# Practical Blind Signatures in Pairing-Free Groups

Michael Klooß<sup>1\*</sup>, Michael Reichle<sup>1</sup>, and Benedikt Wagner<sup>2</sup>

<sup>1</sup> Department of Computer Science ETH Zurich, Switzerland {michael.klooss, michael.reichle}@inf.ethz.ch <sup>2</sup> Ethereum Foundation benedikt.wagner@ethereum.org

**Abstract.** Blind signatures have garnered significant attention in recent years, with several efficient constructions in the random oracle model relying on well-understood assumptions. However, this progress does not apply to pairing-free cyclic groups: fully secure constructions over cyclic groups rely on pairings, remain inefficient, or depend on the algebraic group model or strong interactive assumptions. To address this gap, Chairattana-Apirom, Tessaro, and Zhu (CTZ, Crypto 2024) proposed a new scheme based on the CDH assumption. Unfortunately, their construction results in large signatures and high communication complexity. In this work, we propose a new blind signature construction in the random oracle model

that significantly improves upon the CTZ scheme. Compared to CTZ, our scheme reduces communication complexity by a factor of more than 10 and decreases the signature size by a factor of more than 45, achieving a compact signature size of only 224 Bytes. The security of our scheme is based on the DDH assumption over pairing-free cyclic groups, and we show how to generalize it to the partially blind setting.

## 1 Introduction

A blind signature scheme [Cha82] is a special digital signature scheme with a two-party signing process. Namely, a Signer, who possesses the secret key, interacts with a User holding the message intended for signing. Once the signing interaction terminates, the User should hold a signature for the message that can be verified with respect to the Signer's public key. It is crucial that the scheme upholds the following security and privacy properties [JLO97, PS00]: One-More Unforgeability asserts that the User can not generate valid signatures on its own, i.e., without engaging in the signing protocol with the Signer; Blindness ensures that during the signing process, the User's message remains undisclosed to the Signer. More precisely, the Signer can not link the message-signature pair to the interaction. These two properties render blind signatures a versatile privacy-preserving tool. They have found use in various domains, including but not limited to anonymous credentials [CG08, CL01] and electronic cash [Cha82, OO92].

**Recent Progress.** For a long time, constructions of blind signatures faced considerable challenges, characterized by prohibitive inefficiency  $[GRS^{+}11]$ , reliance on strong assumptions [Cha82, Bol03, BNPS03, FHS15, Gha17], complexity leveraging  $[GRS^{+}11, GG14]$  or limited security guarantees [PS00, AO00, HKL19, KLX22a]. Even in the random oracle model [BR93], a fully secure and efficient blind signature scheme based on well-studied assumptions remained an elusive goal. Recently, progress in two distinct directions has been made, both contributing significantly to the resolution of this longstanding issue: The first line of work  $[KLR21, CHL^+22, HLW23]$  uses cut-and-choose techniques to turn weakly secure but efficient constructions into fully secure ones while avoiding the use of strong assumptions. The second line of work [dK22, KRS23, KNR24] draws inspiration from a generic construction due to Fischlin [Fis06]. By carefully exploiting the algebraic structure of specific instantiations and with major modifications of Fischlin's proof technique, these works provide practical blind signatures based on established assumptions. Notably, among the aforementioned constructions, the practical ones heavily rely on algebraic properties of lattices [dK22], pairings  $[CHL^+22, KRS23, HLW23]$ , or the RSA setting  $[CHL^+22, KNR24]$ .

<sup>©</sup>IACR 2024. This is the full version of an article that will be published in the proceedings of ASIACRYPT 2024.

<sup>&</sup>lt;sup>\*</sup> Work done at Aalto University. The author's affiliation changed before publication.

The Pairing-Free Setting. A long-standing goal in the realm of digital signature variants and in cryptography in general is to understand if and how pairings can be avoided [BGH07, GHKW16, CKU20, CFGG22, TZ22, PW23, TZ23]. This endeavor holds both scientific intrigue and practical merit: operations in pairing-friendly groups are more expensive, and pairing-free groups enjoy a broader library support.

Unfortunately, as we have seen above, the lines of work [KLR21, CHL<sup>+</sup>22, HLW23] and [dK22, KRS23] so far did not yield practical blind signatures over pairing-free cyclic groups. And while there are promising works trying to close this gap, they all fall short in meeting the desired objectives entirely. Specifically, while some works [KLX22b, TZ22, CKM<sup>+</sup>23] yield very efficient pairing-free blind signature based on established assumptions, their analysis relies on the algebraic group model [FKL18]. Conversely, the very recent work by Chairattana-Apirom, Tessaro, and Zhu [CTZ24] avoids the use of the algebraic group model. They give efficient constructions based on interactive variants of CDH, along with a non-interactive CDH-based construction utilizing techniques from [HLW23]. Unfortunately, this latter construction has signatures containing  $\Theta(\lambda)$ many group elements, where  $\lambda$  is the security parameter. In contrast to that, signatures in the most efficient pairing-based construction [KRS23] contain only a small constant number of group elements.

**Our Goal.** The goal of this work is to close this gap by providing a new blind signature scheme over pairing-free cyclic groups, that (1) is based on well-studied cryptographic hardness assumptions, (2) avoids idealizations other than the random oracle model, and (3) is practically efficient, without the  $\lambda$  overhead in signature size.

## 1.1 Our Contribution

We achieve our goal by constructing a practical blind signature scheme in pairing-free groups, which we compare with the state-of-the-art in Tables 1 and 2. To summarize, our scheme comes with the following key characteristics:

- Unforgeability. One-more unforgeability holds based on the DDH assumption in the random oracle model. Notably, our proof avoids the need for rewinding, resulting in a tighter security bound in contrast to CTZ-3 [CTZ24], which is the only other scheme achieving full one-more unforgeability in pairing-free groups without the algebraic group model.
- Blindness. Our scheme is statistically blind, and we show that it naturally generalizes to the setting of partially blind signatures [AF96]. This is the first scheme supporting partial blindness in this regime<sup>3</sup>.
- Efficiency. Our scheme is both concretely and asymptotically efficient. Especially, comparing to CTZ-3 [CTZ24], we reduce the communication complexity by a factor of more than 10, and the signature size from 10.5 Kilobytes to 224 Bytes, see Table 2).

Technically, our starting point is the pairing-based construction by Katsumata, Reichle, and Sakai [KRS23]. We turn this construction into a pairing-free variant by replacing the pairing with a (blindly issued) non-interactive proof. It is worth noting that a straightforward substitution would yield only a weaker notion of one-more unforgeability, similar to CTZ-1 and CTZ-2 [CTZ24]. However, through a strategically devised security proof, we can circumvent this issue, achieving full one-more unforgeability. A second twist allows us to avoid rewinding, another improvement over CTZ-3 [CTZ24]. Further, we revist the security of Fischlin's straightline extractable proof to achieve statistical blindness for our scheme. Notably, this is in contrast to [KRS23], which is only computationally blind.

#### 1.2 Related Work

Here, we discuss related work on blind signatures. We focus primarily on recent efficient and secure constructions in the random oracle model [BR93]. We give a comparison of blind signature schemes in pairing-free cyclic groups in Table 1.

 $<sup>^{3}</sup>$  It is not obvious how to modify CTZ [CTZ24] to achieve partial blindness.

Scheme	Assumption	Full OMUF	Moves	Communication	Signature
Cl-Schnorr [FPS20]	OMDL, mROS	1	3	$2\mathbb{G} + 3\mathbb{Z}_p$	$1\mathbb{G} + 1\mathbb{Z}_p$
Abe [Abe01, KLX22b]	DLOG	1	3	$\lambda + 3\mathbb{G} + 6\mathbb{Z}_p$	$2\mathbb{G}+6\mathbb{Z}_p$
TZ [TZ22]	DLOG	1	3	$2\mathbb{G} + 4\mathbb{Z}_p$	$4\mathbb{Z}_p$
Snowblind [CKM <sup>+</sup> 23]	DLOG	1	3	$2\mathbb{G} + 4\mathbb{Z}_p$	$1\mathbb{G}+2\mathbb{Z}_p$
CTZ-1 [CTZ24]	CT-OMCDH	X	4	$5\mathbb{G} + 5\mathbb{Z}_p$	$1\mathbb{G} + 4\mathbb{Z}_p$
CTZ-2 [CTZ24]	CT-OMCDH	×	5	$5\mathbb{G} + 5\mathbb{Z}_p$	$1\mathbb{G} + 4\mathbb{Z}_p$
CTZ-3 [CTZ24]	CDH	1	4	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$
Ours	DDH	1	4	$\Theta(\lambda)(\lambda + \mathbb{G} + \mathbb{Z}_p)$	$2\mathbb{G} + 5\mathbb{Z}_p$

**Table 1.** Comparison of concurrently secure blind signature schemes in the discrete logarithm setting without pairings. All constructions rely on the random oracle model, and schemes above the line additionally require the algebraic group model. We compare the assumptions and security and the communication complexity and signature size in terms of number of group elements and number of field elements. The schemes CTZ-1 and CTZ-2 [CTZ24] only satisfy a weaker variant of one-more unforgeability.

Scheme	Assumption	Full OMUF	Moves	Communication	Signature
CTZ-3 [CTZ24]	CDH	1	4	27.12 Kilobytes	10.50 Kilobytes
Ours	DDH	1	4	2.46 Kilobytes	224 Bytes

Table 2. Comparison of the concrete efficiency of concurrently secure blind signature schemes in the discrete logarithm setting without pairings. We exclude constructions in the algebraic group model and constructions that do not achieve full one-more unforgeability for this comparison. We assume  $\lambda = 128$  and that group and field elements are represented using 256 bit. Numbers are computed using the script in Appendix F.

Foundations of Blind Signatures. Blind signatures have been introduced by Chaum in 1982 [Cha82] in the context of electronic cash. Pioneering works are by Juels, Luby, and Ostrovsky [JLO97], by Fischlin [Fis06], and by Pointcheval and Stern [PS00]. Namely, Juels, Luby, and Ostrovsky have demonstrated that blind signatures can generically (and inefficiently) be constructed from one-way trapdoor permutations via secure two-party computation. Fischlin also gave a generic construction of round-optimal (i.e., two moves) blind signatures. On the other hand, Pointcheval and Stern have shown the security of efficient three-move blind signatures in the random oracle model, as long as only polylogarithmically many signatures are issued concurrently. Since then, several impossibility results have been proven [FS10, Pas11, BL13]. For example, Fischlin and Schröder have ruled out certain statistically blind three-move constructions from non-interactive assumptions in the standard model [FS10].

Strong Assumptions or Idealized Models. In addition to generic constructions mentioned earlier, several more direct constructions exist, relying on complexity leveraging [GRS<sup>+</sup>11, GG14] or non-standard q-type or interactive assumptions [Oka06, GRS<sup>+</sup>11, FHS15, Gha17]. Also, there are blind variants of BLS signatures and RSA-full-domain hash signatures [Bol03, BNPS03], which are very efficient and round-optimal. However, these constructions rely on interactive one-more variants of the underlying assumptions, e.g., one-more CDH. If one is willing to rely on the algebraic group model [FKL18], there are several efficient constructions of blind signatures in pairing-free groups [KLX22b, TZ22, CKM<sup>+</sup>23]. A recent scheme due to Fuchsbauer and Wolf [FW24] outputs regular Schnorr signatures [Sch91]. In terms of assumptions, their result can be interpreted in two ways: one can assume the security of Schnorr signatures [Sch91] with respect to a fixed hash function, which is an interactive assumption; alternatively, one can rely on the discrete logarithm assumption by treating the hash function as a random oracle. In this latter case, however, their protocol proves relations defined by the random oracle in generic SNARK, which has unclear security implications and is highly non-standard.

Cut-and-Choose Constructions. The starting point of this line of work lies in efficient constructions of blind signatures with weak security guarantees [PS00, AO00, HKL19, HKLN20, KLX22a]

based on witness indistinguishable linear identification schemes [Oka93]. Specifically, these schemes are insecure if more than polylogarithmically many signatures are issued concurrently. This is not only an artifact of the security proof but can be exploited in a practical attack [Sch01, Wag02, BLL+21]. By extending a classical construction of Pointcheval [Poi98], Katz, Loss, and Rosenberg have introduced boosting [KLR21], a technique to turn the aforementioned polylogarithmically-secure blind signatures into fully secure ones: during the Nth signing interaction, the Signer and User engage in a 1-of-N cut-and-choose, which results in communication and computation scaling linearly in N. Subsequently, Chairattana-Apirom et al.  $[CHL^+22]$  have improved communication to scale logarithmically in N. They have also developed two concretely efficient constructions leveraging the cut-and-choose idea. Building on one of these constructions (called PI-Cut-Choo), Hanzlik, Loss, and Wagner [HLW23] have proposed a construction called Rai-Choo. This scheme is stateless and round-optimal, and both computation and communication are independent of N. It relies on the CDH assumption in the pairing-setting. On the downside, signatures in Rai-Choo contain  $\Theta(\lambda)$ many group elements. The latest achievements in this line of work are the pairing-free constructions by Chairattana-Apirom, Tessaro, and Zhu [CTZ24]. While two of their constructions are very efficient, they rely on interactive assumptions and only achieve a weaker version of unforgeability. The third construction, which achieves full unforgeability and relies only on CDH has signatures containing  $\Theta(\lambda)$  many group elements due to techniques inherited from Rai-Choo. Hence, this line of work did not yet result in a fully secure and efficient scheme with constant<sup>4</sup> signature size over pairing-free groups.

**Fischlin and its Descendants.** In addition to the line of work using cut-and-choose outlined above, a second line of constructions managed to construct schemes that are practical and rely on conservative assumptions. This line of work draws inspiration from Fischlin's generic construction [Fis06] but introduces several modifications to the proof technique to enable efficient implementations. Specifically, del Pino and Katsumata [dK22] efficiently instantiate this framework from lattice assumptions, while Katsumata, Reichle, and Sakai [KRS23] give two constructions utilizing pairings. Kastner, Nguyen, and Reichle [KNR24] present a construction relying on pairing-free groups and the strong RSA assumption.

However, this line of research has not yet yielded an efficient scheme with constant signature size over pairing-free groups alone. Our contribution can be viewed as adapting the second construction proposed by Katsumata, Reichle, and Sakai to the pairing-free setting. Doing this naively would result in a weaker form of unforgeability as for the first two constructions in [CTZ24]. With a clever twist, we can prove full unforgeability.

#### 1.3 Technical Overview

Here, we give an informal overview of our techniques. Our starting point will be the pairing-based blind signature scheme by Katsumata, Reichle, and Sakai [KRS23]. As this scheme is already very efficient, our main technical goal is to eliminate the use of pairings.

Our Starting Point: Pairing-based Blind Signatures. Let us briefly recall the construction by Katsumata et al. [KRS23]. To this end, let  $\mathbb{G}$  be a pairing-friendly group generated by  $G \in \mathbb{G}$ . The basis of the scheme is a signature scheme obtained from the Boneh-Boyen identity-based encryption [BB04], for which signatures  $\sigma = (S_1, S_2)$  have the structure

$$S_1 = uV + s(\overline{m}U + H), \quad S_2 = sG. \tag{1}$$

Here,  $s \in \mathbb{Z}_p$  is sampled uniformly during the signing process,  $u \in \mathbb{Z}_p$  is the secret key, U = uG, V, and H are public group elements, and  $\overline{m}$  is a hash of the message to be signed. Verification leverages the pairing. In the construction by Katsumata, Reichle, and Sakai, such signatures are issued blindly as follows:

- 1. The User sends a Pedersen commitment C to  $\overline{m}$ . It also includes a proof  $\pi_{\mathsf{Ped}}$ , proving knowledge of the commitment randomness and  $\overline{m}$ ;
- 2. The Signer homomorphically computes a blinded version  $\sigma_C$  of the signature  $\sigma$  from the commitment C and sends it to the User;

 $<sup>^4</sup>$  Constant signature size here means a constant (in  $\lambda)$  number of group elements.

3. The User can remove the commitment randomness to obtain an actual signature  $\sigma'$ . For blindness, it is also essential that the User rerandomizes the signature into a fresh valid signature  $\sigma$  before outputting it.

In [KRS23] and in this work,  $\pi_{\text{Ped}}$  has to be straightline-extractable. Due to their instantiation of  $\pi_{\text{Ped}}$ , [KRS23] relies on DDH for blindness. We follow a different approach and instantiate  $\pi_{\text{Ped}}$ by revisiting the security of randomized Fischlin's transform [Fis05, Ks22]. Consequently, our instantiation is statistically blind. We refer to the technical part of this paper (and Appendix C) for details.

Towards a Pairing-Free Scheme. We now want to eliminate the use of the pairing from the scheme. For that, we first observe that we can port the underlying signature scheme into the pairing-free setting. Essentially, we include a proof  $\pi$  in the signature that proves that Equation (1) holds. That is, the signature is now  $\sigma = (S_1, S_2, \pi)$ . We observe that such a proof can be constructed very efficiently from a  $\Sigma$ -protocol.

While this works in the non-blind setting, computing such a signature  $\sigma$  interactively and blindly turns out to be challenging: the User needs to obtain  $\pi$ , but it does not know a suitable witness to do so. Especially, the witness includes the secret key  $u \in \mathbb{Z}_p$ . On the other hand, we cannot just let the Signer generate the proof  $\pi$ , because the statement is the rerandomized signature  $\sigma$ , which we want to keep hidden from the Signer.

To overcome this first challenge, our starting point is an approach similar to [CTZ24]. Namely, as  $\pi$  is constructed from an appropriate  $\Sigma$ -protocol, we can issue  $\pi$  interactively and blindly. Roughly, we adapt the techniques of [CTZ24] to our setting, and obtain a pairing-free variant of the construction by Katsumata et al. [KRS23] with blind issuance.

**Full Unforgeability Fails.** Equipped with (a sketch of) our scheme, let us now move our attention to the security proof, concretely, the proof of one-more unforgeability. The natural idea would be to translate the security proof from the pairing-based construction [KRS23] to our setting. Unfortunately, when doing that naively, we can not achieve full one-more unforgeability. To understand this, the reader may first recall that in the one-more unforgeability game, the adversary can interact with the Signer in multiple signing sessions. It wins the game, if it outputs valid signatures for more messages than it *completed* signing interactions<sup>5</sup>. Additionally, the reader may recall the structure of our current blind signing protocol:

- 1. The Signer and the User interact similarly to the pairing-based scheme sketched above. This means that the Signer sends  $\sigma_C$  to the User which allows the User to compute a signature  $(S_1, S_2)$  of the underlying pairing-based scheme [KRS23];
- 2. The Signer and the User interactively (and blindly) compute the proof  $\pi$ ;

Now, assume that we have an adversary interacting 20 times with the Signer, but only completing 7 interactions. Say the remaining 13 interactions end after the first of the two stages above. Now, if an adversary outputs 8 valid message-signature pairs, it is deemed successful in the one-more unforgeability game. However, the reduction from [KRS23] does not apply, as the adversary essentially finished 13 > 8 interactions of the pairing-based protocol and learned  $\sigma_C$ . Conceptually, the reduction would leak  $\sigma_C$  to the adversary too early, and  $\sigma_C$  contains a solution to a hard problem (specifically, CDH). A similar issue with a different underlying scheme also appeared in [CTZ24]. The authors manage to circumvent the issue by outputting a commitment to the signature at first, in the second, and then opening the commitment only in the very last message of the protocol. While this is elegant, it also causes some overhead in terms of efficiency. As we will see next, for our scheme it is possible to prove unforgeability *without* further modifying the signing protocol.

Achieving Full Unforgeability. Our high-level approach for showing full one-more unforgeability is to eliminate information about  $\sigma_C$  from singing interactions that are not finished. To this end, we observe that  $\sigma_C$  is pseudorandom as long as the adversary never learns  $\pi$ , so intuitively, it should not give the adversary any information it can use for its forgeries. To be more concrete, let us assume for the sake of this overview that the reduction knows ahead of time which signing

<sup>&</sup>lt;sup>5</sup> We could show a weaker form of one-more unforgeability similar to CTZ-1 and CTZ-2 [CTZ24], in which the adversary has to output valid signatures for more messages than it *started* signing interactions.

interactions are not finished<sup>6</sup>. Then, the reduction will simply send a random  $\sigma_C$  to the adversary. To get an intuition for why that works, observe from Equation (1) that  $(S_1, S_2)$  is indistinguishable from random by the DDH assumption applied to (sG, H, sH). Coming back to our example from above, the adversary would now only learn 7 < 8 such  $\sigma_C$ 's, and the proof of the pairing-based scheme applies.

Avoiding Rewinding. So far, we have omitted an important detail: the reduction of the pairingbased scheme [KRS23] does (of course) not know the secret key, which is part of the witness for the proof  $\pi$ . It is thus not clear how the reduction can issue  $\pi$  to the User interactively<sup>7</sup>. A similar problem appears in [CTZ24], so let us briefly review their solution. Roughly [CTZ24] employs an OR proof for  $\pi$ . That is,  $\pi$  ensures that the signature is valid or the Signer knows the discrete logarithm of some group element  $X \in \mathbb{G}$  output by the random oracle. The reduction then makes sure to know this discrete logarithm, which allows simulating  $\pi$ . Finally, the reduction either obtains a valid signature, which allows to finish the proof as before, or the discrete logarithm of X. For the latter, the reduction is required to rewind the adversary, leading to a highly non-tight security bound.

To avoid rewinding, we make the following twist: we replace X with a Diffie-Hellman (DH) tuple **D**. In particular,  $\pi$  now ensures that either  $\sigma$  is well-formed or **D** is a DH tuple. Interestingly, this comes at *no* additional cost in signature size. Intuitively, as we are no longer proving knowledge of a witness, but rather membership in a language, rewinding should not be needed.

Turning this into a formal proof requires a careful sequence of hybrid games, as outlined next. Initially, the game simulates the Signer as in the real protocol, which means that the proof  $\pi$  is computed via the *signature branch*, i.e., using the witness which testifies the validity of the signature. Also, **D** is not a DH tuple. Then, soundness (not knowledge soundness!) of the proofs  $\pi$  contained in the forgery guarantees that all signatures in the forgery are valid (because **D** is *not* a DH tuple). Call this event E, our strategy is to preserve E while simulating  $\pi$  using the DH branch. If so, we can argue as above that CDH is solved if the adversary is successful. To carry this out, we need to switch **D** to a valid DH tuple. We want to use DDH to argue that the probability of E does not change significantly when we make this change. To do this formally, we need to present a reduction that interpolates between the two games and *efficiently* evaluates whether E occurs. Doing this naively is equivalent to solving DDH in the first place! The crucial insight here is that this can be done efficiently using the signing key u and the discrete logarithm h of H. Once we are in a game where **D** is a valid DH tuple, we can use the corresponding DH witness to simulate  $\pi$ . Then, if E occurs in this last game, we can reduce to CDH as discussed above<sup>8</sup>.

Generalizing to Partial Blindness. Partial blindness allows the Signer and User to agree on a common message  $\tau$  that is signed together with the (hidden) message m. This property is useful for many privacy-preserving applications. To obtain partial blindness, we employ the design principle from Abe and Okamoto [AO00]. That is, the vector **D** is output by a random oracle  $H_{ddh}$  on input  $\tau$ . Otherwise, the entire protocol remains unchanged. By carefully applying the techniques sketched above, we can prove partial blindness.

#### 1.4 Organization of this Paper

In Section 2, we provide the relevant cryptographic definitions. In Section 3, we sketch the pairingfree signature scheme that underlies our construction. To improve readability, we first provide an unblinded version of our protocol in Section 4 and prove one-more unforgeability. In Section 5, we provide the full protocol and its blindness proof. The Signer in this protocol is the same as in Section 4, which means that one-more unforgeability follows as in Section 4.

<sup>&</sup>lt;sup>6</sup> Naively, this requires guessing aborted sessions which leads to an exponential security loss. Our approach actually relies on a slightly more sophisticated argument.

<sup>&</sup>lt;sup>7</sup> Non-interactively and without blindness, this can be done in a standard way, using honest-verifier zero-knowledge and by programming the random oracle.

<sup>&</sup>lt;sup>8</sup> The final reduction does not need to check if E occurs, and hence it neither needs the secret key u nor the discrete logarithm h of H.

## 2 Preliminaries

Let  $\lambda \in \mathbb{N}$  be the security parameter. We use standard notations for probability, algorithms and distributions <sup>9</sup>. We write  $A(in_A) \longleftrightarrow B(in_B)$  for interactive protocols between parties A and B with input  $in_A$  and  $in_B$ , respectively. Within algorithmic descriptions, we denote by req C that the algorithm outputs  $\perp$  if the condition C is false. When describing games, we denote by **abort if** C that the game outputs 0 if the condition C is false. Throughout, we denote by  $\mathbb{G}$  a group of prime order p with generator  $G \in \mathbb{G}$ . We generally use additive notation for  $\mathbb{G}$ . Throughout, group elements G are capital, whereas elements x in  $\mathbb{N}$  or  $\mathbb{Z}_p$  are lowercase. Vectors of elements  $\mathbb{G}$  or  $\mathbb{X}$  are bold, and generally indexed  $\mathbb{G} = (G_1, \dots, G_n)$  or  $\mathbb{X} = (x_1, \dots, x_n)$ , respectively.

Assumptions. Throughout the paper, we let  $\mathbb{G}$  be a group of prime order p with generator  $G \in \mathbb{G}$ . As common, this should be understood as implicitly being a family of groups, i.e.,  $\mathbb{G} = \mathbb{G}_{\lambda}$  is implicitly parameterized by the security parameter  $\lambda$ . We briefly recall the DL, CDH and (Q-)DDH assumptions and refer to Appendix A for formal definitions. While DDH implies Q-DDH, CDH and DL tightly, these assumptions will be convenient to prove security later. Below, let  $a, b, c \stackrel{\$}{=} \mathbb{Z}_p$ . The DL assumption states that given G and aG it is hard to compute a. The CDH assumption states that it is hard given (G, aG, bB) to compute (ab)G. The DDH assumption states that it is hard to distinguish a real Diffie-Hellman tuple (G, aG, bB, (ab)G) from a random tuple (G, aG, bB, cG). The Q-DDH assumption states that it is hard to distinguish Q random Diffie-Hellman tuples from random Q tuples.

(Partially) Blind Signatures. We define the primitive of interest, namely, blind signatures [Cha82]. For convenience, we directly define partially blind signatures [AF96] and note that plain blind signatures are the special case in which  $\tau$  is fixed, i.e.,  $|\mathcal{T}| = 1$ .

**Definition 1 (Partially Blind Signature Scheme).** A partially blind signature scheme with message space  $\mathcal{M}$  and common message space  $\mathcal{T}$  is a tuple of PPT algorithms BS = (KeyGen, S, U, Verify) with the following syntax:

- KeyGen $(1^{\lambda})$ : outputs a pair of keys (vk, sk). We assume that sk includes vk implicitly.
- $\mathsf{S}(\mathsf{sk}, \tau) \longleftrightarrow \mathsf{U}(\mathsf{vk}, m, \tau): \mathsf{S} \text{ takes as input a secret key } \mathsf{sk} \text{ and common message } \tau \in \mathcal{T}. \text{ U takes as input a key } \mathsf{vk}, \text{ a message } m \in \mathcal{M} \text{ and common message } \tau \in \mathcal{T}. \text{ After the execution, U returns a signature } \sigma \text{ and we write } \sigma \leftarrow \langle \mathsf{S}(\mathsf{sk}, \tau), \mathsf{U}(\mathsf{vk}, m, \tau) \rangle.$
- Verify(vk,  $m, \tau, \sigma$ ) is deterministic and takes as input public key vk, message  $m \in \mathcal{M}$ , a common message  $\tau$ , and a signature  $\sigma$ , and outputs  $b \in \{0, 1\}$ .

**Definition 2 (Correctness).** A partially blind signature BS is correct with correctness error  $\gamma_{\text{err}}$  if for all  $(vk, sk) \in \text{KeyGen}(1^{\lambda})$  and all  $m \in \mathcal{M}, \tau \in \mathcal{T}$ , it holds that

$$\Pr[\sigma \leftarrow \langle \mathsf{S}(\mathsf{sk},\tau), \mathsf{U}(\mathsf{vk},m,\tau) \rangle \colon \mathsf{Verify}(\mathsf{vk},m,\tau,\sigma) = 1] \ge 1 - \gamma_{\mathsf{err}}(\lambda).$$

Intuitively, a (partially) blind signature scheme should not allow any user to obtain signatures without interacting with the Signer. This is modeled by the notion of one-more unforgeability, which states that after completing k-1 signing sessions on some common message  $\tau^*$ , an adversary can not output valid signatures on k messages with common message  $\tau^*$ .

**Definition 3 (One-More Unforgeability).** Let BS = (KeyGen, S, U, Verify) be a blind signature scheme. Consider an algorithm A and the following game:

- 1. Run  $(vk, sk) \leftarrow KeyGen(1^{\lambda})$  and let  $\mathcal{O}$  be an interactive oracle simulating  $S(sk, \cdot)$ .
- 2. Run  $\tau$ ,  $((m_1, \sigma_1), \ldots, (m_k, \sigma_k)) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{vk})$ , where  $\mathcal{A}$  can query  $\mathcal{O}$  in an arbitrarily interleaved way.
- 3. Output 1 if and only if all  $m_i, i \in [k]$  are pairwise distinct,  $\mathcal{A}$  completed at most k-1 interactions with  $\mathcal{O}$  on input  $\tau$ , and for each  $i \in [k]$  it holds that  $\mathsf{Verify}(\mathsf{vk}, m_i, \tau, \sigma_i) = 1$ .

<sup>&</sup>lt;sup>9</sup> We use  $x \coloneqq v$  for assignment of value v to x (and  $x \leftarrow v$  if x is updated with value v),  $x \leftarrow A(in)$  for (probabilistic) algorithms A on input in, and  $x \stackrel{\$}{\leftarrow} \mathcal{D}$  for sampling from distribution  $\mathcal{D}$ . (If  $\mathcal{D}$  is a set, this denotes sampling from  $\mathcal{D}$  uniformly and independently at random).

We denote by  $\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$  the probability that the above game outputs 1. We say that BS is onemore unforgeable (OMUF), if for every PPT algorithm  $\mathcal{A}$ , it holds that  $\mathsf{AdvOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda) = \operatorname{negl}(\lambda)$ .

To protect the privacy of users, blind signatures should satisfy blindness. Intuitively, blindness states that a malicious signer can not link the signing interaction to the message-signature pair (except for the common message  $\tau$ ). We emphasize that we consider the malicious signer blindness, i.e., the malicious signer can freely choose the public key and arbitrarily deviate from the protocol.

**Definition 4 (Partial Blindness).** Let BS = (KeyGen, S, U, Verify) be a blind signature scheme. For an algorithm  $\mathcal{A}$  and bit  $b \in \{0, 1\}$ , consider the following game:

- 1. Run  $(\mathsf{vk}, m_0, m_1, \tau, st) \leftarrow \mathcal{A}(1^{\lambda}).$
- 2. Let  $\mathcal{O}_0$  be an interactive oracle simulating  $U(v\mathbf{k}, m_b, \tau)$  and  $\mathcal{O}_1$  be an interactive oracle simulating
- $U(\mathsf{vk}, m_{1-b}, \tau).$ 3. Run st'  $\leftarrow \mathcal{A}^{\mathcal{O}_0, \mathcal{O}_1}(st)$ , where  $\mathcal{A}$  has arbitrary interleaved one-time access to  $\mathcal{O}_0$  and  $\mathcal{O}_1$ . Let  $\sigma_b, \sigma_{1-b}$  be the local outputs of  $\mathcal{O}_0, \mathcal{O}_1$ , respectively.
- 4. If  $\sigma_0 = \bot$  or  $\sigma_1 = \bot$ , run  $b' \leftarrow \mathcal{A}(st', \bot, \bot)$ . Else, run  $b' \leftarrow \mathcal{A}(st', \sigma_0, \sigma_1)$ .
- 5. Output b'.

We denote by  $\operatorname{AdvPBlind}_{\mathcal{A}}^{\mathsf{BS}}(\lambda)$  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  $\mathsf{AdvPBlind}_{\mathcal{A}}^{\mathsf{BS}}(\lambda) = \operatorname{negl}(\lambda)$ .

**Relations and**  $\Sigma$ -**Protocols.** Next, we define  $\Sigma$ -protocols for NP-relations. We start by defining NP-relations.

**Definition 5 (NP-Relation and Language).** Let  $\mathsf{R} \subseteq \{0,1\}^* \times \{0,1\}^*$  be a binary relation. We say that R is an NP-relation, if there are polynomials p and q such that R can efficiently be decided and for every  $(\mathbf{x}, \mathbf{w}) \in \mathsf{R}$ , we have  $|\mathbf{x}| \leq p(\lambda)$  and  $|\mathbf{w}| \leq q(|\mathbf{x}|)$ . We denote by  $\mathscr{L}_{\mathsf{R}} = \{\mathbf{x} \in \mathsf{R}\}$  $\{0,1\}^* \mid \exists w \ s.t. \ (x,w) \in \mathsf{R}\}$  the language induced by  $\mathsf{R}$ .

Let R be an NP-relation with statements x and witnesses w. A  $\Sigma$ -protocol for an NP-relation R for language  $\mathscr{L}_R$  with challenge space  $\mathcal{CH}$  is a tuple of PPT algorithms  $\Sigma = (\mathsf{Init}, \mathsf{Resp}, \mathsf{Verify})$  such that

- lnit(x, w): given a statement x  $\in \mathscr{L}_R$  and a witness w, outputs a first flow message (i.e., commitment) A and a state st, where we assume st includes (x, w),
- $\mathsf{Resp}(\mathsf{st}, c)$ : given a state st and a challenge  $c \in \mathcal{CH}$ , outputs a third flow message (i.e., response) z,
- Verify  $(\mathbf{x}, A, c, z)$ : given a statement  $\mathbf{x} \in \mathscr{L}_{\mathsf{R}}$ , a commitment A, a challenge  $c \in \mathcal{CH}$ , and a response z, outputs a bit  $b \in \{0, 1\}$ .

We call the tuple (A, c, z) the transcript and say that they are valid for x if Verify(x, A, c, z) outputs 1. When the context is clear, we simply say it is valid and omit  $\mathbf{x}$ . Next, we define the standard notions of correctness, special honest-verifier zero-knowledge, and (2-)special soundness.

**Definition 6 (Correctness).** Let R be an NP-relation and  $\Sigma = (\text{Init}, \text{Resp}, \text{Verify})$  be an  $\Sigma$ -protocol for R. We say  $\Sigma$  is correct, if for all  $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}$ ,  $(A, \mathsf{st}) \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w})$ ,  $c \in \mathcal{CH}$ , and  $z \leftarrow \mathsf{Resp}(\mathsf{st}, c)$ , it holds that Verify(x, A, c, z) = 1.

**Definition 7 (Special HVZK).** Let R be an NP-relation and  $\Sigma = (Init, Resp, Verify)$  be a  $\Sigma$ protocol for R. We say that  $\Sigma$  is special honest-verifier zero-knowledge (HVZK), if there exists a PPT zero-knowledge simulator Sim such that for any (potentially unbounded) adversary  $\mathcal{A}$ , it holds that for any  $(x, w) \in \mathsf{R}$  and  $c \in \mathcal{CH}$  that  $\mathsf{D}_{\mathsf{real}} = \mathsf{D}_{\mathsf{sim}}$  for

$$\begin{split} \mathsf{D}_{\mathsf{real}} &\coloneqq \{(A,c,z) \mid A \leftarrow \mathsf{Init}(\mathtt{x}, \mathtt{w}), z \leftarrow \mathsf{Resp}(\mathsf{st}, c)\}, \\ \mathsf{D}_{\mathsf{sim}} &\coloneqq \{(A,c,z) \mid (A,z) \leftarrow \mathsf{Sim}(\mathtt{x}, c)\}. \end{split}$$

In this work, we write HVZK for short.

**Definition 8 (Special Soundness).** Let R be an NP-relation and  $\Sigma = (\text{Init}, \text{Resp}, \text{Verify})$  be a  $\Sigma$ -protocol for R. We say that  $\Sigma$  is (2-)special sound, if there exists a deterministic PT extractor Ext such that given two valid transcripts  $\{(A, c_b, z_b)\}_{b \in [2]}$  for statement x with  $c_0 \neq c_1$ , along with x, outputs a witness w such that  $(x, w) \in R$ .

Non-Interactive Proof Systems. Here, we define straightline-extractable non-interactive zeroknowledge proofs. We limit ourselves to security in the random oracle model. Efficient constructions are known in this case, e.g., using the Fischlin transformation [Fis05], but also [Pas03, Kat21, Ks22].

Definition 9 (Non-Interactive Proof System). A non-interactive proof system NIPS for NPrelation R using a random oracle H is a pair NIPS = (Prove, Ver) of PPT algorithms with access to a random oracle, where

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

Note that out definitions of zero-knowledge simulator and knowledge extractor are independent of an adversary, in particular, they are straightline by definition.

**Definition 10 (Correctness).** Let NIPS = (Prove, Ver) be a non-interactive proof system for a relation R. It has correctness error  $\gamma_{err}$  if for all  $(x, w) \in R$ , it holds that

$$\Pr\left[\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathbb{x}, \mathbb{w}) \ : \ \mathsf{Ver}^{\mathsf{H}}(\mathbb{x}, \pi) = 1 \right] \ge 1 - \gamma_{\mathsf{err}}(\lambda),$$

where the probability is over the choice of H and the randomness of Prove, Ver. We call NIPS correct if  $\gamma_{\mathsf{err}}(\lambda) = \operatorname{negl}(\lambda)$ . We say it is perfectly correct if  $\gamma_{\mathsf{err}} = 0$ .

**Definition 11 (Witness Indistinguishability).** Let  $NIPS = (Prove^{H}, Ver^{H})$  be a non-interactive proof system for a relation R in the random oracle model. Let A be an algorithm which makes at most  $Q = Q(\lambda)$  queries to H and let

$$\mathsf{AdvWI}_{\mathcal{A}}^{\mathsf{NIPS}}(Q,\lambda) = \Pr\left[b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_0}(1^{\lambda}) \colon b = 1\right] - \Pr\left[b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_1}(1^{\lambda}) \colon b = 1\right],$$

where  $\mathcal{O}_i(\mathbf{x}, \mathbf{w}_0, \mathbf{w}_1)$  returns  $\mathsf{Prove}^{\mathsf{H}}(\mathbf{x}, \mathbf{w}_i)$  for  $i \in \{0, 1\}$ . We call NIPS statistically (resp. computationally) witness indistinguishable (WI), if for any unbounded (resp. PPT) adversary A, the advantage  $\operatorname{AdvWl}_{A}^{\operatorname{NIPS}}(\lambda)$  is negligible.

For knowledge soundness, the extractor must compute a witness from an accepting proof and all adversarial random oracle queries. In particular, extraction is straightline. We say that knowledge soundness is relaxed if the witness is for a relaxed relation  $\hat{R} \supseteq R$ . We refer to  $\hat{R}$  as the knowledge relation.

**Definition 12 (Relaxed Knowledge Soundness).** Let NIPS = (Prove, Ver) be a non-interactive proof system for a relation R and let  $\tilde{R} \supset R$  be an NP-relation. Let Ext be a PPT algorithm. Let A be an oracle algorithm and let

$$\begin{split} & \operatorname{\mathsf{Real}}_{\mathcal{A}}(\lambda) := \Pr\left[ b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_{\mathsf{Ver}}}(1^{\lambda}) \colon b = 1 \right], \\ & \operatorname{\mathsf{Ideal}}_{\mathcal{A}}(\lambda) := \Pr\left[ b \leftarrow \mathcal{A}^{\mathsf{H},\mathcal{O}_{\mathsf{Ext}}}(1^{\lambda}) \colon b = 1 \right]. \end{split}$$

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

- $\mathcal{O}_{\mathsf{Ver}}(\mathbf{x}, \pi)$ : Return  $\mathsf{Ver}(\mathbf{x}, \pi)$ .
- $-\mathcal{O}_{\mathsf{Ext}}(\mathbb{x},\pi)$ : If  $\mathsf{Ver}(\mathbb{x},\pi) = 1$  and  $(\mathbb{x},\mathbb{w}) \notin \tilde{\mathsf{R}}$  for  $\mathbb{w} \leftarrow \mathsf{Ext}(\mathcal{Q},\mathbb{x},\pi)$ , return 0. Else, return 1. Here, Q denotes the set of A's H queries.

The advantage of  $\mathcal{A}$  against knowledge soundness is  $\mathsf{AdvKS}^{\mathsf{NIPS},\tilde{\mathsf{R}}}_{\mathcal{A}}(\lambda) \coloneqq |\mathsf{Real}_{\mathcal{A}}(\lambda) - \mathsf{Ideal}_{\mathcal{A}}(\lambda)|$ . We say that Ext is a knowledge extractor for NIPS and knowledge relation  $\tilde{\mathsf{R}}$ , if for every PPT algorithm  $\mathcal{A}$ , the advantage AdvKS<sup>NIPS,  $\tilde{\mathsf{R}}(\lambda)$ </sup> is negligible in  $\lambda$ . We say that NIPS is knowledge sound, if there is a knowledge extractor for NIPS.

*Remark 1.* Any non-interactive proof system meeting above requirements is sufficient for our blind signature construction, and we present it using the proof system in a black-box way. For concreteness, we will use a variant of the Fischlin transformation [Fis05, Ks22], which detail in Appendix C.

### 3 Signatures based on the Boneh-Boyen IBE

It is well-known that signatures can generically be constructed from identity-based encryption (IBE) [BF01]. Our starting point towards constructing blind signatures is the Boneh-Boyen identity-based encryption scheme [BB04]. Note that without any modification this scheme would rely on pairings, and so would the derived signature scheme. Here, we provide a pairing-free variant of this signature scheme. As this scheme is the basis for our partially blind signature, we also provide a common message  $\tau \in \{0, 1\}^*$  as parameter.

**Overview.** Let  $\mathsf{H}_{\mathsf{M}} \colon \{0,1\}^* \to \mathbb{Z}_p$  be a random oracle. For any  $m \in \{0,1\}^*$ , denote by  $\overline{m} \coloneqq \mathsf{H}_{\mathsf{M}}(m)$ . A signature on a message  $m \in \{0,1\}^*$  consists of two group elements  $S_1$  and  $S_2$  such that

$$S_1 = uV + s(\overline{m}U + H), \quad S_2 = sG,$$
(2)

where  $V, H, U = uG \in \mathbb{G}$  are part of the public key and  $s \in \mathbb{Z}_p$ . To verify such a signature without a pairing, signatures in our variant also contain a proof  $\pi$ , which informally shows that one of the following holds:

- (i)  $(S_1, S_2)$  satisfy Equation (2) for U = uG, or
- (ii)  $\mathbf{D} = \mathsf{H}_{\mathsf{ddh}}(\tau)$  is a DDH-tuple, where  $\mathsf{H}_{\mathsf{ddh}} \colon \{0,1\}^* \to \mathbb{G}^2$  is a random oracle for common message  $\tau \in \{0,1\}^*$ .

Point (ii) is technically not required for the signature scheme itself but will be useful for the security proof of our (partially) blind signature construction (cf. Section 4), where it allows simulating the signer.

**Notation.** To improve readability, we introduce two functions below, where the reader should think of the element X as representing  $X = \overline{m} \cdot U + H$ . We define a function that captures the statement (i). For  $V \in \mathbb{G}$ , we define  $\phi_{G,V}^{\mathsf{BB}} \colon \mathbb{G} \times \mathbb{Z}_p^2 \to \mathbb{G}^3$  as follows:

$$\phi_{G,V}^{\mathsf{BB}}(X,(s,u)) = \begin{pmatrix} u \cdot V + s \cdot X \\ s \cdot G \\ u \cdot G \end{pmatrix}.$$
(3)

If (X, G) are clear from the context, we write  $\phi_0 = \phi_{G,V}^{\mathsf{BB}}$  for short. Note that  $\phi_0(X, \cdot)$  is linear for fixed input X. We also define  $\mathsf{R}_{\mathsf{bb}}$  with induced language  $\mathscr{L}_{\mathsf{bb}}$  as follows:

$$\mathsf{R}_{\mathsf{bb}} \coloneqq \left\{ (\mathfrak{x}_0, \mathfrak{w}_0) \mid \mathbf{S} = \phi_0(X, (s, u)) \right\},$$
  
where  $\mathfrak{x}_0 = (G, V, X, \mathbf{S}) \in \mathbb{G}^6, \ \mathfrak{w}_0 = (s, u) \in \mathbb{Z}_p^3.$ 

We also define a linear function that captures statement (ii). That is, for  $D_1 \in \mathbb{G}$ , we define  $\phi_{D_1}^{\mathsf{DDH}} \colon \mathbb{Z}_p \to \mathbb{G}$  as follows:

$$\phi_{G,D_1}^{\mathsf{DDH}}(d_2) = \begin{pmatrix} d_2 \cdot G \\ d_2 \cdot D_1 \end{pmatrix}.$$
(4)

If  $D_1$  is clear from the context, we write  $\phi_1 = \phi_{G,D_1}^{\mathsf{DDH}}$  for short. Similarly, we define  $\mathsf{R}_{\mathsf{ddh}}$  with induced language  $\mathscr{L}_{\mathsf{ddh}}$  as follows:

$$\mathsf{R}_{\mathsf{ddh}} \coloneqq \{ (\mathfrak{X}_1, \mathfrak{W}_1) \mid (D_2, D_3) = \phi_1(d_2) \},$$
  
where  $\mathfrak{X}_1 = (G, D_1, D_2, D_3) \in \mathbb{G}^4, \ \mathfrak{W}_1 = d_2 \in \mathbb{Z}_p.$ 

#### 3.1 Construction

Let  $\Sigma_0 = (Init_0, Resp_0, Verify_0)$  and  $\Sigma_1 = (Init_1, Resp_1, Verify_1)$  be  $\Sigma$ -protocols with challenge space  $\mathbb{Z}_p$  for the relations  $R_{bb}$  and  $R_{ddh}$  defined above, respectively. We provide concrete instantiations of both  $\Sigma$ -protocols in Appendix B. Denote by Sim<sub>1</sub> the HVZK simulator of Verify<sub>1</sub>. Let  $H_{\Sigma}$ ,  $H_M$ ,  $H_{ddh}$  be random oracles mapping into  $\mathbb{Z}_p$ ,  $\mathbb{Z}_p$  and  $\mathbb{G}^2$ , respectively. We define the signature BBSig in the following.

BBSig: Pairing-free signature based on Boneh-Boyen IBE [BB04] - KeyGen $(1^{\lambda})$ : 1. Sample  $u \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and set  $U \coloneqq uG$ . Sample  $H, V, D_1 \stackrel{*}{\leftarrow} \mathbb{G}$ . 2. Output  $\mathsf{vk} \coloneqq (G, U, H, V, D_1)$  and  $\mathsf{sk} \coloneqq u$ . - Sign(sk,  $m, \tau$ ): 1. Set  $(D_2^{\tau}, D_3^{\tau}) \coloneqq \mathsf{H}_{\mathsf{ddh}}(\tau)$  and  $\mathbf{D}^{\tau} \coloneqq (D_1, D_2^{\tau}, D_3^{\tau})$ . 2. Set  $\overline{m} \coloneqq \mathsf{H}_{\mathsf{M}}(m)$  and  $X \coloneqq \overline{m}U + H$ . 3. Sample  $s \leftarrow \mathbb{Z}_p$  and set  $\mathbf{S} \coloneqq \phi_0(X, (s, u))$ . 4. Compute a proof  $\pi$  as follows: (a) Let  $(\mathbb{X}_0, \mathbb{X}_1)$  be as above <sup>*a*</sup> and set  $\mathbb{W}_0 := (s, u)$ . (b) Sample  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and set  $(\mathbf{A}_1, z_1) \leftarrow \mathsf{Sim}_1(\mathbf{x}_1, c_1)$ . (c) Run  $(\mathbf{A}_0, \mathsf{st}_0) \leftarrow \mathsf{Init}_0(\mathbb{x}_0, \mathbb{w}_0).$ (d) Set  $c \coloneqq \mathsf{H}_{\Sigma}((\mathbf{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m)$  and  $c_0 \coloneqq c - c_1$ . (e) Run  $\mathbf{z}_0 \leftarrow \mathsf{Resp}_0(\mathsf{st}_0, c_0)$ . (f) Set  $\pi := (\mathbf{A}_0, \mathbf{A}_1, c, c_0, \mathbf{z}_0, z_1).$ 5. Output  $\sigma_{\mathsf{bb}} \coloneqq (S_1, S_2, \pi)$ . - Verify(vk,  $m, \tau, \sigma_{bb}$ ): 1. Parse  $\sigma_{bb}$  as  $\sigma_{bb} = (S_1, S_2, \pi)$  and  $\pi$  as  $\pi = (\mathbf{A}_0, \mathbf{A}_1, c, c_0, \mathbf{z}_0, z_1)$ . 2. Set  $(D_2^{\tau}, D_3^{\tau}) \coloneqq \mathsf{H}_{\mathsf{ddh}}(\tau)$  and  $\mathbf{D}^{\tau} \coloneqq (D_1, D_2^{\tau}, D_3^{\tau})$ . 3. Let  $(\mathbf{x}_0, \mathbf{x}_1)$  be as above <sup>*a*</sup>. 4. Set  $\mathbf{S} \coloneqq (S_1, S_2, U), \overline{m} \coloneqq \mathsf{H}_\mathsf{M}(m)$  and  $X \coloneqq \overline{m}U + H$ . 5. Set  $c' := \mathsf{H}((\mathbf{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m)$  and  $c_1 := c' - c_0$ . 6. Output 0 if  $Verify_0(x_0, A_0, c_0, z_0) = 0$ . 7. Output 0 if  $Verify_1(x_1, A_1, c_1, z_1) = 0$ . 8. Otherwise, output 1. <sup>*a*</sup> That is,  $\mathfrak{x}_0 \coloneqq (G, V, X, \mathbf{S})$  and  $\mathfrak{x}_1 \coloneqq (G, \mathbf{D}^{\tau})$ .

Note that above,  $\pi$  functions essentially as proof for the disjunctive relation  $\mathsf{R}_{bb} \cup \mathsf{R}_{ddh}$ . Also, the first flow  $\mathbf{A}_0, \mathbf{A}_1$  can be omitted from the proof  $\pi$  since these values can be recomputed given  $(c, c_0, \mathbf{z}_0, z_1)$ .

## 3.2 Security Analysis

We provide a useful lemma which we employ in our proof of one-more unforgeability (cf. Theorem 1). Roughly, it shows that it is hard to output a tuple  $(S_1, S_2)$  such that  $(G, V, X_{\overline{m}^*}, S_1, S_2, U) \in \mathscr{L}_{bb}$ , where  $\overline{m}^* \in \mathbb{Z}_p$  is chosen selectively and  $X_{m^*} := \overline{m}^*U + H$ . This even holds if the adversary is given oracle access to an oracle that outputs  $(S_1, S_2)$  such that  $(G, V, X_{\overline{m}}, S_1, S_2, U) \in \mathscr{L}_{bb}$  for  $\overline{m} \neq \overline{m}^*$ . Note that this corresponds almost to selective unforgeability of BBSig except that the common message  $\tau$  and the proof  $\pi$  is ignored. This can be shown via the puncturing strategy from [BB04] and we provide a formal proof in Appendix D.

**Lemma 1 (Selective Security of BBSig).** For any algorithm  $\mathcal{A}$ , let  $\epsilon_{\mathcal{A}}^{\mathsf{BB}}$  be the probability that the following game outputs 1:

- 1. Run  $(\overline{m}^*, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}(1^{\lambda}).$
- 2. Sample  $u \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and set  $U \coloneqq uG$ .
- 3. Sample  $(H, V) \xleftarrow{\hspace{0.1cm} \$} \mathbb{G}$  and set  $X_{m^*} \coloneqq \overline{m}^* U + H$ .
- 4. Run  $(S_1^*, S_2^*) \leftarrow \mathcal{A}^{\mathcal{O}}(G, U, H, V, \mathsf{st}_{\mathcal{A}})$ , where  $\mathcal{O}$  is given as:
- $\mathcal{O}(\overline{m})$ : Output  $\perp$  if  $\overline{m} = \overline{m}^*$ . Otherwise, sample  $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ , set  $X_{\overline{m}} = \overline{m} \cdot U + H$ , and compute  $\mathbf{S} := \phi_0(X_{\overline{m}}, (s, u))$ . Then return  $(S_1, S_2)$ .
- 5. Set  $\mathbb{x}_0^* \coloneqq (G, V, X_{\overline{m}^*}, S_1^*, S_2^*, U)$  and output 1 if and only if  $\mathbb{x}_0^* \in \mathscr{L}_{bb}$ .

Then, for any PPT algorithm  $\mathcal{A}$ , there exists some PPT algorithm  $\mathcal{B}$  with running time similar to  $\mathcal{A}$  such that

$$\epsilon_{\mathcal{A}}^{\mathsf{BB}} \leq \mathsf{Adv}\mathsf{CDH}_{\mathcal{B}}^{\mathbb{G}}(\lambda).$$

## 4 Non-Blind Interactive Signing Protocol

With BBSig signatures as introduced in Section 3 at hand, we now move closer to our blind signature construction. The goal of this section is to define an interactive protocol for obtaining BBSig signatures from the Signer. More precisely, what we construct here is a blind signature scheme that satisfies one-more unforgeability, but is *not* blind at this point. We stress that consequently, the protocol presented in this section is not our final blind signature scheme. We will take care of blindness and present our final signing protocol in Section 5.



Fig. 1. An (unblinded) signing session for PreBS for message  $m \in \{0,1\}^*$  and common message  $\tau \in \{0,1\}^*$ . The signer aborts (i.e., outputs  $\perp$ ) if for condition C, req C is evaluated for false C. Recall that  $\mathsf{H}_{\Sigma}$  maps into  $\mathbb{Z}_p$  and that  $\mathbf{D}^{\tau} := (D_1, D_2^{\tau}, D_3^{\tau})$  for  $(D_2^{\tau}, D_3^{\tau}) := \mathsf{H}_{\mathsf{ddh}}(\tau)$ . Also, recall that  $\mathsf{vk} = (G, U, H, V, D_1) \in \mathbb{G}^5$  and  $\mathsf{sk} = u \in \mathbb{Z}_p$ .

#### 4.1 Construction

Let  $NIPS_{Ped} = (NIPS_{Ped}.Prove^{H_{Ped}}, NIPS_{Ped}.Ver^{H_{Ped}})$  be a NIPS proof system with random oracle  $H_{Ped} : \{0,1\}^* \to \mathcal{Y}_{Ped}$  with image space  $\mathcal{Y}_{Ped}$  for the relation

$$\mathsf{R}_{\mathsf{Ped}} \coloneqq \{(\mathbf{x}, \mathbf{w}) \mid C = \overline{m}U + tG\}, \text{ where } \mathbf{x} = (C, U, G), \ \mathbf{w} = (\overline{m}, t).$$
(5)

In addition, our construction makes use of random oracles  $H_M: \{0,1\}^* \to \mathbb{Z}_p$ , and  $H_{\Sigma}: \{0,1\}^* \to \mathbb{Z}_p$ , and  $H_{\mathsf{ddh}}: \{0,1\}^* \to \mathbb{G}^2$ . We now present our construction.

# PreBS: Unblinded interactive signing protocol for BBSig signatures

- KeyGen $(1^{\lambda})$ : Output  $(vk, sk) \leftarrow BBSig.KeyGen<math>(1^{\lambda})$ . - S $(sk, \tau) \longleftrightarrow U(vk, m, \tau)$ : The signing protocol proceeds in 4 moves and is given in Figure 1. - Verify $(vk, m, \tau, \sigma_{bb})$ : Output  $b \leftarrow BBSig.Verify}(vk, m, \tau, \sigma_{bb})$ .

Remark 2 (Optimizations). The signer can omit sending  $T_3 = U$ , as this value is specified in vk. Also, as discussed in Section 3, the values  $(\mathbf{A}_0, A_1)$  can be omitted from the proof  $\pi$  within the output signature  $\sigma_{bb}$ .

#### 4.2 Security Analysis

Correctness follows from inspection and we will give a correctness proof for our final scheme later. As already mentioned, PreBS is not blind. Here, we show one-more unforgeability. As we instantiate  $NIPS_{Ped}$  (cf. Appendix C) with a relaxed knowledge sound NIPS, the extractor  $Ext_{Ped}$  only extracts a witness for the relaxed knowledge relation

$$\mathbf{\hat{R}}_{\mathsf{Ped}} \coloneqq \{(\mathbf{x}, \mathbf{w}) \mid \mathbf{w}G = U \lor (\mathbf{x}, \mathbf{w}) \in \mathsf{R}_{\mathsf{Ped}}\}, \quad \text{where } \mathbf{x} = (C, U, G).$$
(6)

In particular, we show that if  $NIPS_{Ped}$  is (straightline) knowledge sound with knowledge relation  $\tilde{R}_{Ped}$ , then PreBS is one-more unforgeable under the DDH assumption.

**Theorem 1 (One-More Unforgeability).** Denote by p the order of group  $\mathbb{G}$ . For any PPT adversary  $\mathcal{A}$  that causes at most Q random oracle queries, there are reductions  $\mathcal{A}_{KS}$ ,  $\mathcal{A}_{DL}$ ,  $\mathcal{A}_{DDH}$ , and  $\mathcal{A}_{CDH}$  with running time similar to  $\mathcal{A}$  such that

$$\begin{split} \mathsf{Adv}\mathsf{OMUF}_{\mathcal{A}}^{\mathsf{PreBS}}(\lambda) &\leq \quad \frac{4 \cdot Q^2 + 3 \cdot Q + 4}{p - 1} + \mathsf{AdvKS}_{\mathcal{A}_{\mathsf{KS}}}^{\mathsf{NIPS}_{\mathsf{Ped}},\tilde{\mathsf{R}}_{\mathsf{Ped}}}(\lambda) + \mathsf{AdvDL}_{\mathsf{AdvDL}}^{\mathbb{G}}(\lambda) \\ &+ Q^2 \left( 10 \cdot \mathsf{AdvDDH}_{\mathcal{A}_{\mathsf{DDH}}}^{\mathbb{G}}(\lambda) + \mathsf{AdvCDH}_{\mathcal{A}_{\mathsf{CDH}}}^{\mathbb{G}}(\lambda) \right). \end{split}$$

Before we give the proof, let us remark that the quadratic loss is due to partial blindness. For standard blindness, there is only a factor Q before the sum instead of  $Q^2$  and  $4 \cdot Q^2$  is replaced with  $4 \cdot Q$ .

*Proof.* Let  $\mathcal{A}$  be a PPT adversary against one-more unforgeability of PreBS. Denote by  $Q_{\Sigma}, Q_{M}$ ,  $Q_{ddh}, Q_{Ped}$  the number of oracle queries to  $H_{\Sigma}, H_{M}, H_{ddh}, H_{Ped}$ , respectively, including the queries made by the game (*e.g.*, during signing queries or during signature verification). Denote by  $Q_{S}$  the number of  $\mathcal{A}$ 's signing queries. Denote by  $\mathsf{Ext}_{\mathsf{Ped}}$  the extractor of  $\mathsf{NIPS}_{\mathsf{Ped}}$ .

We proceed with a sequence of games. For each game Game i, we denote the probability that the game outputs 1 by  $\varepsilon_i$ .

Game 0 (Honest). This game is the real one-more unforgeability experiment for scheme PreBS and adversary  $\mathcal{A}$  with random oracles  $\mathsf{H}_{\mathsf{ddh}}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\mathsf{Ped}}$  and  $\mathsf{H}_{\Sigma}$ . The game first samples  $\mathsf{vk} = (G, U, H, V, D_1)$  and sk via PreBS.KeyGen. The adversary  $\mathcal{A}$  obtains verification key vk as input and access to the random oracles, as well as both signing oracles  $\mathcal{O}_{\mathsf{S}_1}, \mathcal{O}_{\mathsf{S}_2}$ , and outputs a common message  $\tau^*$  and forgeries  $(m_j^*, \sigma_j^*)_{j \in [k]}$ . The game outputs 1 iff  $\mathcal{O}_{\mathsf{S}_2}$  was queried at most k - 1 times with common message  $\tau^*$ , all messages  $(m_j^*)_{j \in [k]}$  are pairwise-distinct, and all signatures verify (i.e.,  $\mathsf{Verify}(\mathsf{vk}, m_j^*, \tau^*, \sigma_j^*) = 1$ ). Note that each signing session is identified by a session identifier sid which is provided as input in  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$ . Recall that the signing oracles behave as follows:

- $\mathcal{O}_{\mathsf{S}_1}(\mathsf{sid}, C, \pi_{\mathsf{Ped}})$ : Check the proof  $\pi_{\mathsf{Ped}}$  and abort if  $\mathsf{NIPS}_{\mathsf{Ped}}.\mathsf{Ver}^{\mathsf{H}_{\mathsf{Ped}}}(\mathbb{x}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}}) = 0$  for  $\mathbb{x}_{\mathsf{Ped}} \coloneqq (C, U, G)$ . Sample  $s \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and set  $\mathbb{w}_0 \coloneqq (s, \mathsf{sk})$ . Next, set  $\mathbf{T} \coloneqq \phi_0(X_C, \mathbb{w}_0)$  for  $X_C \coloneqq C + H$  and  $\mathbf{D}^{\tau} \coloneqq (D_1, \mathsf{H}_{\mathsf{ddh}}(\tau))$ . Prepare both statements  $\mathbb{x}_0^C \coloneqq (G, V, X_C, \mathbf{S})$  and  $\mathbb{x}_1 \coloneqq (G, \mathbf{D}^{\tau})$ . For the  $\Sigma_1$  proof, sample  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and simulate  $(\mathbf{A}_1, z_1) \leftarrow \mathsf{Sim}_1(\mathbb{x}_1, c_1)$ . For the  $\Sigma_0$  proof, sample first flow  $(A_0^C, \mathsf{st}_0) \leftarrow \mathsf{Init}_0(\mathbb{x}_0^C, \mathbb{w}_0)$ . Store  $(\mathbf{z}_1, c_1, \mathsf{st}_0)$  as state for session sid and output  $(\mathbf{T}, \mathbf{A}_0^C, \mathbf{A}_1)$ .
- $\mathcal{O}_{S_2}(\mathsf{sid}, c)$ : Retrieve  $(\mathbf{z}_1, c_1, \mathsf{st}_0)$  from the state for sid (and abort if this is not possible). Compute challenge  $c_0 \coloneqq c c_1$  and response  $\mathbf{z}_0 \leftarrow \mathsf{Resp}_0(\mathsf{st}_0, c_0)$  for the  $\Sigma_0$  proof. Output  $(\mathbf{z}_0, z_1, c_0)$

For convenience, we provide a detailed description in Figure 3 in Appendix E. By definition, we have

AdvOMUF<sup>PreBS</sup><sub>$$\mathcal{A}$$</sub> $(\lambda) = \varepsilon_0.$ 

**Game 1 (Abort if H\_M collision).** The game aborts if there are collisions for  $H_M$ . More precisely, it aborts if there are queries  $x \neq x'$  such that  $H_M(x) = H_M(x')$ . A standard birthday-bound argument yields that

$$|\varepsilon_0 - \varepsilon_1| \le \frac{Q_{\mathsf{M}}^2}{p}.$$

Game 2 (Extract  $(\overline{m}, t)$  from C). We change the first signer oracle  $\mathcal{O}_{S_1}$ . Namely, whenever the adversary sends a commitment C with a proof  $\pi_{\mathsf{Ped}}$  in its first message of a signing interaction, the game uses the extractor of NIPS<sub>Ped</sub> to extract a preimage  $(\overline{m}, t) \in \mathbb{Z}_p^2$  for C, and aborts its entire execution if  $\pi_{\mathsf{Ped}}$  verifies but extraction fails. In more detail, we modify the first part of the signer oracle (oracle  $\mathcal{O}_{S_1}$ ) as follows. Initially, it proceeds as in Game 1 until the check NIPS<sub>Ped</sub>. Ver<sup>H<sub>Ped</sub>( $\mathbb{X}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}}$ ) = 1. If the check fails, it outputs  $\bot$  as before. Else, the game sets  $\mathbb{W}_{\mathsf{Ped}} \leftarrow \mathsf{Ext}_{\mathsf{Ped}}(\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}, \pi_{\mathsf{Ped}})$  and parses  $(\overline{m}, t) \coloneqq \mathbb{W}_{\mathsf{Ped}}$ . (Recall that  $\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}$  denotes the queries to  $\mathsf{H}_{\mathsf{Ped}}$  made by  $\mathcal{A}$  and the game.) The game aborts its entire execution if parsing  $\mathbb{W}_{\mathsf{Ped}}$  fails or  $C \neq \overline{m}U + tG$ .</sup>

Let us analyze the advantage of  $\mathcal{A}$  in Game 2. Roughly, we need to ensure that extraction succeeds and that the extracted witness  $w_{\mathsf{Ped}}$  is an opening for C, i.e.,  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \in \mathsf{R}_{\mathsf{Ped}}$ . But because the soundness relation is *relaxed*, it is possible that  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \notin \mathsf{R}_{\mathsf{Ped}}$ . Instead, the extracted witness for relation  $\tilde{\mathsf{R}}_{\mathsf{Ped}}$  might be the discrete logarithm u of U (cf. Equation (6)). Since the adversary  $\mathcal{A}$  provides the proof  $\pi_{\mathsf{Ped}}$  and u is kept (computationally) hidden from  $\mathcal{A}$ , this should occur with negligible probability. But because u is also required to simulate the signing oracles, we cannot immediately reduce to the DLOG assumption, and a few intermediate games are required. Roughly, starting with Game 1, we first move to a game where the signing oracles can be simulated without the secret key  $\mathsf{sk} = u$ . Then, we add an abort condition if  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \notin \mathsf{R}_{\mathsf{Ped}}$ , where we can now upper bound the abort probability under the DL assumption. Finally, we revert back the changes (keeping the abort condition), and obtain Game 2. This is formalized in Lemma 2 below. We provide a proof in Appendix D.

**Lemma 2.** There are reductions  $\mathcal{A}_{KS}$ ,  $\mathcal{A}^i_{DDH}$  for  $i \in [2]$  and  $\mathcal{A}_{DL}$  with running time close to that of  $\mathcal{A}$  such that

$$|\varepsilon_1 - \varepsilon_2| \leq \mathsf{AdvKS}_{\mathcal{A}_{\mathsf{KS}}}^{\mathsf{NIPS}_{\mathsf{Ped}}, \tilde{\mathsf{R}}_{\mathsf{Ped}}}(\lambda) + \mathsf{AdvDL}_{\mathsf{AdvDL}}^{\mathbb{G}}(\lambda) + \frac{4}{p-1} + \sum_{i \in [2]} \mathsf{AdvDDH}_{\mathcal{A}_{\mathsf{DDH}}^i}^{\mathbb{G}}(\lambda).$$

Game 3 (Guess  $\tau^*$ ). Informally, we now guess the first query to  $\mathsf{H}_{\mathsf{ddh}}$  such that  $\tau^*$  is provided as input. More formally, the game samples  $q_{\tau}^* \stackrel{s}{\leftarrow} [Q_{\mathsf{ddh}}]$  at its start. When the adversary outputs its forgeries with common message  $\tau^*$  in the end, the challenger additionally checks whether  $\tau^*$  was queried for the first time to  $\mathsf{H}_{\mathsf{ddh}}$  on the  $q_{\tau}^*$ -th query to  $\mathsf{H}_{\mathsf{ddh}}$ . If not, the game aborts its execution.

Recall that the challenger's  $\mathsf{H}_{\mathsf{ddh}}$  queries count towards  $Q_{\mathsf{ddh}}$ . Furthermore, note that the challenger sets  $(D_2^{\tau^*}, D_3^{\tau^*}) \coloneqq \mathsf{H}_{\mathsf{ddh}}(\tau^*)$  when it verifies  $\mathcal{A}$ 's forgeries, so such a query exists. Next, observe that the guess  $q_{\tau}^*$  is hidden from  $\mathcal{A}$ , and so the probability that the challenger guesses this query is at least  $1/Q_{\mathsf{ddh}}$ , even conditioned on Game 2 outputting 1. We get that

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

Observe that the game evaluates  $\mathsf{H}_{\mathsf{ddh}}$  on  $\tau$  in the first signer oracle  $\mathcal{O}_{\mathsf{S}_1}$  to compute  $(D_2^{\tau}, D_3^{\tau}) := \mathsf{H}_{\mathsf{ddh}}(\tau)$ . Thus, if  $\mathcal{A}$  succeeds, the game knows the forgery's common message  $\tau^*$  when the first query to  $\mathcal{O}_{\mathsf{S}_1}$  with  $\tau^*$  is made. This will be useful later.

Game 4 (Guess unsigned  $\overline{m}^*$  in forgery). We guess the first query  $q_m^*$  to  $H_M$  such that the following two conditions hold:

- 1. The input  $m_{q_m^*}$  to the  $q_m^*$ -th  $H_M$  query is part of  $\mathcal{A}$ 's forgeries.
- 2. No session with common message  $\tau^*$  is completed if  $\overline{m} = H_M(m_{q_m^*})$  is extracted from the commitment C (see Game 2).

Again, the game aborts its execution if the guess was incorrect. If  $\mathcal{A}$  is successful, then  $\mathcal{A}$ 's forgeries  $(m_j^*, \sigma_j^*)_{j \in [k]}$  with common message  $\tau^*$  contain k distinct messages. Because we have ruled out collisions for  $\mathsf{H}_{\mathsf{M}}$  (see Game 1), the hashed messages  $(\overline{m}_j^*)_{j \in [k]}$  are also pairwise distinct, where  $\overline{m}_j^* = \mathsf{H}_{\mathsf{M}}(m_j^*)$ . Furthermore, there are at most k-1 completed sessions with common message  $\tau^*$  and each corresponding call to the first oracle  $\mathcal{O}_{\mathsf{S}_1}$ , exactly one message  $\overline{m} \in \mathbb{Z}_p$  is extracted from C via  $\mathsf{Ext}_{\mathsf{Ped}}$ . In conclusion, one of the k distinct  $\overline{m}_j^*$  was never extracted from C within a completed session. Thus, there is an index  $j \in [k]$  such that  $m_{q_m^*} \coloneqq m_j^*$  fulfils the above conditions (also counting the challenger's queries).

The probability that the challenger guesses  $q_m^*$  correctly is  $1/Q_M$  and the guess  $q_m^*$  is hidden from the adversary. Thus, we have

$$\varepsilon_3 \leq Q_{\mathsf{M}} \cdot \varepsilon_4.$$

In the following, we denote by  $\overline{m}^* := \mathsf{H}_{\mathsf{M}}(m_{q_m^*})$ . Note that if  $\mathcal{A}$  is successful, we can assume that  $\overline{m}^*$  is known by the game from the start on <sup>10</sup>. Also, we stress that the game aborts only if both signer oracles  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$  are executed with extracted  $\overline{m}^*$  and common message  $\tau^*$ , i.e., such a signing interaction is completed. In particular, it is possible that  $\overline{m}^*$  is extracted in  $\mathcal{O}_{\mathsf{S}_1}$  if the session will not be completed.

Game 5 (Sample DDH tuples if  $\tau \neq \tau^*$ ). From now on, the game samples real DDH tuples in  $H_{ddh}$  except in the  $q_{\tau}^*$ -th query. That is, the game now holds an initially empty table  $T_{ddh}[\cdot] := \bot$ . Whenever random oracle  $H_{ddh}$  is queried on an input  $\tau$  and the hash value is not yet defined, the game samples  $d_2 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and sets  $(D_2^{\tau}, D_3^{\tau}) := (d_2G, d_2D_1)$  instead of  $(D_2, D_3) \stackrel{*}{\leftarrow} \mathbb{G}^2$ . Additionally, witness  $d_2$  is stored in the table, i.e.,  $T_{ddh}[\tau] := d_2$ . Importantly, the output on the  $q_{\tau}^*$ -th  $H_{ddh}$  query (i.e.,  $(D_2^{\tau^*}, D_3^{\tau^*})$ ) and all subsequent queries on the same input remain unchanged. Note that by design, we have  $\mathbf{D}^{\tau} \in \mathscr{L}_{ddh}$  for  $\tau \neq \tau^*$ . Clearly, there is a reduction  $\mathcal{B}_1$  on Q-DDH with  $Q = Q_{ddh}$  with running time similar to  $\mathcal{A}$  such that

$$|\varepsilon_4 - \varepsilon_5| \leq \mathsf{AdvQDDH}_{\mathcal{B}_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda).$$

Game 6 (Use DDH witness for  $\Sigma_1$  if  $\tau \neq \tau^*$ ). Now, the game computes the  $\Sigma_1$  transcript  $(\mathbf{A}_1, c, z_1)$  for  $\tau \neq \tau^*$  via the witness  $\mathsf{T}_{\mathsf{ddh}}[\tau]$ . More precisely, in  $\mathcal{O}_{\mathsf{S}_1}$  with  $\tau \neq \tau^*$ , the game samples  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and  $(\mathbf{A}_1, \mathsf{st}_1) \leftarrow \mathsf{lnit}_1(\mathbb{X}_1, \mathbb{W}_1)$ , where  $\mathbb{W}_1 \coloneqq \mathsf{T}_{\mathsf{ddh}}[\tau]$  and  $\mathbb{X}_1 \coloneqq (G, \mathbf{D}^{\tau})$ . In  $\mathcal{O}_{\mathsf{S}_2}$  with  $\tau \neq \tau^*$ , the game computes  $z_1 \leftarrow \mathsf{Resp}_1(\mathsf{st}_1, c_1)$ .

Recall that in Game 5, the game samples  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and  $(\mathbf{A}_1, z_1) \leftarrow \mathsf{Sim}_1(\mathbb{x}_1, c_1)$ . It follows by perfect HVZK of  $\Sigma_1$ , that the  $\Sigma_1$  transcripts  $(\mathbf{A}_1, c_1, z_1)$  in Game 5 and Game 6 are identically distributed. Thus, we have

$$\varepsilon_5 = \varepsilon_6.$$

Game 7 (Simulate  $\Sigma_0$  if  $\tau \neq \tau^*$ ). The game now simulates the  $\Sigma_0$  transcript  $(\mathbf{A}_0, c_0, \mathbf{z}_0)$  via HVZK in  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$  for all  $\tau \neq \tau^*$ . In more detail, if  $\tau \neq \tau^*$  in  $\mathcal{O}_{\mathsf{S}_1}$ , the game computes  $\mathbf{A}_0^C$  via

$$c_0 \stackrel{*}{\leftarrow} \mathbb{Z}_p, \ (\mathbf{A}_0^C, \mathbf{z}_0) \leftarrow \mathsf{Sim}_0(\mathbf{x}_0, c_0).$$

In  $\mathcal{O}_{S_2}$  for  $\tau \neq \tau^*$ , the game sets  $c_1 \coloneqq c - c_0$  and outputs  $\mathbf{z}_0$  from  $\mathcal{O}_{S_1}$ . The other response  $z_1$  is computed via  $w_1$  as introduced in Game 6.

<sup>&</sup>lt;sup>10</sup> The game samples  $\overline{m}^*$  at random at the beginning of the game and outputs  $\overline{m}^*$  in the  $q_m^*$ -th query to  $H_M$ .

Recall that in Game 6, the game sets  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and  $c_0 \coloneqq c - c_1$ . Thus, the challenges  $(c_0, c_1)$  follow the same distribution in Game 6 and Game 7. Also, observe that in Game 6, the  $\Sigma_0$  transcript is generated honestly. Thus, by perfect HVZK of  $\Sigma_0$ , we have that

$$\varepsilon_6 = \varepsilon_7.$$

Let us take a closer look at the signer oracle in Game 7 for two specific cases, namely if (1)  $\tau \neq \tau^*$ , or (2)  $\tau = \tau^*$  and the game has extracted  $m = \overline{m}^*$  from C. Recall that in the signer oracle  $\mathcal{O}_{\mathsf{S}_1}$ , the game defines the vector  $\mathbf{T} \coloneqq \phi_0(X_C, w_0)$  for  $X_C = C + H$  and  $w_0 = (s, \mathsf{sk})$  where  $s \in \mathbb{Z}_p$  is random. Precisely, this means that  $\mathbf{T} = (\mathsf{sk}V + s(C + H), sG, U)$ . Now, observe that if (1) occurs, then due to the change in Game 7, the challenger uses the witness  $w_0 = (s, \mathsf{sk})$  only to sample  $\mathbf{T}$ . Similarly, if (2) occurs, then the challenger uses the witness  $w_0 = (s, \mathsf{sk})$  to sample  $\mathbf{T}$  and in  $\mathcal{O}_{\mathsf{S}_2}$  to compute  $\mathbf{z}_0$ . Due to the abort condition in Game 4,  $\mathcal{O}_{\mathsf{S}_2}$  is never invoked in case (2). In conclusion, if (1) or (2) occurs, the challenger only uses  $w_0$  to sample  $\mathbf{T}$  in the signing oracles.

**Game 8 (Send random T in some sessions).** We change the signer oracle  $\mathcal{O}_{S_1}$  again, for the cases (1) and (2) mentioned above. Namely, recall that until now, the signer oracle defined the vector  $\mathbf{T} := \phi_0(X_C, \mathbf{w}_0)$ . In this game,  $\mathbf{T}$  is sampled differently. Namely, if (1) or (2) occurs, then the game samples  $T_1, T_2 \stackrel{s}{\leftarrow} \mathbb{G}$  at random and sets  $T_3 := U$ . Intuitively, since (H, sG, sH) form Diffie-Hellman tuples and are included in the definition of  $\mathbf{T}$  in Game 7, replacing sH by a random element should be indistinguishable and make the first component of  $\mathbf{T}$  random.

More formally, we construct a reduction  $\mathcal{B}_2$  that breaks Q-DDH if  $\mathcal{A}$  can distinguish between Game 7 and Game 8. The reduction  $\mathcal{B}_2$  obtains tuples  $(G, H_1, (H_{2,i}, H_{3,i})_{i \in [Q_S]})$  from the Q-DDH game and samples  $\mathsf{vk} = (G, U, H, V, D_1)$  as in Game 7, except that  $H \coloneqq H_1$ . Then,  $\mathcal{B}_2$  proceeds to simulate Game 7 to adversary  $\mathcal{A}$  with the aforementioned  $\mathsf{vk}$  except that in the *i*-th invocation of  $\mathcal{O}_{\mathsf{S}_1}$ , it also checks whether either case (1) or case (2) occurs. If so,  $\mathcal{A}$  sets

$$\mathbf{T} \coloneqq (uV + (m \cdot \mathsf{sk})H_{2,i} + tH_{2,i} + H_{3,i}, H_{2,i}, U),$$

else it sets  $\mathbf{T} = \phi_0(X_C, \mathbf{w}_0)$  for  $\mathbf{w}_0 = (s, \mathsf{sk})$  and random  $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$  as before. As discussed above,  $\mathcal{B}_2$  can proceed as before. That is,  $\mathcal{B}_2$  computes  $\mathbf{A}_0^C$  and  $\mathbf{A}_1$  as in Game 7 and outputs  $(\mathbf{T}, \mathbf{A}_0^C, \mathbf{A}_1)$ . Also,  $\mathcal{B}_2$  simulates  $\mathcal{O}_{S_2}$  as in Game 7. When  $\mathcal{A}$  outputs its forgeries,  $\mathcal{B}_2$  outputs  $b' \coloneqq 1$  if  $\mathcal{A}$  succeeds, and  $b' \coloneqq 0$  otherwise.

Note that the verification key vk that is output by  $\mathcal{B}_2$  is identically distributed to vk in Game 7 and Game 8. Further, if we have H = hG,  $H_{2,i} = s_iG$  and  $H_{3,i} = (h \cdot s_i)G$  for all  $i \in [Q_S]$ , then if event (1) or (2) occurs in the *i*-th  $\mathcal{O}_{S_1}$  query, it holds that

$$T_{1} = uV + (m \cdot \mathsf{sk})H_{2,i} + tH_{2,i} + H_{3,i}$$
  
=  $uV + (s_{i} \cdot m)U + (s_{i} \cdot t)G + (h \cdot s_{i})G$   
=  $uV + s_{i}(mU + tG) + s_{i}H$   
=  $uV + s_{i}(C + H).$ 

and  $T_2 = s_i G, T_3 = U$ . This is exactly the distribution of **T** in Game 7. Otherwise, we have  $H = hG, H_{2,i} = s_i G$  and  $H_{3,i} \stackrel{\$}{\leftarrow} \mathbb{G}$  for all  $i \in [Q_S]$ . If event (1) or (2) occurs in the *i*-th  $\mathcal{O}_{S_1}$  query, then **T** follows the distribution of **T** in Game 8, as  $H_{3,i}$  functions as a one-time pad. In case neither event (1) nor (2) occurs, **T** follows the distribution in Game 7 and Game 8 in  $\mathcal{O}_{S_1}$  by design. The running time of  $\mathcal{B}_2$  is roughly that of  $\mathcal{A}$ . In conclusion, we have

$$|\varepsilon_7 - \varepsilon_8| \leq \mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{B}_2}(Q_S, \lambda).$$

Game 9 (Abort if forgeries not in  $\mathscr{L}_{bb}$ ). Now, we make the game abort if one of the adversary's forgeries  $\sigma_j^* = (S_{1,j}^*, S_{2,j}^*, \pi_j)$  for message  $m_j^*$  satisfies  $(G, V, X_j^*, S_{1,j}^*, S_{2,j}^*, U) \notin \mathscr{L}_{bb}$  with  $X_j^* = \overline{m}_j^* U + H$ . Here,  $\overline{m}_j^* := \mathsf{H}_{\mathsf{M}}(m_j^*)$  denotes the hashed message as before. In more detail, this is done efficiently as follows: The game initially samples  $h \stackrel{\$}{\leftarrow} \mathbb{Z}_p$  and sets H = hG to set up the verification key vk. Further, when  $\mathcal{A}$  outputs its forgeries  $(m_j^*, \sigma_j^*)_{j \in [k]}$  with common message  $\tau^*$ , the game parses  $\sigma_j^* = (S_{1,j}^*, S_{2,j}^*, \pi_j)$ . Then, the game checks that for all  $j \in [k]$ , it holds that

$$S_{1,j}^* = uV + (\overline{m}_j^* \cdot u)S_{2,j}^* + hS_{2,j}^*.$$
<sup>(7)</sup>

This check is efficient using knowledge of h and u. The game aborts if the check fails. Otherwise it proceeds as before.

Denote  $\mathbf{x}_1^* \coloneqq (G, \mathbf{D}^{\tau^*})$  and  $\mathbf{x}_{0,j}^* \coloneqq (G, V, X_j^*, S_{1,j}^*, S_{2,j}^*, U)$ . Roughly, we have  $\mathbf{x}_1 \notin \mathscr{L}_{\mathsf{ddh}}$  except with probability 1/p. Then, soundness of  $\pi_j$  ensures except with negligible probability that  $\mathbf{x}_{0,j} \in \mathscr{L}_{\mathsf{bb}}$  which is equivalent to Equation (7).

More formally, let us analyze the probability that for some  $j \in [k]$ , Equation (7) does not hold. First, we proof two useful claims. The first claim follows from soundness of the Fiat-Shamir transformation and the second claim links Equation (7) with  $\mathscr{L}_{bb}$ .

**Proposition 1.** For every  $H_{\Sigma}$  query  $((\mathbf{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m)$  with  $\mathbf{x}_0 \notin \mathcal{L}_{bb}$  and  $\mathbf{x}_1 \notin \mathcal{L}_{ddh}$ , there exists  $(c_0, c_1, \mathbf{z}_0, z_1)$  such that

$$tr_0 \coloneqq (\mathbf{A}_0, c_0, \mathbf{z}_0) \text{ is valid for } \mathbf{x}_0 \tag{8}$$

$$tr_1 \coloneqq (\mathbf{A}_1, c_1, z_1) \text{ is valid for } \mathbf{x}_1$$

$$\tag{9}$$

$$c^* \coloneqq \mathsf{H}_{\Sigma}((\mathbf{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m_i^*) = c_0 + c_1 \tag{10}$$

with probability at most 1/p.

Proof (Proposition 1). Observe that due to special soundness of  $\Sigma_1$  and because  $\mathfrak{x}_1 \notin \mathscr{L}_{ddh}$ , there is at most one challenge  $c_1 \in \mathbb{Z}_p$  such that there exists a response  $z_1$  with valid transcript  $tr_1 = (\mathbf{A}_1, c_1, z_1)$  for  $\mathfrak{x}_1$ . Similarly, since  $\mathfrak{x}_0 \notin \mathscr{L}_{bb}$ , the same argument applies: There exists exactly one challenge  $c_0$  such that there exists a response  $z_1$  with valid transcript  $tr_0 = (\mathbf{A}_0, c_0, \mathbf{z}_0)$ . Thus, the pair  $(c_0, c_1)$  is determined by  $(\mathfrak{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}$  due to Equations (8) and (9). Further, because  $(\mathfrak{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}$  is part of the input of the  $\mathsf{H}_{\Sigma}$  query that determines  $c^*$ , the value  $c^*$  is distributed uniformly and independently from  $(c_0, c_1)$ . Then, the probability that Equation (10) holds is at most 1/p.

**Proposition 2.** Equation (7) holds if and only if  $(G, V, X_j^*, S_{1,j}^*, S_{2,j}^*, U) \in \mathscr{L}_{bb}$ .

*Proof (Proposition 2).* Denote U = uG. The claim follows due to

$$\begin{aligned} Equation~(7) &\iff S_{1,j}^{*} = u \cdot V + (\overline{m}_{j}^{*} \cdot u) S_{2,j}^{*} + h S_{2,j}^{*} \\ &\iff S_{1,j}^{*} = u \cdot V + (\overline{m}_{j}^{*} \cdot u \cdot s_{j,2}^{*}) G + (h \cdot s_{j,2}^{*}) G \wedge S_{2,j}^{*} = s_{j,2}^{*} G \\ &\iff S_{1,j}^{*} = u \cdot V + s_{j,2}^{*} (\overline{m}_{j}^{*} U + H) \wedge S_{2,j}^{*} = s_{j,2}^{*} G \\ &\iff S_{1,j}^{*} = u \cdot V + s_{j,2}^{*} \cdot X_{j}^{*} \wedge S_{2,j}^{*} = s_{j,2}^{*} G \\ &\iff (G, V, X_{j}^{*}, S_{1,j}^{*}, S_{2,j}^{*}, U) \in \mathscr{L}_{bb} \end{aligned}$$

Let us assume that  $\mathcal{A}$  outputs forgeries with common message  $\tau^*$  such that Game 8 outputs 1. This occurs with probability  $\varepsilon_8$  by definition. Further, let us assume that  $(G, \mathbf{D}^{\tau^*}) \notin \mathscr{L}_{ddh}$  (which holds except with probability 1/p). Denote by  $\mathbb{x}_{0,j}^* = (G, V, X_j^*, S_{1,j}^*, S_{2,j}^*, U)$  the statements within  $\mathcal{A}$ 's forgeries as above. Observe that Equations (8) to (10) are satisfied because all k forgeries are valid. Thus, Proposition 1 yields via a union bound over all  $H_{\Sigma}$  queries that except with probability  $Q_{\Sigma}/p$ , it holds for all  $j \in [k]$  that  $\mathbb{x}_{0,j}^* \in \mathscr{L}_{bb}$ . Due to Proposition 2 this implies that Equation (7) holds for  $j \in [k]$ . In total, the above considerations yield that

$$|\varepsilon_8 - \varepsilon_9| \le \frac{Q_{\Sigma} + 1}{p}.$$

We emphasize that it will be essential for the following changes that the winning condition of this game can still be evaluated efficiently.

**Game 10 (Sample DDH tuple if**  $\tau = \tau^*$ ). In this game, we change how the  $q_{\tau}^*$ -th query to  $\mathsf{H}_{\mathsf{ddh}}$  (i.e., the query with  $\tau = \tau^*$ ) is answered. Namely, on this query, the challenger samples  $d_2 \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p$  and sets  $(D_2^{\tau^*}, D_3^{\tau^*}) \coloneqq (d_2G, d_2D_1)$  instead of  $(D_2^{\tau^*}, D_3^{\tau^*}) \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{G}^2$ . The witness  $d_2$  is stored in the

table  $T_{ddh}[\tau^*]$ . Other outputs of  $H_{ddh}$  remain unchanged. We can easily construct a reduction  $\mathcal{A}^3_{DDH}$  against DDH with running time similar to  $\mathcal{A}$  and with

$$|\varepsilon_9 - \varepsilon_{10}| \leq \mathsf{AdvDDH}_{\mathcal{A}^3_{\mathsf{DDU}}}^{\mathbb{G}}(\lambda).$$

Note that now, we have  $\mathbf{D}^{\tau} \in \mathscr{L}_{\mathsf{ddh}}$  for all common messages  $\tau$ .

Game 11 (Use DDH witness for  $\Sigma_1$  if  $\tau = \tau^*$ ). We change the signer oracle again, for the case that  $\tau = \tau^*$ . Namely, the  $\Sigma_1$  transcript  $(\mathbf{A}_1, c, z_1)$  is now computed via the witness  $\mathbf{w}_1^* \coloneqq \mathsf{T}_{\mathsf{ddh}}[\tau^*]$  and is no longer simulated via HVZK. That is, in  $\mathcal{O}_{\mathsf{S}_1}$  with  $\tau^*$ , the game samples  $c_1 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and  $(\mathbf{A}_1, \mathsf{st}_1) \leftarrow \mathsf{Init}_1(\mathbf{x}_1, \mathbf{w}_1)$ , where  $\mathbf{w}_1 \coloneqq \mathsf{T}_{\mathsf{ddh}}[\tau]$  and  $\mathbf{x}_1 \coloneqq (G, \mathbf{D}^{\tau^*})$ . In  $\mathcal{O}_{\mathsf{S}_2}$  with  $\tau^*$ , the game then computes  $z_1 \leftarrow \mathsf{Resp}_1(\mathsf{st}_1, c_1)$ .

It follows (as in Game 6) from HVZK of  $\Sigma_1$  that the  $\Sigma_1$  transcripts  $(\mathbf{A}_1, c, z_1)$  for  $\tau = \tau^*$  in Game 10 and Game 11 are distributed identically. In conclusion, we have

$$\varepsilon_{10} = \varepsilon_{11}.$$

**Game 12 (Simulate**  $\Sigma_0$  if  $\tau = \tau^*$ ). We change the signer oracle a final time, for the case that  $\tau = \tau^*$ . Concretely, in  $\mathcal{O}_{S_1}$  with  $\tau^*$ , the game computes  $c_0 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and  $(\mathbf{A}_0^C, \mathbf{z}_0) \leftarrow \operatorname{Sim}_0(\mathbb{x}_0^C, c_0)$  for  $\mathbb{x}_0^c := X_C := C + H$ . In  $\mathcal{O}_{S_2}$  with  $\tau^*$ , the game sets  $c_1 := c - c_0$  and outputs  $\mathbf{z}_0$  from  $\mathcal{O}_{S_1}$ . The other branch (i.e.,  $z_1$ ) is computed via  $\mathbb{w}_1$  as in Game 11.

It follows (as in Game 7) that in Game 11 and Game 12, the challenges  $c_0$  and  $c_1$  follow the same distribution and that the  $\Sigma_0$  transcripts  $(\mathbf{A}_0, c_0, \mathbf{z}_0)$  are identically distributed (by HVZK of  $\Sigma_0$ ). Thus, we have that

 $\varepsilon_{11} = \varepsilon_{12}.$ 

A complete description of Game 12 is given in Figure 4 in Appendix E. The game sets up the verification key vk as in KeyGen, except that it knows the discrete logarithm h of H. It also guesses a hash value  $\overline{m}^* = \mathsf{H}_{\mathsf{M}}(m_j^*)$  such that  $m_j^*$  is a forgery's message but no signing session with common message  $\tau^*$  is finished such that  $\overline{m}^*$  is extracted from  $\pi_{\mathsf{Ped}}$ , where  $\tau^*$  is the forgeries' common message. Roughly, the game then simulates the signing oracles as follows. In  $\mathcal{O}_{\mathsf{S}_1}$ , the game outputs  $\mathbf{T} = (T_1, T_2, U)$  computed honestly only if  $\tau = \tau^*$  and  $\overline{m} \neq \overline{m}^*$  (otherwise random  $T_1, T_2 \stackrel{*}{\leftarrow} \mathbb{G}$  are chosen). The  $\Sigma_1$  transcripts ( $\mathbf{A}_1, c_1, z_1$ ) in  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$  are computed via a DDH witness for  $\mathbf{D}^{\tau}$  via ( $\Sigma_1.\mathsf{Init}, \Sigma_1.\mathsf{Resp}$ ), and the  $\Sigma_0$  transcripts ( $\mathbf{A}_0, c_0, \mathbf{z}_0$ ) are simulated via HVZK of  $\Sigma_0$ . In the end, the challenger aborts if the forgeries are not in  $\mathscr{L}_{\mathsf{bb}}$  (as in Equation (7) via h). We stress that h is only required to check Equation (7) in Game 12 and this is only done after  $\mathcal{A}$  output its forgeries.

**Reduction to CDH.** Finally, there exists a reduction  $\mathcal{A}_{\mathsf{CDH}}$  such that  $\varepsilon_{12} \leq \mathsf{AdvCDH}_{\mathcal{A}_{\mathsf{CDH}}}^{\mathbb{G}}(\lambda)$ . This follows via Lemma 1. In more detail, let us construct a reduction  $\mathcal{B}_3$  that outputs  $\mathbb{x}_0^* \in \mathscr{L}_{\mathsf{bb}}$  for the game described in Lemma 1, hereafter denoted by Game BB. Note that  $\mathcal{B}_3$  has access to an oracle  $\mathcal{O}(\lambda)$  that on input  $\overline{m}$  outputs values  $(S_1, S_2)$ . First,  $\mathcal{B}_3$  samples  $\overline{m}^* \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and obtains (G, U, H, V) after providing  $\overline{m}^*$  to Game BB. Next,  $\mathcal{B}_3$  samples  $D_1 \stackrel{*}{\leftarrow} \mathbb{G}$  and sets  $\mathsf{vk} = (G, U, H, V, D_1)$ . Also,  $\mathcal{B}_3$  initializes the counters  $\mathsf{ctr}_{\mathsf{ddh}}$  and  $\mathsf{ctr}_{\mathsf{M}}$  to 0, samples  $q_m^*$  and  $q_\tau^*$  at random, and initializes  $\tau_{q_\tau^*} := \bot$ . It then invokes  $\mathcal{A}$  on input  $\mathsf{vk}$  and simulates the oracles in Game 12 to  $\mathcal{A}$  as follows.

- $-H_M, H_{ddh}, H_{\Sigma}, H_{Ped}, Next, \mathcal{O}_{S_2}: \mbox{Simulated as in Game 12. We remark that for } H_M, \mbox{ the value } \overline{m}^*$
- is output on the  $q_m^*$ -th query. Also, note that Game 12 aborts if  $\overline{m}^*$  is extracted from C in  $\mathcal{O}_{\mathsf{S}_1}$ .  $-\mathcal{O}_{\mathsf{S}_1}$ : Check that  $\pi_{\mathsf{Ped}}$  verifies and output  $\perp$  if not. Else, extract  $(\overline{m}, t)$  such that  $C = \overline{m}U + tG$ from  $\pi_{\mathsf{Ped}}$ . If  $\tau \neq \tau_{q_{\tau}^*}$ , then set  $\mathbf{T} \stackrel{\$}{\leftarrow} \mathbb{G}^2 \times \{U\}$  as in Game 12, else set  $(S_1, S_2) \leftarrow \mathcal{O}(\overline{m})$  and

 $\mathbf{T} \coloneqq (S_1 + t \cdot S_2, S_2, U)$ . Then, proceed as in Game 12.

When  $\mathcal{A}$  outputs its forgeries  $(m_j^*, \sigma_j^*)$  on common message  $\tau^*$ ,  $\mathcal{B}_3$  checks whether there is a message  $m_j^*$  such that  $\mathsf{H}_\mathsf{M}(m_j^*) = \overline{m}^*$ . Finally,  $\mathcal{B}_3$  parses  $(S_1^*, S_2^*, \pi^*) = \sigma_j^*$  and outputs  $(S_1^*, S_2^*)$  to Game BB.

Clearly, the simulated vk is distributed as in Game 12. Also, it is easy to check that  $\mathcal{B}_3$ 's simulation of Game 12 is efficient and the running time of  $\mathcal{B}_3$  is roughly that of  $\mathcal{A}$ . It remains to show that **T** is identically distributed if  $\tau = \tau_{q_{\tau}^*}$ . Denote by u the (unknown) discrete logarithm of U. Recall that by definition (cf. Lemma 1),  $\mathcal{O}(\overline{m})$  outputs values  $(S_1, S_2)$  with  $S_1 = uV + sX_{\overline{m}}$ 

and  $S_2 = sG$ , where  $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$  and  $X_{\overline{m}} = \overline{m}U + H$ . Note that the simulated  $(T_2, T_3)$  follow the distribution of Game 12, and due to

$$T_1 = S_1 + tS_2 = uV + sX_{\overline{m}} + (t \cdot s)G$$
$$= uV + s(\overline{m}U + H + tG) = uV + s(C + H),$$

the simulated  $T_1$  is also distributed as in Game 12. In conclusion, the view of  $\mathcal{A}$  is as in Game 12 and with probability at least  $\varepsilon_{12}$ , there is a message  $m_j^*$  with  $\mathsf{H}_\mathsf{M}(m_j^*) = \overline{m}^*$  and Equation (7) holds (where  $h = \mathsf{DLOG}_G(H)$ ). As shown in Game 9, this implies that  $(G, V, X_{\overline{m}^*}, S_1^*, S_2^*, U) \in \mathscr{L}_{\mathsf{bb}}$  for  $X_{\overline{m}^*} := \overline{m}^*U + H$ . Due to Lemma 1, there is an adversary  $\mathcal{A}_{\mathsf{CDH}}$  with running time similar to  $\mathcal{A}$ such that  $\varepsilon_{12} \leq \mathsf{AdvCDH}_{\mathcal{A}_{\mathsf{CDH}}}^{\mathbb{G}}(\lambda)$ . By collecting all the bounds and applying Remark 5, we obtain the statement.

## 5 Blind Interactive Signing Protocol

In this section, we explain how the unblinded protocol from Section 4 can be turned into a partially blind signature BS.

### 5.1 Construction

We construct a partially blind signature BS by blinding the signing protocol of PreBS (cf. Section 4). The requirements are identical to PreBS. That is, let NIPS<sub>Ped</sub> be a NIPS proof system with random oracle  $H_{Ped}$  for Pedersen openings (see Equation (5) for the exact relation). Also, let  $H_{M}: \{0,1\}^* \to \mathbb{Z}_p$ , and  $H_{\Sigma}: \{0,1\}^* \to \mathbb{Z}_p$ , and  $H_{ddh}: \{0,1\}^* \to \mathbb{G}^2$  be random oracles.

Our blinding essentially follows the same approach as prior works. The blinding of the statement X as  $X_C$  is already present in the unblinded signature, as the proof  $\pi_{\mathsf{Ped}}$  is constructed relative to it and required for the OMUF reduction; the statement  $\mathbf{D}^{\tau}$  corresponding to common message  $\tau$  remains unblinded throughout. Except for the blinding, the only additional change is that the user now verifies the signer's response. Otherwise, it may output invalid "signatures", making interactions linkable.

## **BS:** Partially blind signature

- KeyGen $(1^{\lambda})$ : Output  $(\mathsf{vk},\mathsf{sk}) \leftarrow \mathsf{BBSig}.\mathsf{KeyGen}(1^{\lambda})$ .
- S(sk, τ) ↔ U(vk, m, τ): The *blinded* signing protocol proceeds in 4 moves and is given in Figure 2.
- Verify(vk,  $m, \tau, \sigma_{bb}$ ): Output  $b \leftarrow \mathsf{BBSig.Verify}(vk, m, \tau, \sigma_{bb})$ .

Remark 3 (Notation). In Figure 2, we follow the convention that variables with a star, such as  $\mathbf{A}_0^*$  or  $c^*$  are sent to the signer or received by the user. Variables with a prime, such as  $\mathbf{A}_0'$  and  $c_0'$  are random masks to ensure blindness. Other variables are usually outputs, such as  $\mathbf{A}$ . Sometimes, this convention is broken for consistency with the unblinded protocol, e.g., for C.

 $\mathsf{S}(\mathsf{sk},\tau)$  $\mathsf{U}(\mathsf{vk},m,\tau)$  $t \stackrel{\$}{\leftarrow} \mathbb{Z}_p; \quad \overline{m} \coloneqq \mathsf{H}_{\mathsf{M}}(m),$ 1:  $C \coloneqq \overline{m}U + tG$ 2:3:  $\mathbb{X}_{\mathsf{Ped}} \coloneqq (C, U, G); \quad \mathbb{W}_{\mathsf{Ped}} \coloneqq (\overline{m}, t)$ 4:  $\pi_{\mathsf{Ped}} \leftarrow \mathsf{NIPS}_{\mathsf{Ped}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{Ped}}}(\mathbb{X}_{\mathsf{Ped}},\mathbb{W}_{\mathsf{Ped}})$  $C, \pi_{\mathsf{Ped}}$ 5:  $\mathbb{X}_{\mathsf{Ped}} \coloneqq (C, U, G)$ 6: **req** NIPS<sub>Ped</sub>.Ver<sup>H<sub>Ped</sub>( $x_{Ped}, \pi_{Ped}$ ) = 1</sup> 7:  $s^* \xleftarrow{\$} \mathbb{Z}_p; \quad \mathbb{W}_0 \coloneqq (s^*, \mathsf{sk})$ 8:  $X_C \coloneqq C + H; \quad \mathbf{T}^* \coloneqq \phi_0(X_C, \mathbf{w}_0)$ 9:  $\mathbb{x}_0^C \coloneqq (G, V, X_C, \mathbf{T}^*); \quad \mathbb{x}_1 \coloneqq (G, \mathbf{D}^\tau)$ 10:  $c_1^* \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p; \quad (\mathbf{A}_1^*, z_1) \leftarrow \mathsf{Sim}_1(\mathbb{x}_1, c_1)$ 11:  $(\mathbf{A}_0^*, \mathsf{st}_0) \leftarrow \mathsf{lnit}_0(\mathbb{x}_0^C, \mathbb{w}_0)$  $(\mathbf{T}^*, \mathbf{A}_0^*, \mathbf{A}_1^*)$  $s' \stackrel{\$}{\leftarrow} \mathbb{Z}_p;$ 12:  $c'_0, c'_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ 13: 14:  $\mathbf{z}'_0 \stackrel{\$}{\leftarrow} \mathbb{Z}^2_p$ ;  $z'_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$ 15:  $X \coloneqq \overline{m}U + H = X_C - tG$ 16:  $\mathbf{S} \coloneqq \mathbf{T}^* - (t \cdot T_2, 0, 0)^\mathsf{T} + \phi_0(X, (s', 0))$ 17:  $\mathbf{A}_0 := \mathbf{A}_0^* - (t \cdot A_{0,2}^*, 0, 0)^\mathsf{T} + \phi_0(X, \mathbf{z}_0') - c_0' \mathbf{S}$ 18:  $\mathbf{A}_1 \coloneqq \mathbf{A}_1^* + \phi_1(z_1') - c_1' \mathbf{D}^{\mathsf{T}}$ 19:  $\mathbf{x}_0 \coloneqq (G, V, X, \mathbf{S}); \quad \mathbf{x}_1 \coloneqq (G, \mathbf{D}^{\tau})$ 20:  $c \coloneqq \mathsf{H}_{\Sigma}((\mathbf{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m)$  $c^* = c - c'_0 - c'_1$ 21:  $c_0^* \coloneqq c^* - c_1^*$ 22:  $\mathbf{z}_0^* \leftarrow \mathsf{Resp}_0(\mathsf{st}_0, c_0)$  $\mathbf{z}_{0}^{*}, z_{1}^{*}, c_{0}^{*}$  $/\!\!/ c_1^* = c^* - c_0^*$ 23: **req**  $\mathbf{A}_0^* = \phi_0(X_C, \mathbf{z}_0) - c_0^* \mathbf{T}^*$ 24: **req**  $\mathbf{A}_{1}^{*} = \phi_{1}(\mathbf{z}_{1}) - c_{1}^{*}\mathbf{D}^{\tau}$ 25:  $c_0 = c_0^* + c_0'$ ;  $c_1 = c_1^* + c_1' \quad // c = c_0 + c_1$  $\mathbf{z}_0 = \mathbf{z}_0^* + \mathbf{z}_0' - c_0 \cdot (s', 0)$ 26:  $\mathbf{z}_1 = \mathbf{z}_1^* + z_1'$ 27:  $\pi \coloneqq (\mathbf{A}_0, \mathbf{A}_1, c, c_0, \mathbf{z}_0, z_1)$ 28: 29:  $\sigma_{\mathsf{bb}} \coloneqq (S_1, S_2, \pi)$ 

Fig. 2. The blinded version of a signing session of BS. As in the unblinded version (cf. Figure 1), we have  $m, \tau \in \{0, 1\}^*$ ,  $\mathsf{vk} = (G, U, H, V, D_1) \in \mathbb{G}^5$  and  $\mathsf{sk} = u \in \mathbb{Z}_p$ . The signer aborts (i.e., outputs  $\bot$ ) if for condition C, req C is evaluated for false C. Recall that  $\mathsf{H}_{\Sigma}$  maps into  $\mathbb{Z}_p$  and that  $\mathbf{D}^{\tau} := (D_1, D_2^{\tau}, D_3^{\tau})$  for  $(D_2^{\tau}, D_3^{\tau}) := \mathsf{H}_{\mathsf{ddh}}(\tau)$ . Visually highlighted are the parts which blind the parameters C of the map  $\phi_0$ , the statement  $\mathbf{T}$  of the map  $\phi_0$ , the challenge c of the OR-composition, the responses  $\mathbf{z}_0$  and  $\mathbf{z}_1$ . If these parts are removed, except for parameter blinding, then one recovers the unblinded protocol (cf. Figure 1).

#### 5.2 Security Analysis

We show that BS is correct and partially blind in the random oracle model. One-more unforgeability follows via Theorem 1 under the same conditions.

*Remark 4 (OMUF of* BS). Observe that in Figure 2 only the user-side was modified compared to the protocol in Figure 1. The signer's code is unchanged. As a consequence, Figure 2 is one-more unforgeable if the unblinded version is. Indeed, one-more unforgeability considers a malicious user, whose code is adversarial, so only the signer's code is specified and part of the one-more unforgeability game.

**Theorem 2** (Correctness). BS is correct with error  $\gamma_{err}$ , where  $\gamma_{err}$  is the correctness error of NIPS<sub>Ped</sub>.

Correctness is straightforward. We provide a formal proof in Appendix D for completeness.

**Theorem 3 (Blindness).** For any (unbounded) adversary  $\mathcal{A}$  that causes at most Q queries to the random oracle  $\mathsf{H}_{\mathsf{Ped}}$  (via its own queries or through the oracles  $\mathcal{O}_0$ ,  $\mathcal{O}_1$ ), then there exists an adversary  $\mathcal{A}_{\mathsf{WI}}$  with running time roughly that of  $\mathcal{A}$ , such that

$$\mathsf{AdvPBlind}_{\mathcal{A}}^{\mathsf{BS}} \leq 2 \cdot \mathsf{AdvWl}_{\mathcal{A}_{\mathsf{WI}}}^{\mathsf{NIPS}}(Q, \lambda) + \frac{2}{p},$$

where  $p = |\mathbb{G}|$  is the group order.

The very high-level idea is that enough randomness is injected to completely randomize the transcript  $\pi$  (which is part of the blind signature), and also the signature  $(S_1, S_2)$ ; here we exploit that  $\mathsf{R}_{\mathsf{bb}}$  yields perfectly randomizable signatures. Moreover, the proof  $\pi_{\mathsf{Ped}}$  can be simulated. We give the formal proof of blindness in Appendix D.

Acknowledgement. This work was supported by the Helsinki Institute for Information Technology (HIIT) and was conducted while the first author was affiliated with Aalto University.

## References

Abe01.	Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures.
	In Birgit Pfitzmann, editor, EUROCRYPT 2001, volume 2045 of LNCS, pages 136-151. Springer,
	Heidelberg, May 2001. (Cited on Page 3)
AF96.	Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and
	Tsutomu Matsumoto, editors, ASIACRYPT'96, volume 1163 of LNCS, pages 244-251. Springer,
	Heidelberg, November 1996. (Cited on Pages 2 and 7)
AO00.	Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir
	Bellare, editor, CRYPTO 2000, volume 1880 of LNCS, pages 271–286. Springer, Heidelberg,
	August 2000. (Cited on Pages 1, 3, and 6)
BB04.	Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without
	random oracles. In Christian Cachin and Jan Camenisch, editors, EUROCRYPT 2004, volume
	3027 of LNCS, pages 223–238. Springer, Heidelberg, May 2004. (Cited on Pages 4, 10, and 11)
BF01.	Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe
	Kilian, editor, CRYPTO 2001, volume 2139 of LNCS, pages 213–229. Springer, Heidelberg,
	August 2001. (Cited on Page 10)
BGH07.	Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption
	without pairings. In 48th FOCS, pages 647-657. IEEE Computer Society Press, October 2007.
	(Cited on Page 2)
BL13.	Foteini Baldimtsi and Anna Lysyanskaya. On the security of one-witness blind signature
	schemes. In Kazue Sako and Palash Sarkar, editors, ASIACRYPT 2013, Part II, volume 8270
	of LNCS, pages 82–99. Springer, Heidelberg, December 2013. (Cited on Page 3)
$BLL^+21.$	Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On

the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EURO-CRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Heidelberg, October 2021. (Cited on Page 4)

- BNPS03. Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The onemore-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology, 16(3):185–215, June 2003. (Cited on Pages 1 and 3)
- Bol03. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Heidelberg, January 2003. (*Cited on Pages 1 and 3*)
- BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, ACM CCS 93, pages 62–73. ACM Press, November 1993. (Cited on Pages 1 and 2)
- CFGG22. Dario Catalano, Dario Fiore, Rosario Gennaro, and Emanuele Giunta. On the impossibility of algebraic vector commitments in pairing-free groups. In Eike Kiltz and Vinod Vaikuntanathan, editors, TCC 2022, Part II, volume 13748 of LNCS, pages 274–299. Springer, Heidelberg, November 2022. (Cited on Page 2)
- CG08. Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, ACM CCS 2008, pages 345–356. ACM Press, October 2008. (Cited on Page 1)
- Cha82. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, CRYPTO'82, pages 199–203. Plenum Press, New York, USA, 1982. (Cited on Pages 1, 3, and 7)
- CHL<sup>+</sup>22. 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 Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 3–31. Springer, Heidelberg, August 2022. (*Cited on Pages 1, 2, and 4*)
- CKM<sup>+</sup>23. Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Snowblind: A threshold blind signature in pairing-free groups. In Helena Handschuh and Anna Lysyanskaya, editors, CRYPTO 2023, Part I, volume 14081 of LNCS, pages 710–742. Springer, Heidelberg, August 2023. (Cited on Pages 2 and 3)
- CKU20. Geoffroy Couteau, Shuichi Katsumata, and Bogdan Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In Anne Canteaut and Yuval Ishai, editors, EUROCRYPT 2020, Part III, volume 12107 of LNCS, pages 442–471. Springer, Heidelberg, May 2020. (Cited on Page 2)
- CL01. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001. (*Cited on Page 1*)
- CTZ24. Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Pairing-free blind signatures from CDH assumptions. In Leonid Reyzin and Douglas Stebila, editors, CRYPTO 2024, LNCS. Springer, Heidelberg, August 18–22, 2024. (Cited on Pages 2, 3, 4, 5, and 6)
- dK22. Rafaël del Pino and Shuichi Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 306–336. Springer, Heidelberg, August 2022. (*Cited on Pages 1, 2, and 4*)
- EHK<sup>+</sup>13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, CRYPTO 2013, Part II, volume 8043 of LNCS, pages 129–147. Springer, Heidelberg, August 2013. (Cited on Page 25)
- FHS15. Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 233–253. Springer, Heidelberg, August 2015. (*Cited on Pages 1 and 3*)
- Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, CRYPTO 2005, volume 3621 of LNCS, pages 152–168. Springer, Heidelberg, August 2005. (Cited on Pages 5, 9, 26, 27, and 28)
- Fis06. Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, CRYPTO 2006, volume 4117 of LNCS, pages 60–77. Springer, Heidelberg, August 2006. (Cited on Pages 1, 3, and 4)
- FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, CRYPTO 2018, Part II, volume 10992 of LNCS, pages 33–62. Springer, Heidelberg, August 2018. (Cited on Pages 2 and 3)
- FPS20. Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors,

EUROCRYPT 2020, Part II, volume 12106 of LNCS, pages 63–95. Springer, Heidelberg, May 2020. (Cited on Page 3)

- FS10. Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, EUROCRYPT 2010, volume 6110 of LNCS, pages 197–215. Springer, Heidelberg, May / June 2010. (Cited on Page 3)
- FW24. Georg Fuchsbauer and Mathias Wolf. Concurrently secure blind schnorr signatures. In Marc Joye and Gregor Leander, editors, EUROCRYPT 2024, Part II, volume 14652 of LNCS, pages 124–160, Zurich, Switherland, May 26–30, 2024. Springer, Heidelberg. (Cited on Page 3)
- GG14. Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, EUROCRYPT 2014, volume 8441 of LNCS, pages 477–495. Springer, Heidelberg, May 2014. (Cited on Pages 1 and 3)
- Gha17. Essam Ghadafi. Efficient round-optimal blind signatures in the standard model. In Aggelos Kiayias, editor, FC 2017, volume 10322 of LNCS, pages 455–473. Springer, Heidelberg, April 2017. (Cited on Pages 1 and 3)
- GHKW16. Romain Gay, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Tightly CCA-secure encryption without pairings. In Marc Fischlin and Jean-Sébastien Coron, editors, EUROCRYPT 2016, Part I, volume 9665 of LNCS, pages 1–27. Springer, Heidelberg, May 2016. (Cited on Page 2)
- GRS<sup>+</sup>11. Sanjam Garg, Vanishree Rao, Amit Sahai, Dominique Schröder, and Dominique Unruh. Round optimal blind signatures. In Phillip Rogaway, editor, CRYPTO 2011, volume 6841 of LNCS, pages 630–648. Springer, Heidelberg, August 2011. (Cited on Pages 1 and 3)
- HKL19. Eduard Hauck, Eike Kiltz, and Julian Loss. A modular treatment of blind signatures from identification schemes. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part III, volume 11478 of LNCS, pages 345–375. Springer, Heidelberg, May 2019. (Cited on Pages 1 and 3)
- HKLN20. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, CRYPTO 2020, Part II, volume 12171 of LNCS, pages 500–529. Springer, Heidelberg, August 2020. (Cited on Page 3)
- HLW23. Lucjan Hanzlik, Julian Loss, and Benedikt Wagner. Rai-choo! Evolving blind signatures to the next level. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 753–783. Springer, Heidelberg, April 2023. (*Cited on Pages 1, 2, and 4*)
- JLO97. Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski Jr., editor, CRYPTO'97, volume 1294 of LNCS, pages 150–164. Springer, Heidelberg, August 1997. (Cited on Pages 1 and 3)
- Kat21. Shuichi Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In Tal Malkin and Chris Peikert, editors, CRYPTO 2021, Part II, volume 12826 of LNCS, pages 580–610, Virtual Event, August 2021. Springer, Heidelberg. (Cited on Page 9)
- KLR21. Jonathan Katz, Julian Loss, and Michael Rosenberg. Boosting the security of blind signature schemes. In Mehdi Tibouchi and Huaxiong Wang, editors, ASIACRYPT 2021, Part IV, volume 13093 of LNCS, pages 468–492. Springer, Heidelberg, December 2021. (Cited on Pages 1, 2, and 4)
- KLX22a. Julia Kastner, Julian Loss, and Jiayu Xu. The Abe-Okamoto partially blind signature scheme revisited. In Shweta Agrawal and Dongdai Lin, editors, ASIACRYPT 2022, Part IV, volume 13794 of LNCS, pages 279–309. Springer, Heidelberg, December 2022. (Cited on Pages 1 and 3)
- KLX22b. Julia Kastner, Julian Loss, and Jiayu Xu. On pairing-free blind signature schemes in the algebraic group model. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part II*, volume 13178 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2022. (Cited on Pages 2 and 3)
- KNR24. Julia Kastner, Ky Nguyen, and Michael Reichle. Pairing-free blind signatures from standard assumptions in the rom. In Leonid Reyzin and Douglas Stebila, editors, CRYPTO 2024, LNCS. Springer, Heidelberg, August 18–22, 2024. (Cited on Pages 1 and 4)
- KRS23. Shuichi Katsumata, Michael Reichle, and Yusuke Sakai. Practical round-optimal blind signatures in the ROM from standard assumptions. In Jian Guo and Ron Steinfeld, editors, ASIACRYPT 2023, Part II, volume 14439 of LNCS, pages 383–417. Springer, Heidelberg, December 2023. (Cited on Pages 1, 2, 4, 5, and 6)
- Ks22. Yashvanth Kondi and abhi shelat. Improved straight-line extraction in the random oracle model with applications to signature aggregation. In Shweta Agrawal and Dongdai Lin, editors, ASIACRYPT 2022, Part II, volume 13792 of LNCS, pages 279–309. Springer, Heidelberg, December 2022. (Cited on Pages 5, 9, 26, 27, and 28)
- Oka93. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, CRYPTO'92, volume 740 of LNCS, pages 31–53. Springer, Heidelberg, August 1993. (Cited on Page 4)

- Oka06. Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, TCC 2006, volume 3876 of LNCS, pages 80–99. Springer, Heidelberg, March 2006. (Cited on Page 3)
- OO92. Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In Joan Feigenbaum, editor, CRYPTO'91, volume 576 of LNCS, pages 324–337. Springer, Heidelberg, August 1992. (Cited on Page 1)
- Pas03. Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, CRYPTO 2003, volume 2729 of LNCS, pages 316–337. Springer, Heidelberg, August 2003. (Cited on Page 9)
- Pas11. Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, 43rd ACM STOC, pages 109–118. ACM Press, June 2011. (Cited on Page 3)
- Poi98. David Pointcheval. Strengthened security for blind signatures. In Kaisa Nyberg, editor, EUROCRYPT'98, volume 1403 of LNCS, pages 391–405. Springer, Heidelberg, May / June 1998. (Cited on Page 4)
- PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 13(3):361–396, June 2000. (Cited on Pages 1 and 3)
- PW23. Jiaxin Pan and Benedikt Wagner. Chopsticks: Fork-free two-round multi-signatures from non-interactive assumptions. In Carmit Hazay and Martijn Stam, editors, EUROCRYPT 2023, Part V, volume 14008 of LNCS, pages 597–627. Springer, Heidelberg, April 2023. (Cited on Page 2)
- Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161–174, January 1991. (Cited on Page 3)
- Sch01. Claus-Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Heidelberg, November 2001. (Cited on Page 4)
- Sch22. Tim Scheurer. Universally composable verifiable random oracles. Master's thesis, Karlsruher Institut für Technologie (KIT), 2022. 46.23.01; LK 01. (*Cited on Pages 26 and 28*)
- TZ22. Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, EUROCRYPT 2022, Part II, volume 13276 of LNCS, pages 782–811. Springer, Heidelberg, May / June 2022. (Cited on Pages 2 and 3)
- TZ23. Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Heidelberg, April 2023. (*Cited on Page 2*)
- Wag02. David Wagner. A generalized birthday problem. In Moti Yung, editor, CRYPTO 2002, volume 2442 of LNCS, pages 288–303. Springer, Heidelberg, August 2002. (Cited on Page 4)

# Appendices

## A Assumptions

Let  $\mathbb{G}$  be a group of prime order p with generator  $G \in \mathbb{G}$ —implicitly parameterized by the security parameter (cf. Section 2). We formally define the DL, CDH and (Q-)DDH assumptions. While both DL and CDH is implied by DDH, and Q-DDH is implied by DDH, it will be convenient for us to use these assumptions explicitly.

**Definition 13 (DL Assumption).** The discrete logarithm (DL) assumption holds in group  $\mathbb{G}$  with generator G if for any PPT adversary A, it holds that

$$\mathsf{AdvDL}^{\mathbb{G}}_{\mathcal{A}}(\lambda) \coloneqq \Pr[x \leftarrow \mathbb{Z}_p, x' \leftarrow \mathcal{A}(G, xG) : x = x'] = \operatorname{negl}(\lambda).$$

**Definition 14 (CDH Assumption).** The computational Diffie-Hellman (CDH) assumption holds in group  $\mathbb{G}$  with generator G if for any PPT adversary A, it holds that

 $\mathsf{AdvCDH}^{\mathbb{G}}_{\mathcal{A}}(\lambda) \coloneqq \Pr[a, b \leftarrow \mathbb{Z}_p, C \leftarrow \mathcal{A}(G, aG, bG) : C = (a \cdot b)G] = \operatorname{negl}(\lambda).$ 

**Definition 15 (DDH Assumption).** The decisional Diffie-Hellman (DDH) assumption holds in group  $\mathbb{G}$  with generator G if for any PPT adversary  $\mathcal{A}$ , it holds that

$$\mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{A}}(\lambda) \coloneqq \left| \Pr[a, b \leftarrow \mathbb{Z}_p : \mathcal{A}(G, aG, bG, (ab)G) = 1] - \Pr[a, b, c \leftarrow \mathbb{Z}_p : \mathcal{A}(G, aG, bG, cG) = 1] \right| = \operatorname{negl}(\lambda).$$

**Definition 16 (Q-DDH Assumption).** The Q-fold decisional Diffie-Hellman (Q-DDH) assumption holds in group  $\mathbb{G}$  with generator g if for any PPT adversary  $\mathcal{A}$ , it holds that

$$\mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{A}}(Q,\lambda) \coloneqq \left| \Pr[a \leftarrow \mathbb{Z}_p, \mathbf{b} \leftarrow \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, (ab_i)G)_{i \in [Q]}) = 1] \right| - \Pr[a \leftarrow \mathbb{Z}_p, \mathbf{b}, \mathbf{c} \leftarrow \mathbb{Z}_p^Q : \mathcal{A}(G, aG, (b_iG, c_iG)_{i \in [Q]}) = 1] \right| = \operatorname{negl}(\lambda).$$

Remark 5. Q-DDH is tightly implied by DDH. Namely, for any PPT adversary  $\mathcal{A}$  on Q-DDH, there is a PPT reduction  $\mathcal{B}$  with running time roughly that of  $\mathcal{A}$ , such that  $\mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{A}}(Q,\lambda) \leq \mathsf{AdvDDH}^{\mathbb{G}}_{\mathcal{B}}(\lambda) + 1/(p-1)$  (cf. [EHK+13]).

## B $\Sigma$ -protocols for Our Construction

Recall that in our blind signature construction, we use two  $\Sigma$ -protocols for relations  $R_{bb}$  and  $R_{ddh}$  defined in Section 3. The construction of these follows well-known techniques, but we present them here for completeness.

 $\Sigma$ -protocol for  $\mathsf{R}_{bb}$  and  $\mathsf{R}_{ddh}$ . We provide  $\Sigma$ -protocols for  $\mathsf{R}_{bb}$  and  $\mathsf{R}_{ddh}$  as defined in Section 3. More generally, let  $\phi: \mathbb{Z}_p^{\omega} \to \mathbb{G}^{\kappa}$  be a linear map. Especially, this covers  $\phi_0$  with fixed statement and  $\phi_1$  as defined in Section 3. The following is a  $\Sigma$ -protocol with challenge space  $\mathbb{Z}_p$  for the relation of statements  $\mathbf{x} = \mathbf{T}$  and witnesses  $\mathbf{w} = \mathbf{w}$  with  $\phi(\mathbf{w}) = \mathbf{T}$ . (Note that if  $\phi$  is parameterized, then the statement  $\mathbf{x}$  also contains the parameters, or we can think of it as a class of  $\Sigma$ -protocols.)  $\Sigma$ -protocol  $\Sigma = (Init, Resp, Verify)$  for  $R_{\phi}$  with  $x = (\phi, T)$  and  $w_0 = w$ 

- Init(x, w):

1. Sample  $\mathbf{r} \stackrel{*}{\leftarrow} \mathbb{Z}_p^{\omega}$  and set  $\mathbf{A} \coloneqq \phi(\mathbf{r})$ .

2. Output commitment **A** and state st := (w, r).

-  $\mathsf{Resp}(\mathsf{st}, c)$ : Output  $\mathbf{z} \coloneqq c \cdot \mathbf{w} + \mathbf{r}$ ,

- Verify( $\mathbf{x}, A, c, z$ ): Output 1 if  $\mathbf{A} = \phi(z) - c \cdot \mathbf{T}$  and 0 otherwise.

**Properties of the**  $\Sigma$ -**protocol.** It is straightforward to see that the  $\Sigma$ -protocol is correct. It is also well-known that it is special sound and HVZK. In terms of efficiency, observe that **A** consists of  $\kappa$  many elements in  $\mathbb{G}$ , c consists of 1 element in  $\mathbb{Z}_p$ , and z consists of  $\omega$  many elements in  $\mathbb{Z}_p$ .

## C Construction of Straightline-Extractable Proofs

To instantiate our blind signature scheme, we need a straightline-extractable non-interactive proof system NIPS<sub>Ped</sub> (cf. Section 4). Note that we wish to avoid computational assumptions for zero-knowledge for statistical blindness. We propose to use the randomized Fischlin transform [Fis05, Ks22] to construct such a proof from a  $\Sigma$ -protocol. Concretely, we use the randomized variant of [Ks22], but with a superpolynomial challenge space. While [Fis05] uses *computationally unique* responses to ensure extractability and zero-knowledge<sup>11</sup>, the randomized variant of [Ks22] requires the simpler strong (2-)special soundness. Unlike [Ks22], we will assume a superpolynomial challenge space to prove zero-knowledge. We do so, because for polynomial size challenge space, there is an attack outlined in [Sch22] on the simpler zero-knowledge simulation given in [Ks22]. We thus also revisit the proof for completeness. We emphasize again that any straightline-extractable non-interactive zero-knowledge proof system can be used to instantiate our blind signature construction.

Additional Preliminaries. In the following, we denote by Bin(n, p) the Binomial distribution with parameters  $n \in \mathbb{N}, p \in [0, 1]$ . Also, we write  $X \sim Y$  if X is distributed as Y. For two random variables X, Y over finite domain  $\mathcal{D}$ , we denote by

$$\mathsf{SD}(X,Y) \coloneqq \frac{1}{2} \sum_{d \in D} |\Pr[X = d] - \Pr[Y = d]|.$$

the statistical distance of X and Y. For some random variable V with (finite) image  $\mathcal{Y}$ , we denote the min-entropy by

$$\mathsf{H}_{\min}(V) \coloneqq -\log \max \Pr_{u \in \mathcal{V}}[V = y].$$

Before we revisit the Randomized Fischlin Transform, let us recall the  $\Sigma$ -protocol properties high min-entropy and strong (2-)special soundness. Note that we allow for a relaxed relation in our definition of strong special soundness.

**Definition 17 (High Min-Entropy).** Let R be an NP-relation and  $\Sigma = (Init, Resp, Verify)$ be a  $\Sigma$ -protocol for R. We say that  $\Sigma$  has high min-entropy if for all  $(x, w) \in R$ , it holds for  $(A, st) \leftarrow Init(x, w)$  that

$$2^{-\mathsf{H}_{\min}(A)} = \operatorname{negl}(\lambda).$$

We denote  $\mathsf{H}_{\min}(\Sigma) \coloneqq \min_{\mathbf{x} \in \mathscr{L}_{\mathsf{R}}} \mathsf{H}_{\min}(A)$ .

**Definition 18 ((Relaxed) Strong (2-)Special Soundness).** Let R be an NP-relation and  $\Sigma = (Init, Resp, Verify)$  be a  $\Sigma$ -protocol for R. We say that  $\Sigma$  is strong (2-)special sound for NP-relation  $\tilde{R} \supseteq R$  if there exists a deterministic PT machine extractor Ext such that: Given as input two accepting transcripts tr = (A, c, z) and tr' = (A, c', z') for statement x with  $tr \neq tr'$ , the extractor  $w \leftarrow Ext(tr, tr')$  outputs a witness w such that  $(x, w) \in \tilde{R}$ .

Simplified Randomized Fischlin Transform. Let R be an NP-relation. Let  $\Sigma = (Init, Resp, Verify)$  be a  $\Sigma$ -protocol for R with challenge space CH. We consider following parameters.

<sup>&</sup>lt;sup>11</sup> Because NIPS<sub>Ped</sub> is a proof of Pedersen opening, the *computationally unique responses* property holds only under the DL assumption for the  $\Sigma$ -protocol in Appendix B. We avoid this by relying on [Ks22].

- Challenge Space.  $k := \log(|\mathcal{CH}|) \in \mathbb{R}$  denotes the bit-size of the challenge space.
- Random Oracle.  $b \in \mathbb{N}$  denotes the bit-size of outputs of a random oracle  $\mathsf{H}: \{0,1\}^* \to \{0,1\}^b$ .
- Repetitions.  $r \in \mathbb{N}$  denotes the number of parallel repetitions the transformation will use.
- *Iterations.*  $t \in \mathbb{N}$  is used to avoid infinite loops; the transformation aborts if it does not succeed within at most  $2^t \in \mathbb{N}$  iterations.

Typically, we set the parameters such that  $r \ge \lceil \lambda/b \rceil$  for  $\lambda$  bits of security, and  $t \in \Theta(\log(\lambda)b\log(r))$ and  $k = \omega(b)$  for negligible correctness error (and zero-knowledge); to ensure polynomial time provers, we need  $2^t = \mathsf{poly}(\lambda)$ . The proof size will only depend on r.

We assume that it is efficiently possible to sample uniformly from CH. For simplicity, we ignore any optimizations of the transformation, such as hashing the vector of commitments **A**. Then, the transform yields a non-interactive proof system NIPS = (Prove, Ver) as follows:

NIPS = (Prove, Ver) for R constructed from  $\Sigma$  = (Init, Resp, Verify) -  $\mathsf{Prove}^{\mathsf{H}}(\mathbb{x},\mathbb{w})$ : 1. For each  $i \in [r]$ , compute  $(A_i, \mathsf{st}_i) \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w})$ . 2. Let  $\mathbf{A} = (A_1, \dots, A_r).$ 3. For i = 1, until i > r, or output  $\perp$  if total of t tries are exceeded: (a) Sample  $c_i \stackrel{s}{\leftarrow} CH$  uniformly without replacement within this iteration. (b) Compute  $z_i \leftarrow \mathsf{Resp}(\mathsf{st}_i, c_i)$ . (c) Query  $h_i = \mathsf{H}(\mathbf{x}, \mathbf{A}, i, c_i, z_i)$ . (d) If  $h_i = 0^b$ , set i := i + 1. (Continue to next repetition.) (e) Else: Go back to Step 3a and try again. 4. Return  $\pi := (A_i, c_i, z_i)_{i=1}^r$ . - Ver<sup>H</sup>( $\mathbf{x}, \pi$ ): 1. Parse  $\pi = (A_i, c_i, z_i)_{i=1}^r$ . 2. Compute  $h_i = \mathsf{H}(\mathbf{x}, \mathbf{A}, i, c_i, z_i)$  for all  $i \in [r]$ . 3. If  $h_i \neq 0^b$  for some  $i \in [r]$ , return 0. 4. If  $\operatorname{Verify}(\mathbf{x}, A_i, c_i, z_i) = 0$  for some  $i \in [r]$ , return 0. 5. Return 1.

**Properties of the Transform.** The original [Fis05] transformation assumes that  $\Sigma$  is correct, has high min-entropy, (special) honest-verifier zero-knowledge (HVZK), special soundness and computationally unique responses. In the randomized [Ks22] Fischlin transformations—which we build upon—special soundness and computationally unique responses are replaced by *strong* special soundness.

**Lemma 3 (Correctness).** Let NIPS be the randomized Fischlin transformation for  $\Sigma$ -protocol  $\Sigma$ . Suppose  $t \leq k$ . Then NIPS has correctness error of at most  $r \cdot e^{-2^{t-b}}$ . Moreover, an honest prover runs in  $poly(2^t)$  steps.

We note that even for  $t = r \cdot |\mathcal{CH}|$ , the expected time is concentrated strongly around  $r2^b$  by Chernoff bounds, assuming  $2^b \gg |\mathcal{CH}|$ . Hence, NIPS is expected polynomial time if  $r2^b = \text{poly}(\lambda)$ . To counteract large correctness error, the prover can simply retry. This will not by relevant in our case, as  $\mathcal{CH}$  must be superpolynomial for security, and thus the correctness error is negligible.

*Proof (Sketch).* Because we sample  $c_i$  without replacement, each oracle query is fresh. Thus, each  $h_i$  within each iteration is  $0^b$  with probability  $2^{-b}$ . The prover fails if no challenge for some round i hashes to  $0^b$ , which happens with probability  $(1 - 2^{-b})^{|\mathcal{CH}|}$ . Or if  $2^t$  queries are exceeded and less than r accepting queries were found, which happens with probability at most  $r \cdot (1 - 2^{-b})^{2^t}$ . Since we assume k > t, the latter case will always be reached first. We bound this probability by

$$r \cdot (1 - 2^{-b})^{2^t} = r \cdot ((1 - 2^{-b})^{2^b})^{2^{t-b}} \le r \cdot e^{2^{t-b}}.$$

**Lemma 4 ((Relaxed) Knowledge Soundness).** Let NIPS be the randomized Fischlin transformation for  $\Sigma$ -protocol  $\Sigma$  for NP-relation R. If  $\Sigma$  is strongly knowledge sound for relation  $\tilde{R} \supseteq R$ , then for any (potentially unbounded) adversary A on relaxed knowledge soundness for relation  $\tilde{R}$ making at most Q queries<sup>12</sup>, we have

$$\mathsf{AdvKS}^{\mathsf{NIPS},\tilde{\mathsf{R}}}_{\mathcal{A}}(Q,\lambda) \leq Q \cdot 2^{-r \cdot b}.$$

In particular, if  $Q = \text{poly}(\lambda)$  and  $2^{-r \cdot b} = \text{negl}(\lambda)$ , then  $\text{AdvKS}_{\mathcal{A}}^{\text{NIPS},\tilde{\mathsf{R}}}(Q,\lambda) = \text{negl}(\lambda)$ .

Proof (Sketch). The argument is identical to [Fis05, Ks22]. Namely, suppose an adversary  $\mathcal{A}$  succeeds to generate  $\pi = (A_i, c_i, z_i)_{i=1}^r$ . Let  $\mathbf{A} = (A_1, \ldots, A_r)$ . By strong special soundness, the extractor succeeds to extract from  $\pi$  if it finds two accepting transcripts completing  $(\mathbf{x}, \mathbf{A})$  among the random oracle queries. Hence, the extractor fails if H was queried as  $\mathsf{H}(\mathbf{x}, \mathbf{A}, i, c_i, z_i)$  with accepting transcript  $(\mathbf{x}, A_i c_i, z_i)$  only once for every *i*. By basic probability theory, this happens with probability at most  $(2^{-b})^r = 2^{-rb}$ . By a union bound over all possible  $\mathbf{A}$  which  $\mathcal{A}$  may have tried, the claim follows.  $\Box$ 

Lemma 5 (Zero-Knowledge and Witness-Indistinguishability). Let NIPS be the randomized Fischlin transformation for  $\Sigma$ -protocol  $\Sigma$ . Suppose  $\Sigma$  is special honest-verifier zero-knowledge and  $N = 2^k = |\mathcal{CH}| \ge 2^4$ . Then NIPS is statistical zero-knowledge (and thus witness-indistinguishable), more precisely, any adversary which makes at most Q queries to H has advantage at most  $Q \cdot 2^{-H_{\min}(\Sigma)} + 3r \cdot 2^{(k-b)/2}$  in the zero-knowledge experiment (resp. witness-indistinguishability experiment).

*Proof (Sketch).* Let Sim be the HVZK simulator for  $\Sigma$ . The simulator NIPS.Sim for the randomized Fischlin transform is straightforward:

- For all  $i \in [r]$ : 1. Pick a random challenge  $c_i \stackrel{\$}{\leftarrow} C\mathcal{H}$ . 2. Run HVZK simulator  $(A_i, z_i) \leftarrow \mathsf{Sim}(\mathbb{x}, c_i)$ . - Let  $\mathbf{A} = (A_1, \dots, A_r)$ . - For all  $i \in [r]$ : 1. Program  $\mathsf{H}(\mathbb{x}, \mathbf{A}, i, c_i, z_i) := 0^b$ . - Output  $\pi = (A_i, c_i, z_i)_{i=1}^r$ .

Before we analyze the simulator's success probability, observe that the simulator biases the random oracle outputs towards 0<sup>b</sup>. In [Sch22] it was shown that for polynomial challenge  $C\mathcal{H} = \operatorname{poly}(\lambda)$  this bias is noticeable. Below, we show that for superpolynomial  $C\mathcal{H}$ , the reprogramming is *not* noticeable. Roughly, we show that the statistical distance which programming H at some input  $(\mathbb{x}, \mathbf{A}, i, c_i, z_i)$  incurred is at most  $4 \cdot 2^{-(k-b)/2}$ , where  $2^k = |C\mathcal{H}|$ . The statement of the lemma is then obtained by a union bound. We proceed with the proof of these claims.

Observe that if no string  $s = (\mathbf{x}, \mathbf{A}, *)$  has ever been queried to H, then H is completely free to program. This happens, except with probability  $Q \cdot 2^{-\mathsf{H}_{\min}(\Sigma)}$ . More precisely, observe that the worst case is when  $(\mathbf{x}, \mathbf{A}, c_i)$  uniquely determines  $z_i$ , as otherwise the space over which the programming happens is only larger. Thus, we fix the values  $(\mathbf{x}, \mathbf{A}, i)$  and assume that the choice of  $c_i$  also fixes the response  $z_i$ . Then, we must analyze the statistical difference between a random oracle  $\mathsf{H}': \mathcal{CH} \to \{0, 1\}^b$  and its programmed<sup>13</sup> variant  $\mathsf{H}'[c \mapsto 0^b]$  for a random  $c \stackrel{\$}{\leftarrow} \mathcal{CH}$ . Let  $N = |\mathcal{CH}|$ ,  $p = 2^{-b}$ , and

$$X \sim \mathsf{Bin}(N,p), \quad Y \sim \mathsf{Bin}(N-1,p) + 1, \quad Z \sim \mathsf{Bin}(N-1,p)$$

That is, X is (distributed as) the number of  $0^b$  in H', and Y is (distributed as) the number of  $0^b$  in H' $[c \mapsto 0^b]$ , while Z is an auxiliary random variable. It is easy to see that, the statistical distance of H' and H' $[c \mapsto 0^b]$  is in fact the statistical distance between X and Y. Moreover, the binomial distribution Bin(n, p) has it maximum probability (i.e., mode) on  $\lfloor (n+1)p \rfloor$ . Thus, some  $\lfloor (N+1)p \rfloor \leq L < \lfloor Np \rfloor + 1$ , is the maximal choice for which<sup>14</sup>

$$\Pr[X = L] - \Pr[Y = L] \ge 0$$
 and  $\Pr[X = L + 1] - \Pr[Y = L + 1] \le 0$ 

<sup>&</sup>lt;sup>12</sup> We also count in Q the queries induced by  $\mathcal{A}$  through calls to  $\mathcal{O}_{\mathsf{Ver}}$ .

<sup>&</sup>lt;sup>13</sup> Here,  $\mathsf{H}'[c \mapsto 0^b]$  denotes the random oracle obtained by reprogramming  $\mathsf{H}'$  at input c with output  $0^b$ .

<sup>&</sup>lt;sup>14</sup> This is clear visually: The peak of the density of X occurs at  $\lfloor (N+1)p \rfloor$  before the peak of Y, which occurs before  $\lfloor Np \rfloor + 1$ .

Picking this L, the statistical distance is then

$$\sum_{\ell=0}^{L} \Pr[X = \ell] - \Pr[Y = \ell].$$

Using Bin(N, p) = Bin(N - 1, p) + Bin(1, p), we simplify the expression as follows:

$$\sum_{\ell=0}^{L} \Pr[X=\ell] - \Pr[Y=\ell] = \sum_{\ell=0}^{L} \left( \Pr[Z=\ell] \cdot (1-p) + \Pr[Z=\ell-1] \cdot p \right) - \Pr[Z=\ell-1]$$
$$= (1-p) \sum_{\ell=0}^{L} \Pr[Z=\ell] + p \sum_{\ell=0}^{L} \Pr[Z=\ell-1] - \sum_{\ell=0}^{L} \Pr[Z=\ell-1]$$
$$= (1-p) \sum_{\ell=0}^{L} \Pr[Z=\ell] + p \sum_{\ell=0}^{L-1} \Pr[Z=\ell] - \sum_{\ell=0}^{L-1} \Pr[Z=\ell]$$
$$= (1-p) \Pr[Z=L]$$

where we use that (1-p) + p = 1 and to telescope the summands. As the above holds for any L, and we know there is a choice for which it equals SD(X, Y), we have shown that

$$SD(X,Y) \le \max_{I} \Pr[Z=L]$$

To analyze  $\Pr[Z = L]$ , we use the Berry-Esseen theorem, which is an explicit version of the central limit theorem. Let  $B_i \sim \text{Bin}(1,p)$ , and  $\sigma = \sqrt{\text{Var}(B_i)} = p(1-p)$ , and  $\rho = \mathbb{E}[|\text{Bin}(1,p) - p|^3]$ . Observe that

$$\rho = \mathbb{E}[|\mathsf{Bin}(1,p) - p|^3] \le \mathbb{E}[|\mathsf{Bin}(1,p) - p|^2] = \mathsf{Var}(\mathsf{Bin}(1,p)) = \sigma^2 \tag{11}$$

holds because  $|\mathsf{Bin}(1,p) - p| \leq 1$ . The Berry-Esseen theorem, with explicit constant C = 1, asserts that for the cumulative distribution function  $F_n$  of the standardized sum  $S_n = \frac{\sum_{i=1}^n B_i}{\sigma \sqrt{n}}$ , it holds that for all  $x \in \mathbb{R}$ 

$$|F_n(x) - \Phi(x)| \le \frac{\rho}{\sigma^3 \cdot \sqrt{n}}.$$
(12)

Going back to bounding  $\Pr[Z = L]$  over all choices of L, observe that for  $x \in \mathbb{Z}$ 

$$\Pr[Z = x] = \Pr[Z \le x] - \Pr[Z \le x - 1]$$
  
=  $F_n\left(\frac{x - np}{\sigma\sqrt{n}}\right) - F_n\left(\frac{x - 1 - np}{\sigma\sqrt{n}}\right)$   
 $\le \Phi\left(\frac{x - np}{\sigma\sqrt{n}}\right) - \Phi\left(\frac{x - 1 - np}{\sigma\sqrt{n}}\right) + 2\frac{\rho}{\sigma^3 \cdot \sqrt{n}}$   
 $\le \Phi\left(\frac{x - np}{\sigma\sqrt{n}}\right) - \Phi\left(\frac{x - 1 - np}{\sigma\sqrt{n}}\right) + 2\frac{1}{\sigma \cdot \sqrt{n}}$ 

where n = N - 1 and we used the definition of a CDF,  $F_n$ , Equation (12) and Equation (11), in that order. Now, we can bound the difference in  $\Phi$  by its density  $\varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$  (i.e., derivative) integrated over any interval of width  $\frac{1}{\sigma\sqrt{n}}$ , which yields

$$\Phi\left(\frac{x-np}{\sigma\sqrt{n}}\right) - \Phi\left(\frac{x-1-np}{\sigma\sqrt{n}}\right) \le \frac{1}{\sigma\cdot\sqrt{n}} \cdot \max_{x\in\mathbb{R}}\varphi(x) \le \frac{1}{\sigma\cdot\sqrt{n}} \cdot \frac{1}{\sqrt{2\pi}}$$

Finally, using  $\sqrt{\frac{15}{16}}\sqrt{n+1} \ge \sqrt{n}$  for  $n+1 = N \ge 2^4$ , and plugging  $p = 2^{-b}$  into  $\sigma = \sqrt{p(1-p)} = 2^{-b/2}\sqrt{1-2^b} \ge 2^{-b/2}\sqrt{\frac{1}{2}}$ , we obtain for all x

$$\max_{x \in \mathbb{Z}} \Pr[Z = x] \le \frac{1}{\sigma \cdot \sqrt{n}} \cdot \frac{1}{\sqrt{2\pi}} + 2\frac{1}{\sigma \cdot \sqrt{n}} \le 2.5 \cdot \frac{1}{\sigma \cdot \sqrt{n}} \le 2.5 \cdot \sqrt{\frac{16}{15}} \frac{1}{\sigma \cdot \sqrt{n+1}} \le 4 \cdot 2^{-(k-b)/2}$$

where we use that  $2.5 \cdot \sqrt{\frac{16}{15}} \cdot \sqrt{2} \leq 4$ . With this bound on  $\Pr[Z = L]$ , we conclude by a union bound over all r repetitions, that the overall the statistical distance is at most  $3r \cdot 2^{-(k-b)/2} + Q \cdot 2^{-H_{\min}(\Sigma)}$ .

**Concrete Instantiation.** In our blind signature scheme, we need a non-interactive zero-knowledge proof for the relation  $R_{Ped}$  defined in Equation (5) with relaxed soundness relation  $\tilde{R}_{Ped}$  (Equation (6)). Note that the relation  $R_{Ped}$  is given via the linear map

$$\phi_{U,G} \colon \mathbb{Z}_n^2 \to \mathbb{G}, \quad (m,t) \mapsto mU + tG, \qquad \text{where } U, G \in \mathbb{G}.$$

Thus, we obtain a suitable  $\Sigma$ -protocol  $\Sigma_{\mathsf{Ped}}$  for  $\mathsf{R}_{\phi_{U,G}}$  as explained in Appendix B. We can then apply the simplified Fischlin transformation on this  $\Sigma$ -protocol if it is strongly special sound for the relation  $\widetilde{\mathsf{R}}_{\mathsf{Ped}}$  and has high min-entropy. We briefly show this.

**Proposition 3.** The  $\Sigma$ -protocol  $\Sigma_{\mathsf{Ped}}$  is strongly special sound for  $\tilde{\mathsf{R}}_{\mathsf{Ped}}$  and has high min-entropy.

*Proof (Sketch).* The high min-entropy property follows immediately because for  $(A, \mathsf{st}) \leftarrow \mathsf{Init}(\mathbb{x}, \mathbb{w})$ , the commitment A is uniformly distributed over  $\mathbb{G}$ . In particular,  $\mathsf{H}_{\mathsf{min}}(\Sigma_{\mathsf{Ped}}) = \log(p)$ .

For strong special soundness, let  $tr = (A, c, \mathbf{z})$  and  $tr' = (A, c', \mathbf{z}')$  be two transcripts for statement  $\mathbf{x}$  with  $tr \neq tr'$ . Recall that  $\tilde{\mathsf{R}}_{\mathsf{Ped}} = \{(\mathbf{x}, \mathbf{w}) \mid \mathbf{w}G = U \lor (\mathbf{x}, \mathbf{w}) \in \mathsf{R}_{\mathsf{Ped}}\}$ , where  $\mathbf{x} = (C, U, G)$ . If  $c \neq c'$ , then (standard) special soundness of  $\Sigma_{\mathsf{Ped}}$  yields a witness  $\mathbf{w}$  such that  $(\mathbf{x}, \mathbf{w}) \in \mathsf{R}_{\mathsf{Ped}} \subset \tilde{\mathsf{R}}_{\mathsf{Ped}}$ . Else, for c = c', we have that  $\Delta \mathbf{z} \coloneqq \mathbf{z} - \mathbf{z}' \neq \mathbf{0}$ , and thus

$$\phi_{U,G}(\mathbf{z}') - c \cdot C = A = \phi_{U,G}(\mathbf{z}) - c \cdot C$$
$$\implies \phi_{U,G}(\mathbf{z}' - \mathbf{z}) = 0$$
$$\implies \Delta z_1 U + \Delta z_2 G = 0.$$

Thus, both  $\Delta z_1$  and  $\Delta z_2$  are non-zero, and the above equation yields  $\mathbb{W}G = U$  for  $\mathbb{W} \coloneqq -\Delta z_2/\Delta z_1$ .

Remark 6 (Concrete Parameters). To estimate proof sizes for our concrete parameter calculations we use b = 8 and  $r = \lceil \lambda/b \rceil = 16$  to determine the size of the proof. The remaining parameters do not influence the proof size.

## D Deferred Proofs

In this section, we provide formal proofs that were omitted in the main body.

#### D.1 Proof of Lemma 1

Proof (Lemma 1). Let  $\mathcal{A}$  be an PPT adversary on the game described in Lemma 1, below denoted by Game BB. The reduction  $\mathcal{B}$  obtains the challenge (G, U, V) from the CDH game and proceeds as follows. Initially,  $\mathcal{B}$  invokes  $\mathcal{A}$  on input  $1^{\lambda}$  to obtain  $(\overline{m}^*, \operatorname{st}_{\mathcal{A}})$ . Then,  $\mathcal{B}$  samples  $\delta \stackrel{s}{\leftarrow} \mathbb{Z}_p$  and sets  $H := -\overline{m}^* \cdot U + \delta G$ . Next,  $\mathcal{B}$  invokes  $\mathcal{A}$  and obtains  $(S_1^*, S_2^*) \leftarrow \mathcal{A}^{\mathcal{O}}(G, U, H, V, \operatorname{st}_{\mathcal{A}})$ , where  $\mathcal{O}$  is simulated by  $\mathcal{B}$  as follows:

 $-\mathcal{O}(\overline{m}): \text{ If } \overline{m} = \overline{m}^*, \text{ output } \bot. \text{ Else, set } \Delta \overline{m} \coloneqq \overline{m} - \overline{m}^* \neq 0 \text{ and sample } r \xleftarrow{\$} \mathbb{Z}_p. \text{ Set } S_2 \coloneqq rG - \frac{1}{\Delta \overline{m}}V \text{ and } S_1 \coloneqq (r \cdot \Delta \overline{m})U + (r \cdot \delta)G - (\frac{v}{\Delta \overline{m}} \cdot \delta)G.$ 

Finally,  $\mathcal{B}$  outputs  $C \coloneqq S_1^* - \delta \cdot S_2^*$  as its CDH solution.

Let us analyze the success probability of  $\mathcal{B}$ . Denote by (u, v) the (unknown) discrete logarithms of (U, V), respectively. First, observe that  $\mathcal{B}$ 's elements (G, U, H, V) are distributed as in Game BB. Recall that in Game BB, the oracle output is  $(S_1, S_2) = (u \cdot V + s \cdot X_{\overline{m}}, s \cdot G)$  for  $X_{\overline{m}} = \overline{m} \cdot U + H$ . It follows that the simulated  $S_2 = (r - \frac{v}{\Delta \overline{m}})G$  follows the distribution of  $S_2$  in Game BB—with implicit  $s = r - \frac{v}{\Delta \overline{m}}$ . Also, the simulated  $S_1$  is distributed as in Game BB due to:

$$S_{1} = (r \cdot \Delta \overline{m})U + (r \cdot \delta)G - \left(\frac{v}{\Delta \overline{m}} \cdot \delta\right)G$$
  
$$= u \cdot V - vU + (r \cdot \Delta \overline{m})U + (r \cdot \delta)G - \left(\frac{v}{\Delta \overline{m}} \cdot \delta\right)G$$
  
$$= u \cdot V + \left(r - \frac{v}{\Delta \overline{m}}\right) \cdot (\Delta \overline{m} \cdot U + \delta G)$$
  
$$= u \cdot V + (r - \frac{v}{\Delta \overline{m}}) \cdot (\overline{m} \cdot U - \overline{m}^{*} \cdot U + \delta G)$$
  
$$= u \cdot V + (r - \frac{v}{\Delta \overline{m}}) \cdot (\overline{m} \cdot U + H)$$
  
$$= u \cdot V + s \cdot (\overline{m} \cdot U + H).$$

Thus, the view of  $\mathcal{A}$  in the interaction with the reduction  $\mathcal{B}$  is as in Game BB. Further, if  $\mathbb{x}_0^* \in \mathscr{L}_{bb}$  there is some  $\mathbb{w}_0^* = (s^*, u^*)$  such that  $(\mathbb{x}_0^*, \mathbb{w}_0^*) \in \mathsf{R}_{bb}$ , so:

$$S_1^* = u^* \cdot V + s^* \cdot X_{\overline{m}^*} \tag{13}$$

$$S_2^* = s^* G \tag{14}$$

$$U = u^* G. \tag{15}$$

Due to Equation (15), it holds that  $u = u^*$ . Also, since  $X_{\overline{m}^*} = \delta G$  by construction, Equations (13) and (14) yield  $S_1^* = u \cdot V + \delta S_2^*$ . Thus,  $\mathcal{B}$ 's output  $C = u \cdot V$  is a valid CDH solution conditioned on  $\mathcal{A}$ 's output satisfying  $\mathbb{x}_0^* \in \mathscr{L}_{bb}$ . The runtime of  $\mathcal{B}$  is roughly that of  $\mathcal{A}$ . This concludes the proof.

#### D.2 Proof of Lemma 2

Proof (Lemma 2). We introduce a series of intermediate games between Game 1 and Game 2 in the proof of Theorem 1. Let us give a brief overview. In Game 1.1, the game also extracts a witness  $w_{\mathsf{Ped}}$  from  $\pi_{\mathsf{Ped}}$  using the knowledge extractor in the first signing oracle. Then, we gradually move to Game 1.5 (with intermediate games Game 1.2, Game 1.3, Game 1.4) such that both  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$  are simulated without secret key  $\mathsf{sk} = u$  in Game 1.5. For this, we follow the techniques in the proof of Theorem 1—specifically in Game 5 to Game 8—except for minor modifications. In Game 1.6, we let the game abort if  $w_{\mathsf{Ped}}$  is the discrete logarithm of U which is justified under the DLOG assumption. Then, we revert the changes made in Game 1.2 to Game 1.5, but we keep the abort condition added in Game 1.6. The resulting game is identical to Game 2.

Game 1.1 (Extract  $w_{\mathsf{Ped}}$  from  $\pi_{\mathsf{Ped}}$ ). This game is identical to Game 1 except that the game extracts a witness  $w_{\mathsf{Ped}}$  from  $\pi_{\mathsf{Ped}}$  in the first signer oracle  $\mathcal{O}_{\mathsf{S}_1}$ . In more detail, we modify the first part of the signer oracle  $\mathcal{O}_{\mathsf{S}_1}$  as follows. Initially, it proceeds as in Game 1 until the check  $\mathsf{NIPS}_{\mathsf{Ped}}.\mathsf{Ver}^{\mathsf{H}_{\mathsf{Ped}}}(\mathbb{x}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}}) = 1$ . If the check fails, it outputs  $\bot$  as before. Else, the game sets  $w_{\mathsf{Ped}} \leftarrow \mathsf{Ext}_{\mathsf{Ped}}(\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}, \pi_{\mathsf{Ped}})$ . (Recall that  $\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}$  denotes the queries to  $\mathsf{H}_{\mathsf{Ped}}$  made by  $\mathcal{A}$  and the game.) The game aborts its entire execution if  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \notin \tilde{\mathsf{R}}_{\mathsf{Ped}}$ .

Clearly, there is a reduction  $\mathcal{A}_{KS}$  on knowledge soundness of  $\mathsf{NIPS}_{\mathsf{Ped}}$  with running time similar to  $\mathcal A$  such that

$$|\varepsilon_1 - \varepsilon_{1.1}| \leq \mathsf{AdvKS}_{\mathcal{A}_{\mathsf{KS}}}^{\mathsf{NIPS}_{\mathsf{Ped}},\mathsf{R}_{\mathsf{Ped}}}(\lambda).$$

**Game 1.2 (Sample DDH tuples).** From now on, the game samples real DDH tuples in  $H_{ddh}$ . That is, the game now holds an initially empty table  $T_{ddh}[\cdot] \coloneqq \bot$ . Whenever random oracle  $H_{ddh}$  is queried on an input  $\tau$  and the hash value is not yet defined, the game samples  $d_2 \stackrel{*}{\leftarrow} \mathbb{Z}_p$  and sets  $(D_2^{\tau}, D_3^{\tau}) \coloneqq (d_2G, d_2D_1)$  instead of  $(D_2, D_3) \stackrel{*}{\leftarrow} \mathbb{G}^2$ . Additionally, witness  $d_2$  is stored in the table, i.e.,  $T_{ddh}[\tau] := d_2$ . Note that by design, we have  $\mathbf{D}^{\tau} \in \mathscr{L}_{ddh}$  for all  $\tau$ .

Clearly, there is a reduction  $\mathcal{B}'_1$  to Q-DDH with  $Q = Q_{ddh}$  with running time similar to  $\mathcal{A}$  such that

$$|\varepsilon_{1.1} - \varepsilon_{1.2}| \leq \mathsf{AdvQDDH}^{\mathbb{G}}_{\mathcal{B}'_1}(Q_{\mathsf{ddh}}, \lambda).$$

Game 1.3 (Use DDH witness for  $\Sigma_1$ ). Now, the game computes the  $\Sigma_1$  transcript  $(\mathbf{A}_1, c, z_1)$  via the witness  $\mathsf{T}_{\mathsf{ddh}}[\tau]$ . That is, the game samples  $c_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_p$  and  $(\mathbf{A}_1, \mathsf{st}_1) \leftarrow \mathsf{lnit}_1(\mathbb{x}_1, \mathbb{w}_1)$  in  $\mathcal{O}_{\mathsf{S}_1}$ , where  $\mathbb{w}_1 \coloneqq \mathsf{T}_{\mathsf{ddh}}[\tau]$  and  $\mathbb{x}_1 \coloneqq (G, \mathbf{D}^{\tau})$ . In  $\mathcal{O}_{\mathsf{S}_2}$ , the game computes  $z_1 \leftarrow \mathsf{Resp}_1(\mathsf{st}_1, c_1)$ .

By perfect HVZK of  $\Sigma_1$ , the  $\Sigma_1$  transcripts  $(\mathbf{A}_1, c_1, z_1)$  in Game 1.2 and Game 1.3 are identically distributed. Thus, we have

 $\varepsilon_{1.2} = \varepsilon_{1.3}.$ 

**Game 1.4 (Simulate**  $\Sigma_0$ ). The game now simulates the  $\Sigma_0$  transcript  $(\mathbf{A}_0, c_0, \mathbf{z}_0)$  via HVZK in  $\mathcal{O}_{\mathsf{S}_1}$  and  $\mathcal{O}_{\mathsf{S}_2}$ . In more detail, the game computes  $\mathbf{A}_0^C$  in  $\mathcal{O}_{\mathsf{S}_1}$  via

$$c_0 \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p, \ (\mathbf{A}_0^C, \mathbf{z}_0) \leftarrow \mathsf{Sim}_0(\mathbf{x}_0, c_0)$$

In  $\mathcal{O}_{S_2}$ , the game sets  $c_1 \coloneqq c - c_0$  and outputs  $\mathbf{z}_0$  from  $\mathcal{O}_{S_1}$ . The other response  $z_1$  is computed via  $w_1$  as introduced in Game 1.3.

As in Game 7, we can show that by perfect HVZK of  $\Sigma_0$ , we have that

$$\varepsilon_{1.3} = \varepsilon_{1.4}.$$

**Game 1.5 (Send random T).** We change the signer oracle  $\mathcal{O}_{S_1}$  again. Recall that until now, the signer oracle defined the vector  $\mathbf{T} \coloneqq \phi_0(X_C, \mathbf{w}_0)$ . In this game,  $\mathbf{T}$  is sampled differently. Namely, the game samples  $T_1, T_2 \xleftarrow{\simes} \mathbb{G}$  at random and sets  $T_3 \coloneqq U$ .

Intuitively, since (H, sG, sH) form Diffie-Hellman tuples and are included in the definition of **T** in Game 1.4, replacing sH by a random element should be indistinguishable and make the first component of **T** random. This intuition can be formalized as in Game 8. That is, there is a reduction  $\mathcal{B}'_2$  with running time similar to  $\mathcal{A}$ , such that

$$|\varepsilon_{1.4} - \varepsilon_{1.5}| \leq \mathsf{AdvQDDH}_{\mathcal{B}'_{o}}^{\mathbb{G}}(Q_S, \lambda).$$

Note that at this point, the game does not need the secret key sk = u to simulate the signing oracles. Instead, the game simulates the signing oracles  $\mathcal{O}_{S_1}$  and  $\mathcal{O}_{S_2}$  via the witness for  $\Sigma_1$  in  $T_{ddh}$ , and randomized **T**. This is important for the following game.

**Game 1.6 (Abort if**  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \notin \mathsf{R}_{\mathsf{Ped}}$ ). We change the signer oracle  $\mathcal{O}_{\mathsf{S}_1}$  such that the execution aborts if  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \notin \mathsf{R}_{\mathsf{Ped}}$ . In more detail, we modify the first part of the signer oracle  $\mathcal{O}_{\mathsf{S}_1}$  as follows. After the game extracts  $\mathbb{w}_{\mathsf{Ped}} \leftarrow \mathsf{Ext}_{\mathsf{Ped}}(\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}, \mathbb{x}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}})$ , the game parses  $(\overline{m}, t) \coloneqq \mathbb{w}_{\mathsf{Ped}}$  and aborts its entire execution if parsing  $\mathbb{w}_{\mathsf{Ped}}$  fails or  $C \neq \overline{m}U + tG$ .

Recall that due to the abort condition added in Game 1.1, it holds that  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \in \mathsf{R}_{\mathsf{Ped}}$ . Also, by definition of  $\mathsf{R}_{\mathsf{Ped}}$  (cf. Equation (5)), if  $(\mathbb{x}_{\mathsf{Ped}}, \mathbb{w}_{\mathsf{Ped}}) \in \mathsf{R}_{\mathsf{Ped}}$  it holds that  $\mathbb{w}_{\mathsf{Ped}} = (\overline{m}, t)$  and  $C = \overline{m}U + tG$ . Thus, it remains to bound the probability that  $\mathbb{x}_{\mathsf{Ped}} = u$  with U = uG. For this, we provide a reduction  $\mathcal{B}'_3$  to DL. Essentially,  $\mathcal{B}'_3$  simulates Game 1.5 to adversary  $\mathcal{A}$  except that it embeds an DL challenge U into the verification key vk. If some extracted  $\mathbb{w}_{\mathsf{Ped}}$  in  $\mathcal{O}_{\mathsf{S}_1}$  fulfils uG = U,  $\mathcal{B}'_3$  outputs U to as its DL solution. Note that  $\mathcal{A}_{\mathsf{DL}}$  is well-defined, as the discrete logarithm u of U is not required to simulate the Game 1.5 anymore. Clearly, it holds that

$$|\varepsilon_{1.5} - \varepsilon_{1.6}| \leq \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DI}}}^{\mathbb{G}}(\lambda).$$

Game 1.7 to Game 1.10 (Revert back the changes). In Game 1.7 to Game 1.10 we revert back the changes made in Game 1.2 to Game 1.5, but we keep the abort condition introduced in Game 1.6. Roughly, in Game 1.7, we set  $\mathbf{T} \coloneqq \phi_0(X_C, \mathbf{w}_0)$  again. This change is justified by the Q-DDH assumption. In Game 1.8, we compute the  $\Sigma_0$  using the witness  $\mathbf{w}_0$  again and in Game 1.9, we simulate the  $\Sigma_1$  transcript via the HVZK simulator. Finally, in Game 1.10, we output random elements in H<sub>ddh</sub> again (which is again justified by Q-DDH). This follows as above and we omit details. Observe that Game 1.10 is identical to Game 2. By summing up all the bounds from above, we obtain

$$|\varepsilon_1 - \varepsilon_2| \leq \mathsf{AdvKS}_{\mathcal{A}_{\mathsf{KS}}}^{\mathsf{NIPS}_{\mathsf{Ped}},\tilde{\mathsf{R}}_{\mathsf{Ped}}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + \mathsf{AdvQDDH}_{\mathcal{B}'_2}^{\mathbb{G}}(Q_S, \lambda)) + \mathsf{AdvDL}_{\mathcal{A}_{\mathsf{DL}}}^{\mathbb{G}}(\lambda) + 2 \cdot (\mathsf{AdvQDDH}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQDD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQDD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQDD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQDD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}, \lambda) + 2 \cdot (\mathsf{AdvQDD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQD}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}}, \lambda) + 2 \cdot (\mathsf{AdvQ}_{\mathcal{B}'_1}^{\mathbb{G}}(Q_{\mathsf{ddh}, \lambda) + 2 \cdot (\mathsf{AdvQ}_{\mathcal{B}'_1}^{\mathbb{$$

The statement now follows via Remark 5.

#### D.3 Proof of Theorem 2

*Proof (Theorem 2).* By NIZK correctness, it may fail with probability at most  $\gamma_{err}$ . If the NIZK does not fail, the user always outputs a correct signature. In the following, we always assume NIZK verification succeeded.

Step 0 (Correctness of PreBS). As a first step, we argue that the unblinded signature PreBS is correct. This is equivalent to the equations

$$\phi_0(X, \mathbf{z}_0) = \mathbf{A}_0 + c_0 \mathbf{S} \quad \text{and} \quad \phi_1(z_1) = \mathbf{A}_1 + c_1 \mathbf{D}^{\tau} \tag{16}$$

being satisfied where  $c'_0, c'_1, s', \mathbf{z}'_0, z'_1$  all set to 0 in Figure 2. To see this, observe that the user and signer engage in an OR-compiled  $\Sigma$ -protocol for  $\mathsf{R}_{\mathsf{bb}}$  or  $\mathsf{R}_{\mathsf{ddh}}$ . Since this  $\Sigma$ -protocol is perfectly correct (Appendix B), It always holds that

$$\phi_0(X_C, \mathbf{z}_0^*) = \mathbf{A}_0^* + c_0 \mathbf{T}^*$$
 and  $\phi_1(z_1) = \mathbf{A}_1^* + c_1 \mathbf{D}^{\tau}$ .

Hence, the user never aborts when checking the response  $\mathbf{z}_0^*, z_1^*$ . It is straightforward to check that unblinding the transcript w.r.t. t yields (16), namely

$$\phi_0(X, \mathbf{z}_0^*) = \phi_0(X + tG, \mathbf{z}_0^*) - t \cdot (\mathbf{z}_0^*G, 0, 0)$$
  
=  $\phi_0(X + tG, \mathbf{z}_0^*) - t \cdot (A_{0,2} + c \cdot T_2, 0, 0)$   
=  $\mathbf{A}_0^* + c\mathbf{T}^* - t \cdot (A_{0,2} + c \cdot T_2, 0, 0)$   
=  $\mathbf{A}_0 + c\mathbf{S}$ 

where the first step uses that  $\phi_0(X + tG, (a, b)) = \mathbf{R} + t(R_2, 0, 0)$  holds, where  $\mathbf{R} = \phi_0(X, (a, b))$  and  $R_2$  is independent of X. Then we use linearity and the definition of  $\mathbf{A}_0$  and  $\mathbf{S}$ . Thus, correctness holds if  $c'_0, c'_1, s', \mathbf{z}'_0, z'_1$  all set to 0.

Now, we argue that the randomization preserves perfect correctness. We argue in 3 separate steps, where each step may assume all other randomizations are set to 0. This holds since, by linearity of each randomization, randomizations do not "interfere" with each other, i.e., consecutive application still preserves perfect correctness (simply by renaming of variables to include prior randomization terms and using that the randomized transcripts are again accepting). We leave this step to the reader.

Step 1 (Randomizing s'). We have for  $\mathbf{S}^* = \mathbf{T}^* - (t \cdot T_2, 0, 0)^{\mathsf{T}}$  that

$$((G, V, X_C, \mathbf{T}^*), (s^*, \mathsf{sk})) \in \mathsf{R}_{\mathsf{bb}} \iff ((G, V, X, \mathbf{S}^*), (s^*, \mathsf{sk})) \in \mathsf{R}_{\mathsf{bb}} \iff ((G, V, X, \mathbf{S}), (s^* + s', \mathsf{sk})) \in \mathsf{R}_{\mathsf{bb}}$$
(17)

and the transcript w.r.t.  $S^*$  is accepting by correctness of the unblinded scheme. The transcript verification remains accepting for blinding with  $s' \neq 0$  since

$$\phi_0(X, \mathbf{z}_0) = \phi_0(X, \mathbf{z}_0^* - c_0(s', 0))$$
  
=  $\phi_0(X, \mathbf{z}_0^*) - c_0\phi(X, s', 0)$   
=  $(\mathbf{A}_0 + c_0(\mathbf{S} + \phi_0(X, (s', 0)))) - c_0\phi(X, s', 0))$   
=  $\mathbf{A}_0 + c_0\mathbf{S}$ .

Thus, perfect correctness is preserved.

Step 2 (Randomizing c). This is a completely standard argument for blind signatures based on  $\Sigma$ -protocols, except that we additionally need to consider sum of challenges due to the OR-compilation. For this, observe that

$$c_0 + c_1 = c_0^* + c_1^* + c_0' + c_1' = c^* + (c_0' + c_1') = c_0'$$

For the randomization of the transcripts, we consider only the branch of  $\phi_0$ , since  $\phi_1$  is completely analogous. Again, accepting transcripts are preserved by the randomization since

$$\phi_0(X, \mathbf{z}_0) = (\mathbf{A}_0 - c'_0 \mathbf{S}) + (c_0^* - c'_0) \mathbf{S} = \mathbf{A}_0 + c_0^* \mathbf{S}.$$

Hence, perfect correctness is preserved.

Step 3 (Randomizing  $A_0, A_1$ ). This is another completely standard argument. Again, we consider only the branch of  $\phi_0$ , since  $\phi_1$  is completely analogous. Again, accepting transcripts are preserved by the randomization since

$$\phi_0(X, \mathbf{z}_0) = \phi_0(X, \mathbf{z}_0^* + \mathbf{z}_0') = (\mathbf{A}_0^* + c_0 \mathbf{S}) + \phi_0(X, \mathbf{z}_0') = \mathbf{A}_0 + c_0 \mathbf{S}$$

Again, perfect correctness is preserved.

#### D.4 Proof of Theorem 3

*Proof (Theorem 3).* We consider an unbounded adversary and reduction (except for the number of random oracle queries). We argue by game hops, and proceed in roughly three phases: Firstly, we self-sign  $\overline{m}$  by brute-forcing the secret key. Secondly, we use our knowledge of a witness to compute  $(\mathbf{A}_0, \mathbf{A}_1)$  fresh and honestly, and run the  $\Sigma$ -protocols straightline instead of randomizing transcripts. Thirdly, we remove any leakage from  $(C, \pi_{\mathsf{Ped}})$  by simulating the proof.

Let  $\mathcal{A}$  be a PPT adversary against blindness of BS. Denote by  $Q_{\Sigma}, Q_{M}, Q_{ddh}, Q_{Ped}$  the number of oracle queries to  $H_{\Sigma}, H_{M}, H_{ddh}, H_{Ped}$ , respectively, including the queries made by the game. We proceed with a sequence of games. For each game Game i, we denote by  $\varepsilon_i$  the advantage of  $\mathcal{A}$  in Game i.

**Game 0 (Honest).** The partial blindness experiment with bit b. In the following, we modify the signing oracle  $\mathcal{O}_0$  (resp.  $\mathcal{O}_1$ ) for interaction with  $\tau$  and  $m_b$  (resp.  $m_{1-b}$ ). Let

$$\varepsilon_0 \coloneqq \mathsf{AdvPBlind}\mathcal{A}(\lambda).$$

Game 1 (Abort if DDH tuple). If for any query  $\tau$  to  $\mathsf{H}_{\mathsf{ddh}}$  with output  $(D_2^{\tau}, D_3^{\tau})$  it ever occurs that  $\mathbf{D}^{\tau} := (D_1, D_2^{\tau}, D_3^{\tau})$  is a DDH tuple, abort the experiment <sup>15</sup>. Denoting  $Q_{\mathsf{ddh}}$  the number of  $\mathsf{H}_{\mathsf{ddh}}$  queries, we get

$$\varepsilon_0 \le \varepsilon_1 + \frac{Q_{\mathsf{ddh}}}{p}.$$

Game 2 (Abort if  $(G, V, X, \mathbf{T}^*) \notin \mathscr{L}_{bb}$ ). Observe that the signer runs an OR-compilation of  $\Sigma$ -protocols for knowledge of a preimage  $(s^*, \mathsf{sk})$  with  $\mathbf{T} = \phi_0(X_C, (s, \mathsf{sk}))$  or a preimage  $d_2$ with  $\mathbf{D}^{\tau} = \phi_1(d_2)$ . Since both  $\Sigma$ -protocols are special sound, so is their OR-compilation (for relation  $\mathsf{R}_{bb} \cup \mathsf{R}_{ddh}$ ). By Game 1,  $\mathbf{D}^{\tau}$  is never a DDH tuple in the challenge interactions. Thus, if  $(G, V, X_C, \mathbf{T}^*) \notin \mathscr{L}_{bb}$ , then for any choice of  $(\mathbf{A}_0, \mathbf{A}_1)$  there is a unique challenge pair  $(c_0^*, c_1^*)$  for which  $c^* = c_0^* + c_1^*$  has an accepting response. The probability that  $c^* = c + c_1' + c_2'$  hits this unique challenge is 1/p in either interaction. Thus, we get

$$\varepsilon_1 \le \varepsilon_2 + \frac{2}{p}.$$

Observe that  $(G, V, X_C, \mathbf{T}^*) \in \mathscr{L}_{bb}$  implies  $(G, V, X, \mathbf{S}) \in \mathscr{L}_{bb}$ , cf. Equation (17).

**Game 3 (Self-sign**  $\overline{m}$ ). By Game 2,  $(G, V, X_C, \mathbf{T}) \in \mathscr{L}_{bb}$  holds, and therefore by construction of  $\mathbf{S} = \phi_0(X, (s^* + s', \mathbf{sk}))$ , we get  $(G, V, X, \mathbf{S}) \in \mathscr{L}_{bb}$  as well. Moreover, by the rerandomization with  $\phi_0(X, (s', 0))$ ,  $(G, V, X, \mathbf{S}) \in \mathscr{L}_{bb}$  is a uniformly random "inefficient signature" on  $\overline{m}$ . Thus, we can equivalently compute  $\mathbf{S} = \phi_0(X, (s, \mathbf{sk}))$ , where  $\mathbf{sk}$  is obtained from  $\mathsf{vk}$  via brute-force and  $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p$  is fresh signing randomness. For completeness we set  $s' = s - s^*$ . Observe that the distribution of  $\mathbf{S}$  and s' remains unchanged. Again, we apply this change to both oracles. We get

$$\varepsilon_2 = \varepsilon_3$$

**Game 4 (Sample**  $(\mathbf{A}_0, \mathbf{A}_1)$  **fresh).** In this game, we sample  $(\mathbf{r}_0, r_1) \leftarrow \mathbb{Z}_p^2 \times \mathbb{Z}_p$  fresh, set  $(\mathbf{A}_0, \mathbf{A}_1) = (\phi_0(X, \mathbf{r}_0), \phi_1(X, r_1))$ , and let

$$\mathbf{z}'_0 = \mathbf{r}_0 - \mathbf{z}^*_0 - c_0 \cdot (s', 0)$$
 and  $z'_1 = r_1 - z^*_1$ .

We make some observations:

<sup>&</sup>lt;sup>15</sup> The probability of this event can be bounded even without an explicit check. But since we consider unbounded adversary and reduction, checking explicitly is simpler.

- $-(\mathbf{r}_0, r_1)$  is uniquely defined by  $(\mathbf{A}_0, \mathbf{A}_1)$ .
- By the above equations, given fixed  $-(t \cdot A_{0,2}^*, 0, 0)^{\mathsf{T}} + \phi_0(X, \mathbf{z}'_0) c'_0 \mathbf{S}$  as well as fixed  $+\phi_1(z'_1) c'_1 \mathbf{D}^{\mathsf{T}}$ , we have a bijection between  $(\mathbf{r}_0, r_1)$  and  $(\mathbf{z}'_0, z'_1)$ . Thus, the distribution of  $(\mathbf{r}_0, r_1, \mathbf{z}'_0, z'_1)$  is unchanged.
- If the user (and game) does not abort, then the derandomized  $\pi$  is accepting for statement  $(G, V, X, \mathbf{S})$ . Which means

$$\mathbf{z}_0 = \mathbf{r}_0 - c_0(s, \mathsf{sk})$$
 and  $\mathbf{z}_1 = \mathbf{r}_1 - c_1 d_2$ .

In particular,  $(\mathbf{r}_0, r_1)$  is uniquely defined by  $(\mathbf{z}_0, z_1)$  and  $(c_0, c_1)$ .

Overall, we see that the distribution of the game remains unchanged, as we can define uniquely all variables if we pick random masks  $(\mathbf{r}_0, r_1)$  instead of  $(\mathbf{z}'_0, z'_1)$ . Thus, we have

$$\varepsilon_3 = \varepsilon_4.$$

An important consequence of this game and our observations, is that now  $\pi$  is computed "honestly" instead of via rerandomization (but with unbounded power to recover sk).

**Game 5 (Sample**  $c^*, c_0 \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p$ ) Observe that we can sample  $c^*, c_0, c_1$  instead as

$$c^*, c_0 \xleftarrow{\hspace{1.5pt}{\text{\circle*{1.5}}}} \mathbb{Z}_p, \quad c_1 = c - c_0$$

and set  $c'_0 = c^*_0 - c_0$ ,  $c'_1 = c^*_1 - c_1$  and that this does not influence the distribution of  $\pi$ . Thus, we get

 $\varepsilon_4 = \varepsilon_5.$ 

After the change in Game 5, it is clear that  $(c_0, c_1)$  is a uniformly random under the constraint  $c_0 + c_1 = c = \mathsf{H}_{\Sigma}((\mathbb{x}_b, \mathbf{A}_b)_{b \in \{0,1\}}, m)$ , i.e., uniform in  $\{(c_0, c_1) \mid c_0 + c_1 = c\}$ . In particular, the choice of  $(c_0, c_1)$  is now clearly independent from messages sent to (or received by) the signer.

**Game 6 (Simulate**  $\pi_{\text{Ped}}$ ). Observe that the Pedersen commitment  $C \in \mathbb{G}$  can always be opened to any value v by brute-forcing t. Thus, we can simulate the proof  $\pi_C$  by brute-forcing an opening  $\hat{t}$  for  $\overline{m} = 0$  as  $C = \hat{t}G$  and providing a proof for that. We do this in both  $\mathcal{O}_0$  and  $\mathcal{O}_1$ , so that the witness  $(\overline{m}, t)$  is not required anymore. By a straight-forward reduction (using the witnessindistinguishability experiment to compute the proof in  $\mathcal{O}_0$ , and then in  $\mathcal{O}_1$ ), we obtain that there exists some  $\mathcal{A}_{WI}$  with

$$\varepsilon_5 \leq \varepsilon_6 + 2 \cdot \mathsf{AdvWI}_{\mathcal{A}_{\mathsf{WI}}}^{\mathsf{NIPS}}(Q_{\mathsf{Ped}}, \lambda).$$

**Game 7** ( $C \stackrel{*}{\leftarrow} \mathbb{G}$ ). We sample C via  $C \stackrel{*}{\leftarrow} \mathbb{G}$  instead of computing it honestly. Thus, C is now independent of the message. Observe that t is not used anymore since Game 6. Therefore, the distribution is unchanged, and we get

$$\varepsilon_6 = \varepsilon_7.$$

**Game 8 (Wrapping up).** After Game 7, the transcript  $\pi = (\mathbf{A}_0, \mathbf{A}_1, c, c_0, \mathbf{z}_0, z_1)$  is computed entirely by information local to the oracles, and the oracles only send an independently random C, simulated  $\pi$ , and random  $c^*$ . In particular, computation of  $(\mathbf{S}, \pi)$  can be *delayed until after* the protocol completed and the outputs  $\sigma_0$  resp.  $\sigma_1$  are required. Thus, it is clear that the interactions are independent of the messages  $m_0, m_1$  (and that even holds if the common message  $\tau$  differs).<sup>16</sup> Since both interactions and thus final signatures use the same  $\tau$ , we deduce that the bit *b* it perfectly hidden from the adversary's view. Hence,

$$\varepsilon_7 = \varepsilon_8 = 0.$$

Putting everything together we arrive at

$$\varepsilon_0 \leq \frac{2}{p} + 2 \cdot \mathsf{AdvWI}_{\mathcal{A}_{\mathsf{WI}}}^{\mathsf{NIPS}}(Q,\lambda)$$

which concludes the proof.

<sup>&</sup>lt;sup>16</sup> In other words, we have an (unbounded) straightline simulator, which (1) takes no input and simulates a signing session, and then (2) takes as input  $(m, \tau)$  and generates a signature  $\sigma$  on m w.r.t  $\tau$ .

## **E** Deferred Figures

In this section, we provide figures that were deferred from the main body.

Game 0 (One-more Unforgeability)  $\mathsf{Next}(\mathsf{sid}, \tau)$ 1:  $\forall \mathsf{H} \in \{\mathsf{H}_{\mathsf{ddh}}, \mathsf{H}_{\Sigma}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\mathsf{Ped}}\}, \mathcal{Q}_{\mathsf{H}}[\cdot] \coloneqq \bot$ 1: if sid  $\in SID$  then return  $\perp$ 2:  $\mathcal{SID} \coloneqq \emptyset, \mathcal{Q}_T[\cdot] \coloneqq 0$ 2:  $\mathcal{SID} \leftarrow \mathcal{SID} \cup \{\mathsf{sid}\}$  $3: \quad \mathsf{common}[\cdot] \coloneqq \bot, \mathsf{state}[\cdot] = \bot, \mathsf{round}[\cdot] = \bot$ 3: common[sid]  $\leftarrow \tau$ , round[sid]  $\leftarrow 0$ 4: **return** 1 4:  $u \stackrel{\$}{\leftarrow} \mathbb{Z}_p, H, V \stackrel{\$}{\leftarrow} \mathbb{G}, U \coloneqq uG$ 5:  $\mathsf{vk} := (G, U, H, V, D_1), \mathsf{sk} := u$  $\mathcal{O}_{S_1}(\mathsf{sid}, C, \pi_{\mathsf{Ped}})$  $\mathbf{6}: \quad \mathsf{oracles} \coloneqq (\mathcal{O}_{\mathsf{S}_1}, \mathcal{O}_{\mathsf{S}_2}, \mathsf{H}_{\mathsf{ddh}}, \mathsf{H}_{\Sigma}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\mathsf{Ped}})$ 1:  $\mathbf{req} \ \mathsf{round}[\mathsf{sid}] = 0$ 7:  $(\tau^*, (m_j^*, \sigma_j^*)_{j \in [k]}) \leftarrow \mathcal{A}^{\mathsf{oracles}}(\mathsf{vk})$ 2:  $\tau \coloneqq \mathsf{common}[\mathsf{sid}]$ 8: **abort** if queried  $[\tau^*] \ge k$ 3:  $\mathbb{X}_{\mathsf{Ped}} \coloneqq (C, U, G)$ 9: **abort** if  $\exists j \in [k]$ , Verify $(vk, m_i^*, \tau^*, \sigma_i^*) = 0$ 4: **req** NIPS<sub>Ped</sub>.Ver<sup>H<sub>Ped</sub>( $x_{Ped}, \pi_{Ped}$ ) = 1</sup> 10: **abort** if  $\exists (i,j) \in [k]^2, i \neq j, m_i^* = m_j^*$ 5:  $s \stackrel{\$}{\leftarrow} \mathbb{Z}_p, \mathbb{W}_0 \coloneqq (s, \mathsf{sk})$ 11 : **return** 1 6:  $X_C \coloneqq C + H, \mathbf{T} \coloneqq \phi_0(X_C, \mathbf{w}_0)$  $H_{ddh}(\tau)$ 7:  $\mathbf{D}^{\tau} \coloneqq (D_1, \mathsf{H}_{\mathsf{ddh}}(\tau))$ 1: if  $\mathcal{Q}_{\mathsf{H}_{\mathsf{ddh}}}[\tau] = \bot$  then 8:  $\mathbb{x}_0^C \coloneqq (G, V, X_C, \mathbf{S}), \mathbb{x}_1 \coloneqq (G, \mathbf{D}^{\tau})$  $(D_2^{\tau}, D_3^{\tau}) \stackrel{\hspace{0.1em}\hspace{0.1em}\hspace{0.1em}}{\leftarrow} \mathbb{G}^2, \mathcal{Q}_{\mathsf{H}_{\mathsf{ddb}}}[\tau] \leftarrow (D_2^{\tau}, D_3^{\tau})$ 9:  $c_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_p, (\mathbf{A}_1, z_1) \leftarrow \mathsf{Sim}_1(\mathbb{X}_1, c_1)$ 2:3: return  $\mathcal{Q}_{\mathsf{H}_{\mathsf{ddh}}}[\tau]$ 10:  $(\mathbf{A}_0^C, \mathsf{st}_0) \leftarrow \mathsf{lnit}_0(\mathbb{x}_0^C, \mathbb{w}_0)$ 11: state[sid]  $\leftarrow (z_1, c_1, st_0)$  $H_{\Sigma}(x)$ 12: round[sid]  $\leftarrow 1$ 1: if  $\mathcal{Q}_{\mathsf{H}_{\Sigma}}[x] = \bot$  then 13: return  $(\mathbf{T}, \mathbf{A}_0^C, \mathbf{A}_1)$ 2:  $c \stackrel{\$}{\leftarrow} \mathbb{Z}_p, \mathcal{Q}_{\mathsf{H}_{\Sigma}}[x] \leftarrow c$  $\mathcal{O}_{S_2}(\mathsf{sid}, c)$ 3: return  $\mathcal{Q}_{\mathsf{H}_{\Sigma}}[x]$ 1: req round[sid] = 1 $H_M(m)$ 2: req  $c \in \mathbb{Z}_p$ 1: if  $\mathcal{Q}_{\mathsf{H}_{\mathsf{M}}}[m] = \bot$  then 3:  $(z_1, c_1, \mathsf{st}_0) \coloneqq \mathsf{state}[\mathsf{sid}]$  $\overline{m} \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p, \mathcal{Q}_{\mathsf{H}_{\mathsf{M}}}[m] \leftarrow \overline{m}$  $4: \quad c_0 \coloneqq c - c_1$ 2:3: return  $Q_{H_M}[m]$ 5:  $\mathbf{z}_0 \leftarrow \mathsf{Resp}_0(\mathsf{st}_0, c_0)$ 6: queried[ $\tau$ ]  $\leftarrow$  queried[ $\tau$ ] + 1  $H_{Ped}(x)$ 7: return  $(\mathbf{z}_0, z_1, c_0)$ 1: if  $\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}[m] = \bot$  then  $y \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathcal{Y}_{\mathsf{Ped}}, \mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}[x] \leftarrow y$ 2:3: return  $Q_{H_{Ped}}[x]$ 

**Fig. 3.** Description of Game 0, identical to the real one-more unforgeability game. For some condition C, **abort if** C makes the game output 0 if C is true and **req** C makes the oracle output  $\perp$  if C is false.

```
Game 12 (One-more Unforgeability)
                                                                                                                                \mathcal{O}_{S_1}(\mathsf{sid}, C, \pi_{\mathsf{Ped}})
 1: \forall \mathsf{H} \in \{\mathsf{H}_{\mathsf{ddh}}, \mathsf{H}_{\Sigma}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\mathsf{Ped}}\}, \mathcal{Q}_{\mathsf{H}}[\cdot] \coloneqq \bot
                                                                                                                                  1: req round[sid] = 1
 2: \mathcal{SID} \coloneqq \emptyset, \mathcal{Q}_T[\cdot] \coloneqq 0
                                                                                                                                  2: \tau \coloneqq \mathsf{common}[\mathsf{sid}]
 3: \operatorname{common}[\cdot] \coloneqq \bot, \operatorname{state}[\cdot] = \bot, \operatorname{round}[\cdot] = \bot
                                                                                                                                  3: \mathbb{X}_{\mathsf{Ped}} \coloneqq (C, U, G)
                                                                                                                                            \mathbf{req} \ \ \mathsf{NIPS}_{\mathsf{Ped}}.\mathsf{Ver}^{\mathsf{H}_{\mathsf{Ped}}}(\mathtt{x}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}}) = 1
 4: \mathsf{ctr}_{\mathsf{M}} \coloneqq 0, q_m^* \xleftarrow{\$} [Q_{\mathsf{M}}], \overline{m}^* \xleftarrow{\$} \mathbb{Z}_p \ // \text{ Game 4}
                                                                                                                                  4:
                                                                                                                                  5: (\overline{m}, t) \leftarrow \mathsf{Ext}_{\mathsf{Ped}}(\mathcal{Q}_{\mathsf{H}_{\mathsf{Ped}}}, \mathbb{x}_{\mathsf{Ped}}, \pi_{\mathsf{Ped}}) \ /\!\!/ \text{ Game 2}
 5: \mathsf{ctr}_{\mathsf{ddh}} \coloneqq 0, q_{\tau}^* \xleftarrow{\$} [Q_{\mathsf{ddh}}], \tau_{q_{\tau}^*} \coloneqq \bot \ // \text{ Game 3}
                                                                                                                                  6: abort if C \neq \overline{m}U + tG // Game 2
 6: \quad h \stackrel{\$}{\leftarrow} \mathbb{Z}_p, H \coloneqq hG
                                                                                                                                  7: X_C \coloneqq C + H
 7: u \stackrel{\$}{\leftarrow} \mathbb{Z}_p, V \stackrel{\$}{\leftarrow} \mathbb{G}, U \coloneqq uG
                                                                                                                                  8: \mathbf{D}^{\tau} \coloneqq (D_1, \mathsf{H}_{\mathsf{ddh}}(\tau))
 8: \mathsf{vk} \coloneqq (G, U, H, V, D_1), \mathsf{sk} \coloneqq u
                                                                                                                                  9: \mathbb{x}_0^C \coloneqq (G, V, X_C, \mathbf{S}), \mathbb{x}_1 \coloneqq (G, \mathbf{D}^{\tau})
 9: oracles := (\mathcal{O}_{S_1}, \mathcal{O}_{S_2}, \mathsf{H}_{\mathsf{ddh}}, \mathsf{H}_{\Sigma}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\mathsf{Ped}})
                                                                                                                                10: if \tau \neq \tau_{q_{\tau}^*} \vee \overline{m}^* = \overline{m} then
10: (\tau^*, (m_j^*, \sigma_j^*)_{j \in [k]}) \leftarrow \mathcal{A}^{\mathsf{oracles}}(\mathsf{vk})
                                                                                                                                                 \mathbf{T} \stackrel{\$}{\leftarrow} \mathbb{G}^2 \times \{U\} \quad /\!\!/ \text{ Game 8}
                                                                                                                                11:
11: abort if queried[\tau^*] \ge k
                                                                                                                                12: else
12: abort if \exists j \in [k], Verify(vk, m_i^*, \tau^*, \sigma_i^*) = 0
                                                                                                                                                 s \stackrel{\$}{\leftarrow} \mathbb{Z}_p, \mathbb{w}_0 \coloneqq (s, \mathsf{sk})
                                                                                                                                13:
13: abort if \exists (i,j) \in [k]^2, i \neq j, m_i^* = m_j^*
                                                                                                                                                 \mathbf{T} := \phi_0(X_C, \mathbf{w}_0)
                                                                                                                                14:
14: abort if \forall j \in [k], \mathsf{H}_{\mathsf{M}}(m_j^*) \neq \overline{m}^* \ /\!\!/ \text{ Game 4}
                                                                                                                                            // Game 7, Game 12
15: abort if \tau^* \neq \tau_{q_{\tau}^*} // \text{Game } 3
                                                                                                                                          c_0 \stackrel{\$}{\leftarrow} \mathbb{Z}_p, (\mathbf{A}_0^C, \mathbf{z}_0) \leftarrow \mathsf{Sim}_0(\mathbb{x}_0^C, c_0)
                                                                                                                                15:
16: parse (S_{1,j}^*, S_{2,j}^*, \pi_j)_{j \in [k]} \leftarrow (\sigma_j^*)_{j \in [k]} / Game 9
                                                                                                                                16: \mathbb{W}_1 := \mathsf{T}_{\mathsf{ddh}}[\tau] \ /\!\!/ \text{ Game 6, Game 11}
            abort if \exists j \in [k], S_{1,j}^* \neq S_j, // Game 9
                                                                                                                                17: (\mathbf{A}_1, \mathsf{st}_1) \leftarrow \mathsf{lnit}_1(\mathbb{x}_1, \mathbb{w}_1) // Game 6, Game 11
17:
                           S_j = uV + (\mathsf{H}_{\mathsf{M}}(m_j^*) \cdot u)S_{2,j}^* + hS_{2,j}^*
                                                                                                                                18: state[sid] \leftarrow (\mathbf{z}_0, c_0, st_1, \overline{m})
18 : return 1
                                                                                                                                19: round[sid] \leftarrow 1
H_{ddh}(\tau)
                                                                                                                                20: return (\mathbf{T}, \mathbf{A}_0^C, \mathbf{A}_1)
 1: \mathsf{ctr}_{\mathsf{ddh}} \leftarrow \mathsf{ctr}_{\mathsf{ddh}} + 1 \ // \text{Game 3}
                                                                                                                                \mathcal{O}_{\mathsf{S}_2}(\mathsf{sid}, c)
 2: if \operatorname{ctr}_{ddh} = q_{\tau}^* then \tau_{q_{\tau}^*} \coloneqq \tau // \operatorname{Game 3}
                                                                                                                                 1: req round[sid] = 1
 3: if Q_{\mathsf{H}_{\mathsf{ddh}}}[\tau] = \bot then
                                                                                                                                  2: req c \in \mathbb{Z}_p
                d_2 \leftarrow \mathbb{Z}_p, \mathsf{T}_{\mathsf{ddh}}[\tau] \leftarrow d_2 \quad /\!\!/ \text{ Game 5, Game 10}
 4:
                                                                                                                                  3:
                                                                                                                                            (\mathbf{z}_0, c_0, \mathsf{st}_1, \overline{m}) \coloneqq \mathsf{state}[\mathsf{sid}]
 5:
           (D_2^{\tau}, D_3^{\tau}) := (d_2 G, d_2 D_1) \ // \text{ Game 5, Game 10}
                                                                                                                                            abort if \overline{m} = \overline{m}^* // Game 4
                 \mathcal{Q}_{\mathsf{H}_{\mathsf{ddh}}}[\tau] \gets (D_2^\tau, D_3^\tau)
                                                                                                                                  4:
 6:
                                                                                                                                  5: c_1 \coloneqq c - c_0 \quad // \text{ Game 7, Game 12}
 7: return \mathcal{Q}_{\mathsf{H}_{\mathsf{ddb}}}[\tau]
                                                                                                                                  6: z_1 \leftarrow \mathsf{Resp}_1(\mathsf{st}_1, c_1) \quad /\!\!/ \text{ Game 6, Game 11}
\mathsf{H}_\mathsf{M}(m)
                                                                                                                                  7: queried[\tau] \leftarrow queried[\tau] + 1
 1: \mathsf{ctr}_\mathsf{M} \leftarrow \mathsf{ctr}_\mathsf{M} + 1 \ // \ \mathrm{Game} \ 4
                                                                                                                                  8: return (\mathbf{z}_0, z_1, c_0)
 2: if Q_{\mathsf{H}_{\mathsf{M}}}[m] = \bot then
 3: if \operatorname{ctr}_{\mathsf{M}} = q_m^* then \overline{m} \coloneqq \overline{m}^* // Game 4
 4:
                 else \overline{m} \stackrel{\hspace{0.1em}\mathsf{\scriptscriptstyle\$}}{\leftarrow} \mathbb{Z}_p
                 abort if \exists m, \mathcal{Q}_{\mathsf{H}_{\mathsf{M}}}[m] = \overline{m} \ /\!\!/ \text{ Game } 1
 5:
                 \mathcal{Q}_{\mathsf{H}_\mathsf{M}}[m] \leftarrow \overline{m}
 6:
 7: return Q_{H_M}[m]
\mathsf{H}_{\Sigma}(x) and \mathsf{H}_{\mathsf{Ped}}(x) and \mathsf{Next}(\mathsf{sid},\tau)
 // Identical to H_{\Sigma} and H_{Ped} and Next in Game 0
```

**Fig. 4.** Description of Game 12, where the differences to Game 0 are highlighted in gray. Note that the changes marked with Game 5, Game 10 (resp. Game 7, Game 12) are introduced for  $\tau \neq \tau_{q_{\tau}^*}$  in Game 5 (resp. Game 7), and later for  $\tau = \tau_{q_{\tau}^*}$  in Game 10 (resp. Game 12). For some condition C, **abort if** C makes the game output 0 if C is true and **req** C makes the oracle output  $\perp$  if C is false.

## F Script for Concrete Efficiency

```
#!/usr/bin/env python
#:/usr/Din/env python
import math
import sys
from tabulate import tabulate
size_ge = 256
size_fe = 256
secpar = 128
#
def size_fischlin_proof(num_fe_vitness, num_ge_statement):
    b = 8 # output bits of the hash function that need to be 0
    # r must satisfly that br = secpar
    r = math.ceil(secpar / b)
                num_coms = r
num_chal = r
               num_chal = r
num_resp = r
size_com = num_ge_statement * size_ge
size_chall = size_fe
size_resp = num_fe_witness * size_fe
size_proof = num_coms * size_com + num_chal * size_chall + num_resp * size_resp
return size_proof
# communication of CTZ-3, according to Table 1 in
# the eprint version https://eprint.iacr.org/2023/1780.pdf
comm_ctz_ge = 3 * secpar + 6
comm_ctz_bits = secpar + 9
comm_ctz_bits = secpar + 3 * secpar * secpar
comm_ctz = comm_ctz_ge * size_ge + comm_ctz_fe * size_fe + comm_ctz_bits
print("Communication CTZ: " + str("{:.2f}".format((comm_ctz/8000))) + " Kilobytes")
# communication for our scheme
comm_ours_wo_proof = 8 * size_ge + 5 * size_fe
comm_ours = comm_ours_wo_proof + size_fischlin_proof(2, 1)
print("Communication Ours (using Fischlin): " + str("{:.2f}".format((comm_ours/8000))) + " Kilobytes")
# # Signature for CTZ and Our Scheme # #

print("")
# signature of CTZ-3, according to Table 1 in
# the eprint version https://eprint.iacr.org/2023/1780.pdf
sig_ctz_ge = secpar + 1
sig_ctz_fe = secpar + 7
sig_ctz_bits = secpar * secpar
sig_ctz = sig_ctz_ge * size_ge + sig_ctz_fe * size_fe + sig_ctz_bits
 print("Signature CTZ: " + str("{:.2f}".format((sig_ctz/8000))) + " Kilobytes")
# signature size for our scheme
sig_ours = 2 * size_ge + 5 * size_fe
print("Signature Ours: " + str("{:.2f}".format((sig_ours/8))) + " Bytes")
```

Listing 1.1. Python script to compute the efficiency metrics in Table 2.