learns $(\bot, \bot)$ to avoid trivial attacks.) If $b' = b$, the game outputs 1 (*i.e.*, $\mathcal{A}$ wins), else 0.

In the following games, let msg_i be the messages $\mathcal{A}$ inputs to $\mathcal{O}_i$, and let $\mu_i = \mathsf{H}_\mu(\mathsf{msg}_i)$ (for $i = 0, 1$). Moreover, let τ be the common message.

Game 1 (Simulate π_μ). The common reference string $\mathsf{crs}_{\mathsf{ped}}$ is setup via $\mathsf{SimSetup}$ and the proofs π_μ of knowledge for C^*_μ are simulated. Recall that Π_μ uses its own random oracle/setup, hence it switching to simulation is trivial. Therefore, we have direct reduction to zero-knowledge of Π_μ, and find adversaries $\mathcal{B}_{\mathsf{stp}}$ and $\mathcal{B}_{\mathsf{zk}}$ such that

$$|\varepsilon_0 - \varepsilon_1| \leq \mathsf{AdvCRS}^{\Pi,\mathsf{SimSetup}}_{\mathcal{B}_{\mathsf{stp}}}(\lambda) + \mathsf{AdvZK}^{\Pi_\mu}_{\mathcal{B}_{\mathsf{zk}}}(\lambda).$$

Game 2 (Abort if H_{ch} was queried before $\mathcal{A}$ finalized, *i.e.*, before receiving (σ_0, σ_1).). When the game queries $(\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{dl}}, A_{\mathsf{ped}})$ to H_{ch} on behalf of $\mathcal{O}_0$ (resp. $\mathcal{O}_1$), abort if H_{ch} was queried on this before (by either $\mathcal{A}$ or the game itself). Due to masking with $\phi_{\mathsf{dl}}(z'_{\mathsf{dl}})$, from $\mathcal{A}$'s view the value A_{dl} is uniformly

[9] The latter is possible as the user employs the randomization algorithm Rand for canonical Σ-protocols from Lemma 3.12 to blind the issued transcripts. By Definition 3.11, the randomized transcripts (from the real execution) and the SHVZK-simulated transcripts are distributed identically.

random in $\mathbb{G}$ (prior to receiving (σ_0, σ_1)). Hence we find

$$|\varepsilon_1 - \varepsilon_2| \leq \frac{2Q_{\mathsf{ch}}}{p}.$$

In the above, the factor 2 is due to a hybrid over $\mathcal{O}_0$ and $\mathcal{O}_1$.

Game 3 (Program H_{ch}). Sample $\gamma^*, \gamma_{\mathsf{dl}}, \gamma_{\mathsf{ped}}$ uniformly at the start of $\mathcal{O}_0$ (resp. $\mathcal{O}_1$), and program $\mathsf{H}_{\mathsf{ch}}(\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{dl}}, A_{\mathsf{ped}}) := \gamma_{\mathsf{dl}} + \gamma_{\mathsf{ped}}$. Due to the abort in Game 2, this change is purely conceptual, and thus

$$\varepsilon_2 = \varepsilon_3.$$

Game 4 (Use SHVZK simulation for $\mathbb{x}_{\mathsf{dl}}$). Replace the transcript $(A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}})$ by a SHVZK simulation for $\mathcal{O}_0$ (resp. $\mathcal{O}_1$). To do so, program H_{ch} with γ_{dl} which is sampled (hence known) at the start of the game. Additionally, we (can) move the whole randomization code of the user to the final step $\mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}_2$. Thus, $\mathcal{O}_0$ (resp. $\mathcal{O}_1$) will only sample and send γ^* in the modified $\mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}_1$ phase, and γ^* is uniform due to Game 3.

We argue indistinguishability as follows: By definition, the user computes its output $(A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}})$ as a randomization of the transcript $(A^*_{\mathsf{dl}}, \gamma^*_{\mathsf{dl}}, z^*_{\mathsf{dl}})$ (w.r.t. fixed statement $\mathbb{x}_{\mathsf{dl}}$). By randomizability of transcripts for Σ_{dl}, we know that the randomized distribution of $(A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}, z_{\mathsf{dl}})$ coincides with a SHVZK simulation for $\gamma_{\mathsf{dl}} \leftarrow \mathbb{Z}_p$. Overall, we find that

$$\varepsilon_3 = \varepsilon_4.$$

Observe that after this step, we do not use τ^*_{dl} anymore.

Game 5 (Use SHVZK simulation for $\mathbb{x}_{\mathsf{ped}}$ (and randomize $\mu_{\$}$)). This step is analogous to Game 4 above, except for $\mathbb{x}_{\mathsf{ped}}$. We replace the transcript $(\mu_{\$}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, \boldsymbol{z}_{\mathsf{ped}})$ by a SHVZK simulation for Σ_{ped} where

- $\mu_{\$} \leftarrow \mathbb{Z}_p^\times$ is sampled uniformly at random,
- $\mathbb{x}_{\mathsf{ped}} = (H, C, \mu_{\$})$ is defined with this randomly chosen $\mu_{\$} \leftarrow \mathbb{Z}_p^\times$, and
- $\tau_{\mathsf{ped}} = (A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, z_{\mathsf{ped}})$ now denotes the simulated transcript.

Now, we argue that these changes are indistinguishable. Firstly, $\mu_{\$} = \alpha' \cdot \mu^*_{\$}$ is distributed uniformly in $\mathbb{Z}_p^\times$ unless $\mu^*_{\$} = 0$. However, $\mu^*_{\$} = 0$ makes the user $\mathsf{BS}_{\mathsf{dl}}.\mathsf{BUser}_2$ abort, hence output $\sigma = \bot$ and $\mathcal{A}$ gets no information about $\mu_{\$}$ anyway. Hence, the distribution of $\mu_{\$}$ is unaffected (from $\mathcal{A}$'s view). Secondly, and analogous to Game 4 above, Σ_{ped} is SHVZK and randomizable, and $\tau_{\mathsf{ped}} = (a_{\mathsf{ped}}, \gamma_{\mathsf{ped}}, z_{\mathsf{ped}})$ is a randomization of $\tau'_{\mathsf{ped}}(a'_{\mathsf{ped}}, \gamma'_{\mathsf{ped}}, z'_{\mathsf{ped}})$. Note that by correctness of $\mathsf{BS}_{\mathsf{dl}}$ (cf. Theorem 5.1), if $\tau^*_{\mathsf{ped}} = (a^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}}, z^*_{\mathsf{ped}})$ verifies for $\mathbb{x}^*_{\mathsf{ped}} = (H, C^*, \mu_{\$})$, then so does τ'_{ped} for $\mathbb{x}'_{\mathsf{ped}}$. Hence, by randomizability of Σ_{ped}, this change is perfectly indistinguishable and we find that

$$\varepsilon_4 = \varepsilon_5.$$

After this step, we do not use $(\mu^*_{\$}, \tau^*_{\mathsf{ped}})$ anymore.

Game 6 (Commit to zero in C^*_μ). Finally, we replace the commitment of μ in C^*_μ with a commitment to 0. Since Pedersen commitments are perfectly hiding, we find

$$\varepsilon_5 = \varepsilon_6.$$

At this point, the game outputs signatures σ_0, σ_1 which are independent from the interaction with $\mathcal{A}$ (except for the decision to output $(\perp, \perp)$ instead). In particular, signature generation can be pushed after the finalizing call of $\mathcal{A}$ and $\mathcal{A}$'s view is completely independent of b. Clearly, the probability that $b = b'$ is $\frac{1}{2}$, and the adversary's distinguishing advantage is therefore 0.

By going these steps backward, we can switch the challenge bit $b = 0$ to $b = 1$ (at most doubling the adversary's advantage). □

5.3 One-more Unforgeability

We establish one-more unforgeability. We refer to Section 2 for an overview.

Theorem 5.3 (One-more Unforgeability). *Denote by p the order of $\mathbb{G}$. For any PPT adversary $\mathcal{A}$ that causes at most $Q_{\mathsf{ch}}, Q_\mu, Q_{\mathsf{par}}, Q_\Pi$ random oracle queries to $\mathsf{H}_{\mathsf{ch}}, \mathsf{H}_\mu, \mathsf{H}_{\mathsf{par}}, \mathsf{H}_\Pi$, respectively, in the game, and that starts at most Q_S signing sessions, there are reductions $\mathcal{B}_{\mathsf{crs}}$, $\mathcal{B}_{\mathsf{ext}}$ and $\mathcal{B}_{\mathsf{dl}}$ whose running time is roughly that of the OMUF game, such that*

$$\mathsf{AdvOMUF}^{\mathsf{BS}_{\mathsf{dl}}}_{\mathcal{A}}(\lambda) \leq \mathsf{AdvCRS}^{\Pi,\mathsf{ExtSetup}}_{\mathcal{B}_{\mathsf{stp}}}(\lambda) + \mathsf{AdvExt}^{\Pi_\mu,\widetilde{\mathsf{R}}_\mu}_{\mathcal{B}_{\mathsf{ext}}}(\lambda) + \mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{B}_{\mathsf{dl}}}(\lambda)$$
$$+ \frac{Q_\mu^2 + Q_{\mathsf{par}} + 1}{p} + Q_\mu \cdot Q_{\mathsf{par}} \cdot \left(\frac{Q_S + 1 + Q_{\mathsf{ch}}}{p} + \sqrt{Q_{\mathsf{ch}} \cdot \mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{B}_{\mathsf{dl}}}(\lambda)} \right).$$

Proof. Let $\mathcal{A}$ be a PPT adversary against one-more unforgeability of $\mathsf{BS}_{\mathsf{dl}}$. Let $\mathbb{G}$ be a group of prime order p with generator G. For random oracle $\mathsf{H}_{\mathsf{xyz}} \in \{\mathsf{H}_\mu, \mathsf{H}_{\mathsf{ch}}, \mathsf{H}_{\mathsf{par}}, \mathsf{H}_\Pi\}$, denote by Q_{xyz} an upper bound on the number of oracle queries to $\mathsf{H}_{\mathsf{xyz}}$ (as in Theorem 5.3). We use the convention that $\mathsf{H}_{\mathsf{xyz}}$ queries made by the game (*e.g.*, during signing queries or verification) count towards Q_{xyz}. Denote by Q_S an upper bound on number of $\mathcal{A}$'s signing queries.

We proceed with a sequence of games Game i and denote by ε_i the advantage of $\mathcal{A}$ in Game i (*i.e.*, the probability that Game i outputs 1). Our games follow the game structure in [39]; the main difference lies in the final reduction.

Game 0 (Real game). This game is the real one-more unforgeability game for scheme $\mathsf{BS}_{\mathsf{dl}}$. We recall the game below.

The game sets $\mathsf{crs}_{\mathsf{ped}} = (G, \mathsf{crs}_\mu)$ for random crs_μ, and samples $\mathsf{vk} := X = xG$ for $\mathsf{sk} := x \leftarrow \mathbb{Z}_p$ as in $\mathsf{BS}_{\mathsf{dl}}.\mathsf{KeyGen}$. Then, the game sends $\mathsf{crs}_{\mathsf{ped}}$ and vk to $\mathcal{A}$, and provides access to the random oracles $\mathsf{H}_\mu, \mathsf{H}_{\mathsf{ch}}, \mathsf{H}_{\mathsf{par}}, \mathsf{H}_\Pi$ and signing oracles $\mathcal{O}_{\mathsf{BSign}_1}, \mathcal{O}_{\mathsf{BSign}_2}$. In the end, $\mathcal{A}$ outputs a common message $\widehat{\tau}$ and forgeries $(\widehat{\mathsf{msg}}_j, \widehat{\sigma}_j)_{j \in [Q_{\mathsf{frg}}]}$. The game outputs 1 if and only if $\mathcal{O}_{\mathsf{BSign}_2}$ was queried at most $Q_{\mathsf{frg}} - 1$ times with common message $\widehat{\tau}$, all messages $\{\widehat{\mathsf{msg}}_j\}_{j \in [Q_{\mathsf{frg}}]}$ are pairwise-distinct, and all signatures verify. The signing oracles in session sid behave as follows:

– $\mathcal{O}_{\mathsf{BSign}_1}(\mathsf{sid}, \tau, C^*_\mu, \pi_\mu)$: The game sets $(H_\tau, C_\tau) \coloneqq \mathsf{H}_{\mathsf{par}}(\tau)$ and verifies the proof π_μ via $\Pi_\mu.\mathsf{Verify}^{\mathsf{H}_\Pi}(\mathsf{crs}_\mu, \mathbb{x}_\mu, \pi_\mu) = 1$ for $\mathbb{x}_\mu \coloneqq (H_\tau, C^*_\mu)$ and outputs $\perp$ if this check fails. Else, it sets $C^* \coloneqq C_\tau - C^*_\mu$ and samples $\mu^*_\$ \leftarrow \mathbb{Z}_p^\times$. Then it sets

$$\mathbb{x}^*_{\mathsf{ped}} \coloneqq (G, H_\tau, C^*, \mu^*_\$); \quad \mathbb{x}_{\mathsf{dl}} \coloneqq (G, X)$$

and sets up the transcripts for Σ_{ped} and Σ_{dl} as follows.
  - The game simulates the Σ_{ped} transcript via $(A^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}}) \leftarrow \mathsf{Sim}_{\mathsf{ped}}(\mathbb{x}^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}})$ for a random challenge $\gamma^*_{\mathsf{ped}} \leftarrow \mathbb{Z}_p$.
  - The game computes the Σ_{dl} transcript honestly via its witness $\mathsf{sk} = x$, *i.e.*, it computes $(A^*_{\mathsf{dl}}, \mathsf{st}_{\mathsf{dl}}) \leftarrow \mathsf{Init}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}}, \mathsf{sk})$.

  The game outputs $(\mu^*_\$, A^*_{\mathsf{ped}}, A^*_{\mathsf{dl}})$.
– $\mathcal{O}_{\mathsf{BSign}_2}(\mathsf{sid}, \gamma^*)$: The game retrieves $\gamma^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}}$ and $\mathsf{st}_{\mathsf{dl}}$ from the state for sid (and outputs $\perp$ if this is not possible). Then, the game sets

$$\gamma^*_{\mathsf{dl}} \coloneqq \gamma^* - \gamma^*_{\mathsf{ped}} \quad \text{and} \quad \boldsymbol{z}^*_{\mathsf{dl}} \leftarrow \mathsf{Resp}_{\mathsf{dl}}(\mathsf{st}_{\mathsf{dl}}, \gamma^*_{\mathsf{dl}})$$

  The game empties its state for sid and outputs $(\boldsymbol{z}^*_{\mathsf{ped}}, z^*_{\mathsf{dl}}, \gamma^*_{\mathsf{ped}})$.

By definition, we have

$$\mathsf{AdvOMUF}^{\mathsf{BS}_{\mathsf{dl}}}_{\mathcal{A}}(\lambda) = \varepsilon_0.$$

Game 1 (Ensure collision-free H_μ). The game aborts if a collision in H_μ occurs. A birthday bound yields

$$|\varepsilon_0 - \varepsilon_1| \leq \frac{Q_\mu^2}{p}.$$

Game 2 (Extract (μ, t) from π_μ). The game sets up crs_μ in extractable mode. In each session sid, the game extracts $\mathbb{w}_{\mathsf{sid}} \in \mathbb{Z}_p^2$ from π_μ in $\mathcal{O}_{\mathsf{BSign}_1}$ if π_μ verifies. Then, it parses $(\mu, t) = \mathbb{w}$ and aborts if $C^*_\mu \neq \mu G + tH_\tau$. In more detail, the game initially sets $(\mathsf{crs}_\mu, \mathsf{td}_\mu) \leftarrow \mathsf{ExtSetup}(1^\lambda)$ and sets $\mathsf{crs}_{\mathsf{ped}} = (G, \mathsf{crs}_\mu)$ which it forwards to $\mathcal{A}$ together with vk sampled as in the previous game. In $\mathcal{O}_{\mathsf{BSign}_1}$, if $\mathsf{Verify}(\mathbb{x}_\mu, \pi_\mu) = 1$ it sets $\mathbb{w} \leftarrow \mathsf{Ext}(\mathsf{td}_\mu, \mathcal{Q}, \mathbb{x}_\mu, \pi_\mu)$, where $\mathcal{Q}$ denote the queries to H_Π. The game aborts its entire execution if $C^*_\mu \neq \mu G + tH_\tau$. Otherwise, it proceeds as before.

We must bound the abort probability, *i.e.*, we must show that indeed $C^*_\mu = \mu G + tH_\tau$ with high probability. This can be shown as follows. Recall that Π_μ is $\widetilde{\mathsf{R}}_\mu$-extractable. It is easy to see that there exists an adversary $\mathcal{B}_{\mathsf{ext}}$ such that $(\mathbb{x}_\mu, \mathbb{w}) \in \widetilde{\mathsf{R}}_\mu$ except with probability $\mathsf{AdvExt}^{\Pi_\mu, \widetilde{\mathsf{R}}_\mu}_{\mathcal{B}_{\mathsf{ext}}}(\lambda)$. If $(\mathbb{x}_\mu, \mathbb{w}) \in \widetilde{\mathsf{R}}_\mu$, it holds that $\mu G = H_\tau$ or $C^*_\mu = \mu G + tH_\tau$ by definition. Under the DL assumption, the latter must hold as desired.

Formally, the reduction to DL obtains a challenge $Y \in \mathbb{G}$. Observe that H_τ is sampled uniform via $\mathsf{H}_{\mathsf{par}}$ in G_2 and G_1. The reduction simulates G_2 to $\mathcal{A}$, except for the added abort condition and that on the i-th fresh query to $\mathsf{H}_{\mathsf{par}}$, it outputs (H_τ, C_τ), where $C_\tau \leftarrow \mathbb{G}$ and $H_\tau \coloneqq s_i Y$ for some $s_i \leftarrow \mathbb{Z}_p$. If $Y \neq 0$ (which happens except with probability $1/p$) and if $C^*_\mu \neq \mu G + tH_\tau$, then the

reduction outputs $y \coloneqq s_i^{-1}\mu$ as its DL-solution, where i corresponds to the first $\mathsf{H}_{\mathsf{par}}$ query on input τ. Since it holds that $\mu G = H_\tau = s_i Y$, if $s_i \neq 0$, we have that $yG = Y$. Note that $s_i \neq 0$ for all i except with probability Q_{par}/p. In total, there are reductions $\mathcal{B}_{\mathsf{crs}}$, $\mathcal{B}_{\mathsf{ext}}$ and $\mathcal{B}_{\mathsf{dl}}$ such that

$$|\varepsilon_1 - \varepsilon_2| \leq \mathsf{AdvCRS}^{\Pi,\mathsf{ExtSetup}}_{\mathcal{B}_{\mathsf{stp}}}(\lambda) + \mathsf{AdvExt}^{\Pi_\mu,\widetilde{\mathsf{R}}_\mu}_{\mathcal{B}_{\mathsf{ext}}}(\lambda) + \mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{B}_{\mathsf{dl}}}(\lambda) + \frac{Q_{\mathsf{par}}+1}{p}.$$

Game 3 (Guess forgery's common message $\widehat{\tau}$). The game samples $i_{\tau,\mathcal{A}} \leftarrow [Q_{\mathsf{par}}]$ at its start. When $\mathcal{A}$ outputs its forgeries for common message $\widehat{\tau}$, the game aborts its entire execution if $\widehat{\tau}$ was *not* queried to $\mathsf{H}_{\mathsf{par}}$ on the $i_{\tau,\mathcal{A}}$-th query for the first time.

Since such a query must exist as the game verifies $\mathcal{A}$'s forgeries—which induces an $\mathsf{H}_{\mathsf{par}}$ query with input $\widehat{\tau}$—and the game's $\mathsf{H}_{\mathsf{par}}$ queries count towards Q_{par}. As the guess is hidden from $\mathcal{A}$, we have

$$\varepsilon_2 \leq Q_{\mathsf{par}} \cdot \varepsilon_3.$$

Game 4 (Guess non-completed $\widehat{\mu}$ among forgeries). The game samples $i_{\mu,\mathcal{A}} \leftarrow [Q_\mu]$ and $\widehat{\mu} \leftarrow \mathbb{Z}_p$ at its start. On the $i_{\mu,\mathcal{A}}$-th query to H_μ on input $\mathsf{in}_{i_{\mu,\mathcal{A}}}$, the game outputs $\widehat{\mu}$. When $\mathcal{A}$ outputs its forgeries for messages $(\widehat{\mathsf{msg}}_j)_{j\in[Q_{\mathsf{frg}}]}$ and common message $\widehat{\tau}$, the game aborts its entire execution if

- for all $j \in [Q_{\mathsf{frg}}]$ it holds that $\widehat{\mu} \neq \widehat{\mu}_j$ for $\widehat{\mu}_j = \mathsf{H}_\mu(\widehat{\mathsf{msg}}_j)$; or
- a signing session with common message $\widehat{\tau}$ is completed, where $\widehat{\mu} = \mathsf{H}_\mu(\mathsf{in}_{i_{\mu,\mathcal{A}}})$ is extracted from π_μ.

Let us bound the abort probability. Denote by $\mathcal{M}_{\widehat{\tau}} \subseteq \mathbb{Z}_p$ the set of messages extracted in $\mathcal{O}_{\mathsf{BSign}_1}$ with common message $\widehat{\tau}$ such that $\mathcal{O}_{\mathsf{BSign}_2}$ is *completed*. Since at most $Q_{\mathsf{frg}} - 1$ signing session with $\widehat{\tau}$ are completed, we have $|\mathcal{M}_{\widehat{\tau}}| \leq Q_{\mathsf{frg}} - 1$. Denote by $\mathcal{M}_{\mathcal{A}} = \{\widehat{\mu}_j\}_{j\in[Q_{\mathsf{frg}}]}$ the set of all hashed messages from $\mathcal{A}$'s forgeries. Since H_μ is collision-free due to G_1 and there are Q_{frg} pairwise-distinct msg_j, we have $|\mathcal{M}_{\mathcal{A}}| \geq Q_{\mathsf{frg}}$. Thus, there exists some $j \in [Q_{\mathsf{frg}}]$ such that $\widehat{\mu}_j \in \mathcal{M}_{\mathcal{A}} \setminus \mathcal{M}_{\widehat{\tau}}$. The probability that the first H_μ query with input msg_j is the $i_{\mu,\mathcal{A}}$-th H_μ query is $1/Q_\mu$. Therefore, we have

$$\varepsilon_3 \leq Q_\mu \cdot \varepsilon_4.$$

Game 5 (Setup C_τ with known openings). The game changes how it samples C_τ in oracle $\mathsf{H}_{\mathsf{par}}$ depending on whether the query corresponds to the forgery's common message $\widehat{\tau}$ or not. In more detail, the game samples μ^* in the beginning and aborts its entire execution if μ^* is extracted in *any* $\mathcal{O}_{\mathsf{BSign}_1}$ query. Further, the game answers $\mathsf{H}_{\mathsf{par}}$ queries as follows.

- The game outputs $(H_{\widehat{\tau}}, C_{\widehat{\tau}})$ on the $i_{\tau,\mathcal{A}}$-th query to $\mathsf{H}_{\mathsf{par}}$, where $H_{\widehat{\tau}} \leftarrow \mathbb{G}$ and $C_{\widehat{\tau}} \coloneqq \widehat{\mu}G + t_{\widehat{\tau}} H_{\widehat{\tau}}$. That is, the game commits to $\widehat{\mu}$ in $C_{\widehat{\tau}}$ and it holds that $\widehat{\mu}G - C_{\widehat{\tau}}$ is a commitment to 0. Note that this query corresponds to the first $\mathsf{H}_{\mathsf{par}}$ query for the forgery's common message $\widehat{\tau}$ (cf. G_3).

– The game outputs (H_τ, C_τ) on other fresh $\mathsf{H}_{\mathsf{par}}$ queries (*i.e.*, on input $\tau \neq \widehat{\tau}$), where $H_\tau \leftarrow \mathbb{G}$ and $C_\tau := \mu^* G + t_\tau H_\tau$ for $t_\tau \leftarrow \mathbb{Z}_p$. As μ^* is never extracted in $\mathcal{O}_{\mathsf{BSign}_1}$, it holds that $C_\tau - C^*_\mu$ is a commitment to $\mu^* - \mu \neq 0$, where μ is the message extracted from π_μ.

Observe that $\mathsf{H}_{\mathsf{par}}$ outputs are distributed as in G_4. Since μ^* remains information-theoretically hidden, a union bound yields

$$|\varepsilon_4 - \varepsilon_5| \leq \frac{Q_S}{p}.$$

Game 6 (Compute Σ_{ped}-transcripts via witness). The game samples $\mu^*_{\$}$ and the Σ_{ped} transcripts via the witness embedded in G_5. That is, it sets up $\mu^*_{\$}$ and the Σ_{ped} transcripts $(A^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}})$ in the signing oracles as follows. All other values are sampled as in G_5.

$\mathcal{O}_{\mathsf{BSign}_1}$**:** If extracted message $\mu = \widehat{\mu}$ and common message $\tau = \widehat{\tau}$, then the game proceeds as before, *i.e.*, it samples $\mu^*_{\$} \leftarrow \mathbb{Z}_p^\times$ and sets $(A^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}}) \leftarrow \mathsf{Sim}_{\mathsf{ped}}(\mathbb{x}^*_{\mathsf{ped}})$ for $\mathbb{x}^*_{\mathsf{ped}} = (G, H_\tau, C^*, \mu^*_{\$})$ with $C^* = C_\tau - C^*_\mu$.
Else, if $\mu \neq \widehat{\mu}$ or $\tau \neq \widehat{\tau}$, then the game samples $\alpha \leftarrow \mathbb{Z}_p^\times$ and sets $\Delta\mu := \mu^* - \mu$ if $\tau \neq \widehat{\tau}$ or else, $\Delta\mu := \widehat{\mu} - \mu$. Similarly, it sets $\Delta t = t_\tau - t$. Then, the game sets

$$\mu^*_{\$} := \alpha \cdot \Delta\mu; \qquad \mathbb{w}^*_{\mathsf{ped}} = (-\alpha \cdot \Delta t, \alpha).$$

Instead of simulating the Σ_{ped} transcript as in G_5, the game sets $(A^*_{\mathsf{ped}}, \mathsf{st}_{\mathsf{ped}}) \leftarrow \mathsf{Init}_{\mathsf{ped}}(\mathbb{x}^*_{\mathsf{ped}}, \mathbb{w}^*_{\mathsf{ped}})$.

$\mathcal{O}_{\mathsf{BSign}_2}$**:** Due to the abort condition added in G_4 which ensures that $\widehat{\mu}$-sessions[10] are not completed if $\tau = \widehat{\tau}$, it must hold that either $\mu \neq \widehat{\mu}$ or $\tau \neq \widehat{\tau}$. The game therefore samples $\gamma^*_{\mathsf{ped}} \leftarrow \mathbb{Z}_p$ and sets $\boldsymbol{z}^*_{\mathsf{ped}} \leftarrow \mathsf{Resp}_{\mathsf{ped}}(\mathsf{st}_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}})$.

Let us show that both games G_6 and G_5 are identically distributed. First, observe that G_6 and G_5 differ only if $\mu \neq \widehat{\mu}$ or $\tau \neq \widehat{\tau}$. We therefore analyze the distribution of $\mu^*_{\$}$ and issued Σ_{ped} transcripts $(A^*_{\mathsf{ped}}, \gamma^*_{\mathsf{ped}}, \boldsymbol{z}^*_{\mathsf{ped}})$ in such signing sessions. First, we show that $(\mathbb{x}^*_{\mathsf{ped}}, \mathbb{w}^*_{\mathsf{ped}}) \in \mathsf{R}_{\mathsf{ped}}$. Let us recap some facts due to prior games. We know that

– $C_{\widehat{\tau}} = \widehat{\mu} G + t_{\widehat{\tau}} H_{\widehat{\tau}}$ for $\tau = \widehat{\tau}$ (cf. G_5);
– $C_\tau = \mu^* G + t_\tau H_\tau$ for $\tau \neq \widehat{\tau}$ (cf. G_5);
– $C^*_\mu = \mu G + t H_\tau$ (cf. G_2).

Furthermore, we have that $\Delta\mu \neq 0$ if $\tau = \widehat{\tau}$ and $\mu \neq \widehat{\mu}$ (cf. G_4); and that $\Delta\mu \neq 0$ if $\tau \neq \widehat{\tau}$ as $\mu \neq \mu^*$ (cf. G_5). Therefore, we have that

$$C^* = C_\tau - C^*_\mu = \Delta\mu G + \Delta t H_\tau$$

with $\Delta\mu \neq 0$. By definition of $\mathsf{R}_{\mathsf{ped}}$, we have $(\mathbb{x}^*_{\mathsf{ped}}, \mathbb{w}^*_{\mathsf{ped}}) \in \mathsf{R}_{\mathsf{ped}}$ (cf. Eq. (4.2) and discussion below). Under perfect SHVZK of Σ_{ped}, the distributions of the

[10] With $\widehat{\mu}$-sessions, we refer to sessions in which $\widehat{\mu}$ was extracted in $\mathcal{O}_{\mathsf{BSign}_1}$.

Σ_{ped} transcripts in G_6 and G_5 is identical. Further, observe that as $\alpha \leftarrow \mathbb{Z}_p^\times$ and $\Delta\mu \neq 0$, the value $\mu^*_{\$}$ is also distributed identically in G_6 and G_5. In conclusion, we have

$$\varepsilon_5 = \varepsilon_6.$$

Game 7 (Simulate Σ_{dl}-transcripts). The game simulates all Σ_{dl} transcripts in the signing sessions. That is, it sets up the Σ_{dl} transcripts $(A^*_{\mathsf{dl}}, \gamma^*_{\mathsf{dl}}, z_{\mathsf{dl}})$ as follows.

$\mathcal{O}_{\mathsf{BSign}_1}$**:** The game sets $(A^*_{\mathsf{dl}}, \gamma^*_{\mathsf{dl}}, z^*_{\mathsf{dl}}) \leftarrow \mathsf{Sim}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}})$ instead of computing A^*_{dl} via $\mathsf{Init}_{\mathsf{dl}}$.

$\mathcal{O}_{\mathsf{BSign}_2}$**:** The game sets $\gamma^*_{\mathsf{ped}} \coloneqq \gamma^* - \gamma^*_{\mathsf{dl}}$ and employs the response z^*_{dl} sampled in $\mathcal{O}_{\mathsf{BSign}_1}$.

The distributions of γ^*_{ped} and γ^*_{dl} remain unchanged except if $\mu = \widehat{\mu}$ and $\tau = \widehat{\tau}$. In the latter case, the challenges are not revealed as such signing sessions are not finished (cf. G_4). Therefore, the distribution is identical. Furthermore, the commitment-and-response pairs $(A^*_{\mathsf{dl}}, z^*_{\mathsf{dl}})$ are distributed identically under perfect HVZK. Consequently, we have

$$\varepsilon_6 = \varepsilon_7.$$

Summary of G_7: This concludes the game hop sequence. We will next rewind the adversary in G_7 to reduce to DL. Before, let us recap the final game G_7. The signing key $\mathsf{sk} = x$ and verification key vk are setup as before, *i.e.*, $x \leftarrow \mathbb{Z}_p$ and $\mathsf{vk} = xG$. The game samples $\mu^* \leftarrow \mathbb{Z}_p$. Also, the game sets up its responses $\widehat{\mu} \leftarrow \mathbb{Z}_p$ and $(H_{\widehat{\tau}}, C_{\widehat{\tau}})$ for the $i_{\mu,\mathcal{A}}$-th and $i_{\tau,\mathcal{A}}$-th query to H_μ and $\mathsf{H}_{\mathsf{par}}$, respectively, where $H_{\widehat{\tau}} \leftarrow \mathbb{G}$ and $C_{\widehat{\tau}} \coloneqq \widehat{\mu}G + t_{\widehat{\tau}}H_{\widehat{\tau}}$ and $t_{\widehat{\tau}} \leftarrow \mathbb{Z}_p$ (cf. G_4 and G_5). Note that this happens at the beginning of the game *before* the adversary is ran. This is important for consistency when rewinding. The game then sends $\mathsf{crs}_{\mathsf{ped}} = (G, \mathsf{crs}_\mu)$ and vk to $\mathcal{A}$, where $(\mathsf{crs}_\mu, \mathsf{td}_\mu) \leftarrow \mathsf{ExtSetup}(1^\lambda)$ is setup in extractable mode (cf. G_2). The oracles Π_μ and H_{ch} oracles are simulated as in G_0. The oracles $\mathsf{H}_{\mathsf{par}}$ and H_μ are simulated as follows.

- $\mathsf{H}_{\mathsf{par}}(\tau)$: On the $i_{\tau,\mathcal{A}}$-th query (*i.e.*, the first query such that $\tau = \widehat{\tau}$), the game outputs $(H_{\widehat{\tau}}, C_{\widehat{\tau}})$ prepared in advance. Else, on a fresh $\tau \neq \widehat{\tau}$ query, the game outputs (H_τ, C_τ), where $H_\tau \leftarrow \mathbb{G}$ and $C_\tau \coloneqq \mu^*G + t_\tau H_\tau$ for $t_\tau \leftarrow \mathbb{Z}_p$ (cf. G_5). If τ was already queried, the game outputs consistent tuples.
- $\mathsf{H}_\mu(\mathsf{msg})$: On the $i_{\mu,\mathcal{A}}$-th query, the game outputs $\widehat{\mu}$ prepared in advance. Else, it outputs $\mu \leftarrow \mathbb{Z}_p$ for fresh queries, and answers repeated queries consistently (cf. G_4).

The signing oracles in session sid behave as follows.

- $\mathcal{O}_{\mathsf{BSign}_1}(\mathsf{sid}, \tau, C^*_\mu, \pi_\mu)$: The game sets $(H_\tau, C_\tau) \coloneqq \mathsf{H}_{\mathsf{par}}(\tau)$ and verifies the proof π_μ via $\Pi_\mu.\mathsf{Verify}^{\mathsf{H}_\Pi}(\mathsf{crs}_\mu, \mathbb{x}_\mu, \pi_\mu) = 1$ for $\mathbb{x}_\mu \coloneqq (H_\tau, C^*_\mu)$ and outputs $\perp$ if this check fails. Else, the game extracts $(\mu, t) \leftarrow \mathsf{Ext}(\mathsf{td}_\mu, \mathcal{Q}, \mathbb{x}_\mu, \pi_\mu)$ and aborts its entire execution if $C^*_\mu \neq \mu G + tH_\tau$ (cf. G_2). Then, it proceeds depending on common message τ and the extracted message μ as follows.

Case $\mu \neq \widehat{\mu}$ or $\tau \neq \widehat{\tau}$: The game computes the Σ_{ped} transcript honestly and simulates the Σ_{dl} transcript as described in G_6 and G_7, respectively. That is, it computes A^*_{ped} via $\mathsf{Init}_{\mathsf{ped}}$ and an appropriate witness and it simulates $(A^*_{\mathsf{dl}}, \gamma^*_{\mathsf{dl}}, z^*_{\mathsf{dl}}) \leftarrow \mathsf{Sim}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}})$.

Case $\mu = \widehat{\mu}$ and $\tau = \widehat{\tau}$: The game simulates both transcripts Σ_{dl} and Σ_{ped}. The game outputs $(\mu^*_{\$}, \boldsymbol{z}^*_{\mathsf{ped}}, A^*_{\mathsf{dl}})$.

– $\mathcal{O}_{\mathsf{BSign}_2}(\mathsf{sid}, \gamma^*)$: The game proceeds depending on common message τ and the extracted message μ in $\mathcal{O}_{\mathsf{BSign}_1}$ as follows.

Case $\mu \neq \widehat{\mu}$ or $\tau \neq \widehat{\tau}$: The game sets $\gamma^*_{\mathsf{ped}} := \gamma^* - \gamma^*_{\mathsf{dl}}$ and computes the Σ_{ped} response $\boldsymbol{z}^*_{\mathsf{ped}}$ honestly via $\mathsf{Resp}_{\mathsf{ped}}$.

Case $\mu = \widehat{\mu}$ and $\tau = \widehat{\tau}$: This case cannot occur due to the abort condition added in G_4.

The game outputs $(\boldsymbol{z}^*_{\mathsf{ped}}, z^*_{\mathsf{dl}}, \gamma^*_{\mathsf{ped}})$, where z^*_{dl} is the simulated response.

Note that due to the abort condition in added in G_4, the adversary outputs forgeries such that $\widehat{\mu} = \mathsf{H}_{\mu}(\mathsf{msg}_j)$ for some $j \in [Q_{\mathsf{frg}}]$. Below, we denote by $\sigma := \sigma_j$ the signature associated to msg_j. We parse $\sigma = (\mu_{\$}, \pi)$ and $\pi = (A_{\mathsf{ped}}, A_{\mathsf{dl}}, \gamma_{\mathsf{ped}}, \gamma_{\mathsf{dl}}, \boldsymbol{z}_{\mathsf{ped}}, z_{\mathsf{dl}})$. Also, denote $\mathbb{x}_{\mathsf{ped}} = (G, H_{\widehat{\tau}}, C, \mu_{\$})$, where $C = C_{\widehat{\tau}} - \widehat{\mu} G$ and $\mathbb{x}_{\mathsf{dl}} = (G, X)$. Also, notice that $\mathsf{sk} = x$ is not required for simulation anymore.

Reduction to DL: In brief, the reduction embeds a DL challenge into X and H_{τ} and rewinds the adversary in G_7 to obtain either the discrete logarithm of X or a non-zero opening for the commitment C associated to (hashed) message $\widehat{\mu}$. As the reduction knows a zero opening for C by construction, the reduction can derive a DL solution in both cases.

In more detail, let us define a wrapper $\mathcal{W}$ that is defined as follows.

$\mathcal{W}(X, \boldsymbol{h})$**:** On input $X \in \mathbb{G} \setminus \{0\}$ and $\boldsymbol{h} = (h_1, \ldots, h_\ell) \in \mathbb{Z}_p^\ell$, where $\ell \leq Q_{\mathsf{ch}}$ denotes the number of *fresh* H_{ch} queries, the wrapper $\mathcal{W}$ simulates G_7 to $\mathcal{A}$ as described above, except for the following changes:

– The wrapper sets $\mathsf{vk} = X$ and $\mathsf{sk} = \bot$.

– The wrapper sets $H_{\widehat{\tau}} \leftarrow s_{\widehat{\tau}} X$ for $s_{\widehat{\tau}} \leftarrow \mathbb{Z}_p$.

– On the i-th fresh H_{ch} query, the wrapper outputs h_i.

If $X = 0$ or the adversary does not win, the wrapper outputs $(\bot, \bot)$.[11] Otherwise, let I denote the index of the first time the tuple $\mathsf{in}_h := (\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, A_{\mathsf{dl}})$ is queried to H_{ch}.

Note that $\pi = (A_{\mathsf{ped}}, A_{\mathsf{dl}}, \gamma_{\mathsf{ped}}, \gamma_{\mathsf{dl}}, \boldsymbol{z}_{\mathsf{ped}}, z_{\mathsf{dl}})$ is defined as above. Therefore, this is the hash query associated to $\mathcal{A}$'s forgery on message $\widehat{\mu}$. Also, note that I is well-defined as the query in_h to H_{ch} is made when σ is verified when the forgery checks are made. Also, note that $h_I = \mathsf{H}_{\mathsf{ch}}(\mathbb{x}_{\mathsf{dl}}, \mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, A_{\mathsf{dl}})$.

Output $(I, (\pi, h_I))$.

Note that if $X \neq 0$, then the wrapper simulates G_7 perfectly in the view of $\mathcal{A}$. If $X = 0$, the reduction knows the DL solution (namely 0), and for completeness,

[11] The adversary $\mathcal{A}$ succeeds if in wrapper $\mathcal{W}$'s simulation of G_7, the adversary $\mathcal{A}$ does not trigger an abort condition and all final checks on $\mathcal{A}$'s forgeries pass.

we let $\mathcal{W}(0, \boldsymbol{h})$ return $(1, 0)$. Therefore, we have

$$\begin{aligned}\mathsf{acc} &\coloneqq \Pr[X \leftarrow \mathbb{G}, \boldsymbol{h} \leftarrow \mathbb{Z}_p^\ell, (I, (\pi, h_I)) \leftarrow \mathcal{W}(X, \boldsymbol{h}) : I \geq 1] \\ &\geq \Pr[(\mathcal{A} \text{ wins in } \mathsf{G}_7 \wedge X \neq 0) \vee X = 0] \geq \varepsilon_7. \end{aligned} \tag{5.1}$$

The reduction $\mathcal{B}_{\mathsf{dl}}$ to DL proceeds as follows. Given $(G, X) \in \mathbb{G}^2$, the reduction runs the wrapper algorithm twice: An initial run and once again via rewinding on the I-th H_{ch} query. Formally, it runs $(b, (\pi_1, h_{I_1}), (\pi_2, h_{I_2})) \leftarrow \mathsf{Fork}_{\mathcal{W}}(X)$ (cf. Fig. 1). If $b = 0$, the reduction outputs $\perp$. If $X = 0$, the reduction returns 0. Else, it parses $\pi_i = (A_{\mathsf{ped}}^{(i)}, A_{\mathsf{dl}}^{(i)}, \gamma_{\mathsf{ped}}^{(i)}, \gamma_{\mathsf{dl}}^{(i)}, \boldsymbol{z}_{\mathsf{ped}}^{(i)}, z_{\mathsf{dl}}^{(i)})$ for $i \in \{1, 2\}$. If $b \neq 0$, we have by definition of $\mathsf{Fork}_{\mathcal{W}}$ that

- $h_I^{(1)} \neq h_I^{(2)}$ except with probability $1/p$; and
- $(\mu_{\$}^{(1)}, A_{\mathsf{ped}}^{(1)}, A_{\mathsf{dl}}^{(1)}) = (\mu_{\$}^{(2)}, A_{\mathsf{ped}}^{(2)}, A_{\mathsf{dl}}^{(2)})$.

Below, we assume that indeed $h_I^{(1)} \neq h_I^{(2)}$, else $\mathcal{B}_{\mathsf{dl}}$ outputs $\perp$. Denote by $\mu_{\$} \coloneqq \mu_{\$}^{(1)}, A_{\mathsf{ped}} \coloneqq A_{\mathsf{ped}}^{(1)}, A_{\mathsf{dl}} \coloneqq A_{\mathsf{dl}}^{(1)}$. Note that $\mu_{\$} \neq 0$. Furthermore, we have for $i \in \{1, 2\}$ and $C = C_{\widehat{\tau}} - \widehat{\mu} G$ that

$$\begin{aligned}&\mathsf{Verify}_{\mathsf{ped}}(\mathbb{x}_{\mathsf{ped}}, A_{\mathsf{ped}}, \gamma_{\mathsf{ped}}^{(i)}, \boldsymbol{z}_{\mathsf{ped}}^{(i)}) = 1, \text{ and} \\ &\mathsf{Verify}_{\mathsf{dl}}(\mathbb{x}_{\mathsf{dl}}, A_{\mathsf{dl}}, \gamma_{\mathsf{dl}}^{(i)}, z_{\mathsf{dl}}^{(i)}) = 1,\end{aligned}$$

where $\mathbb{x}_{\mathsf{dl}} = X$ and $\mathbb{x}_{\mathsf{ped}} = (G, H_{\widehat{\tau}}, C, \mu_{\$})$. Also, we know that $h_I^{(i)} = \gamma_{\mathsf{ped}}^{(i)} + \gamma_{\mathsf{dl}}^{(i)}$. As $h_I^{(1)} \neq h_I^{(2)}$, we have that

$$\gamma_{\mathsf{ped}}^{(1)} \neq \gamma_{\mathsf{ped}}^{(2)} \quad \text{or} \quad \gamma_{\mathsf{dl}}^{(1)} \neq \gamma_{\mathsf{dl}}^{(2)}.$$

In the latter case, we immediately obtain a witness $\mathbb{w}_{\mathsf{dl}}$ such that $(\mathbb{x}_{\mathsf{dl}}, \mathbb{w}_{\mathsf{dl}}) \in \mathsf{R}_{\mathsf{dl}}$ by 2-special soundness of Σ_{dl}. The reduction $\mathcal{B}_{\mathsf{dl}}$ outputs $x \coloneqq \mathbb{w}_{\mathsf{dl}}$.

Otherwise, we obtain a witness $\mathbb{w}_{\mathsf{ped}} = (r, \alpha)$ such that $(\mathbb{x}_{\mathsf{ped}}, \mathbb{w}_{\mathsf{ped}}) \in \mathsf{R}_{\mathsf{ped}}$ by 2-special soundness of Σ_{ped}. Recall that $C_{\widehat{\tau}} = \widehat{\mu} G + t_{\widehat{\tau}} H_{\widehat{\tau}}$ by construction. Therefore, we have

$$C = C_{\widehat{\tau}} - \widehat{\mu} G = t_{\widehat{\tau}} H_{\widehat{\tau}}.$$

We stress that this holds in both the initial and rewound run, since $C_{\widehat{\tau}}$ and $\widehat{\mu}$ is sampled before rewinding occurs. Further, recall that $H_{\widehat{\tau}} = s_{\widehat{\tau}} X$ (which is also sampled in advance, before rewinding occurs). By definition of $\mathsf{R}_{\mathsf{ped}}$ (cf. Eq. (4.1)), we obtain

$$\begin{aligned}\mu_{\$} G &= \alpha C + r H_{\widehat{\tau}} \\ &= \alpha (t_{\widehat{\tau}} H_{\widehat{\tau}}) + r H_{\widehat{\tau}} \\ &= \alpha (t_{\widehat{\tau}} (s_{\widehat{\tau}} X)) + r (s_{\widehat{\tau}} X) \\ &= s_{\widehat{\tau}} \cdot (\alpha t_{\widehat{\tau}} + r) X.\end{aligned}$$

Finally, as $\mu_{\$} \neq 0$, both sides are non-zero. The reduction outputs $x \coloneqq \mu_{\$} \cdot (s_{\widehat{\tau}} \cdot (\alpha t_{\widehat{\tau}} + r))^{-1}$.

By construction, we have $xG = X$ if $\mathcal{B}_{\mathsf{dl}}$ outputs some non-$\perp$ value x, *i.e.*, if $b \neq 0$ and $h_I^{(1)} \neq h_I^{(2)}$ (which is implied by $b = 1$). Therefore, we have

$$\mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{B}_{\mathsf{dl}}}(\lambda) = \Pr[x \leftarrow \mathcal{B}_{\mathsf{dl}}(G, X) : X = xG] \geq \mathsf{frk}.$$

where $\mathsf{frk} \coloneqq \Pr\left[(b, (\pi_1, h_{I_1}), (\pi_2, h_{I_2})) \leftarrow \mathsf{Fork}_{\mathcal{W}}(X) \ : \ b = 1\right]$. By Lemma 3.1 and Eq. (5.1), we have

$$\varepsilon_7 \leq \mathsf{acc} \leq \ell/p + \sqrt{\ell \cdot \mathsf{frk}} \leq \ell/p + \sqrt{\ell \cdot \mathsf{AdvDL}^{\mathbb{G},G}_{\mathcal{B}_{\mathsf{dl}}}(\lambda)}.$$

Since $\ell \leq Q_{\mathsf{ch}}$, collecting the bounds completes the proof. □

Acknowledgements

The research of Michael Klooß is supported by KASTEL Security Research Labs. The research of Russell W. F. Lai is supported by the Research Council of Finland grant 358951.

References

[1] Masayuki Abe. "A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures". In: *EUROCRYPT 2001*. LNCS. Springer, Berlin, Heidelberg, 2001. DOI: 10.1007/3-540-44987-6_9.

[2] Masayuki Abe and Eiichiro Fujisaki. "How to Date Blind Signatures". In: *ASIACRYPT'96*. LNCS. Springer, Berlin, Heidelberg, 1996. DOI: 10.1007/BFb0034851.

[3] Masayuki Abe and Tatsuaki Okamoto. "Provably Secure Partially Blind Signatures". In: *CRYPTO 2000*. LNCS. Springer, Berlin, Heidelberg, 2000. DOI: 10.1007/3-540-44598-6_17.

[4] Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. "BlindOR: an Efficient Lattice-Based Blind Signature Scheme from OR-Proofs". In: *CANS 21*. LNCS. Springer, Cham, 2021. DOI: 10.1007/978-3-030-92548-2_6.

[5] Mihir Bellare and Gregory Neven. "Multi-signatures in the plain public-Key model and a general forking lemma". In: *ACM CCS 2006*. ACM Press, 2006. DOI: 10.1145/1180405.1180453.

[6] Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. "On the (in)security of ROS". In: *EUROCRYPT 2021, Part I*. LNCS. Springer, Cham, 2021. DOI: 10.1007/978-3-030-77870-5_2.

[7] Bert den Boer. "Diffie-Hellman is as Strong as Discrete Log for Certain Primes (Rump Session)". In: *CRYPTO'88*. LNCS. Springer, New York, 1990. DOI: 10.1007/0-387-34799-2_38.

[8] Maiara F. Bollauf, Roberto Parisella, and Janno Siim. "Revisiting Discrete Logarithm Reductions". In: *CiC* 2 (2025). DOI: 10.62056/a0c3c3c2h.

[9] Dan Boneh and Richard J. Lipton. "Algorithms for Black-Box Fields and their Application to Cryptography (Extended Abstract)". In: *CRYPTO'96*. LNCS. Springer, Berlin, Heidelberg, 1996. DOI: 10.1007/3-540-68697-5_22.

[10] Stefan Brands. "Untraceable Off-line Cash in Wallets with Observers (Extended Abstract)". In: *CRYPTO'93*. LNCS. Springer, Berlin, Heidelberg, 1994. DOI: 10.1007/3-540-48329-2_26.

[11] Nicholas Brandt, Dennis Hofheinz, Michael Klooß, and Michael Reichle. "Tightly-Secure Blind Signatures in Pairing-Free Groups". In: *ASIACRYPT 2025, Part VI*. LNCS. Springer, Singapore, 2025. DOI: 10.1007/978-981-95-5119-4_11.

[12] Jan Camenisch and Anna Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *EUROCRYPT 2001*. LNCS. Springer, Berlin, Heidelberg, 2001. DOI: 10.1007/3-540-44987-6_7.

[13] Rutchathon Chairattana-Apirom, Lucjan Hanzlik, Julian Loss, Anna Lysyanskaya, and Benedikt Wagner. "PI-Cut-Choo and Friends: Compact Blind Signatures via Parallel Instance Cut-and-Choose and More". In: *CRYPTO 2022, Part III*. LNCS. Springer, Cham, 2022. DOI: 10.1007/978-3-031-15982-4_1.

[14] Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. "Pairing-Free Blind Signatures from CDH Assumptions". In: *CRYPTO 2024, Part I*. LNCS. Springer, Cham, 2024. DOI: 10.1007/978-3-031-68376-3_6.

[15] David Chaum. "Blind Signatures for Untraceable Payments". In: *CRYPTO'82*. Plenum Press, New York, USA, 1982. DOI: 10.1007/978-1-4757-0602-4_18.

[16] David Chaum. "Elections with Unconditionally-Secret Ballots and Disruption Equivalent to Breaking RSA". In: *EUROCRYPT'88*. LNCS. Springer, Berlin, Heidelberg, 1988. DOI: 10.1007/3-540-45961-8_15.

[17] David Chaum, Amos Fiat, and Moni Naor. "Untraceable Electronic Cash". In: *CRYPTO'88*. LNCS. Springer, New York, 1990. DOI: 10.1007/0-387-34799-2_25.

[18] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO'94*. LNCS. Springer, Berlin, Heidelberg, 1994. DOI: 10.1007/3-540-48658-5_19.

[19] Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. "Snowblind: A Threshold Blind Signature in Pairing-Free Groups". In: *CRYPTO 2023, Part I*. LNCS. Springer, Cham, 2023. DOI: 10.1007/978-3-031-38557-5_23.

[20] Alex Davidson, Ian Goldberg, Nick Sullivan, George Tankersley, and Filippo Valsorda. "Privacy Pass: Bypassing Internet Challenges Anonymously". In: *PoPETs* 3 (2018). DOI: 10.1515/popets-2018-0026.

[21] Khue Do, Lucjan Hanzlik, and Eugenio Paracucchi. "M&M'S: Mix and Match Attacks on Schnorr-Type Blind Signatures with Repetition". In:

EUROCRYPT 2024, Part VI. LNCS. Springer, Cham, 2024. DOI: 10.1007/978-3-031-58751-1_13.

[22] Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. "Generic Models for Group Actions". In: *PKC 2023, Part I*. LNCS. Springer, Cham, 2023. DOI: 10.1007/978-3-031-31368-4_15.

[23] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. LNCS. Springer, Berlin, Heidelberg, 1987. DOI: 10.1007/3-540-47721-7_12.

[24] Marc Fischlin. "Round-Optimal Composable Blind Signatures in the Common Reference String Model". In: *CRYPTO 2006*. LNCS. Springer, Berlin, Heidelberg, 2006. DOI: 10.1007/11818175_4.

[25] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. "The Algebraic Group Model and its Applications". In: *CRYPTO 2018, Part II*. LNCS. Springer, Cham, 2018. DOI: 10.1007/978-3-319-96881-0_2.

[26] Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. "Blind Schnorr Signatures and Signed ElGamal Encryption in the Algebraic Group Model". In: *EUROCRYPT 2020, Part II*. LNCS. Springer, Cham, 2020. DOI: 10.1007/978-3-030-45724-2_3.

[27] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. "A practical secret voting scheme for large scale elections". In: *AUSCRYPT*. Springer. 1992.

[28] 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: *ASIACRYPT 2025, Part IV*. LNCS. Springer, Singapore, 2025. DOI: 10.1007/978-981-95-5113-2_2.

[29] Lucjan Hanzlik, Julian Loss, and Benedikt Wagner. "Rai-Choo! Evolving Blind Signatures to the Next Level". In: *EUROCRYPT 2023, Part V*. LNCS. Springer, Cham, 2023. DOI: 10.1007/978-3-031-30589-4_26.

[30] Eduard Hauck, Eike Kiltz, and Julian Loss. "A Modular Treatment of Blind Signatures from Identification Schemes". In: *EUROCRYPT 2019, Part III*. LNCS. Springer, Cham, 2019. DOI: 10.1007/978-3-030-17659-4_12.

[31] Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. "Lattice-Based Blind Signatures, Revisited". In: *CRYPTO 2020, Part II*. LNCS. Springer, Cham, 2020. DOI: 10.1007/978-3-030-56880-1_18.

[32] Scott Hendrickson, Jana Iyengar, Tommy Pauly, Steven Valdez, and Christopher A. Wood. *Private Access Tokens. Internet-Draft draft-private-access-tokens-01*. Work in Progress. 2022. URL: https://datatracker.ietf.org/doc/draft-private-access-tokens/.

[33] Julia Kastner, Julian Loss, and Jiayu Xu. "On Pairing-Free Blind Signature Schemes in the Algebraic Group Model". In: *PKC 2022, Part II*. LNCS. Springer, Cham, 2022. DOI: 10.1007/978-3-030-97131-1_16.

[34] Julia Kastner, Julian Loss, and Jiayu Xu. "The Abe-Okamoto Partially Blind Signature Scheme Revisited". In: *ASIACRYPT 2022, Part IV*. LNCS. Springer, Cham, 2022. DOI: 10.1007/978-3-031-22972-5_10.

[35] Julia Kastner, Stefano Tessaro, and Greg Zaverucha. *Round-Optimal Pairing-Free Blind Signatures*. Cryptology ePrint Archive, Paper 2026/091. 2026. URL: https://eprint.iacr.org/2026/091.

[36] Shuichi Katsumata, Yi-Fu Lai, and Michael Reichle. "Breaking Parallel ROS: Implication for Isogeny and Lattice-Based Blind Signatures". In: *PKC 2024, Part I*. LNCS. Springer, Cham, 2024. DOI: 10.1007/978-3-031-57718-5_11.

[37] Shuichi Katsumata, Michael Reichle, and Yusuke Sakai. "Practical Round-Optimal Blind Signatures in the ROM from Standard Assumptions". In: *ASIACRYPT 2023, Part II*. LNCS. Springer, Singapore, 2023. DOI: 10.1007/978-981-99-8724-5_12.

[38] Jonathan Katz, Julian Loss, and Michael Rosenberg. "Boosting the Security of Blind Signature Schemes". In: *ASIACRYPT 2021, Part IV*. LNCS. Springer, Cham, 2021. DOI: 10.1007/978-3-030-92068-5_16.

[39] Michael Klooß and Michael Reichle. "Blind Signatures from Proofs of Inequality". In: *CRYPTO 2025, Part VI*. LNCS. Springer, Cham, 2025. DOI: 10.1007/978-3-032-01887-8_6.

[40] Michael Klooß, Michael Reichle, and Benedikt Wagner. "Practical Blind Signatures in Pairing-Free Groups". In: *ASIACRYPT 2024, Part I*. LNCS. Springer, Singapore, 2024. DOI: 10.1007/978-981-96-0875-1_12.

[41] Ueli Maurer. "Zero-knowledge proofs of knowledge for group homomorphisms". In: *DCC* 2-3 (2015). DOI: 10.1007/s10623-015-0103-5.

[42] Ueli M. Maurer. "Towards the Equivalence of Breaking the Diffie-Hellman Protocol and Computing Discrete Algorithms". In: *CRYPTO'94*. LNCS. Springer, Berlin, Heidelberg, 1994. DOI: 10.1007/3-540-48658-5_26.

[43] Ueli M. Maurer and Stefan Wolf. "Diffie-Hellman Oracles". In: *CRYPTO'96*. LNCS. Springer, Berlin, Heidelberg, 1996. DOI: 10.1007/3-540-68697-5_21.

[44] Tatsuaki Okamoto and Kazuo Ohta. "Universal Electronic Cash". In: *CRYPTO'91*. LNCS. Springer, Berlin, Heidelberg, 1992. DOI: 10.1007/3-540-46766-1_27.

[45] David Pointcheval. "Strengthened Security for Blind Signatures". In: *EUROCRYPT'98*. LNCS. Springer, Berlin, Heidelberg, 1998. DOI: 10.1007/BFb0054141.

[46] David Pointcheval and Jacques Stern. "Security Arguments for Digital Signatures and Blind Signatures". In: *Journal of Cryptology* 3 (2000). DOI: 10.1007/s001450010003.

[47] Michael Reichle and Zoé Reinke. *Threshold Blind Signatures from CDH*. Cryptology ePrint Archive, Report 2025/1798. 2025. URL: https://eprint.iacr.org/2025/1798.

[48] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO'89*. LNCS. Springer, New York, 1990. DOI: 10.1007/0-387-34805-0_22.

[49] Stefano Tessaro and Chenzhi Zhu. "Short Pairing-Free Blind Signatures with Exponential Security". In: *EUROCRYPT 2022, Part II*. LNCS. Springer, Cham, 2022. DOI: 10.1007/978-3-031-07085-3_27.

[50] David Wagner. "A Generalized Birthday Problem". In: *CRYPTO 2002*. LNCS. Springer, Berlin, Heidelberg, 2002. DOI: 10.1007/3-540-45708-9_19.