Pairing-Free Blind Signatures from Standard Assumptions in the ROM

Julia Kastner[1], Ky Nguyen[2], and Michael Reichle[1,3]

[1] Department of Computer Science
ETH Zürich, Switzerland
{julia.kastner,michael.reichle}@inf.ethz.ch
[2] DIENS, École normale supérieure, CNRS, Inria, PSL University,
Paris, France
ky.nguyen@ens.psl.eu
[3] Work done partially while at ENS Paris.

Abstract. Blind Signatures are a useful primitive for privacy preserving applications such as electronic payments, e-voting, anonymous credentials, and more. However, existing practical blind signature schemes based on standard assumptions require either pairings or lattices. We present the first construction of a round-optimal blind signature in the random oracle model based on standard assumptions without resorting to pairings or lattices. In particular, our construction is secure under the strong RSA assumption and DDH (in pairing-free groups). For our construction, we provide a NIZK-friendly signature based on strong RSA, and efficiently instantiate Fischlin's generic framework (CRYPTO'06). Our Blind Signature scheme has signatures of size 4.28 KB and communication cost 62.19 KB. On the way, we develop techniques that might be of independent interest. In particular, we provide efficient *relaxed* range-proofs with subversion zero-knowledge and compact commitments to elements of arbitrary groups.

1 Introduction

In privacy-preserving authentication of data, a central question is how to authenticate without compromising one's private information. A *blind signature* solves this question by allowing a *user* to obtain signatures blindly from a *signer* while satisfying strong security guarantees. The property of *blindness* ensures that the signer cannot learn anything about the message when signing and cannot link signatures to the signing sessions of a user. This must hold even when the signer's public key is chosen *maliciously*. On the other hand, the property *one-more unforgeability* imposes that after ℓ completed signing sessions, a user cannot obtain more than ℓ valid signatures (*i.e.*, it cannot forge an additional signature).

Due to the strong security guarantees, blind signatures have applications in e-cash [26, 29, 67], e-voting [28, 45], or anonymous credentials [27, 20], and more. In the past few years blind signatures also play an important role in new applications such as blockchains [79, 22] or private access tokens [55, 49].

Initial constructions. Since their introduction by Chaum [26], many variants of blind signatures were proposed. The first proposed construction—blind RSA [26]—was proven secure under one-more RSA [12]. A similar construction secure under one-more CDH was proposed in [19]. These constructions have great efficiency—a signing interaction requires only two rounds—but require both the random oracle model (ROM) and an interactive security assumption. (These assumptions are non-falsifiable and considered *non-standard*.)

Protocols with three or more rounds. Historically, the main alternative to blind RSA and blind BLS are signatures based on linear identification protocols such as blind Schnorr [71] or similar constructions [66, 5, 51, 59]. These blind signatures are shown to be secure under falsifiable assumptions (*e.g.*, DLOG, RSA) for poly-logarithmically many concurrent signing sessions in the ROM. But for a polynomially large number of concurrent sessions, there are efficient attacks on such protocols [74, 15]. Since, interesting mechanisms that bind an obtained signature to a signing session were proposed, and the resulting schemes are secure in the ROM for an unbounded number

of concurrent sessions [1, 44, 58, 73, 34] under standard assumptions. Unfortunately, the security proof also requires the generic group model (*i.e.*, it is assumed that the adversary interacts with the group in a black-box manner). Alternatively, stateful blind signatures can be obtained for an a-priori bounded number of signatures via a cut-and-choose technique [23, 69, 62]. Generally, these Schnorr-style approaches require more than 2 rounds of interaction, *i.e.*, are not round-optimal.

Round optimality. Round optimality is a desirable efficiency measure as it removes the requirement of storing a state for each signing session and less interaction is required to obtain a signature. Another advantage of round optimal blind signatures is that sequential security implies concurrent security [63, 52]. However, it is difficult to construct round-optimal blind signatures in the plain model under standard assumptions which is supported by several impossibility results [63, 40, 68]. Katsumata *et al.* [60] shows that this is possible under *classical and quantum* standard assumptions. While this result is of theoretical interest, the construction is impractical, due to its reliance on general-purpose cryptographic primitives, namely garbled circuit. More commonly, constructions circumvent such hurdles via a trusted setup [39, 4, 65, 16, 18, 72, 2], idealized models (*e.g.*, generic groups and/or the random oracle model [50, 3, 17, 35, 50, 61]), complexity leveraging [47, 46], or interactive assumptions [10, 64, 7, 43, 42, 48]. All such constructions require pairings or lattices [4] with the exception of blind RSA [26, 12, 64, 7].

But over large networks and complex web applications, existing implementations of pairings (*e.g.*, [76]) seem to remain a significant bottleneck. Another disadvantage of pairing-based constructions is that highly-verified standard cryptographic libraries (for instance BoringSSL and NSS) do not support pairing-friendly curves. Similarly, lattice-based constructions are still in the process of being standardized [75]. On the other hand, plain groups (without pairings) and RSA-based constructions have found widespread use in practice, *e.g.*, in Apple's Proposal for Click Fraud Prevention in Safari [77] or SSH [78]. The only efficient round-optimal blind signature in this setting is blind RSA [26, 12] and its variants [64, 7]. The latter are covered in an RTF draft [36] and blind RSA is still a recommended nowadays [25]. Unfortunately, these schemes require both an interactive assumption (tailored to the scheme itself) and the ROM. This brings us to the following natural question:

> *Can we construct* efficient round optimal *blind signatures in the* ROM*, based on standard non-interactive assumptions without resorting to pairings or lattices?*

1.1 Our Contributions

In this paper we improve the state of the art of blind signatures by answering the above question affirmatively. We construct a *round optimal* blind signature scheme with *competitive efficiency*, whose security is proven in the ROM under standard assumptions in the RSA setting *and* group setting (without pairings) simultaneously. Concretely, our construction is secure under the strong RSA (sRSA) assumption and DDH in ordinary prime-order groups.

Our starting point for our construction is the Fischlin framework [39] and more concretely, the framework proposed in [61] that instantiates Fischlin efficiently via a signature scheme with an *all-but-one* reduction [5]. We instantiate the framework with a variant of the signature proposed in [38] (hereafter denoted by $\mathsf{S}_{\mathsf{fis}}$), and obtain blind signatures with 62.19 KB communication and signatures of size 4.28 KB. We provide a comparison to prior works in Table 1. Notably, our signatures are only 11.15 times larger than blind RSA [26] (which relies on both the ROM and an *interactive* security assumption).

We emphasize that our instantiation is non-trivial and requires several new techniques to achieve round-optimality and malicious blindness (*i.e.*, blindness holds even if the signer's verification key was setup maliciously). Also, since the security proof of $\mathsf{S}_{\mathsf{fis}}$ does not fit the framework as it has no *all-but-one* reduction, the one-more unforgeability proof requires new insights. We refer to the

[4] The framework of Fischlin [39] yields round-optimal blind signatures with trusted setup generically, but efficient instantiations rely either on pairings [17, 3, 61] or lattices [35, 6].

[5] In the context of signatures, an all-but-one reduction allows to puncture the verification key in such a way that all-but-one message m^* can be signed and given a signature on m^*, a hard problem can be solved.

technical overview in Section 1.2 for more details. Along the way, we provide techniques that might be of independent interest, such as:

- Easy-to-use notions for subversion zero-knowledge in the ROM (*i.e.*, zero-knowledge holds even for a maliciously setup crs).
- Efficient relaxed range proofs, *i.e.*, zero-knowledge proofs that prove to a verifier that a given value $x \in [0, B]$ lies in a range $[-TB, BT]$, where $T \in \mathbb{N}$ is the slack, with subversion zero-knowledge.
- Compact commitments to elements of arbitrary groups based on DDH in an independent prime-order group. Our commitments can be opened efficiently in zero-knowledge using our relaxed range proofs.
- A zero-knowledge-friendly variant of the $\mathsf{S}_{\mathsf{fis}}$ signature [38]. Knowledge of a signature on a given message m can be shown efficiently in zero-knowledge using our relaxed range proofs.

Table 1. Round-optimal blind signatures in the ROM under standard assumptions

Reference	Sig. size	Comm. size	Setting	Assumption
del Pino et al. [35]	100 KB	850 KB	Lattices	DSMR, MLWE, MSIS
Blazy et al. [17]	96 B	220 KB †	Pairings	SXDH, CDH
Abe et al. [3]	5.5 KB	1 KB	Pairings	SXDH
Hanzlik et al. [50]‡	5 KB 9 KB	72 KB 36 KB	Pairings	CDH
Katsumata et al. [61]	447 B 96 B	303 B 2.2 KB	Pairings	SXDH DDH, CDH
This work	4.28 KB	62.19 KB	RSA, Groups	sRSA, DDH

We provide signature size, communication size, the algebraic setting, and the underlying assumptions for known blind signatures in the ROM. We do not consider schemes that rely on non-falsifiable assumptions (*e.g.*, interactive assumptions). We stress that our work relies on assumptions in prime-order groups *without* pairing.
(†): Communication of [17] scales linearly with the message size, and is given here for 256 bit messages. (‡): [50] offers tradeoffs between signature and communication sizes.

1.2 Technical Overview

We provide an overview of our construction. Since our blind signature builds on the framework proposed in [61], we give a brief recap.

The framework. The framework of [61] is based on an additively homomorphic commitment scheme $\overline{\mathsf{Com}}$ (*i.e.*, $\overline{\mathsf{Com}}(m;r)+\overline{\mathsf{Com}}(m';r') = \overline{\mathsf{Com}}(m+m';r+r')$) with a *compatible* signature scheme. That is, the signing algorithm $\mathsf{Sign}(\mathsf{sk}, m)$ can be rewritten as $\widehat{\mathsf{Sig}}(\mathsf{sk}, \overline{\mathsf{Com}}(m;r)) - \overline{\mathsf{Com}}(0;r)$. Namely, $\widehat{\mathsf{Sig}}$ first commits to the message m with randomness r, proceeds with signing and then removes the randomness r homomorphically. To turn this into a blind signature, a user can generate a valid commitment $c = \overline{\mathsf{Com}}(m;r)$, send it to the signer, and the signer can simply return $\sigma_r \leftarrow \widehat{\mathsf{Sig}}(\mathsf{sk}, c)$. Then, the user obtains a valid signature $\sigma \leftarrow \sigma_r - \overline{\mathsf{Com}}(0;r)$. This approach hides the message m during signing, and if the scheme is rerandomizable, then a user can produce a fresh signature σ' on the message m to ensure blindness. For one-more unforgeability (OMUF), the proof relies on the all-but-one reduction of the signature scheme. Recall that an all-but-one reduction allows to setup a punctured verification key vk^* along with a trapdoor td in such a way that all-but-one message m^* can be signed via an algorithm $\sigma = \widehat{\mathsf{SimSig}}(\mathsf{td}, m)$ for $m \neq m^*$. Importantly, given a signature σ^* on m^* that verifies with respect to vk^*, a hard problem can be solved. Also, the user needs to send a zero-knowledge proof π along with c that proves knowledge of (m, r) such that $c = \overline{\mathsf{Com}}(m;r)$ via an online-extractable NIZK [6]. With this in mind, the OMUF reduction

[6] This is a non-interactive zero-knowledge proof (NIZK) which allows to extract in an on-the-fly manner by embedding a trapdoor in the crs, *i.e.*, without rewinding.

simply punctures the verification key vk^* for some message m^* with trapdoor td. Then, to sign a commitment c, it extracts (m, r) from π on-the-fly, and generates $\sigma_r = \widehat{\mathsf{SimSig}}(\mathsf{td}, m) + \mathsf{Com}(0; r)$ via the trapdoor. If the messages are hashed via a random oracle, then the reduction can guess m^* initially among the oracle outputs and with sufficient probability, the adversary provides a signature on m^* among the $\ell + 1$ forgeries that it outputs. This allows to solve a hard problem. Katsumata *et al.* [61] instantiates this framework with Boneh-Boyen signatures and Pedersen commitments in the pairing setting.

A compatible RSA-based commitment. A natural approach to construct pairing-free blind signatures is to identify a signature scheme in the RSA setting which suffices these requirements. Unfortunately, there are none. To the best of our knowledge, all RSA-based signatures are not rerandomizable or have no all-but-one reduction. For the start, we choose a signature scheme that is *compatible* with Pedersen commitments over $\mathbb{Z}_N$. A natural scheme for this is the signature scheme $\mathsf{S}_{\mathsf{fis}}$ [38] based on Cramer-Shoup signatures [33]. Here, the verification key $\mathsf{vk} = (N, h, h_1, h_2)$ consists of an RSA modulus N and three QR_N generators h, h_1, h_2. As usual, the secret key is the factorization of N. To sign a message $m \in [0, 2^{2\lambda}]$, the signer chooses a $2\lambda + 1$ bit prime e and a 2λ bit value a, and computes y such that

$$y^e \equiv h \cdot h_1^a \cdot h_2^{a \oplus m} \bmod N \tag{1}$$

using its secret key. A valid signature is a tuple (e, a, y) that satisfies Equation (1) and with $a \in [0, 2^{2\lambda} - 1]$ and odd $e \in [2^{2\lambda}, 2^{2\lambda+1}]$. While the scheme is not quite compatible with Pedersen commitments, we observe that a functions as a mask for m in the security proof. Thus, instead of masking m via $\oplus$, we mask additively via noise flooding, and modify Equation (1) as follows:

$$y^e \equiv h \cdot h_1^a \cdot h_2^{a+m} \bmod N, \tag{2}$$

where $a \in [0, 2^{3\lambda} - 1]$ and $e \in [2^{3\lambda}, 2^{3\lambda+1}]$. Since in Equation (2), m is masked statistically by a the security proof can be adapted in a straightforward manner. Note that we still require that e is prime during signature generation and odd during verification.

Turning it into a blind signature. Let g be another QR_N generator. Then, we can sign a commitment $c \equiv h_2^m \cdot g^r \bmod N$ for some random r by first choosing an appropriate e and a, then computing y such that

$$y_r^e \equiv h \cdot h_1^a \cdot h_2^a \cdot c \bmod N. \tag{3}$$

But now, the user cannot derive a valid signature from y, since it requires computing $y \equiv y_r \cdot g^{-r/e \bmod \Phi(N)} \bmod N$. But taking e-th roots is assumed to be hard in the first place! To fix this, we can let the signer send e first, and let the user commit via $c \equiv h_2^m \cdot g^{e \cdot r} \bmod N$. Then, it is easy to compute $y \equiv y_r \cdot g^{-r} \bmod N$, where y_r is generated as in Equation (3) as before. Then, as in [61], the user proves with a proof π_{ped} that she committed to $m \in [0, 2^{2\lambda} - 1]$ with randomness $e \cdot r$ to the signer via an online-extractable NIZK Π_{ped}. Since the $\mathsf{S}_{\mathsf{fis}}$ signatures are not rerandomizable, to present a signature, the user generates a proof π_{fis} via an additional NIZK Π_{fis} that proves that it knows a $\mathsf{S}_{\mathsf{fis}}$ signature on message m (instead of presenting (e, a, y) directly).

Making it round optimal. Unfortunately, the above construction requires an additional round of communication. Note that the user cannot generate e itself, as then the security proof of $\mathsf{S}_{\mathsf{fis}}$ fails. Indeed, it is required that a fresh e and a is picked for each fresh message m to be signed. A natural idea is to let the user generate it via a hash function mapping $\mathsf{H}_{\mathbb{P}}$ into $2\lambda + 1$ bit primes. But it is unclear how to derive e via $\mathsf{H}_{\mathbb{P}}$. We cannot set $e \leftarrow \mathsf{H}_{\mathbb{P}}(c)$ since the user needs e to setup c in the first place. We also cannot derive $e \leftarrow \mathsf{H}_{\mathbb{P}}(m)$ itself, as m is supposed to be hidden from the signer, but the signer needs to derive e to sign c.

Our idea is to derive e based on an integer commitment $c_Z \leftarrow \mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(m, r; r_z)$ to (m, r) instead. Since c_Z fixes c implicitly, this ensures that for each fresh commitment c, we use a fresh e. Under binding of C_{Z}, this implies that for each distinct message m, a fresh e is picked as desired. For technical reasons, we also need that if e is reused, *e.g.*, if the same commitment is sent twice, the signer reuses the same mask a. This can be guaranteed by deriving a from a pseudorandom function PRF via $a \leftarrow \mathsf{PRF}(c \| c_Z)$. Also, we let the user hash the message m first, *i.e.*, commit to $\overline{m} \leftarrow \mathsf{H}(m)$ in c. Then, the blind signature works for arbitrary messages $m \in \{0, 1\}^*$.

In summary, the user hashes the message m via $\overline{m} \leftarrow \mathsf{H}(m)$, commits to $(\overline{m}, r)$ in c_Z, computes $e \leftarrow \mathsf{H}_{\mathsf{pp}}(c_Z)$ and sets $c \equiv h_2^{\overline{m}} \cdot g^{r \cdot e} \bmod N$. Then, the user proves in π_{ped} generated via a NIZK Π_{ped} that the commitment c is constructed based on the values committed in c_Z, and sends $(c, c_Z, \pi_{\mathsf{ped}})$ to the signer. The signer verifies π_{ped}, sets $e \leftarrow \mathsf{H}_{\mathsf{pp}}(c_Z)$ and $a \leftarrow \mathsf{PRF}(c \| c_Z)$, then computes y_r as in Equation (3). Finally, the user sets $y \leftarrow y_r \cdot g^{-r}$ and obtains a valid $\mathsf{S}_{\mathsf{fis}}$ signature (e, a, y) for $\overline{m}$. The blind signature is π_{fis} generated via Π_{fis}.

Proving one-more unforgeability. While the unforgeability reduction of $\mathsf{S}_{\mathsf{fis}}$ has no one-more flavor, we can show one-more unforgeability for our blind signature with the above modifications if the NIZK Π_{fis} is adaptively knowledge sound [7].

For this, we first sketch how the unforgeability proof of $\mathsf{S}_{\mathsf{fis}}$ works. The reduction first punctures the verification key vk^*. Roughly, this is done by generating all primes $\mathcal{E} = \{e_1, \cdots, e_Q\}$ chosen during signing in advance, and setting up h, h_1, h_2 with respect to $\mathcal{E}$. There are two cases for the punctured setup [8]:

1. The reduction guesses that the forgery's e was used during signing, *i.e.*, $e \in \mathcal{E}$.
2. The reduction guesses that the forgery contains a fresh e, *i.e.*, $e \notin \mathcal{E}$.

Then, the reduction sets up h, h_1, h_2 in such a way that it can sign Q arbitrary messages via a trapdoor td but without knowing the factorization of N. This is done by embedding $\mathcal{E}$ into h, h_1, h_2 depending on the guess. Note the punctured setup is indistinguishable from the real setup in both cases. Also, signing via the trapdoor td reveals no information about the guess. Then, it answers all Q signing queries via td and hopes that its guess was correct. If so, the reduction can derive a sRSA solution. Since the guess remains hidden, this happens with sufficient probability.

In the proof of one-more unforgeability we apply the same technique, *i.e.*, we generate all outputs $\mathcal{E}$ for $\mathsf{H}_{\mathbb{P}}$ in advance. Then, we guess that $e \in \mathcal{E}$ or $e \notin \mathcal{E}$, and puncture vk^* with trapdoor td accordingly. To sign a commitment c, the signer extracts $(\overline{m}, r)$ on-the-fly from the proof π_{ped}, then uses the td to sign $\overline{m}$ and blinds the obtained y with g^r to generate $y_r \leftarrow y \cdot g^r$. When the adversary outputs its forgeries, we look for a signature π_{fis} on a message $\overline{m}$ that we never signed [9], and then extract a valid signature (e, a, y) from π_{fis}.

One would assume that now the adversary's output e confirms our guess with sufficient probability as before, but there is a subtlety. While the punctured signing algorithm with td reveals no information about our guess during a single run, this information might be revealed during a second run with the same setup. This happens, *e.g.*, if Π_{fis} is rewinding-based and the adversary asks signing queries in a specific manner during rewinding. Thus, the extracted signature might depend on our guess now, and we might not be able to reduce to sRSA anymore.

We resolve this by asking that the user commits to (e, a) in c_I with a perfectly binding commitment $\mathsf{C}_{\mathsf{RInt}}$ [10]. Then, the extracted values (e, a) are fixed during the initial run, and at that point our guess is hidden. Even if our guess is revealed during extraction, the extractor still succeeds in finding a valid signature with fixed (e, a). With this modification, we can conclude that we guess correctly with sufficient probability which allows to solve sRSA.

Making it maliciously blind. An observant reader might realize that our scheme is *not* blind yet. Concretely, there are two types of problems: (1) Pedersen commitments over $\mathbb{Z}_N^*$ are not hiding for malicious modulus N and $\langle h_2 \rangle \neq \langle g \rangle$. To illustrate this, observe that to distinguish whether m_0 or m_1 is signed, the blindness adversary could raise c to the power of $\mathsf{ord}(g)$ to remove the part g^{re} and then check whether $c^{\mathsf{ord}(g)} = (h_2^{\overline{m}_0})^{\mathsf{ord}(g)}$ or $c^{\mathsf{ord}(g)} = (h_2^{\overline{m}_1})^{\mathsf{ord}(g)}$, where $\overline{m}_b = \mathsf{H}(m_b)$. If $\overline{m}_1 \cdot \mathsf{ord}(g) \neq \overline{m}_0 \cdot \mathsf{ord}(g) \bmod \mathsf{ord}(h_2)$, then this leads to an efficient attack. (2) Also, since we wish to avoid a trusted setup, we let the signer include the common reference string crs into its verification key vk. But in that case, zero-knowledge is not sufficient as its privacy guarantee holds only if crs is setup in a trusted manner.

Fortunately, the fixes are rather straightforward. For the first problem, we simply let the signer prove that $\langle h_2 \rangle = \langle g \rangle$ via a NIZK with subversion soundness [11] (*i.e.*, soundness holds even

[7] That is, there is an extractor that can extract a witness via black-box access to the prover, *e.g.*, via rewinding.

[8] The first case has two additional sub cases, but for the sake of exposition we simplify the proof sketch.

[9] Since we sign at most ℓ messages but there are $\ell + 1$ forgeries on distinct messages, such a proof exists.

[10] The case we omitted also depends on which a is part of the forgery, so we also fix a.

for malicious crs). Then, the Pedersen commitment is statistically hiding if r is sampled from a sufficiently large interval. A similar but more subtle problem appears in our instantiation. Here, we need that $y \in \langle y_r \rangle$ and thus, we let the signer prove this statement, again via an appropriate subversion sound NIZK.

For the second problem, we simply require that the NIZKs Π_{ped} and Π_{fis} are subversion zero-knowledge, *i.e.*, zero-knowledge holds even for a malicious setup [11]. Yet, there remains one problem since this notion is difficult to instantiate in our setting. To the best of our knowledge, all instantiations of subversion zero-knowledge NIZKs [11, 41] require strong knowledge assumptions (which we wish to avoid). Instead, we give a simplified definition which yields the similar guarantees in the ROM. Importantly, our notion can be instantiated under standard assumption (*e.g.*, DDH in pairing-free groups). Roughly, we split the $\mathsf{crs} = (\mathsf{urs}, \mathsf{srs})$ into a uniform part $\mathsf{urs} \in \{0,1\}^\ell$ of length ℓ and structured part $\mathsf{srs} \in \mathcal{SRS}$. In our notion, we ask that (a) membership in $\mathcal{SRS}$ is testable efficiently and (b) zero-knowledge holds with respect to $\mathsf{crs} = (\mathsf{urs}, \mathsf{srs})$ for some random $\mathsf{urs} \leftarrow \{0,1\}^\ell$ and *any* malicious $\mathsf{srs} \in \mathcal{SRS}$. For this notion, we can instantiate efficient NIZKs for our relations with subversion zero-knowledge. Roughly, we embed the structure that guarantees zero-knowledge into the uniform urs and make sure that for any $\mathsf{srs} \in \mathcal{SRS}$, zero-knowledge remains intact (*e.g.*, via zero-knowledge proofs). Our notion is sufficient to instantiate our blind signature framework. Here, the uniform part urs is setup via a random oracle (as is common for NIZKs with uniform crs) and the signer can choose an arbitrary structured part srs as part of its verification key.

Given these modifications, we can show blindness. Essentially, observe that with the above adaptions, the commitments c_Z and c leak no information about $\overline{m}$. Also, a blind signature $(\pi_{\mathsf{fis}}, c_I)$ is distributed independently from the signing interaction, since c_I reveals no information about (e, a) from the interaction and π_{fis} leaks nothing beyond the fact that the user knows a valid $\mathsf{S}_{\mathsf{fis}}$ signature on $\overline{m} = \mathsf{H}(m)$.

Lastly, we note that sampling urs via a random oracle is a well-known technique to obtain zero-knowledge with a trustless setup. In our case, we also require a structured part srs for our instantiations (*e.g.*, to argue that relations hold over the integers in our NIZKs.) Below, we give more details.

Instantiation. There are several challenges when instantiating the NIZKs required for our blind signature. While it is somewhat straightforward to obtain an instantiation with generic techniques, our goal is to keep the instantiation as efficient as possible. These instantiations form a core technical contribution of this work. We give a brief overview of the challenges and our solutions.

Online-extraction and integer commitments. Recall that we require an integer commitment scheme C_{Z} to commit to $(\overline{m}, r) \in \mathbb{Z}^2$ in combination with an efficient online-extractable NIZK for the statement

$$c_Z = \mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(\overline{m}, r; r_z) \wedge c \equiv h_2^{\overline{m}} \cdot g^{r \cdot e} \bmod N. \tag{4}$$

For online-extraction, we use the approach of [61] (cf. Section 6). Let $\mathbb{G}$ be a pairing-free group of prime order p with generators G, H. The values $(\overline{m}, r)$ are decomposed into $(e_i)_i = ((m_i)_i, (r_i)_i)$ via B-ary decomposition (*e.g.*, $B = 2^{64}$), committed in ElGamal commitments $E_i = e_i G + s_i H$ for $s \leftarrow \mathbb{Z}_p$, and a range proof proves that $e_i \in [0, B-1]$ (*e.g.*, Bulletproofs [21]). We then interpret $(E_i)_i$ as the integer commitment. This approach suffices for online-extraction of e_i from which $(\overline{m}, r)$ can be reconstructed, but we also need to extract the randomness $s_i \in \mathbb{Z}_p$ to reduce to binding. But since the commitments are perfectly binding, we avoid this by relaxing online-extraction: we ask that an extractor can extract $(\overline{m}, r)$ such that there exists an opening $r_Z = (s_i)_i$ for which Equation (4) holds. A more subtle problem is that the public parameters G, H are now part of the statement but online-extraction requires that we embed a trapdoor into G, H. Thus, we further modify the notion to enable embedding the trapdoor into the parts of the statement, so long it is picked uniformly. With this notion, we can prove one-more unforgeability and avoid the considerable overhead of online-extracting r_Z. We also need a structured random string srs to ensure that certain relations hold over the integers, but we refer to Appendix E for more details.

Proof for $\mathsf{S}_{\mathsf{fis}}$ signatures. To derive a blind signature, we need a perfectly binding commitment $\mathsf{C}_{\mathsf{RInt}}$ and a NIZK Π_{fis} for the relation in Equation (2) with $c_I = \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(e, a, r_I)$ and $a \in [0, 2^{3\lambda} - 1]$, $e \in [2^{3\lambda}, 2^{3\lambda+1}] \cap \mathbb{Z}_{\mathsf{odd}}$. While it is fine to employ range proofs during the (one-time) signing

interaction, it is somewhat undesirable to include a range proof for presenting the signature (as the overhead is noticeable for such large ranges).

Instead, we relax the range requirements in such a way that the unforgeability proof of $\mathsf{S}_{\mathsf{fis}}$ still goes through, *i.e.*, we allow that a and e lie in larger (but distinct) intervals for verification. Then, we construct very efficient *relaxed* range proofs with subversion zero-knowledge for $\mathsf{C}_{\mathsf{RInt}}$ consisting of ElGamal commitments over a prime-order $\mathbb{G}$ (for perfect binding). Roughly, the range proof is a simple Σ-protocol to open ElGamal in zero-knowledge, where we also add range checks for the messages sent in third flow, compiled with Fiat-Shamir. In addition, we add a fresh RSA modulus $\tilde{N}$ to the crs and commit to a and e in a commitment over $\mathbb{Z}^*_{\tilde{N}}$ (similar to [30]). This technique guarantees that extracted values are short integers (but within a larger range). The overhead over simply opening the ElGamal commitment in zero-knowledge—which we need anyway to instantiate the NIZK—is just 784 Byte for a modulus of size 3072. For comparison, a Bulletproof for the above ranges requires 932 Byte [21]. Our relaxed range proofs are smaller and allow seamless integration into more complex Σ-protocols.

To construct Π_{fis}, we combine our relaxed range proofs for $\mathsf{C}_{\mathsf{RInt}}$ with standard commit-then-prove Σ-protocol techniques to show the remaining equations. For this, we require commitments over $\mathbb{Z}^*_N$ for potentially malicious N to commit to y. Using the above techniques, we construct such commitments and provide efficient openings in zero-knowledge. Roughly, such a commitment is of the form $y \cdot g^s$ for $s \in [N \cdot 2^\lambda]$ with $y \in \langle g \rangle$, in conjunction with a $\mathsf{C}_{\mathsf{RInt}}$ commitment to fix s over the integers. Especially for this purpose our relaxed range proofs shine, since s lies in a large interval. (For such ranges, *e.g.*, Bulletproofs requires 1.6 seconds for proof generation and almost 5 ms for verification.) We generalize the construction for arbitrary untrusted groups.

The remaining NIZKs are straightforward to instantiate. In total, we obtain blind signatures with 62.19 KB communication of size 4.28 KB. We remark that the bulk of communication is required for the NIZK to show $y \in \langle y_r \rangle$ (*i.e.*, roughly 50 KB) as we require subversion soundness. A more efficient NIZK for this statement would largely improve communication.

1.3 Concurrent Works

There is an independent and concurrent work that also constructs a pairing-free blind signature in the ROM from standard assumptions [24]. We give a brief comparison. [24] presents three blind signatures: two constructions BS_1 and BS_2 based on an interactive assumption, and BS_3 based on a non-interactive assumption in the ROM. When instantiating BS_3 with a group of order 256 bit, it has communication and signature size of roughly 26 KB and 10 KB, respectively. Compared to our construction, their signature size is roughly 2.3 times larger than ours, whereas their communication size is smaller by roughly the same factor. Their construction relies on weaker assumptions, namely CDH, but has 4 rounds of interaction. Our construction is round-optimal, but relies on DDH and sRSA. We believe it to be non-trivial to reduce the number of rounds in the protocol of BS_3 as it relies on a Schnorr-style proof of knowledge that is interactively computed.

2 Preliminaries

Notations. We denote the security parameter by λ. A probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ runs in time polynomial in the (implicit) security parameter λ. We write $\mathsf{Time}(\mathcal{A})$ for the runtime of $\mathcal{A}$. A function $f(\lambda)$ is *negligible* in λ if it is $\mathcal{O}(\lambda^{-c})$ for every $c \in \mathbb{N}$. We write $f = \mathrm{negl}(\lambda)$ for short. Similarly, we write $f = \mathsf{poly}(\lambda)$ if $f(\lambda)$ is a polynomial with variable λ. If D is a probability distribution, $x \leftarrow D$ means that x is sampled from D and if S is a set, $x \leftarrow S$ means that x is sampled uniformly and independently at random from S. We also write $|S|$ for the cardinality of set S. Further, we write $D_0 \overset{c}{\approx} D_1$ for distributions D_0, D_1, if for all PPT adversaries $\mathcal{A}$, we have $|\Pr[x_0 \leftarrow D_0 : \mathcal{A}(1^\lambda, x_0) = 1] - \Pr[x_1 \leftarrow D_1 : \mathcal{A}(1^\lambda, x_1) = 1]| = \mathrm{negl}(\lambda)$. Similarly, we write $D_0 \overset{s}{\approx} D_1$ if the above holds even for unbounded adversaries. For some PPT algorithm $\mathcal{A}$, we write $\mathcal{A}^{\mathcal{O}}$ if $\mathcal{A}$ has oracle access to the oracle $\mathcal{O}$. If $\mathcal{A}$ performs some check, and the check fails, we assume that $\mathcal{A}$ outputs $\perp$ immediately. Generally, we assume that adversaries are implicitly stateful. We denote with $[n]$ the set $\{1, \ldots, n\}$ for $n \in \mathbb{N}$. We write $\mathbb{P}$ for the set of primes and $\mathbb{P}_I$ for the set of primes in the interval I. For some odd prime p, we use the representatives $\{-\frac{p-1}{2}, \cdots, \frac{p-1}{2}\}$ for

$\mathbb{Z}_p$. For a group $\mathbb{G}$ we write $\mathsf{ord}(\mathbb{G})$ to denote the order of $\mathbb{G}$ and unless stated otherwise we write $\mathbb{G}$ with additive notation. For a group element g we write $\mathsf{ord}(g)$ to denote the order of the group element. We denote by $\mathsf{QR}_N = \{a \in \mathbb{Z}_N^* : \exists\, b \in \mathbb{Z}_N^*, b^2 \equiv a \bmod N\}$ the quadratic residues $\bmod N$. For some $N \in \mathbb{N}$, the group QR_N is a cyclic subgroup of $\mathbb{Z}_N^*$ and we denote by $\mathsf{Gen}(\mathsf{QR}_N)$ the set of generators of QR_N. Some properties of QR_N are recalled in lemma 2 in Appendix A

Probability Let $V, L \in \mathbb{N}$. We define *uniform rejection sampling* for the interval $[0, V]$ with masking overhead L as in [30]. Let $v \in [0, V]$. To mask v additively with a mask μ via rejection sampling, perform the following steps.

1. Draw a random mask $\mu \leftarrow [0, (V+1)L]$.
2. Abort if $v + \mu \notin [V, (V+1)L]$.
3. Output $w = v + \mu$.

The value w is uniform over $[V, (V+1)L]$ conditioned on no abort and the abort probability is at most $1/L$. We use a version of Forking Lemma that fits our usage of it. The lemma was first introduced by Pointcheval and Stern [70] then generalized by Bellare and Neven [13]. The formal statement can be found in Appendix A.2.

Hardness Asssumptions. We use the following assumptions in this paper. Let GenG be a PPT algorithm that on input 1^λ and prime order p, outputs (a description of) a group $\mathbb{G} \leftarrow \mathsf{GenG}(1^\lambda)$ of order p. We generally use additive notations for prime order groups and capital letters for elements. Also, we assume that given the description, group operations and membership tests are efficient. We write $g \leftarrow \mathbb{G}$ for drawing elements from some group $\mathbb{G}$ at random. In the following, we assume that prime order groups are setup with GenG implicitly.

Let GenRSA be a PPT algorithm that on input 1^λ outputs $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$ such that $N = P \cdot Q$ with $P, Q \in \mathbb{P}$, where $P = 2P' + 1$ and $Q = 2Q' + 1$ are strong primes (*i.e.*, P', Q' are also primes). We assume that $P', Q' > 2^\lambda$.

First of all, the (D, ℓ)-*relaxed DLOG assumption with regards to* $\vec{g}$, where $\vec{g} = (g_0, \ldots, g_\ell) \in \mathsf{QR}_N^{\ell+1}$, assumes that for any PPT adversary, given $(N, g_0, \ldots, g_\ell)$ it is only with negligible probability to output $(c, d, x_0, \ldots, x_\ell)$ satisfying $c^d = \prod_{i=0}^{\ell} g_i^{x_i} \wedge \exists i \colon \frac{x_i}{d} \notin \mathbb{Z} \wedge d \in [0, D] \wedge x_i \in \mathbb{Z}$. The (D, ℓ)-relaxed DLOG assumption holds under the strong RSA assumption for all $D \leq 2^{\lambda+1}$. Next, the *Decisional Diffie-Hellman* (DDH) assumption in a cyclic group $\mathbb{G}$ assumes that for all PPT adversary it is only with negligible probability that the adversary can distinguish (aG, bG, abG) from (aG, bG, cG) where $G \leftarrow \mathbb{G}$ and $a, b, c \leftarrow \mathsf{ord}(\mathbb{G})$. Finally, the *strong* RSA (sRSA) assumes that it is only with negligible probability for any PPT adversary to output (e, z) such that $z^e \equiv y \bmod N$.

Explaining Random Group Elements as Random Strings. For our framework, we require commitments with uniform public parameters pp. For readability, we allow pp (and also uniform random strings urs of NIZKs) to contain (uniform) group elements g of prime-order groups $\mathbb{G}$ with known order p. This is without loss of generality because with *explainable sampling*, we can explain $g \leftarrow \mathbb{G}$ as a random bitstring. We refer to, *e.g.*, [61, Appendix B] for more details.

2.1 Cryptographic Primitives

Commitment Scheme. A *commitment scheme* is a PPT algorithm $\mathsf{C} = \mathsf{C.Commit}$ such that

- $\mathsf{C.Setup}(1^\lambda)$: generates the public parameters pp,
- $\mathsf{C.Commit}(\mathsf{pp}, m)$: given the public parameters pp, message $m \in \mathcal{C}_{\mathsf{msg}}$, computes a commitment $c \in \mathcal{C}_{\mathsf{com}}$ with opening randomness d, and outputs the pair (c, d),
- $\mathsf{Verify}(\mathsf{pp}, c, m, d)$: given the public parameters pp, message $m \in \mathcal{C}_{\mathsf{msg}}$, and opening randomness d, outputs a bit $b \in \{0, 1\}$ which depends on the validity of the opening (m, d) with respect to the commitment c.

Here, $\mathcal{C}_{\mathsf{msg}}$, $\mathcal{C}_{\mathsf{rnd}}$, $\mathcal{C}_{\mathsf{com}}$, are message, randomness, and commitment spaces, respectively. If the public parameters are uniform or explainable (*i.e.*, Setup outputs some $\mathsf{pp} \leftarrow \{0,1\}^\ell$ for $\ell \in \mathbb{N}$) we omit Setup without loss of generality.

We require the correctness, hiding and binding properties for a commitment scheme. A commitment schemes is *correct*, if honest commitments $(c, d) \leftarrow \mathsf{Commit}(\mathsf{pp}, m; r)$ always verify, *i.e.* it holds that $\mathsf{Verify}(\mathsf{pp}, c, m, d) = 1$ where pp is the public parameters. It is *hiding* if it is hard to decide whether an unopened commitment c commits to message m_0 or m_1, and it is *binding* if it is hard to open commitments c to distinct messages. We can have *computational, statistical, perfect* variants for hiding and biding properties. The formal definitions can be found in Appendix A.5.

Signature Scheme. A signature scheme is a tuple of PPT algorithms $\mathsf{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ such that

- $\mathsf{KeyGen}(1^\lambda)$: generates a verification key vk and a signing key sk,
- $\mathsf{Sign}(\mathsf{sk}, m)$: given a signing key sk and a message $m \in \mathcal{S}_{\mathsf{msg}}$, *deterministically* outputs a signature σ,
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$: given a verification key pk and a signature σ on message m, *deterministically* outputs a bit $b \in \{0, 1\}$.

Here, $\mathcal{S}_{\mathsf{msg}}$ is the message space. We define the standard notion of *correctness* and $\mathtt{euf\text{-}cma}$ security. Correctness requires that any honestly generated signature $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$ verifies, *i.e.* $\mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1$. The $\mathtt{euf\text{-}cma}$ security imposes that even with oracle accesses to $\mathsf{Sign}(\mathsf{sk}, \cdot)$, no PPT adversary will be able to forge a valid signature σ on a message m that is not queried to $\mathsf{Sign}(\mathsf{sk}, \cdot)$.

Blind Signature Scheme. A blind signature scheme is a tuple of PPT algorithms $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ such that

- $\mathsf{KG}(1^\lambda)$: generates the verification key bvk and signing key bsk,
- $\mathsf{User}(\mathsf{bvk}, m)$: given verification key bvk and message $m \in \mathcal{BS}_{msg}$, outputs a first message ρ_1 and a state st,
- $\mathsf{Signer}(\mathsf{bsk}, \rho_1)$: given signing key bsk and first message ρ_1, outputs a second message ρ_2,
- $\mathsf{Derive}(\mathsf{st}, \rho_2)$: given state st and second message ρ_2, outputs a signature σ,
- $\mathsf{Verify}(\mathsf{bvk}, m, \sigma)$: given verification key bvk and signature σ on message $m \in \mathcal{BS}_{msg}$, outputs a bit $b \in \{0, 1\}$.

Here, $\mathcal{BS}_{msg}$ is the message spaces. We consider the standard security notions for blind signatures [57]. Below, we define correctness, blindness under *malicious keys*, and one-more unforgeability of a blind signature scheme. Moreover, we assume the state is kept implicit in the following for better readability. A blind signature BS is *correct* if for all $m \in \mathcal{BS}_{msg}$, $(\mathsf{bvk}, \mathsf{bsk}) \leftarrow \mathsf{KG}(1^\lambda)$, $(\rho_1, \mathsf{st}) \leftarrow \mathsf{User}(\mathsf{bvk}, m)$, $\rho_2 \leftarrow \mathsf{Signer}(\mathsf{bsk}, \rho_1)$, and $\sigma \leftarrow \mathsf{Derive}(\mathsf{st}, \rho_2)$, it holds that $\mathsf{Verify}(\mathsf{bvk}, m, \sigma) = 1$.

It is blind under malicious keys if a malicious signer cannot distinguish whether it first signed m_0 or m_1, after engaging with a honest user in two signing sessions and being presented the obtained signatures on messages (m_0, m_1) in a fixed order. Here, the honest user permutes the order of the signing sessions at random, and the verification key bvk is adversarially chosen. It is one-more unforgeable if a malicious user that engages in at most Q_S signing sessions with the signer, can output at most Q_S valid distinct signature- message pairs. The formal definitions can be found in Appendix A.7.

Σ-Protocol. Let R be an NP relation with statements x and witnesses w. We denote by $\mathscr{L}_{\mathsf{R}} = \{x \mid \exists w \text{ s.t. } (x, w) \in \mathsf{R}\}$ the language induced by R. A Σ-protocol for an NP relation R for language $\mathscr{L}_{\mathsf{R}}$ is a tuple of PPT algorithms $\Sigma = (\mathsf{Init}, \mathsf{Chall}, \mathsf{Resp}, \mathsf{Verify})$ such that

- $\mathsf{Init}(x, w)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, and a witness w such that $(x, w) \in \mathsf{R}$, outputs a first flow message (*i.e.*, commitment) α and a state st, where we assume st includes x, w,
- $\mathsf{Chall}()$: samples a challenge $\beta \leftarrow \mathcal{CH}$ (without taking any input),
- $\mathsf{Resp}(\mathsf{st}, \beta)$: given a state st and a challenge $\beta \in \mathcal{CH}$, outputs a third flow message (*i.e.*, response) γ,
- $\mathsf{Verify}(x, \alpha, \beta, \gamma)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, a commitment α, a challenge $\beta \in \mathcal{CH}$, and a response γ, outputs a bit $b \in \{0, 1\}$.

We recall the standard notions of *correctness, high-min entropy, honest- verifier zero-knowledge*, and 2-*special soundness*. A Σ-protocol is correct, if for all $(x, w) \in \mathsf{R}$, if for any honestly generated transcripts (α, β, γ), the verifier accepts, *i.e.* $\mathsf{Verify}(x, \alpha, \beta, \gamma) = 1$. It has high min-entropy if for all $(x, w) \in \mathsf{R}$, it is *statistically* hard to predict a honestly generated first flow α. It is honest-verifier zero-knowledge (HVZK), if there exists a PPT zero-knowledge simulator Sim such that the distributions of $\mathsf{Sim}(x, \beta)$ and the honestly generated transcript with Init initialized with (x, w) are computationally indistinguishable for any $x \in \mathscr{L}_{\mathsf{R}}$, and $\beta \in \mathcal{CH}$, where the honest execution is conditioned on β being used as the challenge. Finally, it is 2-special sound, if there exists a *deterministic* PT extractor Ext such that given 2 valid transcripts $\{(\alpha, \beta_i, \gamma_i)\}_{i\in[2]}$ for statement x with pairwise distinct challenges $(\beta_i)_i$, outputs a witness w such that $(x, w) \in \mathsf{R}$.

Non-Interactive Zero Knowledge All formal definitions of the following can be found in Appendix A.9. Let $\mathcal{URS} = \{0,1\}^\ell$ be a set of *uniform random strings* for some $\ell \in \mathbb{N}$ and $\mathcal{SRS}$ be some set of *structured random strings* with efficient membership test [11]. An NIZK for a relation R with common reference string space $\mathcal{CRS} = \mathcal{SRS} \times \mathcal{URS}$ is a tuple of PPT algorithms $(\mathsf{GenCRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$, where the latter two are oracle-calling, such that:

- $\mathsf{GenCRS}(1^\lambda)$: outputs a structured reference string $\mathsf{srs} \in \mathcal{SRS}$,
- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$: receives a $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs}) \in \mathcal{CRS}$, a statement x and a witness w, and outputs a proof π,
- $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi)$: receives a $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs}) \in \mathcal{CRS}$, a statement x and a proof π, and outputs a bit $b \in \{0, 1\}$.

We recall that $\mathscr{L}_{\mathsf{R}} = \{x \mid \exists w \ : \ (x, w) \in \mathsf{R}\}$ denotes the language induced by R. If there is no crs needed, *i.e.* $\mathcal{CRS} = \varnothing$, we then omit crs as an input to Prove and Verify. A NIZK is *correct* if for any $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ with $\mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda)$ and $\mathsf{urs} \leftarrow \mathcal{URS}$, $(x, w) \in \mathsf{R}$, and $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$, it holds that $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi) = 1$. It is *zero-knowledge* if there exists a PPT simulator $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{crs}}, \mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_{\pi})$ such that the distributions of $\pi' \leftarrow \mathsf{Sim}_{\pi}(\mathsf{crs}, x)$ and $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ are computationally indistinguishable for any $(x, w) \in \mathsf{R}$. For simulated proofs, the algorithm $\mathsf{Sim}_{\mathsf{H}}$ simulates the random oracle and $\mathsf{Sim}_{\mathsf{crs}}$ simulates the $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$, where there is an structured part srs. We also define a notion of *subversion zero-knowledge*, inspired by the notion introduced in [11]. To recall, the second part of the $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ is a random reference string which can later be sampled via a random oracle, and the first part is a structured string srs. For *subversion zero-knowledge*, there is no $\mathsf{Sim}_{\mathsf{crs}}$ anymore and the structured srs can be chosen by $\mathcal{A}$, while urs is sampled uniformly at random by H for the real proofs $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ or by $\mathsf{Sim}_{\mathsf{H}}$ for the simulated proofs $\pi' \leftarrow \mathsf{Sim}_{\pi}(\mathsf{crs}, x)$. Here we also require that the subverted srs belong to $\mathcal{SRS}$.

We define *adaptive knowledge soundness*. We remark that the soundness relation $\widetilde{\mathsf{R}}$ can be different from the (correctness) relation R. If $\widetilde{\mathsf{R}}$ is not explicitly defined, we implicitly set $\widetilde{\mathsf{R}} = \mathsf{R}$. An NIZK is adaptively knowledge sound for relation $\widetilde{\mathsf{R}}$ if there exist positive polynomials $p_{\mathsf{T}}, p_{\mathsf{P}}$, a PPT algorithm Ext and a PPT SimCRS so that for any $(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda)$, given oracle access to any PPT $\mathcal{A}$ (with explicit random tape ρ and making $Q_{\mathsf{H}} = \mathsf{poly}(\lambda)$ RO queries) that cannot distinguish $\overline{\mathsf{crs}} \in \mathcal{CRS}$ from a real $\mathsf{crs} := (\mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda), \mathsf{urs} \leftarrow \mathcal{URS})$, given $(x, \pi) \leftarrow \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}}; \rho)$, with probability at least $\frac{\mu(\lambda) - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)}$ the extractor finds $w \leftarrow \mathsf{Ext}(\overline{\mathsf{crs}}, \mathsf{td}, x, \pi, \rho, \vec{h})$ where $(x, w) \in \widetilde{\mathsf{R}}$. Here, $\vec{h}$ contains the outputs of H, the probability is over the random tape ρ of $\mathcal{A}$, the random tape of SimCRS, and the random choices of H. Also, we require that the runtime of Ext is bounded by $p_{\mathsf{T}}(\lambda, Q_H) \cdot \mathsf{Time}(\mathcal{A})$.

We further define *partial online-extractability* for NIZKs over a relation with statements $x = (x_0, x_1)$ and witnesses $w = (w_0, w_1)$. A NIZK is partially online-extractable if there extists an algorithm $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$ such that Ext_1 samples a partial statement x_0 uniformly at random along with a trapdoor td and for any PPT adversary that outputs pairs of partial statements $x_{1,i}$ and proofs π_i such that all $((x_0, x_{1,i}), \pi_i)$ verify with probability $\mu(\lambda)$, the extraction algorithm Ext_1 can use the trapdoor to extract partial witnesses $w_{1,i}$ for all statements such that there exist

[11] This membership test is required for our definition of subversion zero-knowledge. Note that in general it is difficult to check that some srs was generated via GenCRS. (We allow that $\mathcal{SRS}$ is not equal to the output space of GenCRS.)

partial witnesses $w_{0,i}$ with probability $\frac{\mu(\lambda)-\mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda,Q_H)}$ where p_{P} is a polynomial and Q_H is the number of hash queries made by the adversary. Looking forward, we will set the first partial statement x_0 to be the public parameters of a commitment scheme and the extracted witness to be the committed values - where the non-extracted witness is the opening of the commitments.

We also define *(statistical) adaptive subversion soundness*. Note that this notion does not require an extractor for the witness and the srs can be maliciously set up by an adversary, which differs from the standard notion of adaptive soundness. An NIZK is *(statistically) adaptively subversion sound* for relation $\widetilde{\mathsf{R}}$ inducing a language $\mathscr{L}_{\widetilde{\mathsf{R}}}$ if no (possibly unbounded) adversary, given a urs and access to the RO H, can output a subverted srs, an instance x, and a proof π such that $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs} \coloneqq (\mathsf{srs},\mathsf{urs}), x, \pi) = 1$ but $x \notin \mathscr{L}_{\widetilde{\mathsf{R}}}$.

Fiat-Shamir transformation. We recall the Fiat-Shamir transformation [37, 9] to turn a Σ-protocol $\Sigma = (\mathsf{Init}, \mathsf{Chall}, \mathsf{Resp}, \mathsf{Verify})$ that satisfies *correctness*, *high-min entropy*, *honest verifier zero-knowledge*, and 2-*Special Soundness*, into a NIZK $FS[\Sigma] = (\mathsf{GenCRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$ below:

- $\mathsf{GenCRS}(1^\lambda)$: outputs the empty string ϵ as we do not require a common reference string and omit crs as an input for other below algorithms,
- $\mathsf{Prove}^{\mathsf{H}}(x, w)$: receives a statement x and a witness w, runs $(\alpha, \mathsf{st}) \leftarrow \mathsf{Init}(x, w)$, computes the challenge $\beta \leftarrow \mathsf{H}(\mathsf{st}, \alpha)$, then computes $\gamma \leftarrow \mathsf{Resp}(\mathsf{st}, \beta)$ and outputs $\pi = (\alpha, \beta, \gamma)$.
- $\mathsf{Verify}^{\mathsf{H}}(x, \pi)$: receives a statement x and a proof $\pi = (\alpha, \beta, \gamma)$, and outputs $b \leftarrow \mathsf{Verify}(x, \alpha, \beta, \gamma)$.

The resulted NIZK satisfies *correctness*, *adaptive knowledge soundness* and *zero-knowledge*.

Pseudorandom Functions. A PRF is a family of efficient keyed functions $F : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ with key space $\mathcal{K}$, domain $\mathcal{D}$, and range $\mathcal{R}$ such that a randomly chosen function from the family should behave like a truly random function with same domain and range. Specifically, *pseudorandomness* of a standard PRF requires that any efficient adversary, given oracle access to $F(K, \cdot)$ where K is secret and chosen uniformly at random, cannot distinguish $F(K, x^*)$ from a uniformly random value in $\mathcal{R}$ for some challenge x^* chosen by the adversary.

3 NIZK-friendly Signature Scheme

3.1 The scheme

We describe a variant of Fischlin's variant of the Cramer-Shoup signature. We adapt it with the goal of constructing an efficient proof of knowledge of a signature later. We denote by $\mathbb{P}_X$ the set of primes within a set X.

The signature consists of three values $y \in \mathbb{Z}_N^*, a \in \mathbb{Z}$ and $e \in \mathbb{Z}$. We define intervals $\mathcal{S}_a$ and $\mathcal{S}_e$ which we use to sample a and e in Sign, respectively. Also, we define the intervals $\mathcal{R}_a$ and $\mathcal{R}_e$ which we use to in Verify to check range membership of a and e, respectively.

Let $A = 2^{3\lambda}$ and $\mathcal{S}_a = [0, A]$. Also, let $\mathcal{R}_a \supseteq \mathcal{S}_a$, $\mathcal{S}_e$ and $\mathcal{R}_e \supseteq \mathcal{S}_e$ be intervals such that for all $a \in \mathcal{R}_a$, we have $a < e$ for any $e \in \mathcal{R}_e$. Further, we require that $|\mathbb{P}_{\mathcal{S}_e}| = \Omega(2^{2\lambda})$ (*i.e.*, $\mathcal{S}_e$ contains at least $\Omega(2^{2\lambda})$ primes).

- $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}(1^\lambda)$: Sets $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$. Samples generators $h, h_1, h_2 \leftarrow \mathsf{Gen}(\mathsf{QR}_N)$ for QR_N at random. Outputs the public key $\mathsf{vk} = (N, h, h_1, h_2)$ and the secret key $\mathsf{sk} = (P, Q)$.
- $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}(\mathsf{sk}, m)$: Parses $\mathsf{sk} = (P, Q)$ and computes $\overline{m} = \mathsf{H}(m)$. Then, picks $e \leftarrow \mathbb{P}_{\mathcal{S}_e}$ and $a \leftarrow \mathcal{S}_a$ at random. Computes y such that

$$y^e = h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N.$$

  Output the signature $\sigma = (e, a, y)$.
- $\mathsf{S}_{\mathsf{fis}}.\mathsf{Verify}(\mathsf{vk}, m, \sigma)$: Parses $\mathsf{vk} = (N, h, h_1, h_2)$ and $\sigma = (e, a, y)$. Checks that $e \in \mathcal{R}_e$ is odd, $a \in \mathcal{R}_a$, and that

$$y^e = h h_1^a h_2^{a+\mathsf{H}(m)} \bmod N.$$

3.2 Proof of Security

This proof mostly follows the proof given in [38], using some alternative ways to set up the verification key with an embedded strong RSA challenge (N, z). These alternative key generation algorithms come with corresponding alternative signing algorithms, and they are indexed by bits b, b' that correspond to cases in a case distinction in the proof. We provide the full proof for completeness in Appendix B.2. As we will re-use this strategy in the proof of the blind signature scheme in Section 5, we describe this alternative key generation and alternative signing in Appendix B.1.

4 Commitment Schemes

We give an overview of the commitment schemes we require in this work.

4.1 Pedersen Commitments in QR_N

We recall Pedersen multi-commitments (MPed) over QR_N with message space $\mathbb{Z}^\ell$ for some $\ell \in \mathbb{N}$ [32].

- $\mathsf{MPed.GenPP}(1^\lambda)$: set $(N, P, Q) \leftarrow \mathsf{Gen}(1^\lambda)$ and sample ℓ random generators g_i of QR_N, and output $\mathsf{pp} = (N, h, g_1, \cdots, g_\ell)$. Note that with (P, Q), we can check whether g_i generates QR_N.
- $\mathsf{MPed.Commit}(\mathsf{pp}, \vec{m})$: sample $r \leftarrow [0, N \cdot 2^\lambda]$, set $c \leftarrow h^r \cdot \prod_{i=1}^{\ell} g_i^{m_i} \bmod N$, and output (c, r).
- $\mathsf{MPed.Verify}(\mathsf{pp}, c, \vec{m}, r)$: check if $c = \pm h^r \cdot \prod_{i=1}^{\ell} g_i^{m_i} \bmod N$.

MPed commitments are correct, statistically hiding and binding under the factoring assumption (which is implied by sRSA). Throughout this work, we use MPed commitments in QR_N to enforce in security proofs that values extracted from NIZKs are integers via Lemma 5.

4.2 Relaxed Integer Commitments with Slack

We define the notion of *relaxed integer commitment schemes* parameterized by $B, T \in \mathbb{N}$. Those are commitments with message space $\mathcal{C}_{\mathsf{msg}} = [0, B]$ that admit efficient opening proofs in zero-knowledge with some *slack*, *i.e.*, soundness guarantees that $x \in [-BT, BT]$. We refer to B as the range and T as the slack.

Definition 1. *A relaxed integer commitment is a commitment scheme* $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ *parameterized by two values* $T \in \mathbb{N}$ *and* $\vec{B} \in \mathbb{N}^\ell$ *for some* $\ell \in \mathbb{N}$*. The value* $\vec{B}$ *defines the message space* $\mathcal{C}_{\mathsf{msg}} = [0, \vec{B}] \subseteq \mathbb{Z}^\ell$*. The value* T *defines a relaxed message space* $\mathcal{C}_{\mathsf{msg}}^{\mathsf{rel}} = [-\vec{B}T, \vec{B}T]$*. We further require that the commitment scheme* $\mathsf{C}_{\mathsf{RInt}}$ *is*

1. *correct and hiding with respect to* $\mathcal{C}_{\mathsf{msg}}$ *(*i.e.*, the messages are sampled from* $\mathcal{C}_{\mathsf{msg}}$ *in the definitions Definitions 6 and 7), and*
2. *binding with respect to* $\mathcal{C}_{\mathsf{msg}}^{\mathsf{rel}}$ *(*i.e.*, the adversarial messages are allowed to be in* $\mathcal{C}_{\mathsf{msg}}^{\mathsf{rel}}$ *instead of* $\mathcal{C}_{\mathsf{msg}}$ *in Definition 8).*

We now instantiate $\mathsf{C}_{\mathsf{RInt}}$ over a group $\mathbb{G}$ with prime order $p \geq 2^{2\lambda}$. Let $\mathsf{pp} = (G, H) \in \mathbb{G}^2$ be the public parameters. Let $B, T \in \mathbb{N}$ such that $BT < \frac{p-1}{2}$. The commitments are ElGamal commitments $c \leftarrow (xG + rH, rG)$ for $r \leftarrow \mathbb{Z}_p$, except that we add the additional requirement of $x \in [-BT, BT]$ in verification. Note that as we have $[-BT, BT] \subset [-\frac{p-1}{2}, \frac{p-1}{2}]$, this condition ensures that no overflows occur (so we commit to a subset of $\mathbb{Z}$). Looking ahead, our zero-knowledge opening proofs leverage the structure of QR_N to ensure that extracted values are integers in the relaxed range.

We naturally generalize our commitment scheme to vectors $\vec{m} = (m_1, \cdots, m_\ell) \in [0, \vec{B}]$ of integers from (potentially different) intervals induced by $\vec{B} = (B_1, \cdots, B_\ell)$. We require that $B_i \cdot T < \frac{p-1}{2}$ for all $i \in [\ell]$. The integer commitment $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}$ with uniform public parameters $\mathsf{pp} = (H, \vec{G})$ is given below, where $\vec{G} = (G_1 \cdots, G_\ell)$. The randomness space is $\mathcal{C}_{\mathsf{rnd}} = \mathbb{Z}_p$. By definition, the message space is $\mathcal{C}_{\mathsf{msg}} = [0, \vec{B}]$ and the relaxed message space is $\mathcal{C}_{\mathsf{msg}}^{\mathsf{rel}} = [-\vec{B}T, \vec{B}T]$.

– $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}.\mathsf{Commit}(\mathsf{pp}, \vec{m})$: Takes as input public parameters pp and $\vec{m} \in [0, \vec{B}]$, samples $r \leftarrow \mathbb{Z}_p$, sets $C_i \leftarrow m_i H + rG_i, \vec{C} \leftarrow (C_1, \cdots, C_\ell)$, $F \leftarrow rH$, and outputs (c, r) for $c = (\vec{C}, F)$.
– $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}.\mathsf{Verify}(\mathsf{pp}, c, \vec{m}, r)$: Takes as input $(c, r) \in \mathbb{G}^{\ell+1} \times \mathbb{Z}_p$, parses $c = (\vec{C}, F)$ and checks that

$$\vec{m} \in [-\vec{B}T, \vec{B}T], \qquad F = rH, \qquad \vec{C} = \vec{m}H + r\vec{G}.$$

If the (relaxed) range (induced by $\vec{B}$ and T) is clear by context, we often write $\mathsf{C}_{\mathsf{RInt}}$ for short.

Theorem 1. *The scheme $\mathsf{C}_{\mathsf{RInt}}$ is correct, hiding under DDH in $\mathbb{G}$, and perfectly binding.*

Proof. Correctness is straightforward.

For hiding, we have $(H, G_i, rH, rG_i) \overset{c}{\approx} (H, G_i, rH, t_i G_i)$ for $t_i \leftarrow \mathbb{Z}_p$ under DDH. Since $t_i G_i$ masks $m_i H$ additively, the value $m_i H + t_i G_i$ is uniform in $\mathbb{G}$. Thus, $(\vec{C}, F) \overset{c}{\approx} \vec{D}$ for $\vec{D} \leftarrow \mathbb{G}^{\ell+1}$ after ℓ game hops.

For binding, observe that since $\vec{m} \in [-\vec{B}T, \vec{B}T] \subset [-\frac{p-1}{2}, \frac{p-1}{2}]^\ell$, the message $\vec{m}$ is uniquely determined by c and the verification equations. In more detail, if $c = (\vec{C}, F)$ verifies correctly, then we have $r = \log_H(F) \in \mathbb{Z}_p$ and $m_i H = C_i - rG_i$. Thus, we have $m_i \equiv \log_H(C_i - rG_i) \bmod p$. Since for every $x \in \mathbb{Z}_p$, there is exactly *one* $m_i \in [-\frac{p-1}{2}, \frac{p-1}{2}]$ such that $m_i \equiv x$, the value m_i is uniquely determined.

Note that we could also set $C \leftarrow rH + \sum_i x_i G_i$ to obtain compact commitments. We choose ElGamal commitments instead of Pedersen commitments as in our applications, we require perfect binding. In our construction, we also require exact integer commitments for some fixed range.

Definition 2 (Integer commitments with bounded range). *If the range in verification is identical to the message space, we say that the commitment is an (exact) integer commitment with $\mathcal{C}_{\mathsf{msg}} = [0, \vec{B}]$ (and $\mathcal{C}_{\mathsf{msg}} = \mathcal{C}_{\mathsf{msg}}^{\mathsf{rel}}$).*

4.3 Commitments in Arbitrary Groups

Let $\hat{\mathbb{G}} = \langle \hat{g} \rangle$ be an arbitrary cyclic group with generator $\hat{g}$. We assume an upper bound U on the order of $\hat{\mathbb{G}}$.

We construct a commitment scheme with message space $\mathcal{C}_{\mathsf{msg}} = \hat{\mathbb{G}}$ (*i.e.*, for messages $\hat{x} \in \hat{\mathbb{G}}$). Looking ahead, we cannot rely on computational hardness assumptions in $\hat{\mathbb{G}}$ (as in our construction, this group can be chosen maliciously by the adversary). As secure (non-interactive) commitments require some type of hardness assumption, we need some additional structure. For this, we use an additional relaxed integer commitment scheme $\mathsf{C}_{\mathsf{RInt}}^{B,T}$ (with parameters B, T defined below) [12]. To commit to $\hat{x} \in \hat{\mathbb{G}}$, a user first sets $\hat{c} \leftarrow \hat{x}\hat{g}^s$ for $s \leftarrow [0, U \cdot 2^\lambda]$. Note that $\hat{c}$ hides $\hat{x}$ statistically, but is not binding to $\hat{x}$. For example, a user can open $\hat{c} = \hat{g}\hat{g}^2$ to message $\hat{g}$ or $\hat{g}^2$.

To achieve binding, the user additionally commits to its randomness s in a commitment c via $\mathsf{C}_{\mathsf{RInt}}^{B,T}$ for $B = U \cdot 2^\lambda$ and T arbitrary. If s is fixed over the integers $\mathbb{Z}$, the user is forced to open the commitment $\hat{c}$ to the message $\hat{x} = \hat{c} \cdot \hat{g}^{-s}$. Note that the commitment c fixes s over a subset of $\mathbb{Z}$ (due to binding of $\mathsf{C}_{\mathsf{RInt}}$) which is sufficient [13]. Since $\mathsf{C}_{\mathsf{RInt}}$ is hiding, the additional commitment c reveals no information about s and thus, the scheme remains hiding.

Since our instantiation of $\mathsf{C}_{\mathsf{RInt}}$ requires a group $\mathbb{G}$ (whose size scales with B), we allow s to be split into a vector $\vec{s}$ with $s_i \in [0, B]$. Then, the user commits to $\vec{s} \in [0, \vec{B}]$ via $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}$ for $\vec{B} = (B, \cdots, B)$ and arbitrary $B \in \mathbb{N}$. Let $\ell = \lceil \log(U \cdot 2^\lambda) / \log(B) \rceil$. The commitment scheme $\mathsf{C}_{\mathsf{Grp}}$ (which is implicitly parameterized by $\mathsf{C}_{\mathsf{RInt}}$) is given below.

– $\mathsf{C}_{\mathsf{Grp}}.\mathsf{Setup}(1^\lambda)$: Outputs $\mathsf{pp} \leftarrow \mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}.\mathsf{Setup}(1^\lambda)$.

[12] If we instantiate $\mathsf{C}_{\mathsf{RInt}}$ as in Section 4.2, then the additional structure is a prime order group $\mathbb{G}$ in which DDH is assumed to be hard.

[13] Note that for our construction, it is important that $\mathsf{C}_{\mathsf{RInt}}$ commits over the integers. For example, a commitment c over $\mathbb{Z}_p$ is not sufficient. To illustrate this, assume that $s \in \mathbb{Z}$ is fixed over $\mathbb{Z}_p$. Then, $\hat{c} = \hat{g}^s \hat{g}^{ps}$ can be opened to $\hat{g}^s$ or $\hat{g}^{ps}$ since $s \equiv ps \bmod p$. But we have $\hat{g}^s = \hat{g}^{ps}$ only if $\mathsf{ord}(\hat{g}) \mid p(s-1)$. Since the order is unknown, this does not hold in general and thus, the commitment is not binding.

– $\mathsf{C}_{\mathsf{Grp}}.\mathsf{Commit}(\mathsf{pp}, \hat{x})$: Takes as input public parameters pp and $\hat{x} \in \hat{\mathbb{G}}$, samples $s \leftarrow [0, U \cdot 2^{\lambda}]$, sets $\hat{c} \leftarrow \hat{x}\hat{g}^{s}$. Then, decomposes $s = \sum_{i=1}^{\ell} s_i B^{i-1}$ with $s_i \in [0, B]$ and commits to $\vec{s} = (s_1, \cdots, s_\ell)$ via $(c, r) \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}, \vec{s})$. Outputs (c_x, r_x) for $c_x = (\hat{c}, c)$ and $r_x = (\vec{s}, r)$.
– $\mathsf{C}_{\mathsf{Grp}}.\mathsf{Verify}(\mathsf{pp}, c_x, \hat{x}, r_x)$: Parses c_x, r_x as above. Then, sets $s = \sum_{i=1}^{\ell} s_i B^{i-1}$ and checks that $\mathsf{C}_{\mathsf{RInt}}.\mathsf{Verify}(\mathsf{pp}, c, \vec{s}, r) = 1$ and $\hat{c} = \hat{x}\hat{g}^{s}$.

Theorem 2. *The scheme $\mathsf{C}_{\mathsf{Grp}}$ is correct, hiding and binding under the hiding and binding property of $\mathsf{C}_{\mathsf{RInt}}$, respectively.*

We defer the security proof to Appendix C.1.

4.4 Efficient Opening in Zero-Knowledge

We construct efficient NIZKs Π_{int} and Π_{grp} to open $\mathsf{C}_{\mathsf{RInt}}$ and $\mathsf{C}_{\mathsf{Grp}}$, respectively, in zero-knowledge. Due to space limitations, we refer to the Section 1.2 for a brief overview. The full schemes are given in Appendix C.2.

5 Blind Signature with Malicious Signer Blindness

In this section, we detail our blind signature construction based on the strong RSA assumption and DDH in prime order groups.

5.1 Primitives

Before we detail our construction, we prepare the required primitives and related parameters.

Remark 1. In the following, we will define several NIZKs. As the reference string crs of these NIZKs are set up by the signer, we need to be careful with the security guarantees of each NIZK. For cases where the signer takes the role of the prover, we require subversion soundness (*i.e.*, the soundness property should hold even with regard to a maliciously generated crs) but standard zero-knowledge is sufficient. If the signer takes on the role of the verifier, we require subversion zero-knowledge (*i.e.*, the zero-knowledge property should still hold even with regard to a maliciously generated crs).

Relaxed integer commitment. For constructing a proof of knowledge of a signature of the scheme $\mathsf{S}_{\mathsf{fis}}$, we require a relaxed integer commitment. We describe the choices of parameters and motivate them in the following.

Let $T \in \mathbb{N}$. Let $A = 2^{3\lambda}$, $E = 2^{3\lambda}$, and $\overline{E} \in \mathbb{N}$ such that the following equations hold.

$$\log(\overline{E}) = \mathsf{poly}(\lambda) \tag{5}$$

$$A \cdot T < \overline{E} - ET. \tag{6}$$

Let $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}$ be a relaxed integer commitment scheme with uniform public parameters of length ℓ_{rint}, statistical binding and computational hiding (cf. Section 4.2) for $\vec{B} \coloneqq (A, E)$ and slack T. We write $\mathsf{C}_{\mathsf{RInt}}$ for short. The choices for these parameters are motivated below.

Recall that $\vec{B}$ defines the message space $[0, \vec{B}]$ and that the slack T dictates the relaxed message space $[-\vec{B}T, \vec{B}T]$, *i.e.*, the message space for verification [14].

For convenience, let $\mathcal{S}_a \coloneqq [0, A]$ and $\mathcal{S}_e \coloneqq [\overline{E}, \overline{E} + E]$. In our construction, we commit to $a \in \mathcal{S}_a$ and $e - \overline{E} \in [0, E]$ for $e \in \mathcal{S}_e$ via $\mathsf{C}_{\mathsf{RInt}}$. The above parameter choices guarantee that for message $(a, e - \overline{E})$ that passes $\mathsf{C}_{\mathsf{RInt}}$ verification, it holds that the values (a, e) pass the range checks in the $\mathsf{S}_{\mathsf{fis}}$ signature.

To illustrate this, set $\mathcal{R}_a \coloneqq [-AT, AT]$ and $\mathcal{R}_e \coloneqq [\overline{E} - ET, \overline{E} + ET]$. By Equation (6), we have that for any $a \in \mathcal{R}_a$ and $e \in \mathcal{R}_e$ that $a < e$. Further, verification of $\mathsf{C}_{\mathsf{RInt}}$ guarantees that the

[14] In our instantiation, we have $T = 2^{\lambda+1}L$ with $L = 2^{10}$. It is sufficient to set $\overline{E} = 2^{5\lambda}$ to have $AT = 2^{4\lambda+11} < 2^{5\lambda} - 2^{4\lambda+11} = \overline{E} - ET$ for Equation (6) if $\lambda \geq 14$.

committed a lies in the interval $a \in \mathcal{S}_a$ as desired. Also, since we commit to $e - \overline{E} \in [-ET, ET]$, we have $e \in \mathcal{S}_e$.

In our instantiation, we can employ our $\mathsf{C}_{\mathsf{RInt}}$ construction from Section 4.2 which can be opened with a simple NIZK Π_{int}. This is the core technique that allows us to construct a proof of knowledge of a $\mathsf{S}_{\mathsf{fis}}$ signature in an efficient manner [15].

In the instantiation, we set $\overline{E} = 2^{5\lambda}$. Then, it is guaranteed that the interval $\mathcal{S}_e = [\overline{E}, \overline{E} + E]$ contains at least $\Omega(2^{2\lambda})$ primes. This follows from a recent refinement [54] of Huxley's bound [56, 53]. We provide a full proof in Appendix D.1. This is required to avoid collisions in a hash function mapping into $\mathcal{S}_e$.

Proof of Knowledge for $\mathsf{S}_{\mathsf{fis}}$ signatures. We require a NIZK to proof knowledge of a valid $\mathsf{S}_{\mathsf{fis}}$ signature (e, a, y) on the hash of a message $\overline{m}$. To prove one-more unforgeability, we require that (e, a) are fixed statistically in the statement. Thus, we add a $\mathsf{C}_{\mathsf{RInt}}$ commitment for (e, a) which also enables efficient proofs for range membership (as discussed above). Let Π_{fis} be an NIZK with oracle $\mathsf{H}_{\mathsf{fis}}$ for the relation

$$\begin{aligned}\mathsf{R}_{\mathsf{fis}} \coloneqq \big\{(x, w) \mid\ & y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N, e \equiv 1 \bmod 2, y \in \langle h_1 \rangle, \\ & (c_I, d_I) = \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_I, (a, e - \overline{E}); r_I), e \in \mathcal{S}_e, a \in \mathcal{S}_a\big\}\end{aligned}$$

for $x = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, c_I), w = (e, a, y, r_I, d_I)$ with subversion zero-knowledge, correctness, and adaptive knowledge soundness for the relation

$$\begin{aligned}\widetilde{\mathsf{R}}_{\mathsf{fis}} \coloneqq \big\{(x, w) \mid\ & y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N, e \equiv 1 \bmod 2, \\ & \mathsf{C}_{\mathsf{RInt}}.\mathsf{Verify}(\mathsf{pp}_I, (a, e - \overline{E}), d_I) = 1\big\}\end{aligned}$$

with x, w as above. Note that the soundness relation $\widetilde{\mathsf{R}}_{\mathsf{fis}}$ implies that $a \in \mathcal{R}_a$ and $e \in \mathcal{R}_e$ (cf. Section 5.1) and thus, (e, a, y) form a valid $\mathsf{S}_{\mathsf{fis}}$ signature. For zero-knowledge and correctness, there are stronger requirements for the witness (which are fulfilled in our construction). Notably, we require that $a \in \mathcal{S}_a, e \in \mathcal{S}_e$, and that $y \in \langle h_1 \rangle$. (The latter is required to commit to y via $\mathsf{C}_{\mathsf{Grp}}$ in our instantiation.)

Integer commitment and opening proof for Pedersen. Let $S \in \mathbb{N}$. Let C_{Z} be an exact integer commitment scheme with message space $\mathsf{C}_{\mathsf{Z}}.\mathcal{C}_{\mathsf{msg}} = [0, 2^\lambda - 1] \times [0, S]$ with uniform public parameters of length ℓ_{z}, correctness, perfect binding, and computational hiding (cf. Definition 2). We denote by $\mathsf{C}_{\mathsf{Z}}.\mathcal{C}_{\mathsf{opn}}$ the opening space of C_{Z}. In the blind signature scheme, we will require the user to both the hash $\overline{m}$ as well as the random coins r that it plans to use to derive the Pedersen commitment using the perfectly binding commitment scheme C_{Z}. This first commitment is hashed to obtain the prime e used for singing. Furthermore, the user is required to attach a proof π_{ped} that the Pedersen commitment c is consistent with the hash $\overline{m}$ and the coins r. The commitment c_Z along with the proof π_{ped} allows the reduction in the one-more unforgeability proof to obtain the value $\overline{m}$ and the coins r which in turn enables it to generate signatures using the alternate signing algorithms from Appendix B.1. In the proof of blindness, we rely on the zero-knowledge property of Π_{ped} as well as the hiding property of the commitment schemes for blindness.

Let Π_{ped} be an NIZK with oracle $\mathsf{H}_{\mathsf{ped}}$ for the relation

$$\begin{aligned}\mathsf{R}_{\mathsf{ped}} \coloneqq \big\{(x, w) \mid\ & c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N, \mathsf{C}_{\mathsf{Z}}.\mathsf{Verify}(\mathsf{pp}, c_Z, (\overline{m}, r), d_Z) = 1, \\ & \overline{m} \in [0, 2^\lambda - 1], r \in [0, S]\big\},\end{aligned}$$

for $x = (\mathsf{pp}, N, e, h_2, g, c, c_Z), w = (\overline{m}, r, d_Z)$ with correctness and subversion zero-knowledge. We also require partial online-extraction for $\mathsf{R}_{\mathsf{ped}}$, where we split the statement x into $x_0 = \mathsf{pp}$ and $x_1 = (N, e, h_2, g, c)$ and the witness into $w_0 = d_Z$ and $w_1 = (\overline{m}, r)$. (This implicitly defines the partial statement space $X_0 = \{0,1\}^{\ell_z}$ and the partial witness space $W_1 = \mathsf{C}_{\mathsf{Z}}.\mathcal{C}_{\mathsf{opn}}$.) The user uses

[15] In our construction, the value $\vec{B}$ is large. Our technique allows to avoid the use of exact range proofs whose efficiency scales with the range $[0, \vec{B}]$.

the NIZK to ensure that the commitment c is indeed formed with the values committed via $\mathsf{C_Z}$. For the security proof, the reduction "punctures" the verification key in such a way that it can sign messages without knowing the secret key. For this reason, online-extraction is required to extract the messages before signing. As mentioned above, we exclude d_Z from the extracted witness for efficiency (as existence is sufficient). Also, we embed the extraction trapdoor in the public parameters (instead of the crs also for efficiency) [16].

NIZKs for group membership. As the factorization of the RSA modulus N is private, it is hard to check whether a given $g \in \mathbb{Z}_N^*$ generates the entire group QR_N. But such a check is required, *e.g.*, to check that a $\mathsf{S_{fis}}$ verification key is setup honestly.

It is necessary to have such a check as the adversary sends a blinded commitment $c = h_2^{\overline{m}} g^{re}$ to the signer during signing. When $\langle h_2 \rangle \neq \langle g \rangle$, the blindness adversary could raise c to the power of $\mathsf{ord}(g)$ to remove the part g^{re} and then check whether the resulting $c^{\mathsf{ord}(g)} = (h_2^{\overline{m}_0})^{\mathsf{ord}(g)}$ or $c^{\mathsf{ord}(g)} = (h_2^{\overline{m}_1})^{\mathsf{ord}(g)}$.

We carefully design our blind signature such that such that the following check is sufficient: given some generator g for a group $\mathbb{G} = \langle g \rangle$ and an arbitrary element h, check whether $\langle h \rangle = \mathbb{G}$.

While this check remains inefficient, we ask that the signer adds a NIZK for $\langle h \rangle = \mathbb{G}$ to the verification key. Since the signer sets up the elements g and h itself, it can set $h = g^x$ for some $x \in \mathbb{Z}_{\mathsf{ord}(g)}$. Knowing x, constructing such a proof for $\langle h \rangle = \mathbb{G}$ is simple. Since the signer sets up multiple such values h, we batch the statement for simplicity.

Let Π_{gen} be an NIZK with oracle $\mathsf{H_{gen}}$ satisfying statistical adaptive subversion soundness, zero-knowledge, and correctness for the relation

$$\mathsf{R_{gen}} = \left\{ (x, w) \mid \forall i \in [k] : h_i^{\alpha_i} \equiv h \bmod N, h^{\beta_i} \equiv h_i \bmod N \right\} ,$$

where $x = (N, k, h, (h_i)_{i \in [k]}), w = ((\alpha_i, \beta_i)_{i \in [k]})$. Note that $\mathsf{R_{gen}}$ implies that $\langle h \rangle = \langle h_i \rangle$ for all i.

Similarly, we need a NIZK to prove subgroup membership of a single element. For this, let Π_{sub} be an NIZK with oracle $\mathsf{H_{sub}}$ satisfying statistical adaptive subversion soundness, zero-knowledge, and correctness for the relation

$$\mathsf{R_{sub}} = \left\{ (x, w) \mid z \equiv h^d \bmod N \right\} ,$$

where $x = (z, N, h), w = d$, and $\mathsf{R_{sub}}$ induces $\mathscr{L}_{\mathsf{R_{sub}}} = \{(z, N, h) \mid z \in \langle h \rangle \subseteq \mathbb{Z}_N^*\}$.

Pseudorandom Function. For technical reasons, we require the signer's choice of a to be deterministic (in particular, during the one-more unforgeability proof, there will be a case where the signer needs to re-use the same a when signing the same message twice with the same e). Therefore, to avoid having the signer keep state, instead of choosing a uniformly at random from $\mathcal{S}_a$, we let the signer use a PRF to derive a.

The input to the PRF is a bitstring concatenated to length $\ell_N + \ell_{\mathsf{C_{RInt}}}$, where ℓ_N denotes bit length of an RSA modulus N and $\ell_{\mathsf{C_{RInt}}}$ denotes the bit length of $\mathsf{C_Z}$ commitments. Let $\{\mathsf{PRF}_K\}_{K \in \mathcal{K}}$ be a PRF consisting of $\mathsf{PRF}_K \colon \{0,1\}^{\ell_N + \ell_{\mathsf{C_{RInt}}}} \to \mathcal{S}_a$, which is index by $K \in \mathcal{K}$ from some key space $\mathcal{K} = \{0,1\}^{\mathsf{poly}(\lambda)}$.

Hash functions. We require the following hash functions in our construction. Each hash function is modeled as random oracle in the security proofs.

- $\mathsf{H_{urs}}$: Let $\mathsf{H_{urs}} : \{0,1\}^* \to \{0,1\}^{\ell_{\mathsf{ped}}} \times \{0,1\}^{\ell_{\mathsf{fis}}} \times \{0,1\}^{\ell_{\mathsf{gen}}} \times \{0,1\}^{\ell_{\mathsf{sub}}}$ be a hash function, where ℓ_{zkp} is the bit-size of the uniform reference string of Π_{zkp}. Later, we use $\mathsf{H_{urs}}$ to setup the random part urs of each crs for the above NIZKs.
- H: Let $\mathsf{H} : \{0,1\}^* \to [0, 2^{2\lambda} - 1]$ be a hash function. Later, we use H to compute a short digest $\overline{m} = \mathsf{H}(m)$ of the message $m \in \{0,1\}^*$.
- $\mathsf{H_{\mathbb{P}}}$: Let $\mathsf{H_{\mathbb{P}}} : \{0,1\}^* \to \mathbb{P}_{\mathcal{S}_e}$ be a hash function mapping into the primes in the interval $\mathcal{S}_e$.

[16] Roughly, the commitment $\mathsf{C_Z}$ is extractable and we embed the extraction trapdoor into pp. But pp is part of the statement, so we make sure that this part is sampled at random (cf. Definition 22).

– H_{pp}: Let $\mathsf{H}_{\mathsf{pp}} : \{0,1\}^* \to \{0,1\}^{\ell_{\mathsf{z}}} \times \{0,1\}^{\ell_{\mathsf{rint}}}$ be a random oracle.

Note that we can instantiate $\mathsf{H}_{\mathbb{P}} : \{0,1\}^* \to \mathbb{P}_{\mathcal{S}_e}$ by picking uniformly random elements in the space $\mathcal{S}_e \coloneqq [\overline{E}, \overline{E} + E]$ until we hit a prime. The distribution of the outputs of $\mathsf{H}_{\mathbb{P}}$ is uniform over $\mathbb{P}_{\mathcal{S}_e}$, which is the set of primes in the interval $\mathcal{S}_e$. The cardinality of $\mathbb{P}_{\mathcal{S}_e}$ satisfies $|\mathbb{P}_{\mathcal{S}_e}| = \Omega(2^{2\lambda})$ following Appendix D.1.

5.2 Construction

Set $S = N \cdot 2^{\lambda}$ which is passed implicitly as parameter in our construction. Also, we set $\mathsf{pp} = (\mathsf{pp}_I, \mathsf{pp}_Z) \leftarrow \mathsf{H}_{\mathsf{pp}}(0)$. We assume that user and signer compute $\mathsf{pp} = \mathsf{H}_{\mathsf{pp}}(0)$ implicitly. The construction is detailed below. We also detail a signing session in Figure 1.

– $\mathsf{BS}_{\mathsf{fis}}.\mathsf{KG}(1^{\lambda})$: First, generates the crs for the NIZKs as $\mathsf{crs}_{\mathsf{zkp}} \leftarrow (\mathsf{srs}_{\mathsf{zkp}}, \mathsf{urs}_{\mathsf{zkp}})$, where $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$ and $\mathsf{srs}_{\mathsf{zkp}} \leftarrow \Pi_{\mathsf{zkp}}.\mathsf{GenCRS}(1^{\lambda})$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. Then, generates a public key for $\mathsf{S}_{\mathsf{fis}}$ as follows. Sets $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^{\lambda})$ and samples $g \in \mathsf{QR}_N$. Samples $\alpha_i \leftarrow \mathbb{Z}_{\mathsf{ord}(g)}$ and sets $\beta_i \leftarrow \alpha_i^{-1} \bmod \mathsf{ord}(g)$ for $i \in [3]$. Computes $h_1 \leftarrow g^{\alpha_1} \bmod N$, $h_2 \leftarrow g^{\alpha_2} \bmod N$ and $h \leftarrow g^{\alpha_3} \bmod N$. Note that h, h_1 and h_2 are generators of QR_N with overwhelming probability. Next, proves that all QR_N elements in bvk key generate the same group via $\pi_{\mathsf{gen}} \leftarrow \Pi_{\mathsf{gen}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, w_{\mathsf{gen}})$, where $x_{\mathsf{gen}} = (N, 3, g, (h, h_1, h_2)), w_{\mathsf{gen}} = (\alpha_i, \beta_i)_{i \in [3]}$. Then, sample a key $K \leftarrow \mathcal{K}$ for PRF. Finally, output

$$\begin{aligned}\mathsf{bvk} &= (\mathsf{crs}_{\mathsf{fis}}, \mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}, N, h, h_1, h_2, g, \pi_{\mathsf{gen}}),\\ \mathsf{bsk} &= (\mathsf{bvk}, P, Q, K).\end{aligned}$$

– $\mathsf{BS}_{\mathsf{fis}}.\mathsf{User}(\mathsf{bvk}, m)$: Given verification key bvk, and message m, checks

$$\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1$$

for $x_{\mathsf{gen}} = (N, 4, (h, h_1, h_2, g))$. Then, sets $\overline{m} \leftarrow \mathsf{H}(m)$ and set up a commitment c to $\overline{m}$ as follows. Samples randomness $r \leftarrow [0, S]$ for c and commits to $(\overline{m}, r)$ via $(c_Z, d_Z) \leftarrow \mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(\mathsf{pp}, (\overline{m}, r))$. Sets $e \leftarrow \mathsf{H}_{\mathbb{P}}(c_Z)$ and compute $c = h_2^{\overline{m}} \cdot g^{re} \bmod N$. Next, generate a proof $\pi_{\mathsf{ped}} \leftarrow \Pi_{\mathsf{ped}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}}, w_{\mathsf{ped}})$ for $x_{\mathsf{ped}} = (\mathsf{pp}, N, e, h_2, g, c, c_Z), w_{\mathsf{ped}} = (\overline{m}, r, d_Z)$. Note that π_{ped} proves that the generation of c was performed honestly with respect to c_Z. Finally, output

$$\begin{aligned}\rho_1 &= (c, c_Z, \pi_{\mathsf{ped}}),\\ \mathsf{st}_U &= (e, r, m).\end{aligned}$$

– $\mathsf{BS}_{\mathsf{fis}}.\mathsf{Signer}(\mathsf{bsk}, \rho_1)$: Given signing key $\mathsf{bsk} = (\mathsf{bvk}, P, Q, K)$ and user's output $\rho_1 = (c, c_Z, \pi_{\mathsf{ped}})$, checks $\Pi_{\mathsf{ped}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}}, \pi_{\mathsf{ped}}) = 1$ for $x_{\mathsf{ped}} = (\mathsf{pp}, N, e, h_2, g, c, c_Z)$. Next, computes $e \leftarrow \mathsf{H}_{\mathbb{P}}(c_Z)$ and sets $d \leftarrow e^{-1} \bmod \phi(N)$. Then, sets $a \leftarrow \mathsf{PRF}_K(c \parallel c_Z)$ which it uses as randomness for the signing process. Using d and a, computes a *presignature* z via $z' \leftarrow h \cdot h_1^a \cdot c \cdot h_2^a \bmod N$ and $z \leftarrow (z')^d \bmod N$. Finally, proves that $z \in \langle z' \rangle$ via $\pi_{\mathsf{sub}} \leftarrow \Pi_{\mathsf{sub}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{sub}}}(\mathsf{crs}_{\mathsf{sub}}, x_{\mathsf{sub}}, w_{\mathsf{sub}})$ for $x_{\mathsf{sub}} = (z, N, z'), w_{\mathsf{sub}} = d$ and outputs

$$\rho_2 = (z, a, \pi_{\mathsf{sub}}).$$

– $\mathsf{BS}_{\mathsf{fis}}.\mathsf{Derive}(\mathsf{st}_U, \rho_2)$: given state st_U and last message $\rho_2 = (z, a, \pi_{\mathsf{sub}})$, sets $z' \leftarrow h \cdot h_1^a \cdot c \cdot h_2^a$, checks $\Pi_{\mathsf{sub}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{sub}}}(\mathsf{crs}_{\mathsf{sub}}, x_{\mathsf{sub}}, \pi_{\mathsf{sub}}) = 1$ for $x_{\mathsf{sub}} = (z, N, z')$ and $a \in \mathcal{S}_a$. Next, computes a $\mathsf{S}_{\mathsf{fis}}$ signature on $\overline{m}$ from the presignature z via $y \leftarrow z \cdot g^{-r} \bmod N$. Then, checks whether $\sigma_{\mathsf{fis}} = (e, a, y)$ indeed forms a correct signature on m via $\mathsf{S}_{\mathsf{fis}}.\mathsf{Verify}(\mathsf{vk}, m, \sigma_{\mathsf{fis}}) = 1$. Next, generates a $\mathsf{BS}_{\mathsf{fis}}$ signature as follows. Sets $\overline{m} = \mathsf{H}(m)$ and $(c_I, d_I) \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_I, (a, e - \overline{E}); r_I)$ for $r_I \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathcal{C}_{\mathsf{rnd}}$. Proves that σ_{fis} verifies correctly via $\pi_{\mathsf{fis}} \leftarrow \Pi_{\mathsf{fis}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}_{\mathsf{fis}}, x_{\mathsf{fis}}, w_{\mathsf{fis}})$ for $x_{\mathsf{fis}} = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, c_I), w_{\mathsf{fis}} = (e, a, y, r_I, d_I)$. Outputs

$$\sigma = (\pi_{\mathsf{fis}}, c_I).$$

– $\mathsf{BS}_{\mathsf{fis}}.\mathsf{Verify}(\mathsf{bvk}, m, \sigma)$: Given verification key bvk, message m, and signature $\sigma = (\pi_{\mathsf{fis}}, c_I)$, computes $\overline{m} = \mathsf{H}(m)$ and checks

$$\Pi_{\mathsf{fis}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}_{\mathsf{fis}}, x_{\mathsf{fis}}, \pi_{\mathsf{fis}}),$$

for $x_{\mathsf{fis}} = (\mathsf{pp}, N, h_1, h_2, h, \overline{m}, c_I)$.

Signer$(\mathsf{bvk}, \mathsf{bsk})$		**User**(bvk, m)
		1 : **check** $\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1$
		2 : $\overline{m} \leftarrow \mathsf{H}(m)$
		3 : $r \leftarrow [0, S]$
		4 : $(c_Z, d_Z) \leftarrow \mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(\mathsf{pp}_Z, (\overline{m}, r))$
		5 : $e \leftarrow \mathsf{H}_{\mathbb{P}}(c_Z)$
		6 : $c = h_2^{\overline{m}} \cdot g^{re} \bmod N$
		7 : $\pi_{\mathsf{ped}} \leftarrow \Pi_{\mathsf{ped}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}}, w_{\mathsf{ped}})$
	$\leftarrow$	$(c, c_Z, \pi_{\mathsf{ped}})$
8 : **check** $\Pi_{\mathsf{ped}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}}, \pi_{\mathsf{ped}}) = 1$		
9 : $e \leftarrow \mathsf{H}_{\mathbb{P}}(c_Z)$		
10 : $d \leftarrow e^{-1} \bmod \phi(N)$		
11 : $a \leftarrow \mathsf{PRF}_K(c \parallel c_Z)$		
12 : $z' \leftarrow h \cdot h_1^a \cdot c \cdot h_2^a \bmod N$		
13 : $z \leftarrow (z')^d \bmod N$		
14 : $\pi_{\mathsf{sub}} \leftarrow \Pi_{\mathsf{sub}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{sub}}}(\mathsf{crs}_{\mathsf{sub}}, x_{\mathsf{sub}}, w_{\mathsf{sub}})$		
	$\rightarrow$	z, a, π_{sub}
		15 : **check** $\Pi_{\mathsf{sub}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{sub}}}(\mathsf{crs}_{\mathsf{sub}}, x_{\mathsf{sub}}, \pi_{\mathsf{sub}}) = 1$
		16 : **check** $a \in \mathcal{S}_a$
		17 : $y \leftarrow z \cdot g^{-r} \bmod N$
		18 : **check** $\mathsf{S}_{\mathsf{fis}}.\mathsf{Verify}(\mathsf{vk}, m, (e, a, y)) = 1$
		19 : $r_I \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathcal{C}_{\mathsf{rnd}}$
		20 : $(c_I, d_I) \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_I, (a, e - \overline{E}), r_I)$
		21 : $\pi_{\mathsf{fis}} \leftarrow \Pi_{\mathsf{fis}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}_{\mathsf{fis}}, x_{\mathsf{fis}}, w_{\mathsf{fis}})$
		22 : **return** $\sigma = (\pi_{\mathsf{fis}}, c_I)$

Fig. 1. A signing session of $\mathsf{BS}_{\mathsf{fis}}$ for message m. We have $(\mathsf{pp}_I, \mathsf{pp}_Z) = \mathsf{H}_{\mathsf{pp}}(0)$, $x_{\mathsf{gen}} = (N, 4, (h, h_1, h_2, g)), x_{\mathsf{ped}} = (\mathsf{pp}, N, e, h_2, g, c, c_Z), w_{\mathsf{ped}} = (\overline{m}, r, d_Z), x_{\mathsf{sub}} = (z, N, z'), w_{\mathsf{sub}} = d, x_{\mathsf{fis}} = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, c_I), w_{\mathsf{fis}} = (e, a, y, r_I, d_I)$. If a check fails, the party aborts.

5.3 Blindness under Malicious Keys

Before proving that our scheme $\mathsf{BS}_{\mathsf{fis}}$ satisfies blindness under malicious keys, we state a lemma that will be used in our proof:

Lemma 1. *Let $\lambda \in \mathbb{N}$ and $N > 3$ be an odd natural number of bitlength polynomially large in λ. We consider $\mathbb{Z}_N^*$ and fix $\mathbb{G} = \langle g \rangle \subseteq \mathbb{Z}_N^*$ where $g \in \mathbb{Z}_N^*$. Given $e \leftarrow \mathcal{S}_e$ where $\mathcal{S}_e$ contains at least $\Omega(2^\lambda)$ primes, we have*

$$\Pr\left[\langle g^e \rangle \neq \mathbb{G} : e \leftarrow \mathcal{S}_e\right] \leq \mathrm{negl}(\lambda)$$

where the probability is taken over the choice of e.

We defer the proof to Appendix D.2. We now state the main theorem for blindness of $\mathsf{BS}_{\mathsf{fis}}$.

Theorem 3. *The scheme $\mathsf{BS}_{\mathsf{fis}}$ is* blind under malicious keys *following the* statistical adaptive soundness *of Π_{sub}, the* statistical adaptive soundness *of Π_{gen}, the* subversion zero-knowledge *property of Π_{fis}, and the* subversion zero-knowledge *property of Π_{ped}.*

We refer to Appendix D.3 for a detailed proof. We give a brief overview below.

Proof Overview. In the proof we use a sequence of games to transition from the blindness game as in Definition 12 with $\mathsf{coin} = 0$ to the blindness game with $\mathsf{coin} = 1$. To achieve this, we first employ the subversion zero-knowledge property of Π_{fis} as well as the adaptive soundness of Π_{sub} and Π_{gen} to switch to simulating the proofs π_{fis}. This allows us to change the commitment c_I of the signature on m_0 to a commitment to 0 which makes the signature independent of the signing session's exponents e and a. We then to turn to exchanging the CRS of Π_{ped} to a simulated one along with simulating the proof π_{ped} using the subversion zero-knowledge property of Π_{ped}. We also rule out that the signer gave us a key with $\langle h_2 \rangle \neq \langle g \rangle$ via the adaptive soundness of Π_{gen} as otherwise the Pedersen commitment would not be perfectly hiding. The previous game hop allows us to switch to a uniformly random Pedersen commitment $c \leftarrow \langle g \rangle$ for the session where m_0 is getting signed. After the Pedersen commitment is independent of the message, we also switch the commitment c_Z to be independent of the message using the hiding property of C_{Z}. We then use the an analogous series of games in the other direction to make end up with the real game for $\mathsf{coin} = 1$.

5.4 One-more Unforgeability

Theorem 4. *If the strong RSA problem is hard, H, $\mathsf{H}_{\mathbb{P}}$, $\mathsf{H}_{\mathsf{urs}}$, and H_{pp} are random oracles, Π_{ped} is a NIZK with Partial Online-Extractability, C_{Z} is a perfectly binding commitment scheme, PRF is a pseudo-random function, Π_{sub} is a NIZK with CRS indistinguishability and subversion zero-knowledge, Π_{fis} is a NIZK with adaptive knowledge-soundness, and $\mathsf{C}_{\mathsf{RInt}}$ is a perfectly binding integer commitment scheme then $\mathsf{BS}_{\mathsf{fis}}$ is secure against one-more-unforgeability.*

Proof Overview. For one-more unforgeability, we want to use similar techniques to generate signatures and solve the strong RSA problem as the scheme in Section 3.1. To do this, we need to do a hybrid argument to transition to a game where

- The reduction can online-extract the hash of the message $\overline{m}$ and the random blinding factor r used to generate the blinded message c. This we achieve by switching to the CRS that allows for extraction and by introducing extraction in a game.
- The reduction can be sure that in signing queries, the adversary uses an exponent e for which the reduction has trapdoored its verification key. This we achieve through programming the hash oracle $\mathsf{H}_{\mathbb{P}}$ accordingly (as well as through online-extraction).
- The reduction needs to be able to obtain an actual fresh signature (like in the EUF-CMA game for the adapted Fischlin scheme from Section 3.1). This we achieve by applying the knowledge extractor of Π_{fis}.
- We need to be sure that the extracted signature is independent of the various signature simulation modes employed by the reduction (i.e. the choices of b, b', j). This is provided by employing a perfectly binding commitment to contain the signature.
- We make additional game hops to rule out corner cases such as collisions in the hash functions.

The proof proceeds as a series of games to apply the changes described above and then we describe the reduction that solves the strong RSA problem (which roughly follows the security proof of $\mathsf{S}_{\mathsf{fis}}$). We refer to Appendix D.4 for a detailed proof.

5.5 Instantiation

We instantiate the primitives from Section 5.1 required for our blind signature $\mathsf{BS}_{\mathsf{fis}}$ as follows. For $\mathsf{C}_{\mathsf{RInt}}$, we use our construction from Section 4.2 which admits efficient opening proofs in zero-knoweldge. For Π_{gen}, we use the construction from Appendix C.2 and for the PRF, an arbitrary choice is sufficient. It remains to instantiate Π_{fis} and Π_{ped}. Our constructions are technically involved. We refer to Section 1.2 for a brief overview. For detailed constructions, we refer to Appendix E.

For our instantiation, we choose a standard RSA modulus of size 3072 bit for $\lambda = 128$. In total, we obtain blind signatures secure under DDH and sRSA of size 4.28 KB with 62.19 KB communication. We remark that Π_{sub} is the largest overhead (51.216 KB) in communication. A more efficient subgroup membership proof over $\mathbb{Z}_N$ with subversion soundness would heavily reduce the communication overhead.

References

1. Abe, M.: A secure three-move blind signature scheme for polynomially many signatures. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 136–151. Springer, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44987-6_9
2. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. Journal of Cryptology **29**(2), 363–421 (Apr 2016). https://doi.org/10.1007/s00145-014-9196-7
3. Abe, M., Jutla, C.S., Ohkubo, M., Roy, A.: Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 627–656. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_21
4. Abe, M., Ohkubo, M.: A framework for universally composable non-committing blind signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 435–450. Springer, Heidelberg (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_26
5. Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 271–286. Springer, Heidelberg (Aug 2000). https://doi.org/10.1007/3-540-44598-6_17
6. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Practical, round-optimal lattice-based blind signatures. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 39–53. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3560650
7. Amjad, G., Yeo, K., Yung, M.: Rsa blind signatures with public metadata. Cryptology ePrint Archive, Paper 2023/1199 (2023), https://eprint.iacr.org/2023/1199, https://eprint.iacr.org/2023/1199
8. Attema, T., Cramer, R.: Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 513–543. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_18
9. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 113–142. Springer, Heidelberg (Nov 2022). https://doi.org/10.1007/978-3-031-22318-1_5
10. Barreto, P.S., Lynn, B., Scott, M.: Constructing elliptic curves with prescribed embedding degrees. In: Security in Communication Networks: Third International Conference, SCN 2002 Amalfi, Italy, September 11–13, 2002 Revised Papers 3. pp. 257–267. Springer (2003)
11. Bellare, M., Fuchsbauer, G., Scafuro, A.: NIZKs with an untrusted CRS: Security in the face of parameter subversion. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 777–804. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_26
12. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. Journal of Cryptology **16**(3), 185–215 (Jun 2003). https://doi.org/10.1007/s00145-002-0120-1
13. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 390–399. ACM Press (Oct / Nov 2006). https://doi.org/10.1145/1180405.1180453
14. Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. In: Pernul, G., Ryan, P.Y.A., Weippl, E.R. (eds.) ESORICS 2015, Part I. LNCS, vol. 9326, pp. 305–325. Springer, Heidelberg (Sep 2015). https://doi.org/10.1007/978-3-319-24174-6_16
15. Benhamouda, F., Lepoint, T., Loss, J., Orrù, M., Raykova, M.: On the (in)security of ROS. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 33–53. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-77870-5_2

16. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Signatures on randomizable ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 403–422. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19379-8_25
17. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Short blind signatures. Journal of computer security **21**(5), 627–661 (2013)
18. Blazy, O., Pointcheval, D., Vergnaud, D.: Compact round-optimal partially-blind signatures. In: Visconti, I., Prisco, R.D. (eds.) SCN 12. LNCS, vol. 7485, pp. 95–112. Springer, Heidelberg (Sep 2012). https://doi.org/10.1007/978-3-642-32928-9_6
19. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (Jan 2003). https://doi.org/10.1007/3-540-36288-6_3
20. Brands, S.: Untraceable off-line cash in wallets with observers (extended abstract). In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2_26
21. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy. pp. 315–334. IEEE Computer Society Press (May 2018). https://doi.org/10.1109/SP.2018.00020
22. Buser, M., Dowsley, R., Esgin, M., Gritti, C., Kasra Kermanshahi, S., Kuchta, V., Legrow, J., Liu, J., Phan, R., Sakzad, A., Steinfeld, R., Yu, J.: A survey on exotic signatures for post-quantum blockchain: Challenges and research directions. ACM Comput. Surv. **55**(12) (mar 2023). https://doi.org/10.1145/3572771, https://doi.org/10.1145/3572771
23. Chairattana-Apirom, R., Hanzlik, L., Loss, J., Lysyanskaya, A., Wagner, B.: PI-cut-choo and friends: Compact blind signatures via parallel instance cut-and-choose and more. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part III. LNCS, vol. 13509, pp. 3–31. Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/978-3-031-15982-4_1
24. Chairattana-Apirom, R., Tessaro, S., Zhu, C.: Pairing-free blind signatures from cdh assumptions. Cryptology ePrint Archive, Paper 2023/1780 (2023), https://eprint.iacr.org/2023/1780, https://eprint.iacr.org/2023/1780
25. Chator, A., Green, M., Tiwari, P.R.: Sok: Privacy-preserving signatures. Cryptology ePrint Archive, Paper 2023/1039 (2023), https://eprint.iacr.org/2023/1039, https://eprint.iacr.org/2023/1039
26. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO'82. pp. 199–203. Plenum Press, New York, USA (1982)
27. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM **28**(10), 1030–1044 (oct 1985). https://doi.org/10.1145/4372.4373, https://doi.org/10.1145/4372.4373
28. Chaum, D.: Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Günther, C.G. (ed.) EUROCRYPT'88. LNCS, vol. 330, pp. 177–182. Springer, Heidelberg (May 1988). https://doi.org/10.1007/3-540-45961-8_15
29. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 319–327. Springer, Heidelberg (Aug 1990). https://doi.org/10.1007/0-387-34799-2_25
30. Couteau, G., Goudarzi, D., Klooß, M., Reichle, M.: Sharp: Short relaxed range proofs. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 609–622. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3560628
31. Couteau, G., Klooß, M., Lin, H., Reichle, M.: Efficient range proofs with transparent setup from bounded integer commitments. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part III. LNCS, vol. 12698, pp. 247–277. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-77883-5_9
32. Couteau, G., Peters, T., Pointcheval, D.: Removing the strong RSA assumption from arguments over the integers. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 321–350. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56614-6_11
33. Cramer, R., Shoup, V.: Signature schemes based on the strong RSA assumption. In: Motiwalla, J., Tsudik, G. (eds.) ACM CCS 99. pp. 46–51. ACM Press (Nov 1999). https://doi.org/10.1145/319709.319716
34. Crites, E., Komlo, C., Maller, M., Tessaro, S., Zhu, C.: Snowblind: A threshold blind signature in pairing-free groups. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 710–742. Springer Nature Switzerland, Cham (2023)
35. del Pino, R., Katsumata, S.: A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 306–336. Springer, Heidelberg (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_11
36. Denis, F., Jacobs, F., Wood, C.A.: RSA Blind Signatures. Internet-Draft draft-irtf-cfrg-rsa-blind-signatures-02, Internet Engineering Task Force (Aug 2021), https://datatracker.ietf.org/doc/draft-irtf-cfrg-rsa-blind-signatures/02/, work in Progress

37. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
38. Fischlin, M.: The Cramer-Shoup strong-RSA signature scheme revisited. In: Desmedt, Y. (ed.) PKC 2003. LNCS, vol. 2567, pp. 116–129. Springer, Heidelberg (Jan 2003). https://doi.org/10.1007/3-540-36288-6_9
39. Fischlin, M.: Round-optimal composable blind signatures in the common reference string model. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 60–77. Springer, Heidelberg (Aug 2006). https://doi.org/10.1007/11818175_4
40. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 197–215. Springer, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_10
41. Fuchsbauer, G.: Subversion-zero-knowledge SNARKs. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part I. LNCS, vol. 10769, pp. 315–347. Springer, Heidelberg (Mar 2018). https://doi.org/10.1007/978-3-319-76578-5_11
42. Fuchsbauer, G., Hanser, C., Kamath, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model from weaker assumptions. In: Zikas, V., De Prisco, R. (eds.) SCN 16. LNCS, vol. 9841, pp. 391–408. Springer, Heidelberg (Aug / Sep 2016). https://doi.org/10.1007/978-3-319-44618-9_21
43. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 233–253. Springer, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_12
44. Fuchsbauer, G., Plouviez, A., Seurin, Y.: Blind schnorr signatures and signed ElGamal encryption in the algebraic group model. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 63–95. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45724-2_3
45. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Seberry, J., Zheng, Y. (eds.) AUSCRYPT'92. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (Dec 1993). https://doi.org/10.1007/3-540-57220-1_66
46. Garg, S., Gupta, D.: Efficient round optimal blind signatures. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 477–495. Springer, Heidelberg (May 2014). https://doi.org/10.1007/978-3-642-55220-5_27
47. Garg, S., Rao, V., Sahai, A., Schröder, D., Unruh, D.: Round optimal blind signatures. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 630–648. Springer, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_36
48. Ghadafi, E.: Efficient round-optimal blind signatures in the standard model. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 455–473. Springer, Heidelberg (Apr 2017)
49. Google: VPN Google One. https://one.google.com/about/vpn/howitworks
50. Hanzlik, L., Loss, J., Wagner, B.: Rai-choo! evolving blind signatures to the next level. EUROCRYPT 2023 (2023), https://eprint.iacr.org/2022/1350, https://eprint.iacr.org/2022/1350
51. Hauck, E., Kiltz, E., Loss, J.: A modular treatment of blind signatures from identification schemes. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 345–375. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17659-4_12
52. Hazay, C., Katz, J., Koo, C.Y., Lindell, Y.: Concurrently-secure blind signatures without random oracles or setup assumptions. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 323–341. Springer, Heidelberg (Feb 2007). https://doi.org/10.1007/978-3-540-70936-7_18
53. Heath-Brown, D.: The number of primes in a short interval. Journal für die reine und angewandte Mathematik **389**, 22–63 (1988), http://eudml.org/doc/153047
54. Heath-Brown, R.: The Differences Between Consecutive Primes, V. International Mathematics Research Notices **2021**(22), 17514–17562 (12 2019). https://doi.org/10.1093/imrn/rnz295, https://doi.org/10.1093/imrn/rnz295
55. Hendrickson, S., Iyengar, J., Pauly, T., Valdez, S., Wood, C.A.: Private Access Tokens. Internet-Draft draft-private-access-tokens-01, Internet Engineering Task Force (Oct 2021), https://datatracker.ietf.org/doc/draft-private-access-tokens/01/, work in Progress
56. Huxley, M.: On the difference between consecutive primes. Inventiones mathematicae **15**, 164–170 (1971/72), http://eudml.org/doc/142126
57. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (Aug 1997). https://doi.org/10.1007/BFb0052233
58. Kastner, J., Loss, J., Xu, J.: On pairing-free blind signature schemes in the algebraic group model. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part II. LNCS, vol. 13178, pp. 468–497. Springer, Heidelberg (Mar 2022). https://doi.org/10.1007/978-3-030-97131-1_16

59. Katsumata, S., Lai, Y.F., LeGrow, J.T., Qin, L.: Csi-otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 729–761. Springer Nature Switzerland, Cham (2023)
60. Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Round-optimal blind signatures in the plain model from classical and quantum standard assumptions. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 404–434. Springer, Heidelberg (Oct 2021). https://doi.org/10.1007/978-3-030-77870-5_15
61. Katsumata, S., Reichle, M., Sakai, Y.: Practical round-optimal blind signatures in the rom from standard assumptions. to appear in Asiacrypt (2023), https://eprint.iacr.org/2023/1447, https://eprint.iacr.org/2023/1447
62. Katz, J., Loss, J., Rosenberg, M.: Boosting the security of blind signature schemes. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part IV. LNCS, vol. 13093, pp. 468–492. Springer, Heidelberg (Dec 2021). https://doi.org/10.1007/978-3-030-92068-5_16
63. Lindell, Y.: Lower bounds and impossibility results for concurrent self composition. Journal of Cryptology **21**(2), 200–249 (Apr 2008). https://doi.org/10.1007/s00145-007-9015-5
64. Lysyanskaya, A.: Security analysis of rsa-bssa (2023), https://eprint.iacr.org/2022/895
65. Meiklejohn, S., Shacham, H., Freeman, D.M.: Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 519–538. Springer, Heidelberg (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_30
66. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (Aug 1993). https://doi.org/10.1007/3-540-48071-4_3
67. Okamoto, T., Ohta, K.: Universal electronic cash. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 324–337. Springer, Heidelberg (Aug 1992). https://doi.org/10.1007/3-540-46766-1_27
68. Pass, R.: Limits of provable security from standard assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 109–118. ACM Press (Jun 2011). https://doi.org/10.1145/1993636.1993652
69. Pointcheval, D.: Strengthened security for blind signatures. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 391–405. Springer, Heidelberg (May / Jun 1998). https://doi.org/10.1007/BFb0054141
70. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology **13**(3), 361–396 (Jun 2000). https://doi.org/10.1007/s001450010003
71. Schnorr, C.P.: Security of blind discrete log signatures against interactive attacks. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 01. LNCS, vol. 2229, pp. 1–12. Springer, Heidelberg (Nov 2001)
72. Seo, J.H., Cheon, J.H.: Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 133–150. Springer, Heidelberg (Mar 2012). https://doi.org/10.1007/978-3-642-28914-9_8
73. Tessaro, S., Zhu, C.: Short pairing-free blind signatures with exponential security. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 782–811. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_27
74. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (Aug 2002). https://doi.org/10.1007/3-540-45708-9_19
75. Nist announcess additional digital signature candidates for the pqc standardization process. https://csrc.nist.gov/projects/pqc-dig-sig (2023), accessed: 2023-10-06
76. mcl-wasm library for pairings. https://github.com/herumi/mcl-wasm (2023), accessed: 2023-10-02
77. PCM: Click fraud prevention and attribution sent to advertiser. https://webkit.org/blog/11940/pcm-click-fraud-prevention-and-attribution-sent-to-advertiser/ (2022), accessed: 2023-10-06
78. Supported ssh algorithms. https://privx.docs.ssh.com/docs/supported-ssh-key-exchange-algorithms (2022), accessed: 2023-10-06
79. Yi, X., Lam, K.Y.: A new blind ECDSA scheme for bitcoin transaction anonymity. In: Galbraith, S.D., Russello, G., Susilo, W., Gollmann, D., Kirda, E., Liang, Z. (eds.) ASIACCS 19. pp. 613–620. ACM Press (Jul 2019). https://doi.org/10.1145/3321705.3329816

A Full Preliminaries

A.1 Notation

Let $\lambda \in \mathbb{N}$ be the security parameter. A probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ runs in time polynomial in the (implicit) security parameter λ. We write $\mathsf{Time}(\mathcal{A})$ for the runtime of $\mathcal{A}$. A function $f(\lambda)$ is *negligible* in λ if it is $\mathcal{O}(\lambda^{-c})$ for every $c \in \mathbb{N}$. We write $f = \mathrm{negl}(\lambda)$ for short. Similarly, we write $f = \mathsf{poly}(\lambda)$ if $f(\lambda)$ is a polynomial with variable λ. If D is a probability

distribution, $x \leftarrow D$ means that x is sampled from D and if S is a set, $x \leftarrow S$ means that x is sampled uniformly and independently at random from S. We also write $|S|$ for the cardinality of set S. Further, we write $D_0 \overset{c}{\approx} D_1$ for distributions D_0, D_1, if for all PPT adversaries $\mathcal{A}$, we have $|\Pr[x_0 \leftarrow D_0 : \mathcal{A}(1^\lambda, x_0) = 1] - \Pr[x_1 \leftarrow D_1 : \mathcal{A}(1^\lambda, x_1) = 1]| = \mathsf{negl}(\lambda)$. Similarly, we write $D_0 \overset{s}{\approx} D_1$ if the above holds even for unbounded adversaries. For some PPT algorithm $\mathcal{A}$, we write $\mathcal{A}^{\mathcal{O}}$ if $\mathcal{A}$ has oracle access to the oracle $\mathcal{O}$. If $\mathcal{A}$ performs some check, and the check fails, we assume that $\mathcal{A}$ outputs $\perp$ immediately. Generally, we assume that adversaries are implicitly stateful.

We denote with $[n]$ the set $\{1, \ldots, n\}$ for $n \in \mathbb{N}$. We write $\mathbb{P}$ for the set of primes and $\mathbb{P}_I$ for the set of primes in the interval I. For some odd prime p, we use the representatives $\{-\frac{p-1}{2}, \cdots, \frac{p-1}{2}\}$ for $\mathbb{Z}_p$. For a group $\mathbb{G}$ we write $\mathsf{ord}(\mathbb{G})$ to denote the order of $\mathbb{G}$ and unless stated otherwise we write $\mathbb{G}$ with additive notation. We denote by $\mathsf{QR}_N = \{a \in \mathbb{Z}_N^* : \exists\, b \in \mathbb{Z}_N^*, b^2 \equiv a \bmod N\}$ the quadratic residues $\bmod N$. For some $N \in \mathbb{N}$, the group QR_N is a cyclic subgroup of $\mathbb{Z}_N^*$ and we denote by $\mathsf{Gen}(\mathsf{QR}_N)$ the set of generators of QR_N. We recall some properties of QR_N

Lemma 2 (Proposition 1, [32]). *Let $\lambda \in \mathbb{N}$ and $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$. Considering QR_N, the following holds:*

- *The group QR_N is cyclic of order $P'Q'$ where $P = 2P' + 1$ and $Q = 2Q' + 1$.*
- *$-1 \notin \mathsf{QR}_N$.*
- *Any square $h \in \mathsf{QR}_N$ has excatly four roots, among which there is exactly one square.*
- *For any element $h \in \mathsf{QR}_N$, finding roots of h is equivalent to factoring N.*
- *For $g, h \leftarrow \mathsf{QR}_N$, finding $a, b \in \mathbb{N} \setminus \{0\}$ such that $g^a \equiv h^b \bmod N$ is equivalent to factoring N.*
- *For any $e \in \mathbb{N}$ coprime with $\phi(N)$ and $y \in \mathbb{Z}_N^*$, finding $x, e' \in \mathbb{N}$ such that $x^e \equiv y^{e'} \bmod N$ is equivalent to finding an e-th root of y in $\mathbb{Z}_N^*$.*

A.2 Probability

Rejection Sampling. Let $V, L \in \mathbb{N}$. We define uniform rejection sampling for the interval $[0, V]$ with masking overhead L as in [30]. Let $v \in [0, V]$. To mask v additively with a mask μ via rejection sampling, perform the following steps.

1. Draw a random mask $\mu \leftarrow [0, (V+1)L]$.
2. Abort if $v + \mu \notin [V, (V+1)L]$.
3. Output $w = v + \mu$.

The value w is uniform over $[V, (V+1)L]$ conditioned on no abort and the abort probability is at most $1/L$.

Forking Lemma. We state here a version of Forking Lemma that fits our usage of it. The lemma was first introduced by Pointcheval and Stern [70] then generalized by Bellare and Neven [13].

Lemma 3 (Forking Lemma). *Let H be a set and let $F : H^q \to [q]$ be a possibly random function. For every $\vec{h} \in H^q$, let $E(\vec{h})$ be a probability event. The probability that when sampling k vectors $\vec{h}_1, \ldots, \vec{h}_k$ uniformly and independently at random (conditioned that vectors are identical on their first $F(\vec{h}_1)$ components), $E(\vec{h}_i)$ happens for all $i \in [k]$ and $F(\vec{h}_1) = F(\vec{h}_2) = \cdots = F(\vec{h}_k)$, is at least $\delta(E)^k / q^{k-1}$, where $\delta(E) \coloneqq \Pr[\vec{h} \leftarrow H^q : E(\vec{h})]$.*

A.3 Assumptions

Groups and RSA. Let GenG be a PPT algorithm that on input 1^λ and prime order p, outputs (a description of) a group $\mathbb{G} \leftarrow \mathsf{GenG}(1^\lambda)$ of order p. We generally use additive notations for prime order groups and capital letters for elements. Also, we assume that given the description, group operations and membership tests are efficient. We write $g \leftarrow \mathbb{G}$ for drawing elements from some group $\mathbb{G}$ at random. In the following, we assume that prime order groups are setup with GenG implicitly.

Let GenRSA be a PPT algorithm that on input 1^λ outputs $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$ such that $N = P \cdot Q$ with $P, Q \in \mathbb{P}$, where $P = 2P' + 1$ and $Q = 2Q' + 1$ are strong primes (*i.e.*, P', Q' are also primes). We assume that $P', Q' > 2^\lambda$.

Before recalling some standard hardness assumptions, let us recall the following well-known lemma.

Lemma 4. *Given $x, y \in \mathbb{Z}_N^*$ with $a, b \in \mathbb{Z}$ such that $x^a = y^b$ and $\gcd(a, b) = 1$, one can efficiently compute $\bar{x} \in \mathbb{Z}_N^*$ such that $\bar{x}^a = y$.*

Remark 2. We need the following well-known fact. Let $\mathbb{G}$ be a group and let $G \leftarrow \mathbb{G}$ be a random element from $\mathbb{G}$. Let $S \in \mathbb{N}$. We consider the problem of distinguishing zG, where $z \leftarrow [0, S]$, from $\tilde{z}G$ where $\tilde{z} \leftarrow \mathbb{Z}_{\mathsf{ord}(G)}$.

If the order p of the group $\mathbb{G}$ is known, then the distinguishing probability is 0 for $S = p - 1$. If only an upper bound U on the order is known, then the distinguishing probability is upper bounded by $1/L$ for $S = L \cdot U$. For the latter, we set $L = 2^\lambda$ throughout to obtain negligible distinguishing probability.

Next, we recall the definition of a relaxed DLOG-relation from [30] (for the hidden order group QR_N).

Definition 3 ((D, ℓ)-relaxed DLOG-relation). *Let $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$, $D, \ell \in \mathbb{N}$, and $\vec{g} = (g_0, \ldots, g_\ell) \in \mathsf{QR}_N^{\ell+1}$. Define the (D, ℓ)-*relaxed DLOG relation with regards to $\vec{g}$ *as*

$$\mathsf{R}_{D,\ell}(\vec{g}) = \left\{ (c, d, \{x_i\}_{i=1}^\ell) \;\middle|\; \begin{array}{c} c^d = \prod_{i=0}^\ell g_i^{x_i} \;\wedge\; \exists i \colon \frac{x_i}{d} \notin \mathbb{Z} \\ \wedge\, d \in [0, D] \;\wedge\; x_i \in \mathbb{Z} \end{array} \right\}$$

We define the advantage of $\mathcal{A}$ against the hardness of the (D, ℓ)-relaxed DLOG-relation as

$$\mathsf{Adv}^{\text{rel-dlog}}_{(D,\ell),\mathcal{A}}(\lambda) \coloneqq \Pr\left[\begin{array}{c} (N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda); g_0, \cdots, g_\ell \leftarrow \mathsf{Gen}(\mathsf{QR}_N); \\ (c, d, x_0, \ldots, x_\ell) \leftarrow \mathcal{A}(N, g_0, \ldots, g_\ell) \colon \\ (c, d, x_0, \ldots, x_\ell) \in \mathsf{R}_{D,\ell}(\vec{g}) \end{array} \right].$$

The following lemma is a simplification of Lemma A.13 of [30] sufficient for our purpose. Note that $\mathsf{ord}(\mathsf{QR}_N) = P'Q'$ and we assume that $P', Q' > 2^{\lambda+1}$.

Lemma 5. *For all $D \leq 2^{\lambda+1}$ and every PPT adversary $\mathcal{A}$ we have that $\mathsf{Adv}^{\text{rel-dlog}}_{(D,\ell),\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda)$ under the strong RSA assumption.*

Definition 4 (Decisional Diffie-Hellman). *In a cyclic group $\mathbb{G}$ of prime order p, which are set up w.r.t a security parameter $\lambda \in \mathbb{N}$, the* Decisional Diffie-Hellman (DDH) *assumption in $\mathbb{G}$ holds if for all PPT adversary $\mathcal{A}$ the advantage*

$$\begin{aligned} &\Pr\left[G \leftarrow \mathbb{G}; a, b \leftarrow \mathbb{Z}_p \colon \mathcal{A}(G, aG, bG, abG) = 1\right] \\ -\,&\Pr\left[G \leftarrow \mathbb{G}; a, b, c \leftarrow \mathbb{Z}_p \colon \mathcal{A}(G, aG, bG, cG) = 1\right] \\ =\,&\mathrm{negl}(\lambda) \end{aligned}$$

is negligible in λ.

Definition 5 (Strong RSA). *Let $\lambda \in \mathbb{N}$ and $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$. The* strong RSA (sRSA) *assumption holds if for all PPT $\mathcal{A}$ the advantage*

$$\mathsf{Adv}^{\text{s-rsa}}_{\mathcal{A}}(\lambda) \coloneqq \Pr\left[\begin{array}{c} (N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda); y \leftarrow \mathbb{Z}_N^* \\ (e, z) \leftarrow \mathcal{A}(N, y) \colon z^e \equiv y \bmod N \end{array} \right].$$

is negligible in λ.

A.4 Explaining Random Group Elements as Random Strings

For our framework, we require commitments with uniform public parameters pp. For readability, we allow pp (and also uniform random strings urs of NIZKs) to contain (uniform) group elements g of prime-order groups $\mathbb{G}$ with known order p. This is without loss of generality because with *explainable sampling*, we can explain $g \leftarrow \mathbb{G}$ as a random bitstring.

A.5 Commitment Scheme

A *commitment scheme* is a PPT algorithm $\mathsf{C} = \mathsf{C.Commit}$ such that

- $\mathsf{C.Setup}(1^\lambda)$: generates the public parameters pp,
- $\mathsf{C.Commit}(\mathsf{pp}, m)$: given the public parameters pp, message $m \in \mathcal{C}_{\mathsf{msg}}$, computes a commitment $c \in \mathcal{C}_{\mathsf{com}}$ with opening randomness d, and outputs the pair (c, d),
- $\mathsf{Verify}(\mathsf{pp}, c, m, d)$: given the public parameters pp, message $m \in \mathcal{C}_{\mathsf{msg}}$, and opening randomness d, outputs a bit $b \in \{0,1\}$ which depends on the validity of the opening (m, d) with respect to the commitment c.

Here, $\mathcal{C}_{\mathsf{msg}}$, $\mathcal{C}_{\mathsf{rnd}}$, $\mathcal{C}_{\mathsf{com}}$, are message, randomness, and commitment spaces, respectively. If the public parameters are uniform or explainable *as per* Appendix A.4 (*i.e.*, Setup outputs some $\mathsf{pp} \leftarrow \{0,1\}^\ell$ for $\ell \in \mathbb{N}$) we omit Setup without loss of generality.

Below, we define the correctness, hiding and binding properties of a commitment scheme.

Definition 6 (Correctness). *A commitment scheme is* correct*, if for all* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), m \in \mathcal{C}_{\mathsf{msg}}, r \in \mathcal{C}_{\mathsf{rnd}}, (c,d) \leftarrow \mathsf{Commit}(\mathsf{pp}, m; r)$*, it holds that* $\mathsf{Verify}(\mathsf{pp}, c, m, d) = 1$*.*

Definition 7 (Hiding). *A commitment scheme is* hiding *if for any PPT adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathrm{hide}}_{\mathcal{A}}(\lambda) = \left| \Pr \begin{bmatrix} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp}), \\ m_0, m_1 \in \mathcal{C}_{\mathsf{msg}}, \mathsf{coin} \leftarrow \{0,1\} \\ (c,d) \leftarrow \mathsf{Commit}(\mathsf{pp}, m_{\mathsf{coin}}), \end{bmatrix} : \mathsf{coin} = \mathcal{A}(c) \right] - \frac{1}{2} \right| = \mathrm{negl}(\lambda).$$

Definition 8 (Binding). *A commitment scheme is* binding *if for any PPT adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathrm{bind}}_{\mathcal{A}}(\lambda) = \Pr \begin{bmatrix} \mathsf{pp} \leftarrow \{0,1\}^{\ell_c}, \\ (c, m_0, m_1, d_0, d_1) \leftarrow \mathcal{A}(\mathsf{pp}), : m_0 \neq m_1 \in \mathcal{C}_{\mathsf{msg}} \\ \mathsf{Verify}(\mathsf{pp}, c, m_b, d_b) = 1, b \in \{0,1\} \end{bmatrix} = \mathrm{negl}(\lambda).$$

Remark 3. A commitment scheme is said to be *perfectly binding* if for any (possibly unbounded) $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathrm{bind}}_{\mathcal{A}}(\lambda) = 0$.

A.6 Signature Scheme

A signature scheme is a tuple of PPT algorithms $\mathsf{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ such that

- $\mathsf{KeyGen}(1^\lambda)$: generates a verification key vk and a signing key sk,
- $\mathsf{Sign}(\mathsf{sk}, m)$: given a signing key sk and a message $m \in \mathcal{S}_{\mathsf{msg}}$, *deterministically* outputs a signature σ,
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$: given a verification key pk and a signature σ on message m, *deterministically* outputs a bit $b \in \{0,1\}$.

Here, $\mathcal{S}_{\mathsf{msg}}$ is the message space. We define the standard notion of correctness and `euf-cma` security

Definition 9 (Correctness). *A signature scheme is* correct*, if for all* $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*,* $m \in \mathcal{S}_{\mathsf{msg}}$*, and* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$*, it holds that* $\mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1$*.*

Definition 10 (EUF-CMA). *A signature scheme is* `euf-cma` *if for any PPT adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\mathrm{euf}}_{\mathcal{A}}(\lambda) = \Pr \left[\begin{bmatrix} (\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk}) \end{bmatrix} : m \notin L \wedge \mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1 \right] = \mathrm{negl}(\lambda),$$

where L *is the list of messages* $\mathcal{A}$ *queried to the* Sign*-oracle.*

A.7 Blind Signature Scheme

A blind signature scheme is a tuple of PPT algorithms $\mathsf{PBS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ such that

- $\mathsf{KG}(1^\lambda)$: generates the verification key bvk and signing key bsk,
- $\mathsf{User}(\mathsf{bvk}, m)$: given verification key bvk and message $m \in \mathcal{BS}_{msg}$, outputs a first message ρ_1 and a state st,
- $\mathsf{Signer}(\mathsf{bsk}, \rho_1)$: given signing key bsk and first message ρ_1, outputs a second message ρ_2,
- $\mathsf{Derive}(\mathsf{st}, \rho_2)$: given state st and second message ρ_2, outputs a signature σ,
- $\mathsf{Verify}(\mathsf{bvk}, m, \sigma)$: given verification key bvk and signature σ on message $m \in \mathcal{BS}_{msg}$, outputs a bit $b \in \{0,1\}$.

Here, $\mathcal{BS}_{msg}$ is the message spaces. We consider the standard security notions for blind signatures [57]. Below, we define correctness, blindness under *malicious keys*, and one-more unforgeability of a blind signature scheme. Moreover, we assume the state is kept implicit in the following for better readability.

Definition 11 (Correctness). *A blind signature scheme is* correct*, if for all messages $m \in \mathcal{BS}_{msg}$, $(\mathsf{bvk}, \mathsf{bsk}) \leftarrow \mathsf{KG}(1^\lambda), (\rho_1, \mathsf{st}) \leftarrow \mathsf{User}(\mathsf{bvk}, m), \rho_2 \leftarrow \mathsf{Signer}(\mathsf{bsk}, \rho_1), \sigma \leftarrow \mathsf{Derive}(\mathsf{st}, \rho_2)$, it holds that $\mathsf{Verify}(\mathsf{bvk}, m, \sigma) = 1$.*

Definition 12 (Blindness Under Malicious Keys). *A blind signature scheme is* blind under malicious keys *if for any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A}}(\lambda) = \left| \Pr \left[\begin{array}{l} (\mathsf{bvk}, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda),\ \mathsf{coin} \leftarrow \{0,1\}, \\ (\rho_{1,b}, \mathsf{st}_b) \leftarrow \mathsf{User}(\mathsf{bvk}, m_b) \text{ for } b \in \{0,1\}, \\ (\rho_{2,\mathsf{coin}}, \rho_{2,1-\mathsf{coin}}) \leftarrow \mathcal{A}(\rho_{1,\mathsf{coin}}, \rho_{1,1-\mathsf{coin}}), \\ \sigma_b \leftarrow \mathsf{Derive}(\mathsf{st}_b, \rho_{2,b}) \text{ for } b \in \{0,1\}, \\ \text{if } \exists b \text{ s.t. } \mathsf{Verify}(\mathsf{bvk}, m_b, \sigma_b) = 0\text{:} \\ \quad \text{then } \sigma_0 = \sigma_1 = \bot, \end{array} : \mathsf{coin} = \mathcal{A}(\sigma_0, \sigma_1) \right] - \frac{1}{2} \right| = \mathrm{negl}(\lambda).$$

Definition 13 (One-more Unforgeability). *A blind signature scheme is* one-more unforgeable *if for any $Q = \mathsf{poly}(\lambda)$ and PPT adversary $\mathcal{A}$ that makes at most Q signing queries, we have*

$$\mathsf{Adv}^{\mathrm{omuf}}_{\mathcal{A}}(\lambda) = \Pr \left[\begin{array}{l} (\mathsf{bvk}, \mathsf{bsk}) \leftarrow \mathsf{KG}(1^\lambda) \\ \{(m_i, \sigma_i)\}_{i \in [Q+1]} \leftarrow \mathcal{A}^{\mathsf{Signer}(\mathsf{bsk}, \cdot)}(\mathsf{bvk}) \end{array} : \begin{array}{l} \forall i \neq j \in [Q+1] : m_i \neq m_j \\ \wedge\ \mathsf{Verify}(\mathsf{bvk}, m_i, \sigma_i) = 1 \end{array} \right] = \mathrm{negl}(\lambda).$$

A.8 Σ-Protocol

Let R be an NP relation with statements x and witnesses w. We denote by $\mathscr{L}_{\mathsf{R}} = \{x \mid \exists w \text{ s.t. } (x,w) \in \mathsf{R}\}$ the language induced by R. A Σ-protocol for an NP relation R for language $\mathscr{L}_{\mathsf{R}}$ is a tuple of PPT algorithms $\Sigma = (\mathsf{Init}, \mathsf{Chall}, \mathsf{Resp}, \mathsf{Verify})$ such that

- $\mathsf{Init}(x, w)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, and a witness w such that $(x,w) \in \mathsf{R}$, outputs a first flow message (*i.e.*, commitment) α and a state st, where we assume st includes x, w,
- $\mathsf{Chall}()$: samples a challenge $\beta \leftarrow \mathcal{CH}$ (without taking any input),
- $\mathsf{Resp}(\mathsf{st}, \beta)$: given a state st and a challenge $\beta \in \mathcal{CH}$, outputs a third flow message (*i.e.*, response) γ,
- $\mathsf{Verify}(x, \alpha, \beta, \gamma)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, a commitment α, a challenge $\beta \in \mathcal{CH}$, and a response γ, outputs a bit $b \in \{0,1\}$.

Definition 14 (Correctness). *A Σ-protocol is* correct*, if for all $(x,w) \in \mathsf{R}$, $(\alpha, \mathsf{st}) \leftarrow \mathsf{Init}(x,w)$, $\beta \in \mathcal{CH}$, and $\gamma \leftarrow \mathsf{Resp}(\mathsf{st}, \beta)$, it holds that $\mathsf{Verify}(x, \alpha, \beta, \gamma) = 1$.*

Definition 15 (High Min-Entropy). *A Σ-protocol has* high min-entropy *if for all $(x,w) \in \mathsf{R}$ and (possibly unbounded) adversary $\mathcal{A}$, it holds that*

$$\Pr[(\alpha, \mathsf{st}) \leftarrow \mathsf{Init}(x,w), \alpha' \leftarrow \mathcal{A}(1^\lambda) : \alpha = \alpha'] = \mathrm{negl}(\lambda).$$

Definition 16 (HVZK). *A Σ-protocol is* honest-verifier zero-knowledge *(HVZK), if there exists a PPT zero-knowledge simulator* Sim *such that the distributions of* $\mathsf{Sim}(x, \beta)$ *and the honestly generated transcript with* Init *initialized with* (x, w) *are computationally indistinguishable for any* $x \in \mathscr{L}_{\mathsf{R}}$*, and* $\beta \in \mathcal{CH}$*, where the honest execution is conditioned on* β *being used as the challenge.*

Definition 17 (k-Special Soundness). *A Σ-protocol is* k-special sound*, if there exists a* deterministic *PT extractor* Ext *such that given* k *valid transcripts* $\{(\alpha, \beta_i, \gamma_i)\}_{i\in[k]}$ *for statement* x *with pairwise distinct challenges* $(\beta_i)_i$*, outputs a witness* w *such that* $(x, w) \in \mathsf{R}$*.*

A.9 Non-Interactive Zero Knowledge

Let $\mathcal{URS} = \{0,1\}^\ell$ be a set of *uniform random strings* for some $\ell \in \mathbb{N}$ and $\mathcal{SRS}$ be some set of *structured random strings* with efficient membership test [17]. An NIZK for a relation R with common reference string space $\mathcal{CRS} = \mathcal{SRS} \times \mathcal{URS}$ is a tuple of PPT algorithms $(\mathsf{GenCRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$, where the latter two are oracle-calling, such that:

- $\mathsf{GenCRS}(1^\lambda)$: outputs a structured reference string $\mathsf{srs} \in \mathcal{SRS}$,
- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$: receives a $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs}) \in \mathcal{CRS}$, a statement x and a witness w, and outputs a proof π,
- $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi)$: receives a $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs}) \in \mathcal{CRS}$, a statement x and a proof π, and outputs a bit $b \in \{0,1\}$.

We recall that $\mathscr{L}_{\mathsf{R}} = \{x \mid \exists w \ : \ (x, w) \in \mathsf{R}\}$ denotes the language induced by R. If there is no crs needed, *i.e.* $\mathcal{CRS} = \varnothing$, we then omit crs as an input to Prove and Verify.

Definition 18 (Correctness). *An* NIZK *is* correct *if for any* $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ *with* $\mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda)$ *and* $\mathsf{urs} \leftarrow \mathcal{URS}$*,* $(x, w) \in \mathsf{R}$*, and* $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$*, it holds that* $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi) = 1$*.*

Definition 19 (Zero-Knowledge). *An* NIZK *is* zero-knowledge *(*ZK*) if there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{crs}}, \mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_\pi)$ *such that for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathrm{zk}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\begin{array}{c} \mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda), \\ \mathsf{crs} = (\mathsf{srs}, \mathsf{urs}), \\ \mathcal{A}^{\mathsf{H},\mathcal{P}}(\mathsf{crs}) = 1 \end{array}\right] - \Pr\left[\begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Sim}_{\mathsf{crs}}(1^\lambda), \\ \mathsf{crs} = (\mathsf{srs}, \mathsf{urs}), \\ \mathcal{A}^{\mathsf{Sim}_{\mathsf{H}},\mathcal{S}}(\mathsf{crs}) = 1 \end{array}\right] \right| = \mathrm{negl}(\lambda),$$

where $\mathcal{P}$ *and* $\mathcal{S}$ *are oracles that on input* (x, w) *return* $\perp$ *if* $(x, w) \notin \mathsf{R}$*, and else output* $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ *or* $\mathsf{Sim}_\pi(\mathsf{crs}, x)$ *respectively. Note that the probability is taken over the randomness of* Sim *and* $\mathcal{A}$*, and the random choices of* H *and* urs*. Also,* $\mathsf{Sim}_{\mathsf{crs}}, \mathsf{Sim}_{\mathsf{H}}$ *and* Sim_π *have a shared state.*

We also define a notion of *subversion zero-knowledge*, inspired by the notion introduced in [11]. Informally, it guarantees that zero-knowledge holds even for a malicious crs.

Definition 20 (Subversion Zero-Knowledge). *An* NIZK *is* subversion zero-knowledge *(*$\mathsf{Sub\text{-}ZK}$*) if there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_\pi)$ *such that for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\text{sub-zk}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[\begin{array}{c} (\mathsf{srs}, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{urs}), \\ \mathsf{crs} = (\mathsf{srs}, \mathsf{urs}), \\ \mathcal{A}^{\mathsf{H},\mathcal{P}}(\mathsf{st}) = 1 \wedge \mathsf{srs} \in \mathcal{SRS} \end{array}\right] - \Pr\left[\begin{array}{c} (\mathsf{srs}, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Sim}_{\mathsf{H}}}(\mathsf{urs}), \\ \mathsf{crs} = (\mathsf{srs}, \mathsf{urs}), \\ \mathcal{A}^{\mathsf{Sim}_{\mathsf{H}},\mathcal{S}}(\mathsf{st}) = 1 \wedge \mathsf{srs} \in \mathcal{SRS} \end{array}\right] \right| = \mathrm{negl}(\lambda),$$

where $\mathcal{P}$ *and* $\mathcal{S}$ *are oracles that on input* (x, w) *return* $\perp$ *if* $(x, w) \notin \mathsf{R}$*, and else output* $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ *or* $\mathsf{Sim}_\pi(\mathsf{crs}, x)$*, respectively. Note that the probability is taken over the randomness of* Sim *and* $\mathcal{A}$*, and the random choices of* H *and* urs*. Also, both* $\mathsf{Sim}_{\mathsf{H}}$ *and* Sim_π *have a shared state.*

We define different notions of soundness. We remark that the soundness relation $\widetilde{\mathsf{R}}$ can be different from the (correctness) relation R. If $\widetilde{\mathsf{R}}$ is not explicitly defined, we implicitly set $\widetilde{\mathsf{R}} = \mathsf{R}$.

Definition 21 (Adaptive Knowledge Soundness). *An* NIZK *is* adaptively knowledge sound *for relation* $\widetilde{\mathsf{R}}$ *if there exists PPT simulator* SimCRS *and extractor* Ext *such that*

[17] This membership test is required for our definition of subversion zero-knowledge. Note that in general it is difficult to check that some srs was generated via GenCRS. (We allow that $\mathcal{SRS}$ is not equal to the output space of GenCRS.)

CRS Indistinguishability. *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathrm{crs}}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\begin{matrix}\mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda), \mathsf{urs} \leftarrow \mathcal{URS}, \\ \mathsf{crs} = (\mathsf{srs}, \mathsf{urs}) : \mathcal{A}^{\mathsf{H}}(\mathsf{crs}) = 1\end{matrix}\right] - \Pr\left[\begin{matrix}(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda) : \\ \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}}) = 1\end{matrix}\right]\right| = \mathrm{negl}(\lambda),$$

Knowledge Soundness. *There exists positive polynomials $p_{\mathsf{T}}, p_{\mathsf{P}}$ such that given oracle access to any PPT adversary $\mathcal{A}$ (with explicit random tape ρ) that makes $Q_H = \mathsf{poly}(\lambda)$ random oracle queries with*

$$\Pr[(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda), (x, \pi) \leftarrow \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}}; \rho) : \mathsf{Verify}^{\mathsf{H}}(\overline{\mathsf{crs}}, x, \pi) = 1] \geq \mu(\lambda),$$

we have

$$\Pr\left[\begin{matrix}(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda), \\ (x, \pi) \leftarrow \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}}; \rho), \\ w \leftarrow \mathsf{Ext}(\overline{\mathsf{crs}}, \mathsf{td}, x, \pi, \rho, \vec{h})\end{matrix} : (x, w) \in \widetilde{\mathsf{R}}\right] \geq \frac{\mu(\lambda) - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)},$$

where $\vec{h}$ are the outputs of H, and the probability is over the random tape ρ of $\mathcal{A}$, the random tape of SimCRS, and the random choices of H. Also, we require that the runtime of Ext is bounded by $p_{\mathsf{T}}(\lambda, Q_H) \cdot \mathsf{Time}(\mathcal{A})$.

We also adapt the standard notion of online-extractability in two ways. Instead of embedding the online-extraction trapdoor td into crs, we allow that the extractor embeds it into specific parts of statement. Also, we relax the requirements in the sense that only a partial witness w_1 is extracted. For extraction, we require that there exists a witness w_0 such that $(x, (w_0, w_1)) \in \widetilde{\mathsf{R}}$.

Definition 22 (Partial Online Extractability). *An* NIZK *is* partially online-extractable *for relation $\widetilde{\mathsf{R}}$ with statements $x = (x_0, x_1)$ and witnesses $w = (w_0, w_1)$, where $w_0 \in W_0$ and $x_0 \in X_0$ for some sets W_0, X_0, if for all PPT adversaries $\mathcal{A}$, there exists a stateful PPT extractor $\mathsf{Ext} = (\mathsf{Ext}_1, \mathsf{Ext}_2)$, such that*

1. *$x_0 \sim \mathcal{U}_{X_0}$ is distributed uniform over X_0 for $(x_0, \mathsf{td}) \leftarrow \mathsf{Ext}_1(1^\lambda)$ and*
2. *there exists positive polynomials $p_{\mathsf{T}}, p_{\mathsf{P}}$ such that for any $Q_H = \mathsf{poly}(\lambda)$ and PPT adversary $\mathcal{A}$ that makes at most Q_H random oracle queries with*

$$\Pr\left[\begin{matrix}(x_0, \mathsf{td}) \leftarrow \mathsf{Ext}_1(1^\lambda), \mathsf{crs} \leftarrow \mathsf{GenCRS}(1^\lambda), \\ \{(x_{1,i}, \pi_i)\}_{i \in [Q_S]} \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{crs}, x_0), x_i \leftarrow (x_0, x_{1,i}) : \\ \forall i \in [Q_S] : \mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x_i, \pi_i) = 1\end{matrix}\right] \geq \mu(\lambda),$$

it holds that

$$\Pr\left[\begin{matrix}(x_0, \mathsf{td}) \leftarrow \mathsf{Ext}_1(1^\lambda), \mathsf{crs} \leftarrow \mathsf{GenCRS}(1^\lambda), \\ \{(x_{1,i}, \pi_i)\}_{i \in [Q_S]} \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{crs}, x_0), x_i \leftarrow (x_0, x_{1,i}) \\ \{w_{1,i} \leftarrow \mathsf{Ext}_2(\mathsf{crs}, \mathsf{td}, x_i, \pi_i)\}_{i \in [Q_S]} : \\ \forall i \in [Q_S]\ \exists w_{0,i} \in W_0 : (x_i, (w_{0,i}, w_{1,i})) \in \widetilde{\mathsf{R}} \\ \wedge\ \mathsf{Verify}^{\mathsf{H}}(\overline{\mathsf{crs}}, x_i, \pi_i) = 1\end{matrix}\right] \geq \frac{\mu(\lambda) - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)},$$

where the runtime of Ext is upper bounded by $p_{\mathsf{T}}(\lambda, Q_H) \cdot \mathsf{Time}(\mathcal{A})$.

Adaptive Subversion Soundness. We also define adaptive subversion soundness, where we allow that srs can be maliciously set up by an adversary. Note that this notion does not require an extractor for the witness.

Definition 23 (Statistical Adaptive Subversion Soundness). *An* NIZK *is* (statistically) adaptively sound *for relation $\widetilde{\mathsf{R}}$ inducing a language $\mathscr{L}_{\widetilde{\mathsf{R}}}$ if for any possibly unbounded $\mathcal{A}$ we have*

$$\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{A}}(\lambda) \coloneqq \Pr\left[\begin{matrix}\mathsf{urs} \leftarrow \mathcal{URS}, \\ (\mathsf{srs}, x, \pi) \leftarrow \mathcal{A}^{\mathsf{H}}(1^\lambda; \mathsf{urs}))\end{matrix} : \begin{matrix}\mathsf{crs} \leftarrow (\mathsf{srs}, \mathsf{urs}), x \notin \mathscr{L}_{\widetilde{\mathsf{R}}}, \\ \mathsf{Verify}^{\mathsf{H}}(\mathsf{crs}, x, \pi) = 1\end{matrix}\right] \leq \mathrm{negl}(\lambda),$$

where the probability is over the random coins of $\mathcal{A}$ and GenCRS, the random choices of urs, and the random choices of H.

Fiat-Shamir transformation. We recall the Fiat-Shamir transformation [37] to turn a Σ-protocol into a NIZK. Sometimes, we require more involved variants of this transformations. In that case, we provide the compiled NIZK explicitly.

Theorem 5. *Let* $\Sigma = (\mathsf{Init}, \mathsf{Chall}, \mathsf{Resp}, \mathsf{Verify})$ *be a* Σ*-protocol that satisfies* correctness*,* high-min entropy*, and* 2-Special Soundness*. The* Fiat-Shamir transformation $FS[\Sigma] = (\mathsf{GenCRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$ *is described below:*

- $\mathsf{GenCRS}(1^\lambda)$*: outputs the empty string* ϵ *as we do not require a common reference string and omit* crs *as an input for other below algorithms,*
- $\mathsf{Prove}^{\mathsf{H}}(x, w)$*: receives a statement* x *and a witness* w*, runs* $(\alpha, \mathsf{st}) \leftarrow \mathsf{Init}(x, w)$*, computes the challenge* $\beta \leftarrow \mathsf{H}(\mathsf{st}, \alpha)$*, then computes* $\gamma \leftarrow \mathsf{Resp}(\mathsf{st}, \beta)$ *and outputs* $\pi = (\alpha, \beta, \gamma)$*.*
- $\mathsf{Verify}^{\mathsf{H}}(x, \pi)$*: receives a statement* x *and a proof* $\pi = (\alpha, \beta, \gamma)$*, and outputs* $b \leftarrow \mathsf{Verify}(x, \alpha, \beta, \gamma)$*.*

In the ROM, $FS[\Sigma]$ *is a* NIZK *that is* correct *and satisfies* adaptive knowledge soundness*.*

A.10 Pseudorandom Functions

Definition 24 (Pseudorandom Functions). *Let* $\mathcal{X}$*,* $\mathcal{Y}$ *and* $\mathcal{K}$ *be sets representing domain, range and key space, respectively. We assume that they are implicitly indexed by a security parameter* $\lambda \in \mathbb{N}$*. Furthermore, let* $\mathcal{R}$ *be the set of all functions with domain* $\mathcal{X}$ *and range* $\mathcal{Y}$*. A family of functions* $\{F_K\}_{K \in \mathcal{K}}$ *that consists of efficiently computable functions* $F_K \colon \mathcal{X} \to \mathcal{Y}$ *is called a* pseudorandom function *(*PRF*) if for any PPT adversary* $\mathcal{A}$*, the following advantage is negligible in* λ*:*

$$\mathsf{Adv}^{\mathsf{PRF}}_{F_K, \mathcal{A}}(1^\lambda) \coloneqq \left| \Pr[\mathcal{A}^{F_K(\cdot)} = 1] - \Pr[\mathcal{A}^{R(\cdot)} = 1] \right| ,$$

where $K \leftarrow \mathcal{K}$ *and* $R \leftarrow \mathcal{R}$*. The probability is taken over the choices of* K, R *and the random coins of* $\mathcal{A}$*.*

B Deferred Content from Section 3

B.1 Alternative Algorithms

For the proof of security, we describe some "alternative" algorithms for signing and key generation.

First, we describe the alternate key generation algorithms:

$\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{1,0}(1^\lambda, N, z)$ sample Q_S primes $e_1, \ldots, e_{Q_S} \leftarrow \mathcal{S}_e$. Sample $\beta \leftarrow \mathcal{S}_a$. Sample $v, w \leftarrow \mathbb{Z}_N^*$. Sample $j \leftarrow \{1, \ldots, Q_S\}$. Set $h_1 \coloneqq z^{2\prod_{i \neq j} e_i}$, $h_2 \coloneqq v^{2\prod_i e_i}$, $h \coloneqq h_1^{-\beta} \cdot w^{2\prod_i e_i}$. Output $\mathsf{vk} = (N, h, h_1, h_2)$ along with $\mathsf{sk}_{0,0} = (\beta, v, w, e_1, \ldots, e_{Q_S}, j)$

$\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{1,1}(1^\lambda, N, z)$ sample Q_S primes $e_1, \ldots, e_{Q_S} \leftarrow \mathcal{S}_e$. Sample $\beta \leftarrow \mathcal{S}_a$. Sample $v, w \leftarrow \mathbb{Z}_N^*$. Sample $v, w \leftarrow \mathbb{Z}_N^*$. Sample $j \leftarrow \{1, \ldots, Q_S\}$. Set $h_1 \coloneqq v^{2\prod_i e_i}$, $h_2 \coloneqq z^{2\prod_{i \neq j} e_i}$, $h \coloneqq h_2^{-\beta} \cdot w^{2\prod_i e_i}$. Output $\mathsf{vk} = (N, h, h_1, h_2)$ along with $\mathsf{sk}_{0,1} = (\beta, v, w, e_1, \ldots, e_{Q_S}, j)$

$\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_0(1^\lambda, N, z)$ sample Q_S primes $e_1, \ldots, e_{Q_S} \leftarrow \mathcal{S}_e$. Sample $a, a' \leftarrow \{1, \ldots, N^2\}$ and set $h_1 \coloneqq z^{2\prod_i e_i}$, $h_2 \coloneqq h_1^{a'}$, $h \coloneqq h_1^a$. Output $\mathsf{vk} = (N, h_1, h_2, h)$ along with $\mathsf{sk}_1 = (a, a', e_1, \ldots, e_{Q_S})$.

Corresponding to these alternate key generation algorithms, we describe how to use the internal state for generating signatures on hashes of messages $\overline{m}$ where k is a counter for the number of signing queries.

$\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{1,0}(\beta, v, w, e_1, \ldots, e_{Q_S}, j, k, \overline{m})$ If $k \neq j$, sample $a_k \leftarrow \mathcal{S}_a$. Compute

$$\begin{aligned} y_k \coloneqq & w^{2\prod_{i \neq k} e_i} \cdot \begin{pmatrix} 2\prod_{i \neq j} e_i \\ z \qquad i \neq k \end{pmatrix}^{a_k - \beta} \left(v^{2\prod_{i \neq k} e_i} \right)^{a_k + \overline{m}} \\ = & \left(h \cdot h_1^{a_k} \cdot h_2^{a_k + \overline{m}} \right)^{\frac{1}{e_k}} \end{aligned}$$

For $k = j$, it sets $a_k = \beta$ and computes

$$\begin{aligned} y_k :=& w^{2\prod_{i\neq k} e_i} \cdot \left(v^{2\prod_{i\neq k} e_i}\right)^{a_k+\overline{m}} \\ =& \left(h \cdot h_1^{a_k} \cdot h_2^{a_k+\overline{m}}\right)^{\frac{1}{e_k}} \end{aligned}$$

Output $\sigma_k = (e_k, a_k, y_k)$.

$\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{1,1}(\beta, v, w, e_1, \ldots, e_{Q_S}, j, k, \overline{m})$ For any $k \neq j$, sample $a_k \leftarrow \mathcal{S}_a$ and compute

$$\begin{aligned} y_k :=& w^{2\prod_{i\neq k} e_i} \cdot \left(z^{2\prod_{i\neq j\ i\neq k} e_i}\right)^{a_k+\overline{m}-\beta} \left(v^{2\prod_{i\neq k} e_i}\right)^{a_k} \\ =& \left(h \cdot h_1^{a_k} \cdot h_2^{a_k+\overline{m}}\right)^{\frac{1}{e_k}} \end{aligned}$$

For $k = j$, it sets $a_k = \beta - \overline{m}$ and computes

$$\begin{aligned} y_k :=& w^{2\prod_{i\neq k} e_i} \cdot \left(v^{2\prod_{i\neq k} e_i}\right)^{a_k+\overline{m}} \\ =& \left(h \cdot h_1^{a_k} \cdot h_2^{a_k+\overline{m}}\right)^{\frac{1}{e_k}} \end{aligned}$$

$\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_0(a, a', e_1, \ldots, e_{Q_S}, k, \overline{m})$ Sample $a_k \leftarrow \mathcal{S}_a$ and compute

$$\begin{aligned} y_k :=& z^{2\cdot(a+a_k\cdot a'+(a_k+\overline{m}))\prod_{i\neq k} e_i} \\ =& \left(h \cdot h_1^{a_k} \cdot h_2^{a_k+\overline{m}}\right)^{\frac{1}{e_k}} \end{aligned}$$

B.2 Proof of security

Theorem 6. *If the sRSA assumption is $(t, \varepsilon_{\mathsf{RSA}})$-hard and the hash function H is $(t, \varepsilon_{\mathsf{coll}})$-collision resistant then the scheme described above is $(t' \approx t, 6Q_S\varepsilon_{\mathsf{RSA}} + \varepsilon_{\mathsf{coll}} + \frac{Q_S}{2^{\lambda-2}})$-EUF-CMA secure against an adversary that makes Q_S signing queries.*

Proof. Let $\mathcal{A}$ be an adversary against the EUF-CMA security of the scheme that runs in time t' and has advantage ε' and makes Q_S queries to the signing oracle.

We denote by m_i the ith message queried to the signing oracle by $\mathcal{A}$, by $\sigma_i = (e_i, a_i, y_i)$ the ith signature output by the signing oracle to $\mathcal{A}$, and by $m^*, \sigma^* = e^*, a^*, y^*)$ we denote $\mathcal{A}$'s forgery. We show security through a series of games.

Game 1: Game 1 is the original EUF-CMA game.

Game 2: In Game 2 the game aborts if for any i, j $m_i \neq m_j$ it holds that $\mathsf{H}(m_i) = \mathsf{H}(m_j)$ or if $\mathsf{H}(m_i) = \mathsf{H}(m^*)$. We can bound the abort probability using the collision resistance, namely,

$$|\Pr[\text{Game } 2 = 1] - \Pr[\text{Game } 1 = 1]| \leq \mathsf{Adv}^{\text{coll}}_{\mathcal{B}_1}$$

for an adversary B_1 against the collision resistance of H.

Game 3: In Game 3, we introduce an abort condition in which our reduction will not be able to simulate. At the end of the game, the game Game 3 samples a bit b and aborts if $b = 0$ and $e^* \in \{e_1, \ldots, e_{Q_S}\}$ or if $b = 1$ and $e^* \notin \{e_1, \ldots, e_{Q_S}\}$. It is easy to see that

$$\Pr[\text{Game } 3 = 1] \geq \frac{1}{2}\Pr[\text{Game } 2 = 1].$$

Game 4: In Game 4, if $b = 1$, the Game samples an index $j \in \{1, \ldots, Q_S\}$. It aborts if $e^* \neq e_j$. It holds that

$$\Pr[\text{Game } 4 = 1] \geq \frac{1}{Q_S} \Pr[\text{Game } 3 = 1]$$

Game 5: In Game 5, if $b = 1$, the Game samples a bit b'. If $b = 0$ and $a_j = a^*$ (where j is as defined in Game 4), the game aborts. If $b = 1$ and $a_j + \mathsf{H}(m_j) = a^* + \mathsf{H}(m^*)$, the game aborts. It holds that

$$\Pr[\text{Game } 5 = 1] \geq \frac{1}{2} \Pr[\text{Game } 4 = 1].$$

Game 6: In Game 6, we change sample b, b', j at the beginning of the game. This is a purely conceptual change, thus

$$\Pr[\text{Game } 6 = 1] = \Pr[\text{Game } 5 = 1]$$

Game 7: In Game 7 we change how the values a_j are sampled during signature generation. If $b = 1$, $b' = 0$, instead of sampling $a_j \leftarrow \mathcal{S}_a$, it first samples $\beta \leftarrow \mathcal{S}_a$ and then sets $a_j = \beta$. If $b = 1$ and $b' = 1$, it samples $\beta \leftarrow \mathcal{S}_a$ and sets $a_j = \beta - \mathsf{H}(m_j)$. A simple argument shows that the distribution of a_j in Game 7 has statistical distance at most $1/2^\lambda$ from the distribution of a in Game 6.

Thus, we get that $|\Pr[\text{Game } 7 = 1] - \Pr[\text{Game } 6 = 1]| \leq \frac{1}{2^\lambda}$.

The Reduction: We now provide a reduction that simulates Game 7 and breaks the strong RSA assumption.

On input $(N, z \in \mathbb{Z}_N^*)$,the reduction behaves as follows:

First, it samples a bit b, b' and an index j. If $b = 0$ (recall that in this case Game 7 aborts if $e^* \in \{e_1, \ldots, e_{Q_S}\}$), the reduction works as follows:

Setup. Runs $\mathsf{S_{fis}.KeyGen}_{0,b'}(1^\lambda, N, z)$ to obtain $\mathsf{vk}, \mathsf{sk}_{0,b'}$ It passes the public key (N, h, h_1, h_2) to the adversary.

Signing Queries. For the kth signing query it runs $\mathsf{S_{fis}.Sign}_{0,b'}(\mathsf{sk}_{0,b'}, k, \mathsf{H}(m))$ to obtain $\sigma_k = (e_k, a_k, y_k)$ and outputs σ_k.

Output Determination. When the adversary outputs a forgery $m^*, \sigma^* = e^*, a^*, y^*)$, the reduction can compute an e_jth root of z. As Game 7 aborts unless $e^* = e_j$, the reduction obtains

$$h_1^{-a_j} h_2^{-(a_j + \mathsf{H}(m_j))} \cdot y_j^{e_j} = h = h_1^{-a^*} h_2^{-(a^* + \mathsf{H}(m^*))} y^{*e_j}$$

solving for z using the preselected values from the public key yields:

$$z^{2 \prod_{i \neq j} e_i \cdot (a^* - a_j)} = \left(v^{2 \prod_{i \neq j} e_i \cdot (a_j + \mathsf{H}(m_j) - a^* - \mathsf{H}(m^*))} \right)^{e_j}$$

Which we can solve for a e_jth root of z if $\gcd(e_j, 2 \prod_{i \neq j} e_i \cdot (a^* - a_j)) = 1$ using Lemma 4. It holds that the gcd is 1 as $a_j < e$ and $a^* < e$ by virtue of the range checks, and thus also their difference is smaller than e_j. As e_j is prime this immediately implies coprimality. Furthermore, all the other e_i are coprime to e_j, and e_j is odd, so it is also coprime with 2.

For the case that $b = 1$, the reduction simulates as follows:

Setup. Given N and $z \in \mathbb{Z}_N$, the reduction runs $\mathsf{S_{fis}.KeyGen}_1(1^\lambda, N, z)$ to obtain $\mathsf{vk} = (N, h, h_1, h_2)$ and sk_1. It outputs the public key vk to the adversary.

Signing Queries. The reduction responds to the kth signing query by running $\mathsf{S_{fis}.Sign}_1(\mathsf{sk}_1, k, \mathsf{H}(m))$ to obtain σ_k. It outputs the signature σ_k.

Output Determination. When the adversary outputs its forgery $m^*, \sigma^* = e^*, a^*, y^*)$, the reduction can learn the following

$$y^{*e^*} = hh_1^{a^*} h_2^{a^* + \mathsf{H}(m^*)} = z^{2 \cdot (a + a^* \cdot a' + (a^* + \mathsf{H}(m^*)))\prod_{i \neq k} e_i}$$

Computing a root of z follows as in [38] where the probability of success is $(1 - 1/r)$ where r is the smallest prime factor dividing e^*. As e^* is odd, r is at least 3.
Putting this together yields

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sRSA}}_{\mathcal{B}} \geq & \frac{2}{3} \Pr[\text{Game } 7 = 1] \\
\geq & \frac{2}{3} \left(\Pr[\text{Game } 6 = 1] - \frac{1}{2^\lambda} \right) \\
\geq & \frac{2}{3} \left(\frac{1}{4Q_S} \Pr[\text{Game } 2 = 1] - \frac{1}{2^\lambda} \right) \\
\geq & \frac{2}{3} \left(\frac{1}{4Q_S} \left(\Pr[\text{Game } 1 = 1] - \mathsf{Adv}^{\mathrm{coll}}_{\mathcal{B}_1} \right) - \frac{1}{2^\lambda} \right) \\
= & \frac{1}{6Q_S} \mathsf{Adv}^{\mathsf{euf-cma}}_{\mathcal{A}} - \frac{1}{6Q_S} \mathsf{Adv}^{\mathrm{coll}}_{\mathcal{B}_1} - \frac{1}{3 \cdot 2^{\lambda - 1}}
\end{aligned}$$

C Deferred Content from Section 4

C.1 Security Proof of $\mathsf{C}_{\mathsf{Grp}}$

Proof. Correctness is straightforward.

Hiding is argued as follows. First, observe that $c \leftarrow \mathsf{C}_{\mathsf{RInt}}(\mathsf{pp}, \vec{s}) \overset{c}{\approx} \mathsf{C}_{\mathsf{RInt}}(\mathsf{pp}, \vec{0})$ under the hiding property of $\mathsf{C}_{\mathsf{RInt}}$. Also, the distribution of $\hat{c} = \hat{x} \cdot \hat{g}^s$ for $s \leftarrow [0, U \cdot 2^\lambda]$ has a statistical distance of at most $2^{-\lambda}$ to the uniform distribution $\mathcal{U}_{\hat{\mathbb{G}}}$ over $\hat{\mathbb{G}}$. Thus, we have $\hat{c} \overset{s}{\approx} \mathcal{U}_{\mathbb{G}}$. In total, $(\hat{c}, c) \overset{c}{\approx} (\mathcal{U}_{\hat{\mathbb{G}}}, \mathsf{C}_{\mathsf{RInt}}(\mathsf{pp}, \vec{0}))$ for $(\hat{c}, c) \leftarrow \mathsf{C}_{\mathsf{Grp}}.\mathsf{Commit}(\mathsf{pp}, \hat{x})$.

Binding follows from the binding property of $\mathsf{C}_{\mathsf{RInt}}$ and since $\hat{x} = \hat{g}^s \hat{c}^{-1}$ is uniquely determined if s is fixed. In more detail, we reduce binding to the binding property of $\mathsf{C}_{\mathsf{RInt}}$. Let $\mathcal{A}$ be an adversary on the binding property of $\mathsf{C}_{\mathsf{Grp}}$. First, obtain pp from a challenger of the $\mathsf{C}_{\mathsf{RInt}}$ binding property. Set $(c_x, \hat{x}^{(0)}, \hat{x}^{(1)}, r_x^{(0)}, r_x^{(1)}) \leftarrow \mathcal{A}(\mathsf{pp})$. Parse $c_x = (\hat{c}, c)$ and $r_x^{(b)} = (\vec{s}^{\,(b)}, r^{(b)})$. Output $(c, \vec{s}^{\,(0)}, \vec{s}^{\,(1)}, r^{(0)}, r^{(1)})$ to the challenger.

To analyze the success probability, assume that $\mathcal{A}$ is successful. Then, we have $\hat{x}^{(0)} \neq \hat{x}^{(1)} \in \hat{\mathbb{G}}$ and $\mathsf{C}_{\mathsf{Grp}}.\mathsf{Verify}(\mathsf{pp}, c_x, \hat{x}^{(b)}, r_x^{(b)}) = 1$ for $b \in \{0, 1\}$. Set $s^{(b)} = \sum_{i=1}^{\ell} s_i^{(b)} B^{i-1}$. If $s^{(0)} = s^{(1)} \coloneqq s$, we have that

$$\hat{c} = \hat{x}^{(0)} \cdot \hat{g}^s = \hat{x}^{(1)} \hat{g}^s.$$

Thus, we have $\hat{x}^{(0)} = \hat{x}^{(1)}$ which contradicts our assumption. Consequently, it holds that $s^{(0)} \neq s^{(1)}$. By construction of $s^{(0)}$ and $s^{(1)}$, it must hold that $\vec{s}^{\,(0)} \neq \vec{s}^{\,(1)}$ (over $\mathbb{Z}$). But since $\mathsf{C}_{\mathsf{RInt}}.\mathsf{Verify}(\mathsf{pp}, c, \vec{s}^{\,(b)}, r^{(b)}) = 1$ for $b \in \{0, 1\}$, the values $(c, \vec{s}^{\,(0)}, \vec{s}^{\,(1)}, r^{(0)}, r^{(1)})$ form a valid solution solution for the binding game of $\mathsf{C}_{\mathsf{RInt}}$.

C.2 Efficient Opening in Zero-Knowledge

We construct efficient NIZKs Π_{int} and Π_{grp} to open $\mathsf{C}_{\mathsf{RInt}}$ and $\mathsf{C}_{\mathsf{Grp}}$, respectively, in zero-knowledge.

Proof for Public Parameters. Before we detail both NIZKs, we construct an additional NIZK Π_{gen} to prove that MPed is statistically hiding under public parameters $\mathsf{pp} = (N, h, \vec{g})$ for MPed and $\vec{g} = (g_1, \cdots, g_\ell)$. This is the case if $\langle h \rangle = \langle g_i \rangle \subseteq \mathbb{Z}_N^*$ for all $i \in [\ell]$. More generally, we construct an NIZK Π_{gen} with oracle $\mathsf{H}_{\mathsf{gen}}$ for the relation

$$\mathsf{R}_{\mathsf{gen}} = \left\{ (x, w) \mid \forall i \in [\ell] : g_i^{\alpha_i} \equiv h \bmod N, h^{\beta_i} \equiv g_i \bmod N \right\},$$

where $x = (N, \ell, h, (g_i)_{i\in[\ell]})$ and $w = ((\alpha_i, \beta_i)_{i\in[\ell]})$ for some $\ell \in \mathbb{N}$. Note that we also use Π_{gen} in Section 5. It is based on the Σ-protocol Σ_{gen} given in Figure 2 with challenge space $[0, C]$ for $C = 2^\lambda - 1$, compiled into a NIZK via Fiat-Shamir. The random oracle is denoted by $\mathsf{H}_{\mathsf{gen}}$. Note that no crs is required (*i.e.*, $\mathcal{SRS} = \mathcal{URS} = \{\bot\}$).

- $\Pi_{\mathsf{gen}}.\mathsf{GenCRS}(1^\lambda)$: Outputs $\bot$.
- $\Pi_{\mathsf{gen}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}, x, w)$: On input crs, statement x, and witness w, outputs the proof π computed as follows

$$\begin{aligned}
&(\Omega_\Sigma, \mathsf{st}) \leftarrow \Sigma_{\mathsf{gen}}.\mathsf{Init}(x, w),\\
&\gamma_\Sigma \leftarrow \mathsf{H}_{\mathsf{gen}}(x, \Omega_\Sigma),\\
&\tau_\Sigma \leftarrow \Sigma_{\mathsf{gen}}.\mathsf{Resp}(x, \mathsf{st}, \gamma_\Sigma),\\
&\pi \leftarrow (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma).
\end{aligned}$$

- $\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}, x, \pi)$: On input crs, statement x, and proof π, checks

$$\begin{aligned}
&\mathsf{H}_{\mathsf{gen}}(x, \Omega_\Sigma) = \gamma_\Sigma,\\
&\Sigma_{\mathsf{ped}}.\mathsf{Verify}(x, \Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma) = 1,
\end{aligned}$$

  where $\pi = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$, and outputs 1 iff all checks succeed.

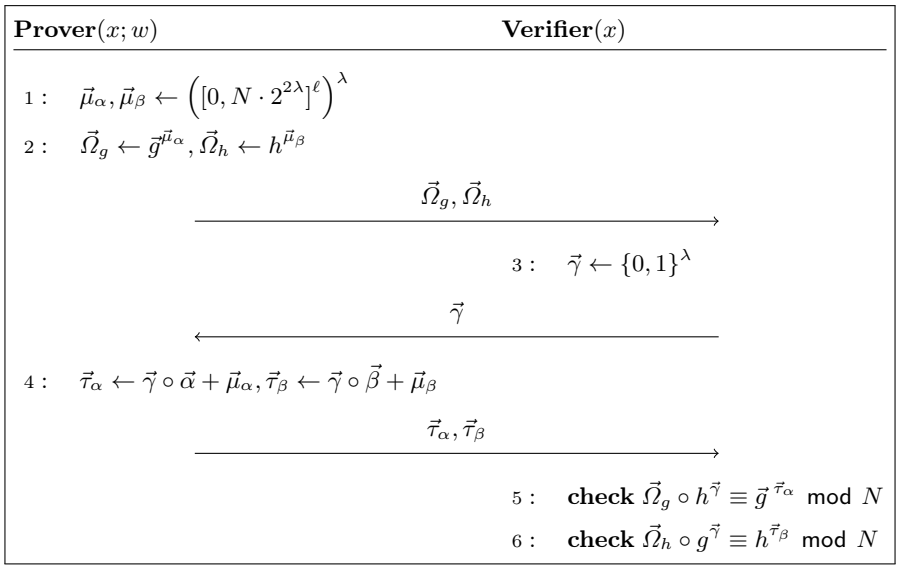

Fig. 2. Description of Σ_{gen} for $x = (N, \ell, h, \vec{g})$ and $w = (\alpha_i, \beta_i)_{i\in[\ell]}$ with $\vec{g} = (g_1, \cdots, g_\ell)$. We denote the Hadamard product by $\circ$.

We first show that the Σ-protocol Σ_{gen} given in Figure 2 satisfies desired properties for the Fiat-Shamir transform.

Theorem 7. *The Σ-protocol Σ_{gen} given in Figure 2 satisfies* correctness*,* 2-special soundness*,* honest verifier zero-knowledge*, and has high min-entropy.*

Proof. For the commitment vectors $\vec{\Omega}_g, \vec{\Omega}_h$ and the response vectors $\vec{\tau}_\alpha, \vec{\tau}_\beta$, their i-th element is denoted by $\vec{\Omega}_{g,i}, \vec{\Omega}_{h,i}, \vec{\tau}_{\alpha,i}, \vec{\tau}_{\beta,i}$. First of all, Σ_{gen} has *high min-entropy* due to the fact that the space $[0, N \cdot 2^{2\lambda}]^\ell$ for commitment vectors is exponentially large in λ. *Correctness* is straightforward, noting that by construction, for $x = (N, \ell, h, \vec{g})$ and $w = (\alpha_i, \beta_i)_{i\in[\ell]}$, for all $i \in [\ell]$

$$\vec{\Omega}_{g,i} \circ h^{\vec{\gamma}} = g_i^{\vec{\mu}_{\alpha,i}} \circ g_i^{\alpha_i \cdot \vec{\gamma}} = g_i^{\vec{\tau}_{\alpha,i}}; \quad \vec{\Omega}_{h,i} \cdot g_i^{\vec{\gamma}} = h^{\vec{\mu}_{\beta,i}} \circ h^{\beta_i \vec{\gamma}} = h_i^{\vec{\tau}_{\beta,i}} \ .$$

For 2-*special soundness*, given two valid transcripts $(\vec{\Omega}_g, \vec{\Omega}_h), \vec{\gamma}^b, \vec{\tau}^b_\alpha, \vec{\tau}^b_\beta$ where $b \in \{0,1\}$ and $\gamma^0 \neq \gamma^1$, a deterministic polynomial-time extractor Ext can executes as follows: First, identify an index j s.t. $\gamma^0_j \neq \gamma^1_j$

1. For each $i \in [\ell]$, Ext sets $\alpha_i \coloneqq \frac{\vec{\tau}^0_{\alpha,i} - \vec{\tau}^1_{\alpha,i}}{\gamma^0_j - \gamma^1_j}$.
2. For each $i \in [\ell]$, Ext sets $\beta_i \coloneqq \frac{\vec{\tau}^0_{\beta,i} - \vec{\tau}^1_{\beta,i}}{\gamma^0_j - \gamma^1_j}$.
3. Outputs $w \coloneqq (\alpha_i, \beta_i)_{i\in[\ell]}$ as a witness for $x = (N, \ell, h, \vec{g})$.

The output w by Ext is well defined and indeed a witness of x because $\gamma^0 \neq \gamma^1$, $\gamma^0_j - \gamma^1_j \in \{-1, 1\}$ (and thus has an efficiently computable multiplicative inverse) and

$$\begin{cases} \vec{\Omega}_{g,i} \cdot h^{\gamma^0_j} &= g_i^{\vec{\tau}^0_{\alpha,i}} \\ \vec{\Omega}_{g,i} \cdot h^{\gamma^1_j} &= g_i^{\vec{\tau}^1_{\alpha,i}} \end{cases}; \quad \begin{cases} \vec{\Omega}_{h,i} \cdot g_i^{\gamma^0_j} = h^{\vec{\tau}^0_{\beta,i}} \\ \vec{\Omega}_{h,i} \cdot g_i^{\gamma^1_j} = h^{\vec{\tau}^1_{\beta,i}} \end{cases} \Rightarrow \begin{cases} h^{\gamma^0_j - \gamma^1_j} &= g_i^{\vec{\tau}^0_{\alpha,i} - \vec{\tau}^1_{\alpha,i}} \\ g_i^{\gamma^0_j - \gamma^1_j} &= h^{\vec{\tau}^0_{\beta,i} - \vec{\tau}^1_{\beta,i}} \end{cases}.$$

A PPT simulator Sim for *honest verifier zero-knowledge* works as follows:

1. For each $i \in [\ell]$
   - Sim samples the challenge $\vec{\gamma} \leftarrow \{0,1\}^\lambda$ as well as the i-th responses $\vec{\tau}_{\alpha,i}, \vec{\tau}_{\beta,i} \leftarrow [0, N \cdot 2^{2\lambda}]$.
   - Sim computes $\vec{\Omega}_{g,i} \coloneqq g_i^{\vec{\tau}_{\alpha,i}}(h^{\vec{\gamma}})^{-1}$ and $\vec{\Omega}_{h,i} \coloneqq h^{\vec{\tau}_{\beta,i}} \circ (g_i^{\vec{\gamma}})^{-1}$. The commitments are defined $\vec{\Omega}_g \coloneqq (\vec{\Omega}_{g,i})_{i\in[\ell]}, \vec{\Omega}_h \coloneqq (\vec{\Omega}_{h,i})_{i\in[\ell]}$.
   - Output $(\vec{\Omega}_g, \vec{\Omega}_h, \gamma, \vec{\tau}_\alpha, \vec{\tau}_\beta)$.

For any $x \in \mathscr{L}_{\mathsf{R}}$, *i.e.* $\langle h \rangle = \langle g_i \rangle \subseteq \mathbb{Z}^*_N$ for all $i \in [\ell]$, the simulator $\mathsf{Sim}(x, C)$ outputs a valid transcript that is distributed identically to the honestly generated transcript with Init initialized with (x, w). We use the fact that because $\langle h \rangle = \langle g_i \rangle$, it holds that $\vec{\Omega}_{g,i} \coloneqq \frac{g_i^{\vec{\tau}_{\alpha,i}}}{h^\gamma} \in \langle g_i \rangle$ and $\vec{\Omega}_{h,i} \coloneqq \frac{h^{\vec{\tau}_{\beta,i}}}{g_i^\gamma} \in \langle h \rangle$ having the same distributions thanks to how Sim samples $\gamma, \vec{\tau}_{\alpha,i}, \vec{\tau}_{\beta,i}$. The proof is completed. □

We now show that the Π_{gen} satisfies statistical adaptive subversion soundness, zero-knowledge, and correctness.

Theorem 8. Π_{gen} *satisfies statistical adaptive subversion soundness, zero-knowledge, and correctness.*

Proof.

Correctness. Correctness directly follows from the correctness of the underlying Σ-protocol.

Soundness. As the CRS of this protocol is empty, it suffices to consider an adversary $\mathcal{A}$ that outputs a pair (x, π) for $x \notin \mathscr{L}_{\mathsf{R}}$. Consider an arbitrary $x \notin \mathscr{L}_{\mathsf{R}}$, i.e. $\langle h \rangle \neq \langle g_i \rangle$ for some $i \in [\ell]$. W.l.o.g. we consider the case that $\langle h \rangle \not\subseteq \langle g_i \rangle$ (the argument for the other direction is symmetrical). This in particular means $h \notin \langle g_i \rangle$. Thus, for any value $\Omega_{g,i,j} \in \mathbb{Z}^*_N$ it cannot hold that both $\Omega_{g,i,j} \cdot h \in \langle g_i \rangle$ as well as $\Omega_{g,i,j} \in \langle g_i \rangle$. We consider a hash query made by the statistical soundness adversary. The adversary submits vectors $\vec{\Omega}_g, \vec{\Omega}_h$ to the random oracle. By what we saw above, for each entry $\vec{\Omega}_{g,i,j}$, it holds that either $\Omega_{g,i,j} \cdot h \in \langle g_i \rangle$ or $\Omega_{g,i,j} \in \langle g_i \rangle$ (if neither is the case the adversary cannot output a proof using this hash query). As the hash oracle is a random oracle, with probability $\leq \frac{1}{2}$, the j-th entry of the hash response is b_j such that $\Omega_{g,i,j} \cdot h^b \in \langle g_i \rangle$. As the b_j are sampled uniformly at random by the random oracle, it follows that the probability that for all $j \in [\lambda]$, $\Omega_{g,i,j} \cdot h^b \in \langle g_i \rangle$ is $\leq \frac{1}{2^\lambda}$. Union bounding over all $Q_{\mathsf{H}_{\mathsf{gen}}}$ hash queries made by the adversary yields that $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{A}}(\lambda) \leq \frac{Q_{\mathsf{H}_{\mathsf{gen}}}}{2^\lambda}$.

Zero-knowledge. The Zero-Knowledge property directly follows from the honest verifier zero-knowledge property of the Σ-protocol and the Fiat-Shamir transform.

Efficient Proof of Opening for $\mathsf{C}_{\mathsf{RInt}}$. We construct a NIZK Π_{int} that allows to open $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}$ in zero-knowledge for arbitrary $B \in \mathbb{N}$ and slack $T = 2^{\lambda+1}L$, where $L \in \mathbb{N}$ is the masking overhead for rejection sampling. Note that the size of T and $\vec{B}$ impact the size of the underlying group $\mathbb{G}$.

To construct Π_{int}, we compile a Schnorr-style Σ-protocol with challenge space $[0, C]$ for $C := 2^{\lambda}-1$ using Fiat-Shamir with abort. To ensure (relaxed) range membership we use techniques from [31, 30]. Roughly, we add an MPed commitment $\bar{c}$ to $\vec{m}$ that in conjunction with a size check ensures that the extracted integers are in the relaxed range $[-\vec{B}T, \vec{B}T]$. The public parameters $\mathsf{pp}_{\mathsf{MPed}} = (N, h, g_1, \cdots, g_\ell)$ for MPed constitute the srs. To obtain subversion zero-knowledge, we add a proof π_{gen} generated via Π_{gen} that $\langle h \rangle = \langle g_i \rangle$ for all $i \in [\ell]$ to ensure that MPed is hiding even for a malicious $\mathsf{pp}_{\mathsf{MPed}}$. We denote by $\mathsf{H}_{\mathsf{gen}}$ the hash function for Π_{gen}.

Formally, the zero-knowledge relation is

$$\mathsf{R} = \{(x, w) \mid (c, d) = \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\vec{m}; r), \vec{m} \in [0, \vec{B}]\}$$

for $x = (\mathsf{pp}, c)$ with $c = (\vec{C}, F)$ and $w = (\vec{m}, r)$, where $d = r \in \mathbb{Z}_p$. The soundness relation is

$$\widetilde{\mathsf{R}} = \{(x, w) \mid \mathsf{C}_{\mathsf{RInt}}.\mathsf{Verify}(\mathsf{pp}, c, \vec{m}, r)\}.$$

The underlying Σ-protocol Σ_{int} is given in Figure 3. Note that the crs is included in the statement of Σ_{int} for technical reasons. The NIZK Π_{int} with hash function $\mathsf{H}_{\mathsf{int}} : \{0,1\}^* \to [0, C]$, urs length $\ell_{\mathsf{int}} = 0$ and

$$\mathcal{SRS} = \big\{(\mathsf{pp}_{\mathsf{MPed}}, \pi_{\mathsf{gen}}) \mid \mathsf{pp}_{\mathsf{MPed}} = (N, h, \vec{g}) \in \mathbb{N} \times (\mathbb{Z}_N^*)^{\ell+1}, \\ \Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(x_{\mathsf{gen}}, \pi_{\mathsf{gen}}), x_{\mathsf{gen}} = (N, \ell, h, \vec{g})\big\}$$

is defined as follows. Note that membership checks for $\mathcal{SRS}$ are efficient by design.

- $\Pi_{\mathsf{int}}.\mathsf{GenCRS}(1^\lambda)$: On input 1^λ, samples $\mathsf{pp}_{\mathsf{MPed}} = (N, h, \vec{g}) \leftarrow \mathsf{MPed.Setup}(1^\lambda)$. Then, sets $\pi_{\mathsf{gen}} \leftarrow \Pi_{\mathsf{gen}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{gen}}}(w_{\mathsf{gen}}, x_{\mathsf{gen}})$ for $x_{\mathsf{gen}} = (N, \ell, h, \vec{g})$ and appropriate w_{gen} (which can be computed explicitly during $\mathsf{MPed.Setup}$). Outputs the structured reference string $\mathsf{srs} = (\mathsf{pp}_{\mathsf{MPed}}, \pi_{\mathsf{gen}})$.
- $\Pi_{\mathsf{int}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{int}}}(\mathsf{crs}, x, w)$: Computes a proof π as follows for $x_\sigma = (x, \mathsf{crs})$.

$$\begin{aligned}
&(\Omega_\Sigma, \mathsf{st}) \leftarrow \Sigma_{\mathsf{int}}.\mathsf{Init}(x, w),\\
&\gamma_\Sigma \leftarrow \mathsf{H}_{\mathsf{int}}(x_\Sigma, \Omega_\Sigma),\\
&\tau_\Sigma \leftarrow \Sigma_{\mathsf{int}}.\mathsf{Resp}(x_\Sigma, \mathsf{st}, \gamma_\Sigma),\\
&\pi \leftarrow (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma).
\end{aligned}$$

  Restarts if if $\Sigma_{\mathsf{int}}.\mathsf{Resp}$ aborted, else outputs π.
- $\Pi_{\mathsf{int}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}, x, \pi)$: On input crs, statement x, and proof π, sets $x_\Sigma = (x, \mathsf{crs})$ and checks

$$\begin{aligned}
&\mathsf{H}_{\mathsf{int}}(x_\Sigma, \Omega_\Sigma) = \gamma_\Sigma,\\
&\Sigma_{\mathsf{ped}}.\mathsf{Verify}(x_\Sigma, \Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma) = 1,
\end{aligned}$$

  where $\pi = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$, and outputs 1 iff all checks succeed.

We show that Π_{int} is secure. We give a brief sketch. Correctness is clear (if the abort probability is sufficiently low). For soundness, we use the forking lemma to obtain 2 accepting transcripts. Then, we compute openings for $\mathsf{C}_{\mathsf{RInt}}$ as usual. Due to Lemma 5 and the shortness checks, the opening is in the right interval. For subversion zero-knowledge, observe that for any $\mathsf{srs} \in \mathcal{SRS}$, the commitment $\bar{c}$ is hiding (under soundness of Π_{gen}).

Theorem 9. *The NIZK is correct if* $(1 - (1 - \frac{1}{L})^\ell)^{-1} = \mathsf{poly}(\lambda)$*, adaptively knowledge sound for* $\widetilde{\mathsf{R}}$ *and subversion zero-knowledge.*

Proof. We give a proof sketch for correctness and subversion zero-knowledge (as the proofs are straightforward) and give a detailed proof for soundness.

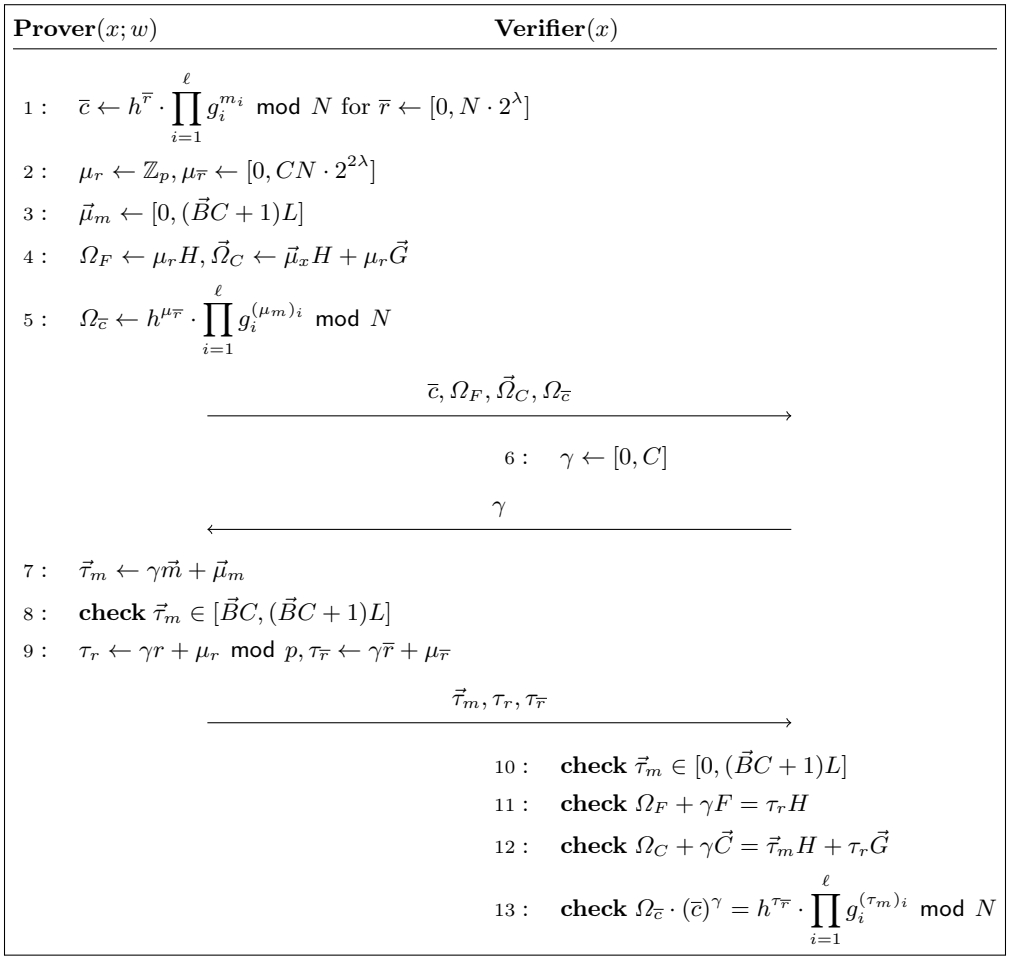

Fig. 3. Description of Σ_{int}, an efficient Σ-protocol for opening $\mathsf{C}_{\mathsf{RInt}}$. Here, $x = (\mathsf{pp}, \vec{C}, F, \mathsf{crs})$ and $w = (\vec{m}, r)$. Also, $\mathsf{crs} = (N, h, \vec{g}, \pi_{\mathsf{gen}})$ for $\vec{g} = (g_1, \cdots, g_\ell)$. If a check fails, the party aborts.

Correctness. Note that a single run succeeds with probability $1 - (1 - \frac{1}{L})^{\ell}$. Thus, proof generation runs in time $\mathcal{O}((1 - (1 - \frac{1}{L})^{\ell})^{-1})$ in expectation. In case of no abort, the verification equations verify by construction.

Subversion zero-knowledge. This follows with standard arguments. We give a sketch and omit details. Observe that for any $\mathsf{srs} \in \mathcal{SRS}$, the commitment $\overline{c}$ is hiding (under soundness of Π_{gen}). Also, $\vec{\tau}_m$ leaks no information about $\vec{m}$ due to rejection sampling (cf. Appendix A.2) and $\tau_r, \tau_{\overline{r}}$ are distributed as random vectors over $\mathbb{Z}_p$. Further, $\Omega_F, \vec{\Omega}_C, \Omega_{\overline{c}}$ is determined by γ, $\overline{c}$ and $\vec{\tau}_m, \tau_r, \tau_{\overline{r}}$ (due to the verification equations). Thus, a proof leaks no information about the witness w except with negligible probability.

Adaptive knowledge soundness. For soundness, we obtain two valid transcripts $tr = (\alpha, \gamma, \omega)$, $tr' = (\alpha, \gamma', \omega')$ with shared $\alpha = (\overline{c}, \Omega_F, \vec{\Omega}_C, \Omega_{\overline{c}})$ but distinct challenges $\gamma \neq \gamma'$ via the forking lemma (cf. Appendix A.2). Parse $\omega = (\vec{\tau}_m, \tau_r, \tau_{\overline{r}})$ and $\omega' = (\vec{\tau}_{m'}, \tau_{r'}, \tau_{\overline{r'}})$. Let us denote $\Delta\vec{m} = \vec{\tau}_m - \vec{\tau}_{m'}, \Delta r = \tau_r - \tau_{r'}, \Delta\overline{r} = \tau_{\overline{r}} - \tau_{\overline{r'}}$, and $\Delta\gamma = \gamma - \gamma' \neq 0$. Without loss of generality, we have $\Delta\gamma \in [0, C]$. Since both transcripts are valid (with shared $\alpha = \alpha'$), we have

$$\Omega_F = \tau_r H - \gamma F = \tau_{r'} H - \gamma' F$$

Rearranging both terms yields

$$\begin{aligned} & \tau_r H - \tau_{r'} H = -\gamma' F + \gamma F \\ \Longrightarrow\ & \Delta\gamma F = \Delta r H \\ \Longrightarrow\ & F = \frac{\Delta r}{\Delta\gamma} H \end{aligned}$$

Similarly, we obtain

$$\vec{C} = \frac{\Delta\vec{m}}{\Delta\gamma} H + \frac{\Delta r}{\Delta\gamma} \vec{G}$$

Thus, $\vec{m} \coloneqq \frac{\Delta\vec{m}}{\Delta\gamma}$ and $r = \frac{\Delta r}{\Delta\gamma}$ form a valid opening for c if $\vec{m} \in [-\vec{B}T, \vec{B}T]$. For this, we use the properties of $\overline{c}$. As above, we obtain

$$\begin{aligned} & h^{\tau_{\overline{r}}} \cdot \prod_{i=1}^{\ell} g_i^{(\tau_m)_i} \cdot (\overline{c})^{-\gamma} = h^{\tau_{\overline{r'}}} \cdot \prod_{i=1}^{\ell} g_i^{(\tau_{m'})_i} \cdot (\overline{c})^{-\gamma'} \ \mathsf{mod}\ N \\ \Longrightarrow\ & h^{\Delta\overline{r}} \cdot \prod_{i=1}^{\ell} g_i^{(\Delta m)_i} = (\overline{c})^{\Delta\gamma} \ \mathsf{mod}\ N \end{aligned}$$

Recall that $\Delta\gamma \in [0, C]$ with $C = 2^\lambda - 1$. Under Lemma 5, we have $\Delta\overline{r}/\Delta\gamma, (\Delta m)_i/\Delta\gamma \in \mathbb{Z}$. Also, since $(\tau_m)_i, (\tau_{m'})_i \in [0, (B_i C+1)L]$ we have that $|(\Delta m)_i/\Delta\gamma| \leq 2(B_i C+1)L$. Since $2(B_i C+1)L \leq 2^{\lambda+1} B_i L = T_i L$, we have $\vec{m} \in [-\vec{B}T, \vec{B}T]$ as desired.

Efficient Proof of Opening for $\mathsf{C_{Grp}}$. A commitment of $\mathsf{C_{Grp}}$ consists of a Pedersen commitment (in $\hat{\mathbb{G}}$) and a $\mathsf{C_{RInt}}$ commitment. If $\mathsf{C_{RInt}}$ is instantiated as in Section 4.2, it is straightforward to obtain a NIZK for opening $\mathsf{C_{Grp}}$ in zero-knowledge using the techniques from Appendix C.2 (since the decomposition of s is linear). We omit details.

D Deferred content from Section 5

D.1 Number of primes in $[2^{5\lambda}, 2^{5\lambda} + 2^{3\lambda}]$

We prove that there are $\Omega(2^{2\lambda})$ in the interval $[2^{5\lambda}, 2^{5\lambda} + 2^{3\lambda}]$. In the following we denote by $\pi(x)$ the number of primes at most x, for any $x \in \mathbb{R}$ is a function of λ. In the following we use $\sim$ to write the limit as $\lambda \to \infty$. We want to estimate

$$\pi(2^{5\lambda} + 2^{3\lambda}) - \pi(2^{5\lambda}) \tag{7}$$

which is the number of primes in $[2^{5\lambda}, 2^{5\lambda} + 2^{3\lambda}]$. First, from a recent result [54], which refines the celebrated Huxley's bound [56, 53], we have

$$\pi(x + y) - \pi(x) \sim y / \log x$$

for Huxley's range $x^{7/12} \leq y \leq x$. Setting $x = 2^{5\lambda}$ and $y = 2^{3\lambda}$, while noticing that $7/12 < 3/5$ yields

$$\pi(2^{5\lambda} + 2^{3\lambda}) - \pi(2^{5\lambda}) \sim \frac{2^{3\lambda}}{5\lambda} \ . \tag{8}$$

The approximation Equation (8) means that for any $\epsilon > 0$, there exists $\lambda_0 \in \mathbb{R}_{>0}$ such that for sufficiently large $\lambda > \lambda_0$, the number of primes between $2^{5\lambda}$ and $2^{5\lambda} + 2^{3\lambda}$ satisfies

$$\left| \pi(2^{5\lambda} + 2^{3\lambda}) - \pi(2^{5\lambda}) - \frac{2^{3\lambda}}{5\lambda} \right| \leq \epsilon \ . \tag{9}$$

We choose $\epsilon \coloneqq \frac{1}{5} > 0$ and (9) implies: for sufficiently large λ

$$\left|\pi(2^{5\lambda}+2^{3\lambda}) - \pi(2^{5\lambda}) - \frac{2^{3\lambda}}{5\lambda}\right| \leq \frac{1}{5} \Rightarrow \frac{2^{3\lambda}}{5} - \frac{1}{5} \leq \pi(2^{5\lambda}+2^{3\lambda}) - \pi(2^{5\lambda})$$
$$\Rightarrow \frac{2^{3\lambda}-\lambda}{5\lambda} \leq \pi(2^{5\lambda}+2^{3\lambda}) - \pi(2^{5\lambda}) \ .$$

In other words, we have

$$\pi(2^{5\lambda}+2^{3\lambda}) - \pi(2^{5\lambda}) = \Omega\left(\frac{2^{3\lambda}-\lambda}{\lambda}\right) = \Omega\left(2^{2\lambda}\right)$$

and the claim is proved.

D.2 Proof of Lemma 1

Proof. We write $N = \prod_{i=1}^{k} p_i^{\nu_i}$ for some $k \in \mathbb{N}$ and $p_i \in \mathcal{S}_e$ where $p_i > 2$ as N is odd. We denote by $\ell(\lambda) : \mathbb{N} \to \mathbb{N}$ a polynomial dictating the bit length of N. Then, since $3 < N$ it holds that

$$2^{\ell(\lambda)} > N > \phi(N) = \prod_{i=1}^{k} p_i^{\nu_i - 1}(p_i - 1)$$
$$> \prod_{i=1}^{k} 2 > 2^k \tag{10}$$

and thus $k < \ell(\lambda)$, *i.e.* the number of distinct prime factors of $\phi(N)$ is at most $\ell(\lambda)$.

Moreover, we have $\langle g^e \rangle \subsetneq \langle g \rangle$ if and only if $e \mid \mathsf{ord}(g)$. Because $\mathbb{G} = \langle g \rangle \subseteq \mathbb{Z}_N^*$, we have $\mathsf{ord}(g) \mid \phi(N)$ and from (10) it follows that the number of distinct prime factors of $\mathsf{ord}(g)$ is also at most $\ell(\lambda)$. Consequently, this implies

$$\Pr\left[\langle g^e \rangle \subsetneq \langle g \rangle : e \leftarrow \mathcal{S}_e\right] \leq \Pr\left[e \mid \mathsf{ord}(g) : e \leftarrow \mathcal{S}_e\right]$$
$$\leq \frac{k}{|\mathcal{S}_e|} = O\left(\frac{\ell(\lambda)}{2^\lambda}\right) = \mathrm{negl}(\lambda)$$

by the fact that $k < \ell(\lambda)$, $\mathcal{S}_e$ contains at least $\Omega(2^\lambda)$, as well as $\ell(\lambda)$ is a polynomial in λ[18]. The proof is completed. □

D.3 Blindness Proof of $\mathsf{BS}_{\mathsf{fis}}$

Proof. We proceed by a sequence of hybrids. We denote by $\mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},Game\ i}(\lambda)$ to be the probability that a PPT adversary $\mathcal{A}$ outputs 1 in *Game i*. Without loss of generality, we assume that all the **check** steps are passed during the execution.

Game 1: We start with the game following Definition 12 where $\mathsf{coin} = 0$.

Game 2: This hybrid is the same as Game 1, except that we use the subversion zero-knowledge simulator $\mathsf{Sim}_{\mathsf{fis}} = (\mathsf{Sim}_{\mathsf{H},\mathsf{fis}}, \mathsf{Sim}_{\pi,\mathsf{fis}})$ of Π_{fis} to simulate π_{fis} in the derived signature $\sigma = (\pi_{\mathsf{fis}}, c_I)$. Game 2 differs from Game 1 in the following details. We program the unstructured reference string $\mathsf{urs}_{\mathsf{fis}}$ in $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$. The blindness adversary $\mathcal{A}$ also sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}$ in bvk. We program $\mathsf{H}_{\mathsf{fis}}$ by $\mathsf{Sim}_{\mathsf{H},\mathsf{fis}}$ for further RO queries. Then, run $\pi_{\mathsf{fis}} \leftarrow \mathsf{Sim}_{\pi,\mathsf{fis}}(\mathsf{crs}_{\mathsf{fis}}, x_{\mathsf{fis}})$. The following Lemma 6 argues that Game 2 and Game 1 are indistinguishable. In particular, for any blindness adversary $\mathcal{A}$, there exist PPT $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ so that

$$|\mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},\mathrm{Game}\ 2}(\lambda) - \mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},\mathrm{Game}\ 1}(\lambda)| \leq \mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{sub}}}(\lambda) + \mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_2,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_3,\Pi_{\mathsf{fis}}}(\lambda)$$

and is negligible in λ.

[18] In our blind signature scheme $\mathsf{BS}_{\mathsf{fis}}$ (Figure 1) we set $\ell(\lambda) \coloneqq 2\lambda$.

Lemma 6. *Under the* subversion zero-knowledge *of* Π_{fis} *as well as the* adaptive soundness *of* Π_{sub} *and* Π_{gen}*, the games Game 2 and Game 1 are indistinguishable. for any blindness adversary* $\mathcal{A}$*, there exist PPT* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *so that*

$$|\mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},Game\ 2}(\lambda) - \mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},Game\ 1}(\lambda)| \leq \mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{sub}}}(\lambda) + \mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_2,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_3,\Pi_{\mathsf{fis}}}(\lambda) \ .$$

Proof. By construction, with respect to the relation $\mathsf{R}_{\mathsf{fis}}$, the values $e, (c_I, d_I), r_I$ determined by the user satisfy:

$$\begin{cases} e \equiv 1 \bmod 2 \\ (c_I, d_I) = \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_I, (a, e - \overline{E}); r_I) = 1, \\ e \in \mathcal{S}_e \end{cases} .$$

We also recall that $c = h_2^{\overline{m}} \cdot g^{re} \bmod N$ in the first message to the blindness adversary $\mathcal{A}$ and $y \leftarrow z \cdot g^{-r} \bmod N$ during the signature derivation are both computed by the user. Moreover, we note that since all the **check** steps are passed during the execution, it holds $a \in \mathcal{S}_a$ as a part in the relation $\mathsf{R}_{\mathsf{fis}}$. Now, using the simulation as described in Game 2, there are three cases to treat as follows:

Case 1**:** Suppose that $x_{\mathsf{fis}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{fis}}}$ and $y^e \not\equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N$. This implies

$$\begin{aligned} & y^e && \not\equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \\ \Rightarrow\ & z^e \cdot g^{-re} && \not\equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \\ \Rightarrow\ & z^e && \not\equiv h \cdot h_1^a \cdot h_2^{\overline{m}} \cdot g^{re} \cdot h_2^a \bmod N \\ \Rightarrow\ & z^e && \not\equiv h \cdot h_1^a \cdot c \cdot h_2^a \bmod N \\ \Rightarrow\ & z && \not\equiv (h \cdot h_1^a \cdot c \cdot h_2^a)^d \bmod N \end{aligned}$$

Therefore, we can define $z' \coloneqq h \cdot h_1^a \cdot c \cdot h_2^a \bmod N$ to obtain an instance $x_{\mathsf{sub}} = (z, N, z')$ that breaks the soundness of Π_{sub}. We recall that the **check** $\Pi_{\mathsf{sub}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{sub}}}(\mathsf{crs}_{\mathsf{sub}}, x_{\mathsf{sub}}, \pi_{\mathsf{sub}}) = 1$ is supposed to hold when the user derives the signature, without loss of generality. We provide a PPT adversary $\mathcal{B}_1$ agaisnt the soundness of Π_{sub} that outputs $x_{\mathsf{sub}} = (z, N, z')$ as follows:

- $\mathcal{B}_1$ simulates Game 2 by programming the unstructured reference string $\mathsf{urs}_{\mathsf{fis}}$ in $\mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{gen}} \in \mathcal{URS}$. Then $\mathcal{B}_1$ receives $\mathsf{urs}_{\mathsf{sub}}$ from its soundness challenger.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}$ in bvk.
- $\mathcal{B}_1$ simulates the rest of Game 2 to $\mathcal{A}$: computes then sends $(c, c_Z, \pi_{\mathsf{ped}})$, simulates $\mathsf{H}_{\mathsf{fis}}$ and queries the RO for other $\mathsf{H}_{\mathsf{sub}}, \mathsf{H}_{\mathsf{ped}}$ queries.
- As soon as $\mathcal{A}$ outputs $(z, a, \pi_{\mathsf{sub}})$, $\mathcal{B}_1$ defines $z' \coloneqq h \cdot h_1^a \cdot c \cdot h_2^a \bmod N$ and outputs the instance $x_{\mathsf{sub}} = (z, N, z')$ to its challenger against the soundness of Π_{sub}.

The probability of this case is bounded by $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{sub}}}(\lambda)$ for some PPT $\mathcal{B}_1$ against the soundness of Π_{sub}.

Case 2**:** Suppose that $x_{\mathsf{fis}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{fis}}}$ and $y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N$ but $y \notin \langle h_1 \rangle$. Due to the hypotheses that $y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N$ and $y \notin \langle h_1 \rangle$, we have $\langle h_1 \rangle \neq \langle h \rangle$ or $\langle h_1 \rangle \neq \langle h_2 \rangle$. Recalling that without loss of generality we are supposing all the **check** steps are passed during the execution, in particular $\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1$. This means we obtain an instance $(N, 3, g, (h, h_1, h_2))$ that breaks the soundness of Π_{gen}.

We provide a PPt adversary $\mathcal{B}_2$ breaking the soundness of Π_{gen} as follows:

- $\mathcal{B}_2$ simulates Game 2 by programming the unstructured reference string $\mathsf{urs}_{\mathsf{fis}}$ in $\mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$. Then $\mathcal{B}_2$ receives $\mathsf{urs}_{\mathsf{gen}}$ from its soundness challenger.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}$ in bvk.
- Specifically, as soon as $\mathcal{A}$ outputs

$$\mathsf{bvk} = (\mathsf{crs}_{\mathsf{fis}}, \mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}, N, h, h_1, h_2, g, \pi_{\mathsf{gen}})$$

$\mathcal{B}_2$ outputs the instance $(N, 3, g, (h, h_1, h_2))$ to its challenger against the soundness of Π_{gen}.

Hence, the probability of this case is bounded by $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_2,\Pi_{\mathsf{gen}}}(\lambda)$ for some PPT $\mathcal{B}_2$ against the soundness of Π_{gen}.

Case 3: Finally, suppose that $x_{\mathsf{fis}} \in \mathscr{L}_{\mathsf{R}_{\mathsf{fis}}}$. The adversary $\mathcal{A}$ can be used to construct a PPT $\mathcal{B}_3$ against the subversion zero-knowledge (S-ZK) game of Π_{fis} as below:

- $\mathcal{B}_3$ receives $\mathsf{urs}_{\mathsf{fis}}$ from the S-ZK challenger and program $\mathsf{urs}_{\mathsf{fis}}$ intp the output of $\mathsf{H}_{\mathsf{urs}}(0)$, together with honest $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{crs}_{\mathsf{fis}}$ as part of bvk. $\mathcal{B}_3$ parses $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ and outputs $\mathsf{srs}_{\mathsf{fis}}$ to the S-ZK challenger for Π_{fis}.
- The blindness game for $\mathcal{A}$ is simulated by $\mathcal{B}_3$: computes then sends $(c, c_Z, \pi_{\mathsf{ped}})$ to $\mathcal{A}$, simulates $\mathsf{H}_{\mathsf{fis}}$ and queries the RO for other $\mathsf{H}_{\mathsf{sub}}, \mathsf{H}_{\mathsf{ped}}$ queries, receives $(z, a, \pi_{\mathsf{sub}})$ from $\mathcal{A}$. At the step of derived signature, $\mathcal{B}_3$ queries its S-ZK challenger on

$$x_{\mathsf{fis}} = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, c_I), w_{\mathsf{fis}} = (e, a, y, r_I, d_I)$$

  to get π_{fis}. We note that $\mathcal{B}_3$ possesses the witness w_{fis} throughout the signing session that is simulated to $\mathcal{A}$ (see Figure 1). Then $\mathcal{B}_3$ outputs $(\pi_{\mathsf{fis}}, c_I)$ as the derived signature.
- $\mathcal{B}_3$ outputs what $\mathcal{A}$ outputs.

We argue that $\mathcal{B}_3$ is breaking S-ZK of Π_{fis}:

- Following Definition 20, Game 2 corresponds to the simulated case in the S-ZK game for Π_{fis}, where $\mathcal{B}_3$ receives $\mathsf{urs}_{\mathsf{fis}}$ and outputs a possibly subverted $\mathsf{srs}_{\mathsf{fis}}$, then interacts with $\mathsf{Sim}_{\mathsf{H},\mathsf{fis}}$. The proofs in the derived signatures by $\mathcal{B}_3$ during signing sessions with $\mathcal{A}$ are simulated by $\pi_{\mathsf{fis}} \leftarrow \mathsf{Sim}_{\pi,\mathsf{fis}}(\mathsf{crs}_{\mathsf{fis}}, x_{\mathsf{fis}})$, where $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$.
- On the other hand Game 1 correspond to the real case in Definition 20, where the adversary receives $\mathsf{urs}_{\mathsf{fis}}$ and output a possibly subverted $\mathsf{srs}_{\mathsf{fis}}$, then interacts with H. The proofs in the derived signatures by $\mathcal{B}_3$ during signing sessions with $\mathcal{A}$ are computed by $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ where $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$.

Conditioned on the foregoing case, the advantage that $\mathcal{A}$ can distinguish Game 2 from Game 1 is bounded by $\mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_3,\Pi_{\mathsf{fis}}}(\lambda)$ against the subversion zero-knowledge property of Π_{fis}.

Totally, the probability that $\mathcal{A}$ can distinguish Game 2 from Game 1 is bounded by

$$\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{sub}}}(\lambda) + \mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_2,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_3,\Pi_{\mathsf{fis}}}(\lambda)$$

for PPT adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ as described above. Assuming the *subversion zero-knowledge* of Π_{fis} as well as the *adaptive soundness* of Π_{sub} and Π_{gen} against all such PPT $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$, Game 2 are indistinguishable from Game 1. □

Game 3: This hybrid is the same as Game 2, except that we make c_I independent of the blindness adversary's response $(z, a, \pi_{\mathsf{sub}})$. More specifically, we change the computation $(c_I, d_I) \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_I, (0,0), r_I)$ for $r_I \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathcal{C}_{\mathsf{rnd}}$. We argue that this change is indistinguishable using the fact that r_I is information theoretically hidden thanks to the simulation of π_{fis} from Game 2 as well as the hiding property of $\mathsf{C}_{\mathsf{RInt}}$. Indeed, we construct a simulator $\mathcal{B}$ against the hiding game of $\mathsf{C}_{\mathsf{RInt}}$ that simulates Game 3. At the time of computing c_I, $\mathcal{B}$ outputs two messages $(a, e - \overline{E})$ and $(0,0)$ when interacting with the hiding game's challenger, to receive c_I. Finally, $\mathcal{B}$ uses c_I in the derived signature $\sigma = (\pi_{\mathsf{fis}}, c_I)$ to the blindness adversary $\mathcal{A}$ and outputs what $\mathcal{A}$ outputs. We have

$$|\mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},\text{Game }3}(\lambda) - \mathsf{Adv}^{\mathrm{blind}}_{\mathcal{A},\text{Game }2}(\lambda)| \leq \mathsf{Adv}^{\mathrm{hide}}_{\mathcal{B},\mathsf{C}_{\mathsf{RInt}}}(\lambda)$$

and is negligible in λ.

Game 4: This hybrid is the same as Game 3, except that we use the subversion zero-knowledge simulator $\mathsf{Sim}_{\mathsf{ped}} = (\mathsf{Sim}_{\mathsf{H},\mathsf{ped}}, \mathsf{Sim}_{\pi,\mathsf{ped}})$ of Π_{ped} to simulate π_{ped} in the first message $(c, c_Z, \pi_{\mathsf{ped}})$. Game 4 differs from Game 3 in the following details. We program the unstructured reference string $\mathsf{urs}_{\mathsf{ped}}$ in $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$. The blindness adversary $\mathcal{A}$ also sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$ along with $\mathsf{crs}_{\mathsf{fis}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}$ in bvk. We also program $\mathsf{H}_{\mathsf{ped}}$ by $\mathsf{Sim}_{\mathsf{H},\mathsf{ped}}$ for further RO queries. We afterwards run $\pi_{\mathsf{ped}} \leftarrow \mathsf{Sim}_{\pi,\mathsf{ped}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}})$.

The following Lemma 7 argues that this simulation of π_{ped} is indistinguishable from the real proofs. The following Lemma 7 argues that Game 4 and Game 3 are indistinguishable. In particular, for any blindness adversary $\mathcal{A}$, there exist PPT $\mathcal{B}_1, \mathcal{B}_2$ so that

$$|\mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game }4}(\lambda) - \mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game }3}(\lambda)| \leq \mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_2,\Pi_{\mathsf{ped}}}(\lambda)$$

and is negligible in λ.

Lemma 7. *Under the* subversion zero-knowledge *of* Π_{ped} *as well as the* adaptive soundness *of* Π_{gen}*, the games Game 4 and Game 3 are indistinguishable. For any blindness adversary* $\mathcal{A}$*, there exist PPT* $\mathcal{B}_1, \mathcal{B}_2$ *so that*

$$|\mathsf{Adv}^{\text{blind}}_{\mathcal{A},Game\ 4}(\lambda) - \mathsf{Adv}^{\text{blind}}_{\mathcal{A},Game\ 3}(\lambda)| \leq \mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_2,\Pi_{\mathsf{ped}}}(\lambda) \ .$$

Proof. By construction, with respect to the relation $\mathsf{R}_{\mathsf{ped}}$, the values $\overline{m}, r, (c_Z, d_Z)$ determined by the user satisfy:

$$\begin{cases} \mathsf{C_Z.Verify}(\mathsf{pp}, c_Z, (\overline{m}, r), d_Z) = 1 \\ \overline{m} \in [0, 2^\lambda - 1] \\ r \in [0, S] \end{cases} .$$

We recall that $c = h_2^{\overline{m}} \cdot g^{re} \bmod N$ is computed by the user in this Game 4, as part of the first message that is sent to the adversarial signer. Now, using the simluation as described in Game 2, there are three cases to treat as follows:

Case 1 Suppose $x_{\mathsf{ped}} \notin \mathscr{L}_{\mathsf{R}_{\mathsf{ped}}}$ and $c \notin \langle g \rangle$. This implies $\langle g \rangle \neq \langle h_2 \rangle$. As we are supposing $\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1$, without loss of generality so that all **check** pass, the instance $(N, 3, g, (h, h_1, h_2))$ breaks the soundness of Π_{gen}. We provide a PPT adversary $\mathcal{B}_1$ breaking the soundness of Π_{gen} as follows:
- $\mathcal{B}_1$ simulates Game 4 by programming the unstructured reference string $\mathsf{urs}_{\mathsf{ped}}$ in $\mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$. Then $\mathcal{B}_1$ receives $\mathsf{urs}_{\mathsf{gen}}$ from its soundness challenger.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}, \mathsf{sub}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}$ in bvk.
- Specifically, as soon as $\mathcal{A}$ outputs

$$\mathsf{bvk} = (\mathsf{crs}_{\mathsf{fis}}, \mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}, \mathsf{crs}_{\mathsf{sub}}, N, h, h_1, h_2, g, \pi_{\mathsf{gen}})$$

  $\mathcal{B}_1$ outputs the instance $(N, 3, g, (h, h_1, h_2))$ to its challenger against the soundness of Π_{gen}.

Hence, the probability of this case is bounded by $\mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda)$ for some PPT $\mathcal{B}_1$ against the soundness of Π_{gen}.

Case 2 Suppose $x_{\mathsf{ped}} \in \mathscr{L}_{\mathsf{R}_{\mathsf{ped}}}$. The adversary $\mathcal{A}$ can be used to construct a PPT $\mathcal{B}_2$ against the subversion zero-knowledge (S-ZK) game of Π_{ped} aas follows:
- $\mathcal{B}_2$ receives $\mathsf{urs}_{\mathsf{ped}}$ from the S-ZK challenger and program $\mathsf{urs}_{\mathsf{ped}}$ intp the output of $\mathsf{H}_{\mathsf{urs}}(0)$, together with honest $\mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}, \mathsf{urs}_{\mathsf{sub}} \in \mathcal{URS}$.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{crs}_{\mathsf{ped}}$ as part of bvk. $\mathcal{B}_2$ parses $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$ and outputs $\mathsf{srs}_{\mathsf{ped}}$ to the S-ZK challenger for Π_{ped}.
- The blindness game for $\mathcal{A}$ is simulated by $\mathcal{B}_2$. First of all $\mathcal{B}_2$ queries its S-ZK challenger on

$$x_{\mathsf{ped}} = (\mathsf{pp}, N, e, h_2, g, c, c_Z), w_{\mathsf{ped}} = (\overline{m}_0, r, d_Z)$$

  to get π_{ped}. We note that $\mathcal{B}_2$ possesses the witness w_{ped}, where $\overline{m}_0 \coloneqq \mathsf{H}(m_0)$, throughout the signing session that is simulated to $\mathcal{A}$ (see Figure 1). Then $\mathcal{B}_2$ sends $(c, c_Z, \pi_{\mathsf{ped}})$ to $\mathcal{A}$, queries the RO for other $\mathsf{H}_{\mathsf{sub}}, \mathsf{H}_{\mathsf{fis}}$ queries, receives $(z, a, \pi_{\mathsf{sub}})$ from $\mathcal{A}$. Finally, $\mathcal{B}_2$ derives and outputs $(\pi_{\mathsf{fis}}, c_I)$ as the derived signature.
- $\mathcal{B}_2$ outputs what $\mathcal{A}$ outputs.

We argue that $\mathcal{B}_2$ is breaking S-ZK of Π_{ped}:

– Following Definition 20, Game 4 corresponds to the simulated case, where the adversary receives $\mathsf{urs}_{\mathsf{ped}}$ and outputs a possibly subverted $\mathsf{srs}_{\mathsf{ped}}$, then interacts with $\mathsf{Sim}_{\mathsf{H},\mathsf{ped}}$. The proofs in the derived signatures by $\mathcal{B}_2$ during signing sessions with $\mathcal{A}$ are simulated by $\pi_{\mathsf{ped}} \leftarrow \mathsf{Sim}_{\pi,\mathsf{ped}}(\mathsf{crs}_{\mathsf{ped}}, x_{\mathsf{ped}})$, where $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$.
– On the other hand Game 3 correspond to the real case in Definition 20, where the adversary receives $\mathsf{urs}_{\mathsf{ped}}$ and output a possibly subverted $\mathsf{srs}_{\mathsf{ped}}$, then interacts with H. The proofs in the derived signatures by $\mathcal{B}_2$ during signing sessions with $\mathcal{A}$ are computed by $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs}, x, w)$ where $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$.

Conditioned on the foregoing case, the advantage that $\mathcal{A}$ can distinguish Game 4 from Game 3 is bounded by $\mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_2,\Pi_{\mathsf{ped}}}(\lambda)$ against the subversion zero-knowledge property of Π_{ped}.

Totally, the probability that $\mathcal{A}$ can distinguish Game 4 from Game 3 is bounded by

$$\mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda) + \mathsf{Adv}^{\text{sub-zk}}_{\mathcal{B}_2,\Pi_{\mathsf{ped}}}(\lambda)$$

for some PPT $\mathcal{B}_1, \mathcal{B}_2$. Assuming the *subversion zero-knowledge* of Π_{ped} as well as the *adaptive soundness* of Π_{gen} against all such PPT $\mathcal{B}_1, \mathcal{B}_2$, Game 4 are indistinguishable from Game 3. □

Game 5: This hybrid is the same as Game 4, except that we replace c in the first user's message by $c \leftarrow \langle g \rangle$. This transition is *statistical*. By the union bound, the advantage of any possibly unbounded adversary $\mathcal{A}$ to distinguish between this Game 5 and the previous Game 4 can be bounded by considering two cases:

Case 1 The replacement $c \leftarrow \langle g \rangle$ is distinguishable from the previous computation

$$c = h_2^{\overline{m}} \cdot g^{re} \bmod N$$

in Game 4 because $\langle g \rangle \neq \{h_2^x \cdot g^y \bmod N \mid x, y \in \mathbb{N}\}$. This implies that $\langle h_2 \rangle \neq \langle g \rangle$ and under our hypothesis that $\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(\mathsf{crs}_{\mathsf{gen}}, x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1$, this implies the adversary $\mathcal{A}$ can output $(N, 3, g, (h, h_1, h_2))$ that breaks the soundness of Π_{gen}. We provide a PPT adversaary $\mathcal{B}_1$ breaking the soundness of Π_{gen} in the same manner as **Case** 1 in Game 4. The probability of this case is bounded by $\mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda)$ for some PPT $\mathcal{B}_1$ against the soundness of Π_{gen}.

Case 2 Else, suppose that $\langle h_2 \rangle = \langle g \rangle$. By Lemma 1 under the fact that $\mathsf{H}_{\mathbb{P}}$ is uniform over $\mathcal{S}_e$ where $|\mathcal{S}_e| = \Omega(2^{2\lambda})$, with overwhelming probability we have $\langle g^{re} \rangle = \langle g \rangle$. This means we can write $c = h_2^{\overline{m}} \cdot \bar{g}^r \bmod N$ for some generator $\bar{g} \coloneqq g^e$ of $\langle g \rangle = \langle h_2 \rangle$, thus has the form of a Pedersen commitment over $\langle g \rangle$. Therefore, because $r \leftarrow [0, S]$, where $S = N \cdot 2^{\lambda}$ is exponentially large in λ, Remark 2 implies the statistical hiding of the commitment $c = h_2^{\overline{m}} \cdot \bar{g}^r \bmod N$ that encures the advantage of distinguishing of $\mathcal{A}$ in this case is $\mathrm{negl}(\lambda)$.

By combining the two cases, we conclude that the probability a blindness adversary $\mathcal{A}$ can distinguish Game 5 from Game 4 is bounded by $\mathsf{Adv}^{\text{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda) + \mathrm{negl}(\lambda)$, for some PPT $\mathcal{B}_1$, and thus negligible under the soundness of Π_{gen}.

Game 6: This hybrid is the same as Game 5, except that we makes c_Z independent of the adversary's response. More specifically, we change the computation $(c_Z, d_Z) \leftarrow \mathsf{C}_{\mathsf{RInt}}.\mathsf{Commit}(\mathsf{pp}_Z, (0, r))$ for $r \leftarrow [0, S]$. We argue that this change is indistinguishable by constructing a simulator $\mathcal{B}$ against the hiding game of $\mathsf{C}_{\mathsf{RInt}}$ that simulates Game 6. At the time of computing c_Z, $\mathcal{B}$ outputs two messages $(\overline{m}_0, r)$ and $(0, 0)$ when interacting with the hiding game's challenger, to receive c_Z. We are using the fact that r is information theoretically hidden thanks to the simulation of π_{fis} from Game 2, the simulation of π_{ped} from Game 4, and the replacement of the commitment $c \leftarrow \langle g \rangle$ from Game 5. Finally, $\mathcal{B}$ uses c_Z in the first message $(c, c_Z, \pi_{\mathsf{ped}})$ to the blindness adversary $\mathcal{A}$ and outputs what $\mathcal{A}$ outputs. We have

$$|\mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game } 6}(\lambda) - \mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game } 5}(\lambda)| \leq \mathsf{Adv}^{\text{hide}}_{\mathcal{B},\mathsf{C}_{\mathsf{RInt}}}(\lambda)$$

and is negligible in λ.

Game 7: We note that after hopping to Game 6, the first message $(c, c_Z, \pi_{\mathsf{ped}})$ as well as the derived signature $(\pi_{\mathsf{fis}}, c_I)$ do not depend on $\overline{m}_0$ anymore. We then apply a similar sequence of hoppings, but symmetrically in a reverse order to go to the game following Definition 12 where $\mathsf{coin} = 1$, *i.e.* $\overline{m}_1$ is used in the first message and the derived signature. The above arguments still apply so that the transitions stay indistinguishable. In total, we have proved that

$$2 \cdot \mathsf{Adv}^{\text{blind}}_{\mathcal{A},\mathsf{BS}_{\mathsf{fis}}}(\lambda) = \left|\mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game }1}(\lambda) - \mathsf{Adv}^{\text{blind}}_{\mathcal{A},\text{Game }7}(\lambda)\right|$$

is neglgible in λ and the proof is completed. □

D.4 One-More Unforgeability Proof of $\mathsf{BS}_{\mathsf{fis}}$

Proof. We prove this using a series of games to rule out some cases in which the reduction won't work.

Game 1: This is the one-more-unforgeability game.

Game 2: In this game we introduce an abort condition. Namely, the game aborts if there is a collision in the hash oracle H, i.e. if the adversary during the game makes two queries ζ, ζ' to H such that $\mathsf{H}(\zeta) = \mathsf{H}(\zeta')$, but $\zeta \neq \zeta'$.

Claim. $\mathsf{Adv}_{\mathcal{A},\text{Game }1}(\lambda) - \mathsf{Adv}_{\mathcal{A},\text{Game }2}(\lambda) \leq Q_{\mathsf{H}}^2/2 \cdot 2^{2\lambda}$

Proof. Birthday bound.

Game 3: In this game we introduce an abort condition. Namely, the game aborts if there is a collision in the hash oracle $\mathsf{H}_{\mathbb{P}}$, i.e. if the adversary during the game makes two queries ζ, ζ' to $\mathsf{H}_{\mathbb{P}}$ such that $\mathsf{H}_{\mathbb{P}}(\zeta) = \mathsf{H}_{\mathbb{P}}(\zeta')$, but $\zeta \neq \zeta'$.

Claim. $\mathsf{Adv}_{\mathcal{A},\text{Game }1}(\lambda) - \mathsf{Adv}_{\mathcal{A},\text{Game }2}(\lambda) \leq \text{negl}(\lambda)$

Proof. Birthday bound.

Game 4: In this game, we alter how the parameters for C_{Z} are set up. Namely, we use the algorithm $\Pi_{\mathsf{ped}}.\mathsf{Ext}_1$ to set up the parameters for C_{Z} as $(x_0, \mathsf{td}) \leftarrow \Pi_{\mathsf{ped}}.\mathsf{Ext}_1(1^\lambda)$ and we program the random oracle H_{pp} so that it returns x_0 as pp_Z. Apart from this, Game 4 behaves identically to Game 3. As the parameters pp_Z are chosen uniformly at random by Ext_1, this game is identically distributed to the previous one and we get $\mathsf{Adv}_{\mathcal{A},Game4}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game3}(\lambda)$.

Game 5: In this game, we introduce another abort condition, namely the game aborts if there exists a signing session where no witness can be extracted from π_{ped}. The game now extracts the values $\overline{m}, r$ for every signing session. This game hop can be bounded by the Partial Online-Extractability of Π_{ped}. We formalize this in the following claim:

Claim. There exists a PPT adversary $\mathcal{B}_1$ against the online-extractability of Π_{ped} such that $\mathsf{Adv}_{\mathcal{A},Game\ 5}(\lambda) \geq \frac{\mathsf{Adv}_{\mathcal{A},Game\ 4}(\lambda) - \text{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)}$ where we plugged in $\mathsf{Adv}_{\mathcal{A},Game4}(\lambda)$ as $\mu(\lambda)$ from Definition 22 and $\text{negl}, p_{\mathsf{P}}$ are as in Definition 22.

Proof. We provide an adversary $\mathcal{B}_1$ against the online-extractability of Π_{ped} to bound the distance between the two games. The adversary receives the simulated CRS $\overline{\mathsf{crs}}$ for Π_{ped}. It then simulates Game 4 to the adversary $\mathcal{A}$ by sampling all the other parts of vk as in Game 4 and answering the signing queries using the secret key. It outputs the proofs of Π_{ped} that the adversary sent when opening a new signing session. The online-extractability of Π_{ped} yields the claim.

Remark 4. We note that as the commitment scheme C_{Z} is perfectly binding, and the above online extraction property guarantees the existence of a full witness, there cannot be two sessions using the same commitment c_Z with different messages $\overline{m}, \overline{m}'$ and different r, r'. Thus, it follows that if $\overline{m} \neq \overline{m}'$, also $c_Z \neq c'_Z$ and $\mathsf{H}_{\mathbb{P}}(c_Z) \neq \mathsf{H}_{\mathbb{P}}(c'_Z)$ due to the abort condition introduced in Game 3.

Game 6: This game aborts if among the message-signature pair in the adversary's output there is a message for which the adversary has never queried $\mathsf{H}(m)$.

Claim. $\mathsf{Adv}_{\mathcal{A},Game5} - \mathsf{Adv}_{\mathcal{A},Game6}(\lambda) \leq \frac{1}{2^{2\lambda}}$.

Proof. This boils down to the adversary having to guess the hash value $\overline{m} = \mathsf{H}(m)$. As H is a random oracle mapping into $\{0,1\}^{2\lambda}$, the probability of guessing a uniformly random value from this space is $2^{2\lambda}$

Game 7: In this game, the PRF is replaced by a truly random function that is sampled via lazy-evaluation. This game hop is justified by the pseudo-randomness of the PRF. More formally:

Claim. There exists a PPT adversary $\mathcal{B}_2$ such that $\mathsf{Adv}_{\mathcal{A},Game\ 7}(\lambda) - \mathsf{Adv}_{\mathcal{A},Game\ 6}(\lambda) \leq \mathsf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_2}(\lambda)$.

Proof. We provide the reduction $\mathcal{B}_2$.

The reduction has access to the PRF real-or-random oracle. It samples all keys like in Game 6 except for sampling a PRF key. Whenever the adversary makes a signing query, instead of evaluating the PRF, the reduction $\mathcal{B}_2$ queries the PRF oracle. If the adversary wins the game it outputs 1, otherwise 0. It is easy to see that this reduction has the advantage as in the claim.

Game 8: In this game, the game samples all random choices that the signer and the random oracle make at the beginning of the game. As this change is purely conceptual, it holds that

$$\mathsf{Adv}_{\mathcal{A},Game8}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game7}(\lambda)$$

Game 9: In this game, we change how the CRS for Π_{fis} is generated. Namely, we instead of generating $\mathsf{crs}_{\mathsf{fis}}$ using GenCRS, we switch to generating $\mathsf{crs}_{\mathsf{fis}}$ using SimCRS. This game hop can be bounded by the CRS indistinguishability property of Π_{fis}.

Claim. There exists a reduction $\mathcal{B}_3$ such that $\mathsf{Adv}_{\mathcal{A},Game9} - \mathsf{Adv}_{\mathcal{A},Game8} \leq \mathsf{Adv}^{\mathrm{crs}}_{\mathcal{B}_3}(\lambda)$

Proof. The reduction $\mathcal{A}_3$ receives a CRS from the CRS indistinguishability challenger.

It samples all other parts of the verification key as in Game 8 and outputs them to the adversary. It answers signing queries as in Game 8. If the adversary wins the game it outputs that the CRS was honest, otherwise that it was simulated. It is easy to see that the claim follows.

Game 10: In this game, we switch from generating proofs for Π_{sub} honestly using $\mathsf{H}_{\mathsf{sub}}$ and $\mathcal{P}$ to using $\mathsf{Sim}_{\mathsf{H}_{\mathsf{sub}}}$ and Sim_{π}. This game hops can be bounded by the subversion zero-knowledge property of Π_{sub}. Thus, by Definition 20, we obtain that $\mathsf{Adv}_{\mathcal{A},Game10} - \mathsf{Adv}_{\mathcal{A},Game9} \leq \mathrm{negl}(\lambda)$.

Game 11: In Game 11 we change how the values a_j are sampled during signature generation. In particular, the game samples bits $b, b' \leftarrow \{0,1\}$ and $j \leftarrow \{1, \ldots, Q_{\mathsf{H}_{\mathbb{P}}}\}$.

If $b = 1$, $b' = 0$, instead of sampling $a_j \leftarrow \{0,1\}^{2\lambda}$, it first samples $\beta \leftarrow \{0,1\}^{3\lambda}$ and then sets $a_j = \beta$. If $b = 1$ and $b' = 1$, it samples $\beta \leftarrow \{0,1\}^{3\lambda}$ and sets $a_j = \beta - \mathsf{H}(m_j)$. A simple argument shows that the distribution of a_j in Game 11 has statistical distance at most $1/2^{\lambda}$ from the distribution of a in Game 10.

Thus, we get that $|\Pr[\text{Game } 11 = 1] - \Pr[\text{Game } 10 = 1]| \leq \frac{1}{2^{\lambda}}$.

Game 12: In Game 12, we change how we set up the key vk Namely, instead of using $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}$, we use the alternate algorithm $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{b,b'}$ using N generated as before and $z \leftarrow \mathbb{Z}_N$ and it programs the random oracle $\mathsf{H}_{\mathbb{P}}$ to return primes from $e_1, \ldots e_{Q_{\mathsf{H}_{\mathbb{P}}}}$ Everything else we do as in Game 11. As the keys are distributed the same, it holds that

$$\mathsf{Adv}_{\mathcal{A},Game12}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game11}(\lambda).$$

Game 13: In Game 13, we change how signatures are created. In particular, the game uses the alternate signing algorithms $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{b,b'}$ as follows. As we introduced extraction of $\overline{m}, r$ in Game 5, the game has access to these two values. It therefore applies $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{b,b'}(\mathsf{sk}_{b,b'}, \overline{m})$ to obtain a signature $\sigma = (e, a, y)$. It then computes $y' = y \cdot g^r$ and π_{sub} using $\Pi_{\mathsf{sub}}.\mathsf{Sim}_\pi$. It then outputs y', a and π_{sub} to the adversary.

As the signatures produced by this game are identically distributed to the ones output by Game 12, we obtain that

$$\mathsf{Adv}_{\mathcal{A},Game13}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game12}(\lambda).$$

Game 14: In this game, we use the knowledge soundness property of the NIZK Π_{fis}. Namely, after the adversary has submitted its signatures, we extract a witness from π_{fis}. In particular, due to the changes made in Game 6, for each message that the adversary outputs a signature for, it has to have made a hash query. Further, as we introduced online-extraction of all witnesses of π_{ped} submitted during signing queries, the game can identify the hashes $\overline{m}$ that it has signed and which messages they belong to. As Game 2 aborts if there are collisions in H, there are no collisions in H in Game 14, and therefore, the game can efficiently identify which of the messages submitted as part of the final message-signature pairs it has not signed. It picks the first such message-signature pair (denoted by $(m^*, \pi^*_{\mathsf{fis}})$) and applies the extractor $\Pi_{\mathsf{fis}}.\mathsf{Ext}$ to the proof π^*_{fis} to obtain a signature σ^*. It aborts if this extraction fails.

Claim. $\mathsf{Adv}_{\mathcal{A},Game14} \geq \frac{\mathsf{Adv}_{\mathcal{A},Game13}(\lambda) - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_{\mathsf{H}_{\mathsf{fis}}})}$ where $\mathrm{negl}, p_{\mathsf{P}}$ are as in Definition 21.

Proof. We describe an algorithm $\mathcal{B}_4$ to extract from. The algorithm $\mathcal{B}_4$ takes as input the CRS and generates all other parts of the verification key as Game 13. It then runs $\mathcal{A}$ and answers signing queries as Game 13. Once the adversary outputs its message signature pairs, it identifies the message-signature pair $(m^*, (c^*_I, \pi^*_{\mathsf{fis}}))$ as described in Game 14 and outputs them.

If $\mathcal{A}$ won the game, π_{fis} is a valid proof and thus $\Pi_{\mathsf{fis}}.\mathsf{Ext}$ will extract a signature σ^* with probability as in Definition 21.

Remark 5. We note that as c_I is perfectly binding, the $\mathsf{S}_{\mathsf{fis}}$-signature contained c_I is already determined after one run of the game and will not be affected by the adversary being rewound.

Game 15: In Game 15 we introduce some abort conditions where the reduction would not be able to solve the sRSA problem.

Namely, if $b = 0$, the game aborts if $e^* \in \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$. If $b = 1$, the game aborts if $e^* \notin \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$. The game aborts if $e^* \neq e_j$ where j is the index j from $\mathsf{sk}_{b,b'}$. Furthermore, if $b = 1$ it identifies the first signing session where e^* was used if such a session exists.

We denote the component a being derived in this session by a_j. If no such session exists, sample $a_j \leftarrow \mathcal{S}_a$. The game aborts if $b' = 0$ and $a^* = a_j$ and if $b' = 1$ and $a^* + \overline{m}^* = a_j + \overline{m}_j$.

We get that

$$\mathsf{Adv}_{\mathcal{A},Game15}(\lambda) = \frac{1}{4Q_{\mathsf{H}_{\mathbb{P}}}} \mathsf{Adv}_{\mathcal{A},Game14}(\lambda).$$

Reduction simulating Game 15 We describe a reduction $\mathcal{B}_5$ that simulates Game 15 amd solves the strong RSA problem.

Setup. The reduction receives a strong RSA challenge N, z from its challenger. The reduction samples $b, b' \leftarrow \{0,1\}$ and then runs $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{b,b'}(1^\lambda, N, z)$ to generate the verification key parts of $\mathsf{S}_{\mathsf{fis}}$. It sets up the NIZK and commitment parameters as in Game 15.

Online Phase. The reduction interacts with the adversary as follows:

Simulation of H This is done via lazy sampling from $\{0,1\}^{2\lambda}$

Simulation of $\mathsf{H}_{\mathbb{P}}$ on the i-th fresh query to $\mathsf{H}_{\mathbb{P}}$, return e_i from $\mathsf{sk}_{b,b'}$.

Simulation of other hash oracles Lazy Sampling apart from whatever is defined through the setup of the NIZKs

Answering Signing Queries The reduction extracts the values $(\overline{m}, r)$ from the proof π_{ped} using $\Pi_{\mathsf{ped}}.\mathsf{Ext}$. It then derives e using $\mathsf{H}_{\mathbb{P}}$. By the programming of $\mathsf{H}_{\mathbb{P}}$, it identifies the index k such that $e = e_k \in \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$. It then uses $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{b,b'}(\mathsf{sk}_{b,b'}, k, \overline{m})$ to generate the signature $\sigma = (e, a, y)$. It then re-blinds the signature as $z = y \cdot g^r$ and generates the proof π_{sub} using the simulator $\Pi_{\mathsf{sub}}.\mathsf{Sim}$ of Π_{sub} and outputs z, a, π_{sub}.

Output Determination When the adversary $\mathcal{A}$ outputs its message-signature pairs, the reduction identifies the first message m^* that it has not signed before (as in Game 14). It then uses the extractor $\Pi_{\mathsf{fis}}.\mathsf{Ext}$ to obtain a signature $\sigma^* = (e^*, a^*, y^*)$ The reduction then solves for $z^{\frac{1}{e^*}}$ using the same techniques as in Section 3.

It is easy to see that $\mathcal{B}_5$ simulates Game 15 perfectly.

We briefly describe why solving sRSA works as in Section 3. First of all, Π_{fis} guarantees that $a \in \mathcal{S}_a$ and $e \in \mathcal{S}_e$. Thus $a < e$ and in the case of $b = 1$, we know e^* is prime and thus co-prime to $a_j - a^*$. In the case of $b = 0$ we cannot guarantee primality of e^* as it is chosen by the adversary, however, Π_{fis} still guarantees that e^* is odd and thus the same strategy as in Section 3 can be applied.

E Instantiations of NIZKs

In this section, we instantiate the remaining NIZKs required for $\mathsf{BS}_{\mathsf{fis}}$.

E.1 Instantiation of Π_{sub}

For this NIZK, we use the same techniques as for Π_{gen}. To this end, let $\mathsf{H}_{\mathsf{sub}} \colon \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function modelled as a random oracle. We describe the algorithms:

- $\Pi_{\mathsf{sub}}.\mathsf{GenCRS}(1^\lambda)$: Outputs $\bot$.
- $\Pi_{\mathsf{sub}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{sub}}}(x = (z, N, h), w = d)$: Samples $\vec{\mu} \leftarrow [0, N \cdot 2^\lambda]^\lambda$. Computes $\vec{\Omega} := h^{\vec{\mu}}$ and $\vec{\gamma} := \mathsf{H}_{\mathsf{sub}}(x, \vec{\Omega})$. Then, computes $\vec{\tau} := \vec{\mu} + d \cdot \vec{\gamma}$. Outputs $\pi_{\mathsf{sub}} = (\vec{\Omega}, \vec{\gamma}, \vec{\tau})$.
- $\Pi_{\mathsf{sub}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{sub}}}(x = (z, N, h), \pi_{\mathsf{sub}} = (\vec{\Omega}, \vec{\gamma}, \vec{\tau}))$: Checks that $\vec{\gamma} = \mathsf{H}_{\mathsf{sub}}(x, \vec{\Omega})$ and $\vec{\Omega} \circ z^{\vec{\gamma}} = h^{\vec{\tau}}$ where $\circ$ denotes the Hadamard product.

The correctness, subversion statistical adaptive soundness, as well as zero-knowledge properties follow using the same arguments as in Appendix C.2.

Efficiency. It is well-known that $\vec{\Omega}$ can be omitted from the proof (as it can be recomputed given $\vec{\gamma}$ and $\vec{\tau}$). For an RSA modulus N of 3072 bits, a proof is of size 51.216 KB.

E.2 Instantiation of Π_{ped}

We instantiate the online-extractable NIZK $\mathsf{R}_{\mathsf{ped}}$. We follow the well-known blueprint of combining an extractable commitment (*e.g.*, ElGamal) with an adaptively knowledge sound NIZK for the relation to obtain online-extraction (see, *e.g.*, [61]). Roughly, we decompose the witnesses into short values committed in ElGamal commitments and show that the relation holds with respect to these values. A range proof (*i.e.*, a variant of Bulletproofs [21, 8]) guarantees that the committed values are short to enable online-extraction via a discrete logarithm computation. (The trapdoor is the ElGamal decryption key.) These ElGamal commitments function as the integer commitment C_{Z}. The commitment and its public parameters pp are part of the statement, but since these are sampled uniform, we can embed a trapdoor into pp (cf. Definition 22).

Integer Commitment. Recall that we want to show that $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$, where $\overline{m} \in [0, 2^\lambda - 1]$ and $r \in [0, S]$ are committed in some integer commitment $(c_Z, d_Z) \leftarrow \mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(\mathsf{pp}, (\overline{m}, r))$ with bounded range. Let $B = \mathsf{poly}(\lambda)$ be a power of two. Let $\overline{\mathbb{G}}_p$ be a group with prime order $p \geq 2^{2\lambda}$. To instantiate C_{Z}, we essentially decompose $\overline{m}, r$ into values $(m_i)_i, (r_i)_i \in [0, B-1]$ via a B-ary decomposition, respectively, and commit to the values via ElGamal commitments over $\overline{\mathbb{G}}_p$. Let $\ell_m = \lceil \lambda / B \rceil$ and $\ell_r = \lceil \log_B(S+1) \rceil$. The scheme C_{Z} is defined below.

– $\mathsf{C_Z}.\mathsf{Setup}(1^\lambda)$: Samples $\overline{G}, \overline{H} \leftarrow \overline{\mathbb{G}}_p$ and outputs $\mathsf{pp} \leftarrow \overline{G}, \overline{H}$.
– $\mathsf{C_Z}.\mathsf{Commit}(\mathsf{pp}, (\overline{m}, r))$: Takes as input public parameters pp and message $(\overline{m}, r)$, where $\overline{m} \in [0, 2^\lambda - 1]$ and $r \in [0, S]$. Decomposes $\overline{m} = \sum_{i=1}^{\ell_m} m_i B^{i-1}$ and $r = \sum_{i=1}^{\ell_r} r_i B^{i-1}$. Let $\vec{e} = (m_1, \ldots, m_{\ell_m}, r_1, \ldots, r_{\ell_r}) \in [0, B-1]^{\ell_m+\ell_r}$. Samples $s_i \leftarrow \mathbb{Z}_p$ and sets $E_i = e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G}$. Outputs $c_Z = (E_i, S_i)_{i=1}^{\ell_m+\ell_r}$ and $d_Z = (s_1, \ldots, s_{\ell_m+\ell_r})$.
– $\mathsf{C_Z}.\mathsf{Verify}(\mathsf{pp}, c_Z, (\overline{m}, r), d_Z)$: Parses c_Z and d_Z as above. Decomposes $\overline{m}$ and r into m_i and r_i, respectively, and defines $\vec{e}$ as above. Checks that $E_i = e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G}$ and $e_i \in [0, B]$ for all $i \in [\ell_m + \ell_r]$.

Lemma 8. *The integer commitment scheme with bounded range $\mathsf{C_Z}$ with message space $\mathsf{C_Z}.\mathcal{C}_{\mathsf{msg}} = [0, 2^\lambda - 1] \times [0, S]$ is correct, hiding under DDH in $\overline{\mathbb{G}}_p$ and perfectly binding (cf. Definition 2).*

Proof. Correctness and hiding are straightforward. We sketch binding. Observe that (E_i, S_i) fixes $e_i \bmod p$ perfectly. As $e_i \in [0, B]$ and $\mathsf{poly}(\lambda) = B < 2^\lambda \leq p$, the values e_i are fixed over the integers. These values determine the message $(\overline{m}, r)$ uniquely within $[0, 2^\lambda - 1] \times [0, S]$.

Online-Extractable NIZK. We are now ready to instantiate Π_{ped}. For the above $\mathsf{C_Z}$, we can rewrite the relation $\mathsf{R}_{\mathsf{ped}}$ as follows.

$$\mathsf{R}_{\mathsf{ped}} = \big\{(x, w) \mid c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N, E_i = e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G}, e_i \in [0, B-1], \overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}\big\},$$

for $x = (B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $w = (\overline{m}, r, (s_i)_{i\in[\ell_m+\ell_r]})$. Note that the above relation implies that $\overline{m} \in [0, 2^\lambda - 1]$ and that $r \in [0, S]$. (Also, e_i is uniquely determined by $\overline{m}$ and r via B-ary decomposition.)

To instantiate Π_{ped}, we construct a standard Σ-protocol Σ_{ped} to show that $\mathsf{R}_{\mathsf{ped}}$ holds, except for the statement $e_i \in [0, B-1]$. For the latter, we later use a range proof Π_{rp} from [8]. Then, we compile Σ_{ped} into an NIZK Π_{ped} via Fiat-Shamir and combine both NIZKs Π_{ped} and Π_{rp} into an NIZK for the full relation $\mathsf{R}_{\mathsf{ped}}$ as in [61], Section 6. This approach was shown to be secure in [61].

There is one difficulty that arises during the construction of Σ-protocol: the relations for $\overline{m}$ and r have to hold over the integers. For example, notice that it is *not* sufficient to show that $r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1} \bmod p$ over $\overline{\mathbb{G}}_p$ since the commitment $\mathsf{C_Z}$ is (perfectly) binding only if this relation holds over $\mathbb{Z}$. To ensure that soundness guarantees that the relations hold over $\mathbb{Z}$, we add an additional MPed commitment $\tilde{c}$ over $\mathsf{QR}_{\tilde{N}}$ for a fresh RSA modulus $\tilde{N}$. If we commit to all witnesses (except s_i since these are defined over $\mathbb{Z}_p$) in $\tilde{c}$ and open it in ZK, the extracted values are integers under sRSA (cf. Lemma 5). We can also use MPed commitments to show the statements over the integers (leveraging the binding property of MPed). To ensure subversion zero-knowledge, we add a Π_{gen} proof (cf. Appendix C.2) which ensures that the public parameters of MPed are setup in a manner that ensures hiding.

Below, we provide the protocols Σ_{ped} and Π_{rp}, and then combine them to construct Π_{ped}.

Step 1: the Σ-protocol. Let $C = 2^\lambda$ (which determines the challenge space). Let $\tilde{N} \in \mathbb{N}$ with $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \ldots, \tilde{g}_{\ell_m+\ell_r}) \in (\mathbb{Z}_{\tilde{N}}^*)^{1+\ell_m+\ell_r}$. Denote by R the relation

$$\mathsf{R} = \big\{(x, w) : c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N, E_i = e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G} \quad \overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}\big\},$$

where $x = (\tilde{N}, \tilde{\mathsf{pp}}, B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $w = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$. The protocol Σ_{ped} for relation

$$\mathsf{R}_{\Sigma_{\mathsf{ped}}} = \big\{(x, w) : (x, w) \in \mathsf{R} \text{ and } e_i \in [0, B-1] \text{ and } \langle\tilde{h}\rangle = \langle\tilde{g}_i\rangle\big\},$$

is given in Figure 4. We include the statements $e_i \in [0, B-1]$ and $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$ in the relation $\mathsf{R}_{\Sigma_{\mathsf{ped}}}$ because this is required for correctness and HVZK [19]. For soundness, we use standard Σ-protocol techniques combined with integer commitments to show the statements. Notably, we relax the soundness relation and omit $e_i \in [0, B-1]$ and $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$ (since these statements are shown via a separate NIZK within Π_{ped} later). We remark that to show the decompositions, we use a standard technique that is often used in lattices to show multiplicative relations (*e.g.*, [14]). That is, we let the verifier recompute the statements with the masked witness τ_{e_i} instead of e_i. This yields a polynomials f_m and f_r (if we interpret the challenge γ as variable). If f_m and f_r are constant (*i.e.*, the term multiplied with γ is 0), then the relation holds. This can be efficiently verified if the prover commits to the constant terms $f_{m,0}$ of $f_{r,0}$ in a separate MPed commitment and the verifier checks that $f_{m,0} = f_m$ and $f_{r,0} = f_r$ using the commitment's linearity. Then, we can show 2-special soundness for the relation

$$\widetilde{\mathsf{R}}_{\Sigma_{\mathsf{ped}}} = \{(x, w) : \ (x, w) \in \mathsf{R} \text{ or } (\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}}) \text{ or } (\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{\mathsf{dlog}}\},$$

where $\mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$ is defined in Definition 3 and $\mathsf{R}_{\mathsf{dlog}}$ denotes the relation that contains all non-trivial DLOG relations in $\tilde{\mathsf{pp}}$ (see [8] for more details). Note that under the factoring assumption, it is hard to find a witness for $\mathsf{R}_{\mathsf{dlog}}$ if the statement $\tilde{\mathsf{pp}}$ are random generators of $\mathsf{QR}_{\tilde{N}}$. For zero-Knowledge, we mask the witnesses via noise flooding to improve readability. Instead, we could use rejection sampling and compile Σ_{ped} with Fiat-Shamir with aborts for better efficiency.

Lemma 9. *The Σ-protocol Σ_{ped} for relation $\mathsf{R}_{\Sigma_{\mathsf{ped}}}$ is correct, HVZK if the elements in $\tilde{\mathsf{pp}}$ generate the same subgroup, and 2-special sound for relation $\widetilde{\mathsf{R}}_{\Sigma_{\mathsf{ped}}}$.*

Proof. For correctness, observe that

$$\begin{aligned} f_m &= (\textstyle\sum_{i=1}^{\ell_m} \tau_{e_i} B^{i-1}) - \tau_{\overline{m}} \\ &= (\textstyle\sum_{i=1}^{\ell_m} (\gamma e_i + \mu_{e_i}) B^{i-1}) - (\gamma \overline{m} + \mu_{\overline{m}}) \\ &= \gamma \underbrace{\left((\textstyle\sum_{i=1}^{\ell_m} B^{i-1} e_i) - \overline{m}\right)}_{=0} + (\textstyle\sum_{i=1}^{\ell_m} B^{i-1} \mu_{e_i}) - \mu_{\overline{m}} = f_{m,0} \end{aligned}$$

Similarly, we have that $f_{r,0} = f_r$. Thus, the last check succeeds. The other checks pass due to linearity and since we mask all values (except s_i) over $\mathbb{Z}$. (Note that s_i is masked over $\mathbb{Z}_p$ but is used exclusively within equations over $\mathbb{Z}_p$ within $\overline{\mathbb{G}}_p$.)

Also, HZVK follows from the observations that (1) the first flow (except $\tilde{c}$) is determined by the verification equations, the challenge and the third flow, (2) the masks ensure that the third flow is distributed statistically close to uniform over $[0, CB \cdot 2^{\lambda}]^{\ell_m + \ell_r} \times \mathbb{Z}_p^{\ell_m + \ell_r} \times [0, C \cdot 2^{3\lambda}] \times [0, CS \cdot 2^{\lambda}] \times [0, C\tilde{N} \cdot 2^{2\lambda}] \times [0, \tilde{N} \cdot 2^{\lambda}]$ and (3) the $\tilde{c}$ commitment is statistically hiding (since all generators in $\mathbb{Z}_{\tilde{N}}^*$ generate the same subgroup).

For soundness, given two related transcripts tr, tr' with $\gamma \neq \gamma'$, set $e_i = \Delta e_i / \Delta\gamma, s_i = \Delta s_i / \Delta\gamma \bmod p, \overline{m} = \Delta\overline{m}/\Delta\gamma, r = \Delta r / \Delta r, \tilde{t}\Delta\tilde{t}/\Delta\gamma$ and $t_q = \Delta t_q / \Delta\gamma$. Note that due to the check in line 22, we either find a relaxed DLOG relation or we have that $e_i, \overline{m}, r, \tilde{t}, t_q \in \mathbb{Z}$. From line 23, we obtain two openings for Ω_q. Thus, we know that $f_m = f'_m$ and $f_r = f'_r$, else we find a non-trivial DLOG relation in $\tilde{\mathsf{pp}}$ as in [32], Section 5.1. By definition, we have

$$\begin{aligned} & f_m = (\textstyle\sum_{i=1}^{\ell_m} B^{i-1} \tau_{e_i}) - \tau_{\overline{m}} = (\textstyle\sum_{i=1}^{\ell_m} B^{i-1} \tau_{e_i}{}') - \tau_{\overline{m}}{}' = f'_m \\ \implies & (\textstyle\sum_{i=1}^{\ell_m} B^{i-1} e_i) - \overline{m} = 0 \\ \implies & \textstyle\sum_{i=1}^{\ell_m} B^{i-1} e_i = \overline{m}. \end{aligned}$$

Similarly, we obtain $r = \sum_{i=1}^{\ell_r} e_{\ell_m + i} B^{i-1}$. The other equalities follow as usual.

[19] We commit to e_i via an MPed commitment over $\tilde{N}$ to ensure that the statements hold over the integers. Thus, we need that the public parameters $\tilde{\mathsf{pp}}$ is setup such that MPed is hiding. This is guaranteed by $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$. Further, if we know that $e_i \in [0, B-1]$ we can use smaller masks.

Step 2: the range proof. Next, let Π_{rp} be the NIZK with random oracle H_{rp} from [61] (cf. Section 6.2). Note that Π_{rp} is obtained by compiling Bulletproofs [21, 8] via Fiat-Shamir. The correctness and zero-knowledge relation is

$$\mathsf{R}_{\mathsf{rp}} = \{(x, w) : \ E_i = e_i\overline{G} + s_i\overline{H}, e_i \in [0, B-1] \text{ for } i \in [\ell_m + \ell_r]\},$$

with $x = (\overline{G}, \overline{H}, B, (E_i)_{i\in[\ell_m+\ell_r]})$ and $w = ((e_i, s_i)_{i\in[\ell_m+\ell_r]})$, where B is a power of two [20]. Note that $\mathsf{srs} = \bot$ and $\mathsf{urs}_{\mathsf{rp}} = ((\widehat{g}_i)_{i\in[\ell_{\mathsf{rp}}]}) \in \overline{\mathbb{G}}_p^{\ell_{\mathsf{rp}}}$ define the $\mathsf{crs} = \mathsf{urs}_{\mathsf{rp}}$ of Π_{rp}, where $\ell_{\mathsf{rp}} \in \mathbb{N}$ is chosen appropriately. The soundness relation is

$$\widetilde{\mathsf{R}}_{\mathsf{rp}} \coloneqq \{(x, w) : \ (x, w) \in \mathsf{R}_{\mathsf{rp}} \text{ or } ((\overline{G}, \overline{H}, \mathsf{urs}_{\mathsf{rp}}), w) \in \mathsf{R}_{\mathsf{dlog}}\},$$

where $\mathsf{R}_{\mathsf{dlog}} = \{((\overline{G}, \overline{H}, \mathsf{urs}_{\mathsf{rp}}), w)\}$ denotes the relation that contains all non-trivial DLOG relations w for $(\overline{G}, \overline{H}, \mathsf{urs}_{\mathsf{rp}})$ (see [8] for more details). Note that for uniform statement, it is hard to find a witness for $\mathsf{R}_{\mathsf{dlog}}$ under the DLOG assumption.

Lemma 10 ([61], Theorem 17). *The* NIZK Π_{rp} *for relation* R_{rp} *is correct, zero-knowledge and adaptively knowledge sound for the relaxed relation* $\widetilde{\mathsf{R}}_{\mathsf{rp}} \supseteq \mathsf{R}_{\mathsf{rp}}$.

Step 3: the online-extractable NIZK. Finally, we combine Σ_{ped} and Π_{rp} to construct Π_{ped}. The construction is similar to the NIZK in Section 6.3 [61] except that our Σ-protocol has relaxed soundness guarantees and requires that MPed with public parameters $(\tilde{N}, \tilde{\mathsf{pp}})$ is setup in a hiding manner. (Note that we require a fresh modulus $\tilde{N}$ because we cannot reduce to assumption with respect to the existing N as it is part of the statement.) These parameters are part of the srs and are used to argue over the integers. As in Appendix C.2, we add a NIZK to prove that shows that $\tilde{\mathsf{pp}}$ is setup in a hiding manner and set

$$\begin{aligned}\mathcal{SRS} = \big\{&(\tilde{N}, \tilde{\mathsf{pp}}, \pi_{\mathsf{gen}}) \mid \tilde{N} \in \mathbb{N}, \tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{\ell_m+\ell_r}) \in (\mathbb{Z}_N^*)^{1+\ell_m+\ell_r},\\ &\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1, x_{\mathsf{gen}} = (\tilde{N}, \ell_m + \ell_r, \tilde{h}, (\tilde{g}_1, \cdots, \tilde{g}_{\ell_m+\ell_r}))\big\}.\end{aligned}$$

This NIZK ensures subversion zero-knowledge. We denote by $\mathsf{H}_{\mathsf{ped}}$ the random oracle of Π_{ped} and by $\mathcal{URS} = \overline{\mathbb{G}}_p^{\ell_{\mathsf{rp}}}$ the space for the urs of Π_{ped}. Below, we have $\mathsf{urs} \in \mathcal{URS}$ and $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ for some $\mathsf{srs} \in \mathcal{SRS}$. Let H_β be a random oracle mapping into $[0, C]$. The random oracle of Π_{ped} is $\mathsf{H}_{\mathsf{ped}} = (\mathsf{H}_{\mathsf{rp}}, \mathsf{H}_\beta)$. Let $x = (B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $w = (\overline{m}, r, (s_i)_{i\in[\ell_m+\ell_r]})$. The scheme is given below

- $\Pi_{\mathsf{ped}}.\mathsf{GenCRS}(1^\lambda)$: Samples $\mathsf{pp}_{\mathsf{MPed}} = (\tilde{N}, \tilde{\mathsf{pp}}) \leftarrow \mathsf{MPed.Setup}(1^\lambda)$ with $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{\ell_m+\ell_r})$. Then, sets $\pi_{\mathsf{gen}} \leftarrow \Pi_{\mathsf{gen}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{gen}}}(w_{\mathsf{gen}}, x_{\mathsf{gen}})$ for x_{gen} as above and appropriate w_{gen}. Outputs the structured reference string $\mathsf{srs} = (\mathsf{pp}_{\mathsf{MPed}}, \pi_{\mathsf{gen}})$.
- $\Pi_{\mathsf{ped}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}, x, w)$: Decomposes $\overline{m}$ and r into $(m_i)_i$ and $(r_i)_i$, and set $(e_i)_i = (m_1, \cdots, m_{\ell_m}, r_1, \cdots, r_{\ell_r})$ and computes

  $$\pi_0 \leftarrow \Pi_{\mathsf{rp}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{rp}}}(\mathsf{crs}, x_0, w_0),$$

  for $x_0 = (\overline{G}, \overline{H}, B, (E_i)_{i\in[\ell_m+\ell_r]})$ and $w_0 = ((e_i, s_i)_{i\in[\ell_m+\ell_r]})$,

  $$\begin{aligned}&(\Omega_\Sigma, \mathsf{st}) \leftarrow \Sigma_{\mathsf{ped}}.\mathsf{Init}(x_1, w_1),\\ &\gamma_\Sigma \leftarrow \mathsf{H}_\beta(x_1, \Omega_\Sigma),\\ &\tau_\Sigma \leftarrow \Sigma_{\mathsf{ped}}.\mathsf{Resp}(x_1, \mathsf{st}, \gamma_\Sigma),\\ &\pi_1 \leftarrow (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma),\end{aligned}$$

  for statement $x_1 = (\tilde{N}, \tilde{\mathsf{pp}}, B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and witness $w_1 = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$. Outputs $\pi = (\pi_0, \pi_1)$.

[20] We moved the generators $\overline{G}$ and $\overline{H}$ to the statement from the uniform reference string $\mathsf{urs}_{\mathsf{rp}}$. This is a purely notational change and we adapted the soundness relation below accordingly.

– $\Pi_{\mathsf{ped}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}, x, \pi)$: On input crs, x, and $\pi = (\pi_0, \pi_1)$, checks

$$\Pi_{\mathsf{rp}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{rp}}}(\mathsf{crs}, x_0, \pi_0) = 1,$$
$$\mathsf{H}_\beta(x_0, \Omega_\Sigma) = \gamma_\Sigma,$$
$$\Sigma_{\mathsf{ped}}.\mathsf{Verify}(x_1, \Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma) = 1,$$

where $\pi_1 = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$ and x_0, x_1 are defined as above, and outputs 1 iff all checks succeed.

We show that the scheme is sufficient to instantiate our framework $\mathsf{BS}_{\mathsf{fis}}$ in Section 5 (*i.e.*, the NIZK is correct, subversion zero-knowledge, and partially online-extractable). Correctness and subversion zero-knowledge are straightforward. For partial online-extraction, recall that statement x and witness w of relation $\mathsf{R}_{\mathsf{ped}}$ are split into $x_0 = (\overline{G}, \overline{H}), w_0 = (s_1, \ldots, s_{\ell_m+\ell_r})$ and $x_1 = (B, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r}), w_1 = (\overline{m}, r)$. For the sake of simplicity, we sketch how the extractor proceeds to extract from a single proof. Let $\mathcal{A}$ be an adversary (*i.e.*, prover) for online-extraction. Since the tuple $x_0 = (\overline{G}, \overline{H})$ is drawn at random from $X_0 = \overline{\mathbb{G}}_p^2$, the extractor samples $\overline{G} \leftarrow \overline{\mathbb{G}}_p$ and $\mathsf{td} \leftarrow \mathbb{Z}_p$ at random, then sets $\overline{H} = \mathsf{td} \cdot \overline{G}$. Then, it outputs $x_0 = (\overline{G}, \overline{H})$ and $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ to $\mathcal{A}$, where $\mathsf{srs} \leftarrow \Pi_{\mathsf{ped}}.\mathsf{GenCRS}(1^\lambda)$ and $\mathsf{urs} \leftarrow \mathcal{URS}$. After obtaining (partial) statement x_1 and proof π from $\mathcal{A}$, the extractor decrypts the ElGamal commitments (E_i, S_i) via a brute-force computation of the discrete logarithm $e_i = \mathsf{DLOG}_{\overline{G}}(E_i')$ of $E_i' \leftarrow E_i - \mathsf{td} \cdot S_i$. If $e_i \notin [0, B-1]$, the extractor aborts. (Since $B = \mathsf{poly}(\lambda)$, the extractor remains efficient and the NIZKs guarantee that aborts happen with low probability.) Using e_i, the adversary recomputes $\overline{m}$ and r via B-ary decomposition and checks that $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$. Note that in that case, the existence of suitable ElGamal openings $w_0 = (s_i)_i$ is guaranteed. (These are the discrete logarithms $s_i = \mathsf{DLOG}_{\overline{G}}(S_i)$ of S_i.) A subtlety of the proof is that we need to extract from both proofs π_0 and π_1 of $\pi = (\pi_0, \pi_1)$ *simultaneously* in the case that extraction fails. Fortunately, this was shown to be possible in [61]. In both extractions succeed, we can reduce to either DLOG in $\overline{\mathbb{G}}_p$ or sRSA.

Theorem 10. *The* NIZK Π_{ped} *is correct, subversion zero-knowledge under the* DDH *assumption, and partially online-extractable under the* sRSA *assumption and the* DLOG *assumption in* $\overline{\mathbb{G}}_p$.

Proof. Correctness is straightforward. Subversion zero-knowledge is also immediate, as we can simulate π_0 using the zero-knowledge property of Π_{rp} and π_1 using standard Σ-protocol techniques. Note that since π_{gen} guarantees that $\tilde{c}$ is hiding, the simulation of π_1 succeeds [21]. The proof is almost identical to the proof of Theorem 19 [61], and we omit details. Online-extraction requires some care, but the proof is similar to the proof of Theorem 20, [61] (taking into account that we embed the trapdoor into the statement $x_0 = (\overline{G}, \overline{H})$ instead of the crs). The extractor Ext proceeds as follows.

– $\mathsf{Ext}(1^\lambda)$: Sets up $\overline{G} \leftarrow \overline{\mathbb{G}}_p$ and $\overline{H} \leftarrow \mathsf{td} \cdot \overline{G}$ for $\mathsf{td} \leftarrow \mathbb{Z}_p$ and outputs $x_0 = (\overline{G}, hp)$
– $\mathsf{Ext}(\mathsf{crs}, \mathsf{td}, x, \pi)$: Parses $x = (x_0, x_1)$ with $x_0 = (\overline{G}, \overline{H})$ and $x_1 = (B, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$. Note that $\overline{H} = \mathsf{td} \cdot \overline{G}$. Decrypts the ElGamal commitments (E_i, S_i) via a discrete logarithm $e_i = \mathsf{DLOG}_{\overline{G}}(E_i')$ computation of $E_i' \leftarrow E_i - \mathsf{td} \cdot S_i$ (but outputs $\perp$ and aborts if there is no such $e_i \in [0, B-1]$). Then, sets $\overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1}$ and $r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}$. Checks that $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$. If the check succeeds, outputs partial witness $w_1 = (\overline{m}, r)$, and $\perp$ otherwise.

Note that $\mathsf{Ext}(1^\lambda)$ outputs uniform $(\overline{G}, \overline{H})$ over $\overline{\mathbb{G}}_p^2 = X_0$ and that Ext runs in polynomial time (since the DLOG computation aborts in case e_i is not short, *i.e.*, of polynomial size). Also, if all check succeed, then the output of Ext is sufficient, *i.e.*, there is a $w_0 = (s_1, \ldots, s_{\ell_m+\ell_r})$ such that $((w_0, w_1), x) \in \mathsf{R}_{\mathsf{ped}}$ due to the following facts.

– We have that $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$ due to the last check.
– We have that $E_i = e_i \overline{G} + s_i \overline{H}$ and $S_i = s_i \overline{G}, e_i \in [0, B-1]$, where $s_i = \mathsf{DLOG}_{\overline{G}}(E_i')$ for some $s_i \in \mathbb{Z}_p$. This holds by construction since for $s_i = \mathsf{DLOG}_{\overline{G}}(S_i)$, we have that $E_i = E_i' + \mathsf{td} \cdot S_i = e_i \overline{G} + s_i \overline{H}$.

[21] In more detail, we need that $x_1 \in \mathscr{L}_{\mathsf{R}_{\Sigma_{\mathsf{ped}}}}$ for the simulation but $\tilde{N}$ and $\tilde{\mathsf{pp}}$ might be setup in a non-hiding manner. Soundness of π_{gen} guarantees that no such malicious setup is possible.

– We have that $\overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}$ by construction.

Now let $\mathcal{A}$ be an adversary that on input (crs, x_0) outputs Q_S pairs $(x_{1,i}, \pi_i)_{i\in[Q_S]}$ that verify (*i.e.*, we have that $\Pi_{\mathsf{ped}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}, (x_0, x_{i,1}), \pi_i) = 1$) with probability at least $\mu(\lambda)$. Here, $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs}_{\mathsf{rp}})$ is setup via $\mathsf{srs} \leftarrow \Pi_{\mathsf{ped}}.\mathsf{Setup}(1^\lambda)$ and $\mathsf{urs}_{\mathsf{rp}} \leftarrow \overline{\mathbb{G}}_p^{\ell_{\mathsf{rp}}}$. Denote with Fail_i the event that the proof $(x_{1,i}, \pi_i)$ verifies but extraction fails for $i \in [Q_S]$. It remains to show $\Pr[\mathsf{Fail}_i] = \mathrm{negl}(\lambda)$. Then, we can conclude that $\Pr[\exists i : \mathsf{Fail}_i] = \mathrm{negl}(\lambda)$ via a union bound.

Assume that Fail_i occurs. Parse $x_{1,i} = (N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $\pi_i = \pi_{i,0}, \pi_{i,1}$ with $\pi_{i,0} = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$. Set $x = (x_0, x_{1,i})$, $x_{\mathsf{rp}} = (\overline{G}, \overline{H}, B, (E_i)_{i\in[\ell_m+\ell_r]})$ and $x_\Sigma = (\tilde{N}, \tilde{\mathsf{pp}}, B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$. Let us assume we have a witness w_{rp} such that $(x_{\mathsf{rp}}, w_{\mathsf{rp}}) \in \widetilde{\mathsf{R}}_{\mathsf{rp}} \supseteq \mathsf{R}_{\mathsf{rp}}$ and two related transcripts tr, tr' of Σ_{ped} for the statement x_Σ. (We refer to Theorem 20 [61] for a detailed extraction strategy that runs in polynomial time and succeeds with probability close to $\Pr[\mathsf{Fail}_i]$). In that case, we have $((\overline{G}, \overline{H}, \mathsf{urs}_{\mathsf{rp}}), w_{\mathsf{rp}}) \in \mathsf{R}_{\mathsf{dlog}}$ with at most negligible probability under the DLOG assumption. Thus, $(x_{\mathsf{rp}}, w_{\mathsf{rp}}) \in \mathsf{R}_{\mathsf{rp}}$ which means that $w_{\mathsf{rp}} = (e_i', s_i')_{i\in[\ell_m+\ell_r]}$ and

$$E_i = e_i'\overline{G} + s_i' G' \text{ and } e_i' \in [0, B-1]. \tag{11}$$

Then, we invoke 2-special soundness of Σ_{ped} on tr, tr' and obtain a witness w_Σ with $(x_\Sigma, w_\Sigma) \in \mathsf{R}_{\Sigma_{\mathsf{ped}}}$. We have that $(x_\Sigma, w_\Sigma) \in \mathsf{R}$ or $(\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$ or $(\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{\mathsf{dlog}}$. Under sRSA, we have that $(\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{C,\tilde{\ell}}(\vec{g})$ or $(\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{\mathsf{dlog}}$ with at most negligible probability. Thus, $(x_\Sigma, w_\Sigma) \in \mathsf{R}$. Parse $w_\Sigma = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$. By definition, it holds that

$$\begin{aligned} c &\equiv h_2^{\overline{m}} \cdot g^{re} \bmod N \\ E_i &= e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G} \\ \overline{m} &= \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1} \end{aligned} \tag{12}$$

Notably, we have $e_i = e_i'$ under DLOG as otherwise, we can compute a non-trivial DLOG relation between $\overline{H}$ and $\overline{G}$. But then, Equations (11) and (12) imply that extraction succeeds and thus, $\Pr[\mathsf{Fail}_i] = \mathrm{negl}(\lambda)$.

Optimizations. We apply standard Σ-protocol optimizations for Σ_{ped}. That is, we omit the first flow of Σ_{ped} (*i.e.*, the values $\Omega_{\tilde{c}}, \Omega_q, (\Omega_{E_i})_i, (\Omega_{S_i})_i, \Omega_c$ except $\tilde{c}$) from the proof π_1. The verification equations are then verified within the hash function H_β.

Efficiency. We set $B = 2^{64}$. Then, the DLOG computation in extraction runs in time $\mathcal{O}(2^{32})$. Further, we use standard RSA moduli and groups for $\lambda = 128$ bit security, *i.e.*, N and $\tilde{N}$ of size 3072 bit and group $\overline{\mathbb{G}}_p$ with order 256 bit. With these parameters, we have $\ell_m + \ell_r = 54$ and an integer commitment C_{Z} is of size 3.46 KB. The online-extractable NIZK is of size 5.62 KB and the range proof Π_{rp} is of size 1088 Byte. In total, the proof size of Π_{ped} is 6.7 KB.

E.3 Instantiation of Π_{fis}

Recall that Π_{fis} allows to prove knowledge of a valid $\mathsf{S}_{\mathsf{fis}}$ signature (e, a, y), where (e, a) are fixed in a $\mathsf{C}_{\mathsf{RInt}}$ commitment c_I. Note that we instantiate $\mathsf{C}_{\mathsf{RInt}}$ with $\mathsf{C}_{\mathsf{RInt}}^{\vec{B},T}$, where $\vec{B} = (2^{3\lambda}, 2^{3\lambda})$ and $T = 2^{\lambda+1}L$ for $L \in \mathbb{N}$. As discussed in Section 5.1, we set $\overline{E} = 2^{5\lambda}$. The public parameters are $\mathsf{pp}_I = (G_1, G_2, H) \in \mathbb{G}$ for some group $\mathbb{G}$ of order p such that $2^{3\lambda}T < \frac{p-1}{2}$. We can rewrite the relation $\mathsf{R}_{\mathsf{fis}}$ as follows

$$\begin{aligned} \mathsf{R}_{\mathsf{fis}} \coloneqq \big\{(x, w) \mid y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N, e \equiv 1 \bmod 2, y \in \langle h_1 \rangle, \\ (e - \overline{E}, a) \in [0, \vec{B}], \vec{C} = (e - \overline{E}, a)H + r\vec{G}, F = rH\big\}, \end{aligned}$$

for $x = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, \vec{C}, F), w = (e, a, y, r)$. The soundness relation can be written as $\widetilde{\mathsf{R}}_{\mathsf{fis}}$

$$\begin{aligned} \mathsf{R}_{\mathsf{fis}} \coloneqq \big\{(x, w) \mid y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N, e \equiv 1 \bmod 2, \\ (e - \overline{E}, a) \in [\vec{B}T, \vec{B}T], \vec{C} = (e - \overline{E}, a)H + r\vec{G}, F = rH\big\}, \end{aligned}$$

We construct a Σ-protocol Σ_{fis} for the relation $\mathsf{R}_{\mathsf{fis}}$, and then compile it via Fiat-Shamir. Note that the relation $\mathsf{R}_{\mathsf{fis}}$ contains the equation

$$y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N.$$

Since both y and e are part of the witness, this equation is non-trivial to prove, especially since y is an element of a group $\langle h_1 \rangle$ that might be setup maliciously. We solve this by committing to y in an additional $\mathsf{C}_{\mathsf{Grp}}$ commitment c_N (cf. Section 4.3). Note that $\mathsf{C}_{\mathsf{Grp}}$ is an ElGamal commitment of the form $c_N = y \cdot h_1^s$, where the randomness s is fixed via an Π_{int} commitment. Then, we show the equivalent equation

$$c_N^e \cdot h_1^{-\omega} \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \text{ with } \omega = e \cdot s$$

instead. For this equation, we can resort to techniques for quadratic equations over $\mathbb{Z}$. In order to use the same group $\mathbb{G}$ for the relaxed integer commitment of $\mathsf{C}_{\mathsf{Grp}}$, we split the randomness s of $\mathsf{C}_{\mathsf{Grp}}$ into ℓ_s values $s_i \in [0, 2^{3\lambda} - 1]$. In more detail, we set $\vec{B}' = (2^{3\lambda}, \cdots, 2^{3\lambda}) \in \mathbb{N}^{\ell_s}$, where $\ell_s = \lceil \frac{\log(N \cdot 2^\lambda)}{3\lambda} \rceil$ and use second relaxed integer commitment $\mathsf{C}'_{\mathsf{RInt}} = \mathsf{C}_{\mathsf{RInt}}^{\vec{B}',T}$ with public parameters $\mathsf{pp}'_I = (H', G'_1, \cdots, G'_{\ell_s})$ to commit to $\vec{s}$.

Step 1: the Σ-protocol. Set $C = 2^\lambda - 1$. The Σ-protocol Σ_{fis} is given in Figure 5. Instead of using witness e, we use $\overline{e} = e - \overline{E}$ and adapt the relations accordingly. As $\overline{e}$ is shorter, this reduces the proof size. We open $\mathsf{C}_{\mathsf{RInt}}$ commitments in zero-knowledge as in Appendix C.2 and $\mathsf{C}_{\mathsf{Grp}}$ commitments as in Appendix C.2. To show that $e \equiv 1 \bmod 2$, we show that we can write $\overline{e} = 2e' + 1$ over $\mathbb{Z}$. (Note that since $\overline{E}$ is even, this is sufficient.) For this and the equality $\omega = \overline{e} \cdot s$, we leverage the properties of MPed commitments in a similar manner as Σ_{ped} (cf. Appendix E.2). That is, we construct a polynomial $f_e = f_{e,2}\gamma^2 + f_{e,1}\gamma + f_{e,0}$ and $f_\omega = f_{\omega,2}\gamma^2 + f_{\omega,1}\gamma + f_{\omega,0}$, where the challenge γ is interpreted as variable, such that $f_{\omega,2} = f_{e,2} = 0$ implies that $\omega = \overline{e} \cdot s$ and $\overline{e} = 2e' + 1$. As before, we show that $f_{\omega,2} = f_{e,2} = 0$ via the linearity of MPed [22]. For this (and the commitment openings), we use a separate modulus $\tilde{N}$ and parameters $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{\tilde{\ell}}) \in \mathbb{N}^{\tilde{\ell}}$ with $\tilde{\ell} = 4 + \ell_s$ for MPed. To keep the security proof modular, we define the relation

$$\begin{aligned}\mathsf{R} \coloneqq \big\{(x,w) \mid c_N^{\overline{e}+\overline{E}} \cdot h_1^{-\omega} \cdot h_1^{-s\cdot\overline{E}} \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N, \overline{E} = 2\overline{E}' + 1, \omega = e \cdot s,\\ \vec{C} = (e - \overline{E}, a)H + r\vec{G}, F = rH, \vec{C}' = \vec{s}H' + r'\vec{G}', F' = r'H,\\ s = \textstyle\sum_{i \in [\ell_s]} s_i, c_N \equiv y \cdot h_1^s \bmod N\big\}\end{aligned}$$

with $x_\Sigma = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}'_I, c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C}', F')$ and $w_\Sigma = (e', e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$. Note that $c_N^{\overline{e}+\overline{E}} \cdot h_1^{-\omega} \cdot h_1^{-s\cdot\overline{E}} \equiv c_N^e \cdot h_1^{-s\cdot e} \bmod N$. Then, we set

$$\mathsf{R}_\Sigma = \big\{(x,w) \mid (x,w) \in \mathsf{R}, y \in \langle h_1 \rangle, (e,a) \in [0, \vec{B}], (s_1, \ldots, s_{\ell_s}) \in [0, \vec{B}'], \langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle\big\}$$

for correctness and HVZK of Σ_{fis} and

$$\begin{aligned}\widetilde{\mathsf{R}}_\Sigma = \big\{(x,w) \mid (x,w) \in \mathsf{R}, (e,a) \in [-\vec{B}T, \vec{B}T], (s_1, \ldots, s_{\ell_s}) \in [-\vec{B}'T, \vec{B}'T],\\ (\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}}) \text{ or } (\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{\mathsf{dlog}}\big\},\end{aligned}$$

for special soundness, where $\mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$ is defined in Definition 3 and $\mathsf{R}_{\mathsf{dlog}}$ denotes the relation that contains all non-trivial DLOG relations in $\tilde{\mathsf{pp}}$ (cf. Appendix E.2). Note that under sRSA (with respect to the setup of $\tilde{N}$ and $\tilde{\mathsf{pp}}$), it is simple to obtain a witness for $\mathsf{R}_{\mathsf{fis}}$ given a witness for $\widetilde{\mathsf{R}}_\Sigma$ for appropriate statements.

Lemma 11. *The Σ-protocol Σ_{fis} for relation R_Σ is correct with abort probability $1 - (1 - \frac{1}{L})^{\ell_s+3}$, HVZK and 3-special sound for $\widetilde{\mathsf{R}}_\Sigma$.*

Proof. We only give a brief sketch as this follows as similarly to security of previous Σ-protocols. Correctness follows from a straightforward but tedious computation. The abort probability can be computed as in Theorem 9.

[22] Since the polynomial is of degree 2, we require 3 transcripts for extraction instead of 2 transcripts.

For HVZK, observe that masking ensures that the third flow is distributed statistically close to uniform. Also, $\tilde{c}$ and $\tilde{c}_q$ are close to uniform over $Z^*_{\tilde{N}}$ due to the guarantee that $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$. The remaining values $\vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}$ and Ω_q are determined by the verification equations, the challenge γ and the third flow.

For 3-special soundness, it follows as in Theorem 9 and Lemma 9 that either we obtain a witness for the relaxed DLOG relation or all extracted values are integers (except r and r'). As in Theorem 9, the range checks guarantee relaxed range membership for $\vec{s}, e', \overline{e}$ and a, and lines 22 and 23 allow us to obtain openings for both $\mathsf{C}_{\mathsf{RInt}}$ commitments. Given 3 related transcripts, we obtain that $\omega = \overline{e} \cdot (\sum_i s_i)$ and $\overline{e} = 2e' + 1$ due to the check in line 26. This follows similarly to the equations over $\mathbb{Z}$ in Lemma 9, else we find a a witness for the hard DLOG relation. (Note that we need 3 related transcripts because the degree of f is 2 instead of 1 but the required adaptions to obtain the result are straightforward). Finally, the check in line 21 ensures that $c_N^e \cdot h_1^{-(\omega + s\overline{E})} \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N$. Note that there is no need to prove that $c_N \equiv y \cdot h_1^s \bmod N$ explicitly since this equation and the pair (c_N, s) already determine $y \equiv c_N \cdot h_1^{-s} \bmod N$, where $s = \sum_{i \in [\ell_s]} s_i$.

Step 2: the NIZK. For the final NIZK, we compile Π_{fis} into an NIZK via Fiat-Shamir with abort. Again, we require public parameters for MPed in the srs, where $\mathcal{SRS}$ is defined as in Appendix E.2, *i.e.*,

$$\mathcal{SRS} = \big\{ (\tilde{N}, \tilde{\mathsf{pp}}, \pi_{\mathsf{gen}}) \mid \tilde{N} \in \mathbb{N}, \tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{3+\ell_s}) \in (\mathbb{Z}^*_N)^{3+\ell_s}, \\ \Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1, x_{\mathsf{gen}} = (\tilde{N}, 3 + \ell_s, \tilde{h}, (\tilde{g}_1, \cdots, \tilde{g}_{\ell_m + \ell_r})) \big\}.$$

Let $\mathcal{URS} = \mathbb{G}^{1+\ell_s}$. Note that $\mathsf{urs} = \mathsf{pp}'_I$ contains public parameters for another $\mathsf{C}_{\mathsf{RInt}}$ commitment. We denote by $\mathsf{H}_{\mathsf{fis}}$ the random oracle of Π_{fis}. Below, we have $\mathsf{urs} \in \mathcal{URS}$ and $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ for some $\mathsf{srs} \in \mathcal{SRS}$. Alos, let $x = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, \vec{C}, F)$ and $w = (e, a, y, r)$. The scheme is given below.

- $\Pi_{\mathsf{fis}}.\mathsf{GenCRS}(1^\lambda)$: Samples $\mathsf{pp}_{\mathsf{MPed}} = (\tilde{N}, \tilde{\mathsf{pp}}) \leftarrow \mathsf{MPed.Setup}(1^\lambda)$ with $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{3+\ell_s})$. Then, sets $\pi_{\mathsf{gen}} \leftarrow \Pi_{\mathsf{gen}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{gen}}}(w_{\mathsf{gen}}, x_{\mathsf{gen}})$ for x_{gen} as above and appropriate w_{gen}. Outputs the structured reference string $\mathsf{srs} = (\mathsf{pp}_{\mathsf{MPed}}, \pi_{\mathsf{gen}})$.
- $\Pi_{\mathsf{fis}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}, x, w)$: Samples $s \leftarrow [N \cdot 2^\lambda]$, splits s into $(s_i)_i \in [0, 2^{3\lambda}]^{\ell_s}$ via $2^{3\lambda}$-ary decomposition and computes $c_N = y \cdot h_1^s$ and $\vec{C}' = \vec{s}H' + r'\vec{G}'$. Then, compiles Σ_{fis} into a proof π via

$$(\Omega_\Sigma, \mathsf{st}) \leftarrow \Sigma_{\mathsf{fis}}.\mathsf{Init}(x_\Sigma, w_\Sigma), \\ \gamma_\Sigma \leftarrow \mathsf{H}_\beta(x_\Sigma, \Omega_\Sigma), \\ \tau_\Sigma \leftarrow \Sigma_{\mathsf{ped}}.\mathsf{Resp}(x_\Sigma, \mathsf{st}, \gamma_\Sigma), \\ \pi_\Sigma \leftarrow (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma),$$

  for statement $x_\Sigma = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}'_I, c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C}', F')$ and witness $w_\Sigma = (e', e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$. Outputs $\pi = (\pi_\Sigma, c_N, \vec{C}', F')$.
- $\Pi_{\mathsf{fis}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}, x, \pi)$: On input crs, x, and $\pi = (\pi_\Sigma, c_N, \vec{C}', F')$, checks

$$\mathsf{H}_{\mathsf{fis}}(x_\Sigma, \Omega_\Sigma) = \gamma_\Sigma, \\ \Sigma_{\mathsf{fis}}.\mathsf{Verify}(x_\Sigma, \Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma) = 1,$$

  where $\pi_\Sigma = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$ and x_Σ is defined as above, and outputs 1 iff all checks succeed.

Theorem 11. *The* NIZK Π_{fis} *is correct, subversion zero-knowledge under the* DDH *assumption, and partially online-extractable under the* sRSA *assumption.*

Proof. Correctness is clear. Subversion zero-knowledge follows in previous NIZKs. For soundness, observe that $(x_\Sigma, w_\Sigma) \in \widetilde{\mathsf{R}}_\Sigma$ allows to obtain a valid witness w for $\widetilde{\mathsf{R}}_{\mathsf{fis}}$ or obtain an sRSA solution. We can rewind a prover twice to obtain 3 related transcripts [23], then we can apply 3-special soundness of Σ_{fis} to obtain such a witness w_Σ.

[23] It is well-known that this is possible in polynomial time.

Optimizations. Again, we omit the first flow of Σ_{fis} (*i.e.*, the values $\vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}, \Omega_q$ except $\tilde{c}$ and $\tilde{c}_q$) from the proof π_Σ. The verification equations are verified within the hash function $\mathsf{H}_{\mathsf{fis}}$.

Efficiency. We use standard RSA moduli and groups for $\lambda = 128$ bit security, *i.e.*, N and $\tilde{N}$ of size 3072 bit. We set $L = 2^{10}$ and thus $T = 2^{\lambda+11}$. Further, we assume that $\mathbb{G}$ is of order $4\lambda + 12$ (which is required for $\mathsf{C}_{\mathsf{RInt}}$). With these parameters, we have $\ell_s = 9$. In total, a proof is of size 4.08 KB.

Prover$(x; w)$		**Verifier**(x)
1 :	$t_q, \tilde{t} \leftarrow [0, \tilde{N} \cdot 2^{\lambda}]$	
2 :	$\mu_{\tilde{t}} \leftarrow [0, C\tilde{N} \cdot 2^{2\lambda}], \mu_{\overline{m}} \leftarrow [0, C2^{3\lambda}], \mu_r \leftarrow [0, CS \cdot 2^{\lambda}]$	
3 :	**for** $i \in [\ell_m + \ell_r]$ **do**	
4 :	$\quad \mu_{e_i} \leftarrow [0, CB \cdot 2^{\lambda}], \mu_{s_i} \leftarrow \mathbb{Z}_p$	
5 :	$\quad \Omega_{E_i} \leftarrow \mu_{e_i}\overline{G} + \mu_{s_i}\overline{H}, \Omega_{S_i} \leftarrow \mu_{s_i}\overline{G}$	
6 :	$\Omega_c \leftarrow h_2^{\mu_{\overline{m}}} \cdot g^{\mu_r e} \bmod N$	
7 :	$f_{m,0} \leftarrow (\sum_{i \in [\ell_m]} B^{i-1}\mu_{e_i}) - \mu_{\overline{m}}$	
8 :	$f_{r,0} \leftarrow (\sum_{i \in [\ell_r]} B^{i-1}\mu_{e_{i+\ell_m}}) - \mu_r$	
9 :	$\tilde{c} \leftarrow \tilde{h}^{\tilde{t}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{e_i} \bmod \tilde{N}$	
10 :	$\Omega_{\tilde{c}} \leftarrow \tilde{h}^{\mu_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{\mu_{e_i}} \bmod \tilde{N}$	
11 :	$\Omega_q \leftarrow \tilde{h}^{t_q} \cdot \tilde{g}_1^{f_{m,0}} \cdot \tilde{g}_2^{f_{r,0}} \bmod \tilde{N}$	
	$\xrightarrow{\tilde{c}, \Omega_{\tilde{c}}, \Omega_q, (\Omega_{E_i})_i, (\Omega_{S_i})_i, \Omega_c}$	
		12 : $\gamma \leftarrow [0, C]$
	$\xleftarrow{\gamma}$	
13 :	$\tau_{\overline{m}} \leftarrow \gamma\overline{m} + \mu_{\overline{m}}, \tau_r \leftarrow \gamma r + \mu_r, \tau_{\tilde{t}} \leftarrow \gamma\tilde{t} + \mu_{\tilde{t}}$	
14 :	**for** $i \in [\ell_m + \ell_r]$ **do**	
15 :	$\quad \tau_{e_i} \leftarrow \gamma e_i + \mu_{e_i}, \tau_{s_i} \leftarrow \gamma s_i + \mu_{s_i}$	
	$\xrightarrow{(\tau_{e_i})_i, (\tau_{s_i})_i, \tau_{\overline{m}}, \tau_r, \tau_{\tilde{t}}, t_q}$	
		16 : **check** $c^{-\gamma} h_2^{\tau_{\overline{m}}} \cdot g^{\tau_r e} \equiv \Omega_c \bmod N$
		17 : **for** $i \in [\ell_m + \ell_r]$ **do**
		18 : $\quad$ **check** $-\gamma E_i + \tau_{e_i}\overline{G} + \tau_{s_i}\overline{H} = \Omega_{E_i}$
		19 : $\quad$ **check** $-\gamma S_i + \tau_{s_i}\overline{G} = \Omega_{S_i}$
		20 : $f_m \leftarrow (\sum_{i=1}^{\ell_m} B^{i-1}\tau_{e_i}) - \tau_{\overline{m}},$
		21 : $f_r \leftarrow (\sum_{i=1}^{\ell_r} B^{i-1}\tau_{e_{\ell_m+i}}) - \tau_r$
		22 : **check** $\tilde{c}^{-\gamma} \cdot \tilde{h}^{\tau_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{\tau_{e_i}} \equiv \Omega_{\tilde{c}} \bmod \tilde{N}$
		23 : **check** $\tilde{h}^{t_q} \cdot \tilde{g}_1^{f_m} \cdot \tilde{g}_2^{f_r} \equiv \Omega_q \bmod \tilde{N}$

Fig. 4. Description of Σ_{ped} for $x = (\tilde{N}, \tilde{\mathsf{pp}}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $w = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$.

Prover$(x;w)$		**Verifier**(x)
1 :	$\tilde{t}, t_q \leftarrow [0, N \cdot 2^{\lambda}], \mu_r, \mu_{r'} \leftarrow \mathbb{Z}_p, \mu_{\tilde{t}}, \mu_{t_q} \leftarrow [0, CN \cdot 2^{2\lambda}]$	
2 :	$(\mu_e, \mu_a) \leftarrow [0, (\vec{B}C+1)L], \mu_{e'} \leftarrow [0, (B_1C+1)L]$	
3 :	$\vec{\mu}_s \leftarrow [0, (\vec{B}'C+1)L]$	
4 :	$\omega \leftarrow e \cdot (\sum_{i\in[\ell_s]} s_i), \mu_\omega \leftarrow [CB_1N \cdot 2^{2\lambda}]$	
5 :	$\Omega_f \leftarrow c_N^{-\mu_e} \cdot h_1^{\mu_\omega} \cdot h_1^{\sum_{i\in[\ell_s]} \mu_{s_i} \overline{E}} \cdot h_1^{\mu_a} \cdot h_2^{\mu_a} \bmod N,$	
6 :	$\vec{\Omega}_C = (\mu_e, \mu_a)H + \mu_r \vec{G}, \Omega_F \leftarrow \mu_r H,$	
7 :	$\vec{\Omega}_{C'} = (\vec{\mu}_s)H + \mu_{r'} \vec{G}, \Omega_{F'} \leftarrow \mu_{r'} H,$	
8 :	$f_{\omega,0} \leftarrow -\sum_{i\in[\ell_s]} \mu_{s_i}\mu_e, f_{\omega,1} \leftarrow \mu_\omega - (\sum_{i\in[\ell_s]} \mu_{s_i} e + s_i \mu_e)$	
9 :	$f_{e,0} \leftarrow 0, f_{e,1} \leftarrow -2\mu_{e'}$	
10 :	$\tilde{c} \leftarrow h^{\tilde{t}} \cdot \tilde{g}_1^e \tilde{g}_2^a \tilde{g}_3^{e'} \prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{s_i}, \Omega_{\tilde{c}} \leftarrow h^{\mu_{\tilde{t}}} \cdot \tilde{g}_1^{\mu_e} \tilde{g}_2^{\mu_a} \tilde{g}_3^{\mu_{e'}} \prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\mu_{s_i}}$	
11 :	$\tilde{c}_q \leftarrow \tilde{h}^{t_q} \cdot \tilde{g}_1^{f_{\omega,1}} \cdot \tilde{g}_2^{f_{e,1}}, \Omega_q \leftarrow \tilde{h}^{\mu_{t_q}} \cdot \tilde{g}_1^{f_{\omega,0}} \cdot \tilde{g}_2^{f_{e,0}}$	
	$\xrightarrow{\tilde{c}, \tilde{c}_q, \vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}, \Omega_q}$	
		12 : $\gamma \leftarrow [0, C]$
	$\xleftarrow{\gamma}$	
13 :	$\tau_r \leftarrow \gamma r + \mu_r, \tau_{r'} \leftarrow \gamma r' + \mu_{r'}$	
14 :	$\tau_{\tilde{t}} \leftarrow \gamma \tilde{t} + \mu_{\tilde{t}}, \tau_{t_q} \leftarrow \gamma t_q + \mu_{t_q}$	
15 :	$\tau_e \leftarrow \gamma e + \mu_e, \tau_a \leftarrow \gamma a + \mu_a, \tau_{e'} \leftarrow \gamma e' + \mu_{e'}$	
16 :	$\vec{\tau}_s \leftarrow \gamma \vec{s} + \vec{\mu}_s, \tau_\omega \leftarrow \gamma\omega + \mu_\omega$	
17 :	**check** $(\tau_e, \tau_a) \in [\vec{B}C, (\vec{B}C+1)L]$	
18 :	**check** $\vec{\tau}_s \in [\vec{B}'C, (\vec{B}'C+1)L], \tau_{e'} \in [B_1C, (B_1C+1)L]$	
	$\xrightarrow{\tau_r, \tau_{r'}, \tau_{\tilde{t}}, \tau_{t_q}, \tau_e, \tau_a, \tau_{e'}, \vec{\tau}_s, \tau_\omega}$	
		19 : **check** $(\tau_e, \tau_a) \in [0, (\vec{B}C+1)L]$
		20 : **check** $\vec{\tau}_s \in [0, (\vec{B}'C+1)L], \tau_{e'} \in [0, (B_1C+1)L]$
		21 : **check** $c_N^{-\tau_e} \cdot h_1^{\tau_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]} \mu_{s_i}} \cdot h^{\gamma} \cdot h_1^{\tau_a} \cdot h_2^{\tau_a + \gamma\overline{m}} \equiv \Omega_f \bmod N$
		22 : **check** $(\tau_e, \tau_a)H + \tau_r \vec{G} - \gamma \vec{C} = \vec{\Omega}_C, \tau_r H - \gamma F = \Omega_F$
		23 : **check** $\vec{\tau}_s H' + \tau_{r'} \vec{G}' - \gamma \vec{C}' = \vec{\Omega}_{C'}, \tau_{r'} H' - \gamma F' = \Omega_{F'}$
		24 : **check** $\tilde{c}_q^{-\gamma} \cdot h^{\tau_{\tilde{t}}} \cdot \tilde{g}_1^{\tau_e} \tilde{g}_2^{\tau_a} \tilde{g}_3^{\tau_{e'}} \prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\tau_{s_i}} \equiv \Omega_{\tilde{c}} \bmod \tilde{N}$
		25 : $f_\omega = \gamma \cdot \tau_\omega - ((\sum_{i\in[\ell_s]} \tau_{s_i}) \cdot \tau_e), f_e = \gamma(\tau_e - (2 \cdot \tau_{e'} + \gamma))$
		26 : **check** $\tilde{c}_q^{\gamma} \cdot \Omega_q \equiv \tilde{h}^{\tau_{t_q}} \cdot \tilde{g}_1^{f_\omega} \cdot \tilde{g}_2^{f_e}$

Fig. 5. Description of Σ_{fis} with $x = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}'_I, c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C}', F')$ and $w = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$.