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 and Inria, 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 practical 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 a variant of Fischlin's generic framework (CRYPTO'06). Our Blind Signature scheme has signatures of size 4.28 KB and communication cost 10.98 KB. On the way, we develop techniques that might be of independent interest. In particular, we provide efficient *relaxed* range-proofs for large ranges 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 [28, 31, 74], e-voting [30, 49], or anonymous credentials [29, 21], and more. In the past few years blind signatures also play an important role in new applications such as blockchains [86, 23] or private access tokens [59, 53].

Initial constructions. Since their introduction by Chaum [28], many variants of blind signatures were proposed. The first proposed construction—blind RSA [28]—was proven secure under one-more RSA [13]. A similar construction secure under one-more CDH was proposed in [20]. 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 strong assumptions are only *somewhat* falsifiable [71] and are tailored to the schemes itself.

Protocols with three or more rounds. Historically, the main alternative to the aforementioned blind signatures are based on linear identification protocols, *e.g.*, blind Schnorr [78] or similar constructions [73, 6, 55, 64]. 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 [81, 16]. 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, 47, 63, 80, 36] 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 [25, 76, 67]. Generally, these Schnorr-style approaches require more than two moves, *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 [68, 56]. However, it is difficult to construct round-optimal blind signatures in the plain model under standard assumptions which is supported by several impossibility results [68, 43, 75]. Katsumata *et al.* [65] 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 circuits. More commonly, constructions circumvent such hurdles via a trusted setup [42, 5, 70, 17, 19, 79, 3], idealized models (*e.g.*, generic groups and/or the random oracle model [54, 4, 18, 37, 54, 66]), complexity leveraging [51, 50], or interactive assumptions [11, 69, 8, 46, 45, 52]. All such constructions require pairings or lattices [4] with the exception of blind RSA [28, 13, 69, 8].

But over large networks and complex web applications, existing implementations of pairings (*e.g.*, [83]) 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 [82]. 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 [84] or SSH [85]. The only efficient round-optimal blind signature in this setting is blind RSA [28, 13] and its variants [69, 8]. The latter are covered in an RTF draft [38] and blind RSA is still a recommended nowadays [27]. 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?* [5]

1.1 Our Contributions

In this paper we answer this question affirmatively. We construct a *round optimal* blind signature scheme with *competitive e ciency*, 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 variant of Fischlin's framework [42] proposed in [66]. Roughly, [66] shows how to construct a blind signature via a signature scheme with an *all-but-one* reduction [6]. We instantiate the framework with a variant of the signature proposed in [41] (hereafter denoted by $\mathsf{S}_{\mathsf{fis}}$), and obtain blind signatures with 10.98 KB communication and signatures of size 4.28 KB. We provide a comparison to prior works in Table 1. Notably, we provide the first round-optimal blind signature without pairings or lattices. On a practical level, our signature is 2 times smaller than the previously most efficient blind signature in this regime [25]—and adds the desirable features of stateless, session-independent efficiency and round optimality. In particular, the computation cost in [25] (measured in group operations) grows linearly with the number of signing sessions opened, whereas in our construction communication and computation is $O(\lambda)$ in

[4] The framework of Fischlin [42] yields round-optimal blind signatures with trusted setup generically, but efficient instantiations rely either on pairings [18, 4, 66] or lattices [37, 7].

[5] Note that due to impossibility results for round optimal blind signatures [68, 43, 75], the reliance on random oracles can likely not be removed *efficiently*.

[6] 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. We refer to [72] for more details.

the number of group elements. Compared to Blind RSA [28], our construction is only around 10 larger times *without* relying on a strong interactive assumption [7].

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 technical overview in Section 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 for large ranges, *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* cyclic 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 [41]. Knowledge of a signature on a given message m can be shown efficiently in zero-knowledge using our relaxed range proofs.

Table 1: Comparison to relevant state-of-the-art blind signatures.

Reference	Sig. size	Comm. size	Setting	Assumption
del Pino et al. [37]	100 KB	850 KB	Lattices	DSMR, MLWE, MSIS
Blazy et al. [18]	96 B	220 KB †	Pairings	SXDH, CDH
Abe et al. [4]	5.5 KB	1 KB	Pairings	SXDH
Hanzlik et al. [54]††	5 KB 9 KB	72 KB 36 KB	Pairings	CDH
Katsumata et al. [66]	447 B 96 B	303 B 2.2 KB	Pairings	SXDH DDH, CDH
This work	4.28 KB	10.98 KB	RSA, Groups	sRSA, DDH

We provide signature size, communication size, the algebraic setting, and the underlying assumptions for known *round-optimal* blind signatures in the ROM secure under *non-interactive* assumptions. We stress that our work relies on assumptions in prime-order groups *without* pairing.
(†): Communication of [18] scales linearly with the message size, and is given here for 256 bit messages. (††): [54] offers tradeoffs between signature and communication sizes.

Reference	Sig. size	Comm. size	#Rounds	Assumption
Blind RSA and variants [28, 69, 8]	768 B	384 B	2	One-more RSA
Chairattana-Apirom et al. [25]‡	8.66 KB	8.08 KB	5	RSA
This work	4.28 KB	10.98 KB	2	sRSA, DDH

We provide signature size, communication size, number of rounds and the underlying assumption of known blind signatures in the RSA setting.
(‡): [25] is not round-optimal and at most an a-priori fixed number of signatures can be issued, here 2^{30}. Also, the signer is required to keep a state and communication scales logarithmically in the number sessions in size but *linearly* in computation.

1.2 Concurrent Works

There is an independent and concurrent work that also constructs a pairing-free blind signature in the ROM from standard assumptions [26] but for a stricter notion of pairing-free. That is, their construction relies exclusively on a prime-order group without pairing, whereas our construction relies on a prime-order group and a hidden order group (namely QR_N). In addition to this difference, we give a brief comparison. [26] 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.

[7] In the pairing setting, the known tradeoffs are similar (cf. [54, 66]) compared to Blind BLS [20].

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 and communication is more than twice as large as ours. 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. Similarly, we believe it to be non-trivial to remove the reliance of hidden order groups in our round-optimal construction.

2 Technical Overview

We provide an overview of our construction. Since our blind signature builds on the framework proposed in [66], we give a brief recap.

The framework. The framework of [66] is based on a signature scheme S with a *compatible* additively homomorphic commitment scheme Com, *i.e.*, $\mathsf{Com}(m;r) + \mathsf{Com}(m';r') = \mathsf{Com}(m+m';r+r')$. Here, compatible means that there exists an algorithm $\widehat{\mathsf{Sig}}$ such that the signing algorithm $\mathsf{Sign}(\mathsf{sk}, m)$ can be rewritten as $\widehat{\mathsf{Sig}}(\mathsf{sk}, \mathsf{Com}(m;r)) - \mathsf{Com}(0;r)$. Namely, $\widehat{\mathsf{Sig}}$ computes an intermediate signature $\sigma_r = \widehat{\mathsf{Sig}}(\mathsf{sk}, \mathsf{Com}(m;r))$ given a commitment to the message m with randomness r. Then, a signature can be computed by removing $\mathsf{Com}(0;r)$ from σ_r homomorphically [8]. To turn this into a blind signature, a user can generate a commitment $c = \mathsf{Com}(m;r)$, send it to the signer, and the signer can simply return $\sigma_r \leftarrow \widehat{\mathsf{Sig}}(\mathsf{sk}, c)$. Finally, the user obtains a valid signature $\sigma \leftarrow \sigma_r - \mathsf{Com}(0;r)$. During the signing process, the user also sends a proof π along with c that proves knowledge of (m, r) such that $c = \mathsf{Com}(m;r)$ via an online-extractable NIZK [9]. This is required for proof of one-more unforgeability (OMUF).

The above 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 OMUF, their proof relies on the all-but-one reduction of their underlying signature scheme—this means the reduction can set up the verification key in an alternate way that allows to sign all but one message m^*. Also, this property naturally yields the desired algorithm $\widehat{\mathsf{Sig}}$. Since our OMUF proof differs, we refer to [66] for more details. We finally remark that Katsumata *et al.* [66] 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 fulfills these requirements (*i.e.*, a signature scheme with an all-but-one reduction that is rerandomizable and has a compatible commitment scheme). Unfortunately, to the best of our knowledge, all RSA-based signatures are not rerandomizable or have no all-but-one reduction. Instead, we choose a signature scheme that is almost *compatible* with Pedersen commitments over $\mathbb{Z}_N$, namely the signature scheme $\mathsf{S}_{\mathsf{fis}}$ [41] based on Cramer-Shoup signatures [35]. 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, the signer chooses a random prime e and a random integer a—both from specific intervals—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 Eq. (1), where a and e lie in the aforementioned intervals. While the scheme is not quite compatible with Pedersen commitments due to the XOR operation in the exponent, we observe that a functions as a mask of m within the security proof. If we replace XOR-based masking with noise flooding [10], then XOR is replaced with

[8] This view is a simplified variant of [66]'s framework for the sake of exposition. We assume implicitly that σ_r contains a component with the same range as $\mathsf{Com}(0;r)$.

[9] Online-extractability allows the reduction to obtain (m, r) in the proof of one-more unforgeability in an *on-the-fly* manner.

[10] With noise flooding, we refer to adding a mask a that is exponentially larger than m.

a simple addition. We adapt the scheme and modify Eq. (1) as follows:

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

where a is chosen from an exponentially larger interval compared to before. Since in Eq. (2), m is masked statistically by a, the security proof can be adapted in a straightforward manner.

Turning it into a blind signature. We observe that the above scheme is *almost* compatible with Pedersen commitments. Let g be another QR_N generator. We can sign a Pedersen 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

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

This corresponds to the algorithm $\widehat{\mathsf{Sig}}$ but unfortunately, the value $y' = y_r \cdot g^{-r}$ does *not* yield a valid $\mathsf{S}_{\mathsf{fis}}$ signature (e, a, y') since Eq. (2) is not satisfied [11]. It even seems that the user cannot derive a valid signature from y_r in another manner, 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, the computable value $y \equiv y_r \cdot g^{-r} \bmod N$ satisfies Eq. (2), where y_r is generated as in Eq. (3) as before. Then, as in [66], the user proves with a proof π_{ped} that she committed to message m 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 to send e. Note that the user cannot generate e itself, as the reduction for $\mathsf{S}_{\mathsf{fis}}$ needs to be able to choose the primes e used in signatures. Indeed, it is required in the security proof that a fresh prime e and mask a are picked for each fresh message m to be signed. A natural idea is to let the user generate it via a hash function $\mathsf{H}_{\mathbb{P}}$ mapping into primes (of desired range).

As the signer also needs to derive the same prime e (i.e. see the input of $\mathsf{H}_{\mathbb{P}}$), we need to make sure that the message m is hidden by the input to $\mathsf{H}_{\mathbb{P}}$ In particular, we cannot simply derive $e = \mathsf{H}_{\mathbb{P}}(m)$. Another idea would be to derive $e = \mathsf{H}_{\mathbb{P}}(c)$ as the commitment c is already hiding the message. However, the user needs to know e already to set up c, so there is a cyclic dependency.

Therefore, we require that the user commits to m as well as the randomness r for generating c using an integer commitment c_Z [12] which can then be hashed to derive e. 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 $a \leftarrow \mathsf{PRF}(c \| c_Z)$[13].

In summary, the user commits to (m, r) in c_Z, computes $e \leftarrow \mathsf{H}_{\mathsf{pp}}(c_Z)$ and sets $c \equiv h_2^m \cdot g^{r \cdot e} \bmod N$. 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 Eq. (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 m. The blind signature is π_{fis} generated via Π_{fis} as before.

Proving one-more unforgeability. While the unforgeability reduction of $\mathsf{S}_{\mathsf{fis}}$ has no all-but-one flavor, we can show one-more unforgeability for our blind signature with the above modifications if the NIZK Π_{fis} is adaptively knowledge sound [14]. We stress that we cannot reduce to unforgeability

[11] Recall that this property is required for the commitment to be compatible.

[12] An integer commitment allows to commit to (vectors) of integers, *i.e.*, has message space $\mathbb{Z}^n$, and knowledge of an opening can be proven in zero-knowledge. Here, that means that (m, r) is fixed over $\mathbb{Z}^2$. For our construction, we can slightly relax this requirement if we hash m first, but we omit details here.

[13] Later, we implement this PRF with a random oracle.

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

the $\mathsf{S}_{\mathsf{fis}}$ signature scheme directly, as the adversary obtains the prime e for the signature before the reduction knows the message m to be signed.

Instead, analogous to $\mathsf{S}_{\mathsf{fis}}$, the reduction for one-more unforgeability sets up the verification key vk in an alternative way so that it can sign without knowing the factorization of N. This involves guessing the "format" of the forgery generated by the adversary $\mathcal{A}$.

To sign a commitment c, the reduction extracts (m, r) on-the-fly from the proof π_{ped}, and uses the alternatively set up key to sign m. It then reapplies the user's blinding. When the adversary outputs its forgeries, the reduction identifies a signature π_{fis} on a message m that it never signed [15], and then extracts a valid signature (e, a, y) from π_{fis}.

Since the final extraction is not performed in a on-the-fly manner, there is a subtlety. For example, if the extractor rewinds the adversary to extract a proof, then the extracted witness might depend on the guess we made during the vk setup, rendering the forgery useless for our reduction. A similar issue was observed, *e.g.*, in [6, 62], since witness indistinguishability is not necessarily preserved when rewinding. To solve this issue, we let the user commit to the signature parts (e, a) in c_I with a perfectly binding integer commitment $\mathsf{C}_{\mathsf{RInt}}$ [16]. Then, the extracted values (e, a) are fixed during the initial run when our guess is still hidden. Even if our guess is revealed during extraction, the extractor still succeeds in finding a valid signature with fixed (e, a). Since our guess depends only on these values, we can conclude that we guess correctly with sufficient probability which allows to solve sRSA with this modification.

Making it maliciously blind. An observant reader might realize that our scheme is *not* blind yet. Concretely, there are two types of problems: (a) we need to embed the crs into the vk, i.e. the crs is chosen by the signer and (b) Pedersen commitments are not hiding over $\mathbb{Z}_N^*$. We show how we deal with both problems below.

(a) Subversion Zero-knowledge. Since the signer can choose vk maliciously, we need to ensure that zero-knowledge holds for arbitrary crs in vk. As in [48], we require that the NIZKs Π_{ped} and Π_{fis} are subversion zero-knowledge, *i.e.*, zero-knowledge holds even for a malicious setup [12]. Unfortunately, this notion is difficult to instantiate in our setting. To the best of our knowledge, all instantiations of subversion zero-knowledge NIZKs [12, 44] require strong knowledge assumptions (which we wish to avoid). Instead, we give a simplified definition which yields similar guarantees in the ROM. 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 $\mathsf{srs} \in \mathcal{SRS}$. Our notion can be instantiated under standard assumptions (*e.g.*, DDH in pairing-free groups) because in the security proof, we can embed trapdoors into the uniform part (which is output by a random oracle). More details on our instantiations are given below.

(b) Subgroup arguments over $\mathbb{Z}_N^*$. Pedersen commitments over $\mathbb{Z}_N^*$ are not hiding for malicious modulus N: if $\langle g \rangle$ is a proper subset of $\langle h_2 \rangle$, there is a concrete attack on blindness. Thus, we let the signer prove that $\langle g \rangle = \langle h_2 \rangle$ with a NIZK with subversion soundness [12] (*i.e.*, soundness holds even for malicious crs). We embed this proof into the verification key to avoid the trivial attack. We also need to ensure that $\mathsf{C}_{\mathsf{RInt}}$ and C_{Z} are hiding. For both, we simply ask that public parameters are uniform (and sample them via a random oracle). There remain two more subtle problems.

(b.1) Recall that the user sets $c \equiv h_2^m \cdot g^{r \cdot e} \mod N$, so even if $\langle h_2 \rangle = \langle g \rangle$, we might have $\langle g^e \rangle \subsetneq \langle h_2 \rangle$. Fortunately, we can show that if $\langle h_2 \rangle = \langle g \rangle$, then $\langle g^e \rangle = \langle h_2 \rangle$ with overwhelming probability over the choice of a random prime e for arbitrary modulus N.

(b.2) The signer sends y_r to the user which again, might not be in the same subgroup. If the signature π_{fis} reveals the subgroup of y_r, blindness is broken. But conditioned on Eq. (3), we can show that $y_r \in \langle g \rangle$ if $\{h, h_1, h_2\} \subseteq \langle g \rangle$ and $c \in \langle g \rangle$. We let the former be proven by the signer in vk. If we let the user check that Eq. (3) holds, then the latter can be shown to hold with overwhelming probability over the choice of e.

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

[16] Again, we can relax the commitment scheme, *i.e.*, we do not require that $\mathsf{C}_{\mathsf{RInt}}$ is a full-fledged integer commitment. We elaborate later.

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. We give a brief overview of the challenges and our solutions.

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

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

For online-extraction, we use the approach of [66] (cf. Section 6). Let $\mathbb{G}$ be a pairing-free group of prime order p with generators G, H. The values (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 ensures that $e_i \in [0, B-1]$ (*e.g.*, Bulletproofs [22]). We then interpret $(E_i)_i$ as a *bounded* integer commitment c_Z, *i.e.*, the committed values must lie in the limited range $[0, B-1]$. This notion suffices for our construction. While we follow the template of [66], our instantiation is considerably more complicated since we need to show a statement over two algebraic structures: prime order groups and $\mathbb{Z}_N^*$. For this, we employ a structured srs to argue over the integers with techniques from [32]. We refer to Appendix E for more details.

Proof for $\mathsf{S_{fis}}$ signatures. To derive a blind signature, we need a perfectly binding commitment $\mathsf{C_{RInt}}$ and a NIZK Π_{fis} for the relation in Eq. (2) and simultaneously:

$$c_I = \mathsf{C_{RInt}.Commit}(e, a, r_I) \ \wedge \ a \in [0, 2^{3\lambda} - 1] \ \wedge \ e \in [2^{3\lambda}, 2^{3\lambda+1}] \cap \mathbb{Z}_{\mathsf{odd}}.$$

Note that these are the specific ranges for $\mathsf{S_{fis}}$ verification. While it is fine to employ range proofs during the (one-time) signing interaction, it is undesirable to include a range proof for presenting the signature (as the verification overhead is noticeable for such large ranges).

Instead, we relax the range requirements in such a way that the unforgeability proof of $\mathsf{S_{fis}}$ 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 $\mathsf{C_{RInt}}$ 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 [32]). 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 bits. For comparison, a Bulletproof for the above ranges requires 932 Byte [22]. 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 $\mathsf{C_{RInt}}$ 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 $\mathsf{C_{RInt}}$ 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 10.98 KB communication of size 4.28 KB.

Alternative View. Instead of viewing our construction as an instantiation of the Fischlin-style framework of [66] without pairings, we can view it as a maliciously-secure and optimized instantiation the construction sketched in [24]. In particular, [24] presents a blinded interactive signing protocol for sRSA-based signatures. Our instantiation of the underlying signature is more efficient, and the

construction in [24, Figure 1] does not seem to yield maliciously-secure blind signatures [17]. We provide techniques to achieve OMUF and malicious blindness simultaneously and efficiently, while preserving round optimality.

3 Preliminaries

Notations. We denote the security parameter by λ. A polynomial time (PT) algorithm $\mathcal{A}$ runs in time polynomial in the (implicit) security parameter λ. We denote "probabilistic polynomial time" by PPT. 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 $f = \mathsf{negl}(\lambda)$ 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. For a group element g we write $\mathsf{ord}(g)$ to denote the order of the group element. We denote by $\mathsf{QR}_N = \{a \in \mathbb{Z}_N^* : \exists\, b \in \mathbb{Z}_N^*, b^2 \equiv a \mod N\}$ the quadratic residues $\mod N$. 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 3 in Appendix A.

Probability. Let $V, L \in \mathbb{N}$. We define *uniform rejection sampling* for the interval $[V, (V+1)L]$ with masking overhead L as in [32]. 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 $1/L$[18]. We use a version of the Forking Lemma from [2, Lemma 1] that fits our usage of it. The lemma was first introduced by Pointcheval and Stern [77] then generalized in [14, 2]. 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 $P', Q' > 2^{\lambda+1}$.

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

[17] For example, observe that C_x is not hiding if N and (a, b) are set up maliciously. In particular, there is an attack if a and b are chosen in different subgroups. Proving that C_x is hiding requires a computational assumption over QR_N, so it is limited (at most) to honest signer blindness.

[18] When $v = 0$, the number of "bad" $\mu \in [0, (V+1)L]$ causing abort is $\#\{[0, V-1]\} = V$. For a fixed $0 < v \in (0, V]$, the number of "bad" μ causing abort is $\#\{[0, V-1-v]\} + \#\{[(V+1)L - v + 1, (V+1)L]\} = V$. In both cases the abort probability over choices of μ is $V/((V+1)L + 1) \leq 1/L$.

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.*, [66, Appendix B] for more details.

3.1 Cryptographic Primitives

Commitment Scheme. A *commitment scheme* is a tuple of PPT algorithms $\mathsf{C} = (\mathsf{C.Setup}, \mathsf{C.Commit}, \mathsf{C.Commit})$ 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 are 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 binding properties. The formal definitions can be found in Appendix A.5.

(Bounded) Integer Commitments. We refer to a commitment scheme with message space $[A, B] \subseteq \mathbb{N}$ as a (bounded) integer commitment scheme. We often omit the term bounded if the message space is clear by context.

ElGamal commitments. We recall ElGamal (EG) over a group $\mathbb{G}$ of prime order p with message space $\mathbb{Z}_p$ [39]. We use additive notation for prime order groups.

- $\mathsf{EG.GenPP}(1^\lambda)$: set $(G, H) \leftarrow \mathbb{G}$ and output $\mathsf{pp} = (G, H)$.
- $\mathsf{EG.Commit}(\mathsf{pp}, m)$: sample $r \leftarrow \mathbb{Z}_p$ and set $c = (mG + rH, rG)$, and output (c, r).
- $\mathsf{EG.Verify}(\mathsf{pp}, c, m, r)$: check if $c = (mG + rH, rG)$.

Note that the public parameters are uniform and we can sample them via a random oracle to avoid trusted setup. EG commitments are correct, hiding under DDH and perfectly binding.

Remark 1. If in verification of EG, we check that $m \in [0, M]$ for $M < p$, then m is fixed over the integers and we can interpret the commitment as an integer commitment with message space $[0, M] \subseteq \mathbb{N}$.

Pedersen Commitments in $\mathbf{QR}_N$. We recall Pedersen multi-commitments (MPed) over QR_N with message space $\mathbb{Z}^\ell$ for some $\ell \in \mathbb{N}$ (cf. [34]).

- $\mathsf{MPed.GenPP}(1^\lambda)$: set $(N, P, Q) \leftarrow \mathsf{GenRSA}(1^\lambda)$ 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 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. 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 `euf-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 (two-move) blind signature scheme is a tuple of PPT algorithms $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ such that

Definition 1 (Blind Signature). *A (two-move)* 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*,*
- Sign *is split into the following algorithms:*
  - $\mathsf{User}(\mathsf{bvk}, m)$*: 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 [61]. Below, we define correctness, blindness under *malicious keys*, and one-more unforgeability of a blind signature scheme. Moreover, we will omit the state for better readability on occasion.

Definition 2 (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 3 (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 4 (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*

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

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 $\mathscr{L}_{\mathsf{R}}$ with challenge space $\mathcal{CH}$ 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) Ω and a state st, where we assume st includes x, w,
- $\mathsf{Chall}()$: samples a challenge $\gamma \leftarrow \mathcal{CH}$ (without taking any input),
- $\mathsf{Resp}(\mathsf{st},\gamma)$: given a state st and a challenge $\gamma \in \mathcal{CH}$, outputs a third flow message (*i.e.*, response) τ,
- $\mathsf{Verify}(x,\Omega,\gamma,\tau)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, a commitment Ω, a challenge $\gamma \in \mathcal{CH}$, and a response τ, outputs a bit $b \in \{0,1\}$.

We recall the standard notions of *correctness, high-min entropy, (non-abort) honest-verifier zero-knowledge*, and *k-special soundness*. A Σ-protocol is correct, if for all $(x,w) \in \mathsf{R}$, if for any honestly generated transcripts (Ω,γ,τ), the verifier accepts, *i.e.* $\mathsf{Verify}(x,\Omega,\gamma,\tau) = 1$. It has high min-entropy if for all $(x,w) \in \mathsf{R}$, it is *statistically* hard to predict a honestly generated first flow Ω. It is honest-verifier zero-knowledge (HVZK), if there exists a PPT zero-knowledge simulator Sim such that the distributions of $\mathsf{Sim}(x,\gamma)$ and a honestly generated *non-aborting* transcript with Init initialized with (x,w) are statistically indistinguishable for any $x \in \mathscr{L}_{\mathsf{R}}$, and $\gamma \in \mathcal{CH}$, where the honest execution is conditioned on γ being used as the challenge. Finally, it is k-special sound, if there exists a *deterministic* PT extractor Ext such that given k valid transcripts $\{(\Omega,\gamma_i,\tau_i)\}_{i\in[k]}$ for statement x with pairwise distinct challenges $(\gamma_i)_i$, 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.8. 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 [19]. A NIZK for a relation R with common reference string space $\mathcal{CRS} = \mathcal{SRS} \times \mathcal{URS}$ is a tuple of PPT algorithms $(\mathsf{GenSRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$, where the latter two are oracle-calling, such that:

- $\mathsf{GenSRS}(1^\lambda)$: 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 $\mathsf{srs} \leftarrow \mathsf{GenSRS}(1^\lambda)$ 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 $(x,w) \in \mathsf{R}$. Note that the sub-algorithms of Sim share state. 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 [12]. 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 $\mathsf{Sim}_{\mathsf{crs}}$

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

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 belongs to $\mathcal{SRS}$.

We define *adaptive knowledge soundness*. An NIZK is adaptively knowledge sound for relation[20] $\widetilde{\mathsf{R}}$ if there exist positive polynomials $p_{\mathsf{T}}, p_{\mathsf{P}}$, constant c, a PPT extractor Ext and a PPT simulator 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 $\mathsf{crs} := (\mathsf{srs} \leftarrow \mathsf{GenSRS}(1^{\lambda}), \mathsf{urs} \leftarrow \mathcal{URS})$, given $(x, \pi) \leftarrow \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}}; \rho)$, with probability at least $\frac{\mu(\lambda)^c - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)}$ 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 exists 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 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 $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs} := (\mathsf{srs}, \mathsf{urs}), x, \pi) = 1$ but $x \notin \mathscr{L}_{\widetilde{\mathsf{R}}}$.

Fiat-Shamir transformation. We recall the Fiat-Shamir transformation [40, 10] 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 k-*special soundness*, into a NIZK $FS[\Sigma] = (\mathsf{GenSRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$ using a random oracle H that maps to the challenge space $\mathcal{CH}$ of Σ:

- $\mathsf{GenSRS}(1^{\lambda})$: 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 $(\Omega, \mathsf{st}) \leftarrow \mathsf{Init}(x, w)$, computes the challenge $\gamma \leftarrow \mathsf{H}(x, \Omega)$, then computes $\tau \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma)$ and outputs $\pi = (\Omega, \gamma, \tau)$.
- $\mathsf{Verify}^{\mathsf{H}}(x, \pi)$: receives a statement x and a proof $\pi = (\Omega, \gamma, \tau)$, checks that and outputs $b \leftarrow \mathsf{Verify}(x, \Omega, \gamma, \tau) \wedge \gamma = \mathsf{H}(x, \Omega)$.

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

4 NIZK-friendly Signature Scheme

In the following, we describe the signature scheme underlying our construction of a blind signature scheme. The scheme is NIZK-friendly, *i.e.*, compatible with efficiently proving statements about signatures (e.g. knowledge of a valid signature) in NIZK proofs due to its algebraic structure. Looking forward, this property will be useful for creating a blind signature scheme using a Fischlin-inspired construction.

4.1 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. The hash function $\mathsf{H}\colon : \{0,1\}^* \to \{0,1\}^{2\lambda}$ is modelled as a random oracle.

[20] 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 $\widetilde{\mathsf{R}} = \mathsf{R}$.

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

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

4.2 Proof of Security

A detailed proof is given in Appendix B.2. We give a brief sketch below.

Our proof mostly follows the proof given in [41]. 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 [21]:

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.

We also use this strategy in the proof of the blind signature scheme in Section 6. In our blind signature, the primes e are not chosen by the signer but output by a hash function $\mathsf{H}_{\mathbb{P}}$. This still allows the reduction to prepare the list $\mathcal{E}$ of primes used during signing. For a modular security proof, we provide the punctured key generation and alternative signing procedure in Appendix B.1. Since these algorithms depend on guesses with respect to the forgery's format, they are indexed by bits (b, b') that correspond to with in a case distinction in the proof.

5 Novel Commitment Schemes

We give an overview of our novel commitment constructions. These are helpful for our instantiation and influence the choice of primitives that our blind signature relies on. We construct two commitments that admit efficient openings in zero-knowledge:

Relaxed Integer Commitments: allows to commit to integers of a specific range I. The proof of knowledge of an opening proves that the committed integer lies in I. The soundness is relaxed however, *i.e.*, it only guarantees membership in a larger range $I_R \supset I$. Importantly, binding still holds with respect to the range I_R.

Commitments in Groups: allows to commit to elements of arbitrary cyclic groups $\mathbb{G}$ given a generator g of $\mathbb{G}$.

[21] In the detailed proof, the first case has two additional sub cases.

To improve readability, we use additive notation for prime-order groups and multiplicative notation otherwise. Recall that we require several types of commitments in our blind signature construction. Before we describe our novel commitments, we recap the required commitments and sketch how they are instantiated to provide context for the reader. We use the terminology from the technical overview (cf. Section 2):

- *Compatible commitment*: instantiated via MPed commitments (cf. Section 3.1). This is the homomorphic commitment that the user sends to the signer to sign. The user can then later derive a signature on the original message from the signature on the commitment.
- C_Z: instantiated via ElGamal commitments via the observation in remark 1. Details are provided in Appendix E.1. Recall that this commitment is used as an input to the hash function $\mathsf{H}_\mathbb{P}$ to obtain the prime exponent e.
- $\mathsf{C}_{\mathsf{RInt}}$: instantiated via our relaxed integer commitments (cf. Section 5.1). This commitment will be used as a building block of the proof of knowledge of a $\mathsf{S}_{\mathsf{fis}}$ signature that the user outputs in the end.
- $\mathsf{C}_{\mathsf{Grp}}$: This commitment to group elements (cf. Section 5.2) is used as a building block in the NIZK that constitutes our blind signature.

5.1 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 5. *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 11 and 12), 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 13).*

We now instantiate $\mathsf{C}_{\mathsf{RInt}}$ over a group $\mathbb{G}$ with prime order $p \geq 2^{2\lambda}$. Let $\mathsf{pp} = (G, H) \in (\mathbb{G} \setminus \{0\})^2$ be the public parameters, where 0 denotes the neutral element. 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 $(H, G_i, rH, rG_i) \overset{c}{\approx} (H, G_i, rH, t_iG_i)$ for $t_i \leftarrow \mathbb{Z}_p$ under DDH. Since t_iG_i masks m_iH additively, the value $m_iH + t_iG_i$ 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 $m_iH = C_i - rG_i$. 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 $C \leftarrow rH + \sum_i x_iG_i$ 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 6 (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* $\mathcal{C}_{\mathsf{msg}} = \mathcal{C}^{\mathsf{rel}}_{\mathsf{msg}}$*).*

5.2 Commitments in Arbitrary Cyclic 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 $\mathsf{C}^{B,T}_{\mathsf{RInt}}$ (with parameters B, T defined below) [22]. 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 $\mathsf{C}^{B,T}_{\mathsf{RInt}}$ 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 [23]. 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 $\mathsf{C}^{\vec{B},T}_{\mathsf{RInt}}$ 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 $\mathsf{pp} \leftarrow \mathsf{C}^{\vec{B},T}_{\mathsf{RInt}}.\mathsf{Setup}(1^\lambda)$.
- $\mathsf{C}_{\mathsf{Grp}}.\mathsf{Commit}(\mathsf{pp}, \hat{x})$: 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.*

[22] If we instantiate $\mathsf{C}_{\mathsf{RInt}}$ as in Section 5.1, then the additional structure is a prime order group $\mathbb{G}$ in which DDH is assumed to be hard.

[23] 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, $\hat{c} = \hat{g}^s\hat{g}^{s^p}$ can be opened to $\hat{g}^s$ or $\hat{g}^{s^p}$ since $s \equiv s^p \bmod p$. But we have $\hat{g}^s = \hat{g}^{s^p}$ only if $\mathsf{ord}(\hat{g}) \mid s(s^{p-1} - 1)$. Since the order is unknown, this does not hold in general and thus, the commitment is not binding.

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 $s^{(b)} = \sum_{i=1}^{\ell} \vec{s}_i^{(b)} B^{i-1}$. 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 $\mathbb{Z}^\ell$. 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 $\mathsf{C}_{\mathsf{RInt}}$. □

5.3 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 Section 2 for a brief overview. The full schemes are given in Appendix C.1.

6 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.

6.1 Primitives

Before we detail our construction, we prepare the required primitives and related parameters. To see how these primitives fit in the larger picture, we refer to Section 2.

Remark 2. 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. To construct a proof of knowledge of a signature of the scheme $\mathsf{S}_{\mathsf{fis}}$, we use a relaxed integer commitment. The scheme is required to be perfectly binding to fix (parts of) the signature before extraction. 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}, perfect binding and computational hiding (cf. Section 5.1) 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 [24].

For convenience, let $\mathcal{S}_a := [0, A]$ and $\mathcal{S}_e := [\overline{E}, \overline{E} + E]$. 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 $\mathcal{R}_a := [-AT, AT]$ and $\mathcal{R}_e := [\overline{E} - ET, \overline{E} + ET]$. By Eq. (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 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 5.1 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 [25].

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 [58] of Huxley's bound [60, 57]. 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

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

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

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

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 6.1) 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 6). 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 signing. 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.

[24] 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 Eq. (6) if $\lambda \geq 14$.

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

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

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

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 $W_1 = \mathsf{C}_Z.\mathcal{C}_{\mathsf{opn}}$.) The user uses the NIZK to ensure that the commitment c is indeed formed with the values committed via 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) [26].

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. This means that we need to prevent the signer from setting up the signing key for $\mathsf{S}_{\mathsf{fis}}$ in a malicious way that allows the following attack against blindness. Recall that the user sends blinded commitment $c = h_2^{\overline{m}} g^{re}$ to the signer during signing. When $\langle h_2 \rangle \neq \langle g \rangle$, a malicious signer 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 $c^{\mathsf{ord}(g)} = (h_2^{\overline{m}_1})^{\mathsf{ord}(g)}$, and thus breaking blindness.

We carefully design our blind signature such that it actually suffices to check that for some group elements h in the verification key and a generator g, it holds that $\langle g \rangle =: \mathbb{G} = \langle h \rangle$.

We describe how the signer can prove this in a NIZK: 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 $\mathsf{H}_{\mathsf{gen}}$ satisfying statistical adaptive subversion soundness, zero-knowledge, and correctness for the relation

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

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

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

- $\mathsf{H}_{\mathsf{urs}}$: Let $\mathsf{H}_{\mathsf{urs}} : \{0,1\}^* \to \{0,1\}^{\ell_{\mathsf{ped}}} \times \{0,1\}^{\ell_{\mathsf{fis}}} \times \{0,1\}^{\ell_{\mathsf{gen}}}$ be a hash function, where ℓ_{zkp} is the bit-size of the uniform reference string of Π_{zkp}. Later, we use $\mathsf{H}_{\mathsf{urs}}$ 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 $\mathcal{S}_e$.
- H_{pp}: Let $\mathsf{H}_{\mathsf{pp}} : \{0,1\}^* \to \{0,1\}^{\ell_z} \times \{0,1\}^{\ell_{\mathsf{rint}}}$ be a random oracle.
- $\mathsf{H}_{\mathsf{prf}}$: Let $\mathsf{H}_{\mathsf{prf}} : \{0,1\}^\lambda \times \{0,1\}^* \to \mathcal{S}_a$ be a random oracle. We use this hash function like a PRF to make the signer deterministic.

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 $\mathcal{S}_e := [\overline{E}, \overline{E} + E]$ 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 $\mathcal{S}_e$. Appendix D.1 proves the following lemma.

Lemma 1. *For $E = 2^{3\lambda}, \overline{E} = 2^{5\lambda}$, there are $\Omega(2^{2\lambda})$ primes in $\mathcal{S}_e = [\overline{E}, \overline{E} + E]$.*

[26] Roughly, the commitment 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 25).

6.2 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 Fig. 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 $\mathsf{srs}_{\mathsf{zkp}} \leftarrow \Pi_{\mathsf{zkp}}.\mathsf{GenSRS}(1^\lambda)$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}\}$ and $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$. 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 $K \leftarrow \{0,1\}^\lambda$ for $\mathsf{H}_{\mathsf{prf}}$. Finally, output

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

- $\mathsf{BS}_{\mathsf{fis}}.\mathsf{User}(\mathsf{bvk}, m)$: 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 $(c_Z, d_Z) \leftarrow \mathsf{C}_Z.\mathsf{Commit}(\mathsf{pp}, (\overline{m}, r))$. 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 $a \leftarrow \mathsf{H}_{\mathsf{prf}}(K, c \parallel c_Z)$ 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, outputs

$$\rho_2 = (z, a)$$

- $\mathsf{BS}_{\mathsf{fis}}.\mathsf{Derive}(\mathsf{st}_U, \rho_2)$: given state st_U and last message $\rho_2 = (z, a)$, sets $z' \leftarrow h \cdot h_1^a \cdot c \cdot h_2^a$, for $a \in \mathcal{S}_a$, and checks $z^e \equiv z' \bmod N$ given e from st_U. 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 $x_{\mathsf{fis}} = (\mathsf{pp}, N, h_1, h_2, h, \overline{m}, c_I)$.

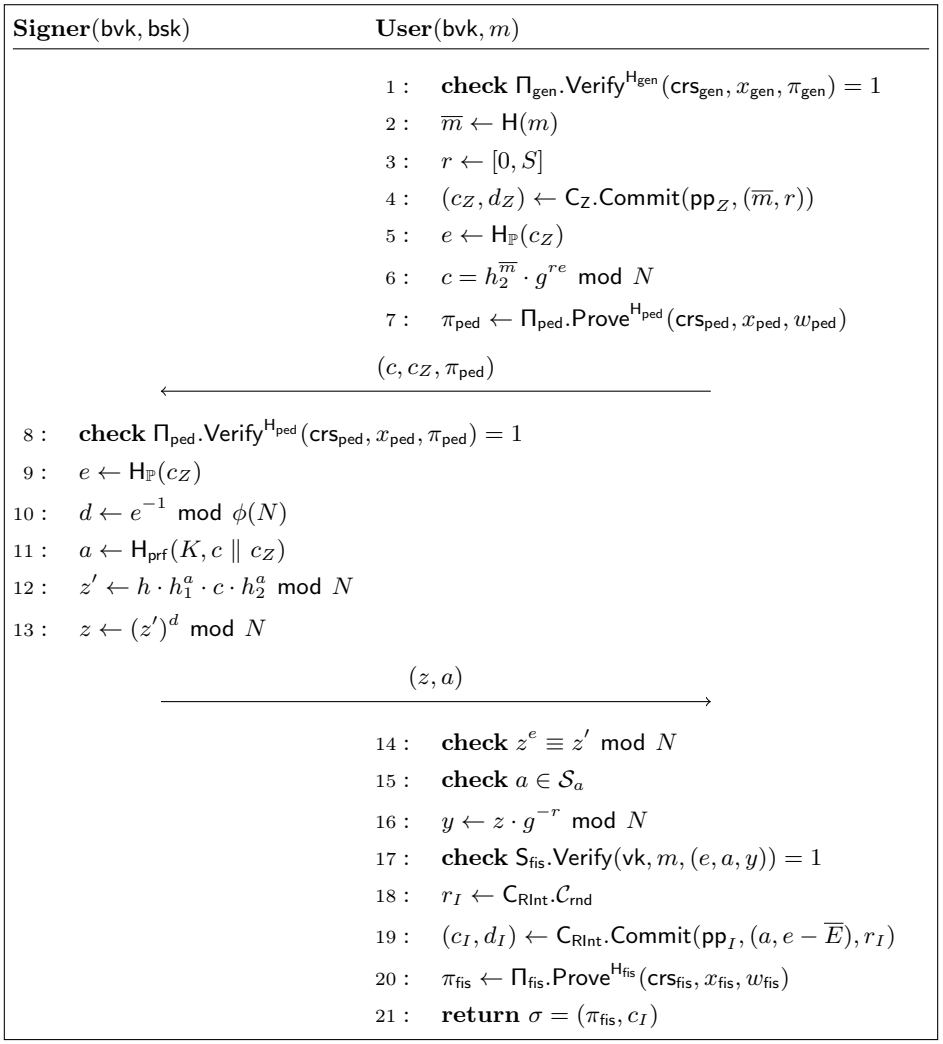

Fig. 1: A signing session of $\mathsf{BS}_{\mathsf{fis}}$ for message m. We have $(\mathsf{pp}_I, \mathsf{pp}_Z) = \mathsf{H}_{\mathsf{pp}}(0)$, $x_{\mathsf{gen}} = (N, 4, (h, h_1, h_2, g))$, $x_{\mathsf{ped}} = (\mathsf{pp}, N, e, h_2, g, c, c_Z)$, $w_{\mathsf{ped}} = (\overline{m}, r, d_Z)$, $x_{\mathsf{fis}}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

6.3 Blindness under Malicious Keys

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

Lemma 2. *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.

Corollary 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, with overwhelming probability over the choice of e*

$$\begin{aligned} \psi : \mathbb{G} &\to \langle g^e \rangle \\ z &\mapsto z^e \bmod N \end{aligned}$$

is a group isomorphism.

We defer the proofs to Appendix D.2. We now state the main theorem for blindness of $\mathsf{BS}_{\mathsf{fis}}$ followed by an overview. The full proof can be found in Appendix D.3.

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

Proof Overview. In the proof we use a sequence of games to transition from the blindness game as in Definition 3 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} for simulating the proofs π_{fis}. Particularly, corollary 1, which ensures that with overwhelming probability the *presignature* z is a unique correct e-th root of a masked $\mathsf{S}_{\mathsf{fis}}$ signature, together with the adaptive soundness of Π_{gen} will guarantee that the derived signature y will be a valid $\mathsf{S}_{\mathsf{fis}}$ signature in order to use the subversion zero-knowledge simulator of Π_{fis}. This allows us to change the commitment c_I of the signature on m_0 to a commitment to 0, under the hiding property of $\mathsf{C}_{\mathsf{RInt}}$, 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. Combining the previous game hop with lemma 2 makes sure that with overwhelming probability over the choices of e the commitment c is Pedersen over $\langle g \rangle$ and its committed values are independent from π_{ped}. Thus c can be switched to a uniformly random $c \leftarrow \langle g \rangle$ 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 end up with the real game for $\mathsf{coin} = 1$.

6.4 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_{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 one-more unforgeable.*

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 4.1. Our final reduction will do the following: It sets up the verification key for $\mathsf{S}_{\mathsf{fis}}$ as the reduction for the scheme in Section 4.1. In particular, this means guessing the format of the "forgery". One guess whether the adversary re-uses a prime e used also in a signing interaction with the signer or whether it picks a new e which may or may not be a prime. This guess we denote by a bit b. In the case that the adversary re-uses the prime e, more guesses are made. The reduction guesses the index j of *which* of the primes will be re-used. The other guess b' concerns the choice of the signature randomness a, namely whether it holds that $a_j \neq a^*$ or $a_j + \overline{m}_j \neq a^* + \overline{m}^*$ where $\overline{m}$ is the hash of a message m. If $\overline{m}_j \neq \overline{m}^*$, at least one of the above will be the case. Analogous to the reduction for plain $\mathsf{S}_{\mathsf{fis}}$ signatures, the reduction manipulates the signature randomness a_j and verification key in such a way that it can sign using exactly one choice of a_j and solve the strong RSA problem for other choices of a^*.

To answer signing queries, it extracts the message and commitment randomness from the NIZK sent by the adversary in the first message of a signing interaction. It then signs the message using the alternate signing key and reapplies the randomness.

To obtain an $\mathsf{S}_{\mathsf{fis}}$ signature from the adversary, it uses the knowledge soundness extractor of Π_{fis}. It then solves the sRSA problem using the same strategy as the reduction described in Appendix B.2.

To apply this reduction we do game hops to arrive 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 4.1). 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.

We refer to Appendix D.4 for a detailed proof.

6.5 Instantiation

We instantiate the primitives from Section 6.1 required for our blind signature $\mathsf{BS}_{\mathsf{fis}}$ as follows. For $\mathsf{C}_{\mathsf{RInt}}$, we use our construction from Section 5.1 which admits efficient opening proofs in zero-knoweldge. For Π_{gen}, we use the construction from Appendix C.1 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 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 10.98 KB communication.

Acknowledgements

We thank Henry Bambury for insightful comments on the Huxley's bounds of the number of primes in a given interval. We also thank the anonymous reviewers for helpful feedback and suggestions. This work was supported in part by the French ANR Project ANR-19-CE39-0011 PRESTO.

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., Ambrona, M., Bogdanov, A., Ohkubo, M., Rosen, A.: Acyclicity programming for sigma-protocols. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part I. LNCS, vol. 13042, pp. 435–465. Springer, Heidelberg (Nov 2021). https://doi.org/10.1007/978-3-030-90459-3_15
3. 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
4. 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
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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

11. 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)
12. 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
13. 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
14. 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
15. 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
16. 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
17. 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
18. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Short blind signatures. Journal of computer security **21**(5), 627–661 (2013)
19. 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
20. 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
21. 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
22. 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
23. 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
24. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 02. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (Sep 2003). https://doi.org/10.1007/3-540-36413-7_20
25. 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
26. 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
27. 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
28. 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)
29. 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
30. 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
31. 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

32. 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
33. 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
34. 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
35. 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
36. 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)
37. 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
38. 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
39. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (Aug 1984)
40. 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
41. 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
42. 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
43. 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
44. 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
45. 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
46. 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
47. 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
48. Fuchsbauer, G., Wolf, M.: Concurrently secure blind schnorr signatures. Cryptology ePrint Archive, Paper 2022/1676 (2022), https://eprint.iacr.org/2022/1676, https://eprint.iacr.org/2022/1676
49. 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
50. 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
51. 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
52. 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)

53. Google: VPN Google One. https://one.google.com/about/vpn/howitworks
54. 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
55. 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
56. 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
57. 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
58. 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
59. 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
60. Huxley, M.: On the difference between consecutive primes. Inventiones mathematicae **15**, 164–170 (1971/72), http://eudml.org/doc/142126
61. 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
62. Kastner, J., Loss, J., Xu, J.: The abe-okamoto partially blind signature scheme revisited. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 279–309. Springer, Heidelberg (Dec 2022). https://doi.org/10.1007/978-3-031-22972-5_10
63. 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
64. 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)
65. 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
66. 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
67. 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
68. 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
69. Lysyanskaya, A.: Security analysis of rsa-bssa (2023), https://eprint.iacr.org/2022/895
70. 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
71. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_6
72. Nishimaki, R.: Equipping public-key cryptographic primitives with watermarking (or: A hole is to watermark). In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 179–209. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64375-1_7
73. 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
74. 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
75. 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

76. 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
77. 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
78. 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)
79. 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
80. 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
81. 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
82. Nist announcess additional digital signature candidates for the pqc standardization process. https://csrc.nist.gov/projects/pqc-dig-sig (2023), accessed: 2023-10-06
83. mcl-wasm library for pairings. https://github.com/herumi/mcl-wasm (2023), accessed: 2023-10-02
84. 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
85. Supported ssh algorithms. https://privx.docs.ssh.com/docs/supported-ssh-key-exchange-algorithms (2022), accessed: 2023-10-06
86. 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

Supplementary Material

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 $f = \mathsf{negl}(\lambda)$ 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 3 (Proposition 1, [34]). *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 [32]. 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 $1/L$. This is easy to see as it is a requirement for the abort that either

Noise Flooding. Let $V, L \in \mathbb{N}$. We define noise flooding for the interval $[-V, V]$ with masking overhead $L = 2^\lambda$. Let $v \in [-V, V]$. To mask v additively with a mask μ via noise flooding, output $w = v + \mu$, where $\mu \leftarrow [0, VL]$ is sampled at random. The value w is distributed close to uniform over $[0, VL]$ with statistical distance at most $1/L = \mathsf{negl}(\lambda)$.

Forking Lemma. We state here a version of Forking Lemma [77, 14] that fits our usage of it.

Lemma 4 (Lemma 1, [2]). *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 $P', Q' > 2^{\lambda+1}$.

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

Lemma 5. *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 e ciently compute $\bar{x} \in \mathbb{Z}_N^*$ such that $\bar{x}^a = y$.*

Remark 3. 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 [32] (for the hidden order group QR_N).

Definition 7 ((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 [32] sufficient for our purpose. Note that $\mathsf{ord}(\mathsf{QR}_N) = P'Q'$ and we assume that $P', Q' > 2^{\lambda+1}$.

Lemma 6. *Let $D \leq 2^{\lambda+1}$ and $\ell = \mathsf{poly}(\lambda)$. For 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 8 (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*

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

is negligible in λ.

Definition 9 (Strong RSA). *Let $\lambda \in \mathbb{N}$. The* strong RSA (sRSA) *assumption holds if for all PPT $\mathcal{A}$ the advantage*

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

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

Definition 10 (Commitment Scheme). *A* commitment scheme *is a tuple of algorithms* $\mathsf{C} = (\mathsf{C.Commit}, \mathsf{C.Verify})$ *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 11 (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 12 (Hiding). *A commitment scheme is* hiding *if for any PPT adversary $\mathcal{A}$, we have*

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

Definition 13 (Binding). *A commitment scheme is* binding *if for any PPT adversary $\mathcal{A}$, we have*

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

Remark 4. A commitment scheme is said to be *perfectly binding* if for any (possibly unbounded) $\mathcal{A}$, it holds that $\mathsf{Adv}^{\text{bind}}_{\mathcal{A}}(\lambda) = 0$.

(Bounded) Integer Commitments. We refer to a commitment scheme with message space $[A,B] \subseteq \mathbb{N}$ as a (bounded) integer commitment scheme. We often omit the term bounded if the message space is clear by context.

ElGamal commitments. We recall ElGamal (EG) over a group $\mathbb{G}$ of prime order p with message space $\mathbb{Z}_p$ [39]. We use additive notation for prime order groups.

- $\mathsf{EG.GenPP}(1^\lambda)$: set $(G, H) \leftarrow \mathbb{G} \setminus \{0\}$ and output $\mathsf{pp} = (G, H)$.
- $\mathsf{EG.Commit}(\mathsf{pp}, m)$: sample $r \leftarrow \mathbb{Z}_p$ and set $c = (mG + rH, rG)$, and output (c, r).
- $\mathsf{EG.Verify}(\mathsf{pp}, c, m, r)$: check if $c = (mG + rH, rG)$.

Note that the public parameters are uniform and we can sample them via a random oracle to avoid trusted setup. EG commitments are correct, hiding under DDH and perfectly binding.

Remark 5. If in verification of EG, we check that $m \in [0, M]$ for $M < p$, then m is fixed over the integers and we can interpret the commitment as an integer commitment with message space $[0, M] \subseteq \mathbb{N}$.

Pedersen Commitments in $\mathbf{QR}_N$ We recall Pedersen multi-commitments (MPed) over QR_N with message space $\mathbb{Z}^\ell$ for some $\ell \in \mathbb{N}$ [34].

- $\mathsf{MPed.GenPP}(1^\lambda)$: 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 $c \leftarrow h^r \cdot \prod_{i=1}^{\ell} g_i^{m_i} \mod N$, and output (c, r).
- $\mathsf{MPed.Verify}(\mathsf{pp}, c, \vec{m}, r)$: check if $c = \pm h^r \cdot \prod_{i=1}^{\ell} g_i^{m_i} \mod N$.

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 6.

A.6 Signature Scheme

Definition 14 (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* $m \in \mathcal{S}_{\mathsf{msg}}$*, 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 15 (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 16 (EUF-CMA). *A signature scheme is* `euf-cma` *if for any PPT adversary* $\mathcal{A}$*, we have*

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

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

A.7 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) Ω and a state st, where we assume st includes x, w,

- $\mathsf{Chall}()$: samples a challenge $\gamma \leftarrow \mathcal{CH}$ (without taking any input),
- $\mathsf{Resp}(\mathsf{st}, \gamma)$: given a state st and a challenge $\gamma \in \mathcal{CH}$, outputs a third flow message (*i.e.*, response) τ,
- $\mathsf{Verify}(x, \Omega, \gamma, \tau)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, a commitment Ω, a challenge $\gamma \in \mathcal{CH}$, and a response τ, outputs a bit $b \in \{0,1\}$.

Definition 17 (Correctness). *A Σ-protocol is* correct*, if for all $(x,w) \in \mathsf{R}$, $(\Omega, \mathsf{st}) \leftarrow \mathsf{Init}(x,w)$, $\gamma \in \mathcal{CH}$, and $\tau \leftarrow \mathsf{Resp}(\mathsf{st}, \gamma)$, it holds that $\mathsf{Verify}(x, \Omega, \gamma, \tau) = 1$.*

Definition 18 (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*

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

Definition 19 (Non-abort HVZK). *A Σ-protocol is* non-abort honest-verifier zero-knowledge *(HVZK), if there exists a PPT zero-knowledge simulator Sim such that the distributions of $\mathsf{Sim}(x,\gamma)$ and the honestly generated transcript with Init initialized with (x,w) are statistically indistinguishable for any $x \in \mathscr{L}_{\mathsf{R}}$, and $\gamma \in \mathcal{CH}$, where the honest execution is conditioned on γ being used as the challenge and no abort occurring.*

We write HVZK for short if the Σ-protocol never aborts.

Definition 20 (k-Special Soundness). *A Σ-protocol is k-*special sound*, if there exists a* deterministic *PT extractor Ext such that given k valid transcripts $\{(\Omega, \gamma_i, \tau_i)\}_{i \in [k]}$ for statement x with pairwise distinct challenges $(\gamma_i)_i$, outputs a witness w such that $(x,w) \in \mathsf{R}$.*

A.8 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 [27]. An NIZK for a relation R with common reference string space $\mathcal{CRS} = \mathcal{SRS} \times \mathcal{URS}$ is a tuple of PPT algorithms $(\mathsf{GenSRS}, \mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$, where the latter two are oracle-calling, such that:

- $\mathsf{GenSRS}(1^\lambda)$: 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 21 (Correctness). *An NIZK is* correct *if for any $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ with $\mathsf{srs} \leftarrow \mathsf{GenSRS}(1^\lambda)$ 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 22 (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*

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

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.

[27] 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 GenSRS. (We allow that $\mathcal{SRS}$ is not equal to the output space of GenSRS.)

We also define a notion of *subversion zero-knowledge*, inspired by the notion introduced in [12]. Informally, it guarantees that zero-knowledge holds even for a malicious crs.

Definition 23 (Subversion Zero-Knowledge). *An* NIZK *is* subversion zero-knowledge *(*Sub-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*

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

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 24 (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*

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

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

Knowledge Soundness. *There exists positive polynomials* $p_{\mathsf{T}}, p_{\mathsf{P}}$ *and a constant* c *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

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

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 25 (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 *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*

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

it holds that

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

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 26 (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*

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

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 [40] 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*,* honest verifier zero-knowledge*, and* 2-Special Soundness*. The* Fiat-Shamir transformation $FS[\Sigma]=(\mathsf{GenSRS},\mathsf{Prove}^{\mathsf{H}},\mathsf{Verify}^{\mathsf{H}})$ *is described below:*

- $\mathsf{GenSRS}(1^\lambda)$*: 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* $(\Omega,\mathsf{st})\leftarrow\mathsf{Init}(x,w)$*, computes the challenge* $\gamma\leftarrow\mathsf{H}(x,\Omega)$*, then computes* $\tau\leftarrow\mathsf{Resp}(\mathsf{st},\gamma)$ *and outputs* $\pi=(\Omega,\gamma,\tau)$*.*
- $\mathsf{Verify}^{\mathsf{H}}(x,\pi)$*: receives a statement* x *and a proof* $\pi=(\Omega,\gamma,\tau)$*, and outputs* $b\leftarrow\mathsf{Verify}(x,\Omega,\gamma,\tau)\wedge\gamma=\mathsf{H}(x,\Omega)$*.*

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

B Deferred Content from Section 4

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:

$\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{0,0}(1^\lambda,N,z)$ 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 $h_1:=z^{2\prod_{i\neq j}e_i}$, $h_2:=v^{2\prod_i e_i}$, $h:=h_1^{-\beta}\cdot w^{2\prod_i e_i}$. Output $\mathsf{vk}=(N,h,h_1,h_2)$ along with $\mathsf{sk}_{0,0}=(\beta,v,w,e_1,\ldots,e_{Q_S},j)$

$\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{0,1}(1^\lambda,N,z)$ 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 $j\leftarrow\{1,\ldots,Q_S\}$. Set $h_1:=v^{2\prod_i e_i}$, $h_2:=z^{2\prod_{i\neq j}e_i}$, $h:=h_2^{-\beta}\cdot w^{2\prod_i e_i}$. Output $\mathsf{vk}=(N,h,h_1,h_2)$ along with $\mathsf{sk}_{0,1}=(\beta,v,w,e_1,\ldots,e_{Q_S},j)$

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

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

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

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

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

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

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

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

$\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{1}(a, a', e_1, \ldots, e_{Q_S}, k, \overline{m})$ 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 holds and the hash function* H *is a random oracle mapping from* $\{0,1\}^*$ *to* $\{0,1\}^{2\lambda}$ *then the scheme described above is EUF-CMA secure.*

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 by bounding the number of collisions in H, namely,

$$|\Pr[\text{Game } 2 = 1] - \Pr[\text{Game } 1 = 1]| \leq \frac{Q_{\mathsf{H}}^2}{2 \cdot 2^{2\lambda}}$$

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 $e^* \notin \{e_1, \ldots, e_{Q_S}\}$ or if $b = 1$ and $e^* \in \{e_1, \ldots, e_{Q_S}\}$. 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 $b = 0$, 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 $b = 0$, 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. We argue why the abort probability is $\frac{1}{2}$. As the adversary is not allowed to re-sign a previously signed message, it holds that $m_j \neq m^*$. This, along with the abort condition induced in Game 2, implies that $\mathsf{H}(m^*) \neq \mathsf{H}(m_j)$. Thus, if the adversary chooses $a^* = a_j$, it must be the case that $a_j + \mathsf{H}(m_j) \neq a^* + \mathsf{H}(m^*)$. Otherwise it holds that $a^* \neq a_j$. The bit b is chosen independently of this choice of the adversary regarding his forgery and therefore,

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

Game 6: In Game 6, we 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 $b = 0$, $b' = 0$, 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 $e^* \notin \{e_1, \ldots, e_{Q_S}\}$), the reduction works as follows:

Setup. Runs $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_{0,b'}(1^\lambda, N, z)$ 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 $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_{0,b'}(\mathsf{sk}_{0,b'}, k, \mathsf{H}(m))$ 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

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

If $b' = 0$, solving for z using the preselected values from the public key yields:

$$z^{2 \prod_{i \neq j} e_i \cdot (a^* - a_j)} = \left(v^{2 \prod_{i \neq j} e_i \cdot (a_j + \mathsf{H}(m_j) - a^* - \mathsf{H}(m^*))} \right)^{e_j} (y^* y_j^{-1})^{e_j}$$

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 5. 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.
Analogously, if $b' = 1$, we swap the roles of h_1 and h_2 and get

$$z^{2\prod_{i\neq j} e_i\cdot(\overline{m}^*+a^*-(\overline{m}_j+a_j))} = \left(v^{2\prod_{i\neq j} e_i\cdot(a_j-a^*)}\right)^{e_j}(y^* y_j^{-1})^{e_j}.$$

Again, using the same reasoning as above, we can solve for an e_jth root of z using lemma 5.

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

Setup. Given N and $z \in \mathbb{Z}_N$, the reduction runs $\mathsf{S}_{\mathsf{fis}}.\mathsf{KeyGen}_1(1^\lambda, N, z)$ 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 $\mathsf{S}_{\mathsf{fis}}.\mathsf{Sign}_1(\mathsf{sk}_1, k, \mathsf{H}(m))$ 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 [41] 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

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

C Deferred Content from Section 5

C.1 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 6. It is based on the Σ-protocol Σ_{gen} given in Fig. 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.*, $\mathcal{SRS} = \mathcal{URS} = \{\bot\}$).

- $\Pi_{\mathsf{gen}}.\mathsf{GenSRS}(1^\lambda)$: 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

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

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

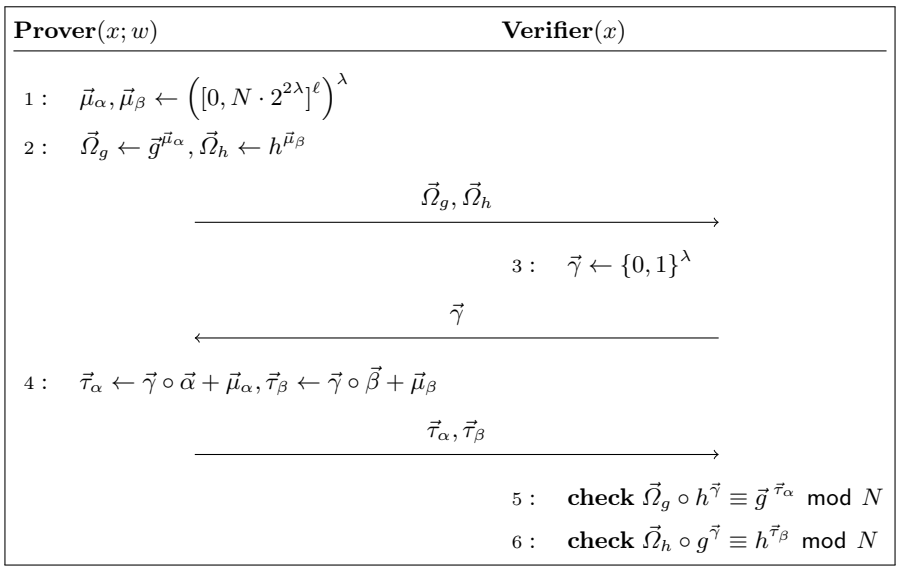

Fig. 2: Description of Σ_{gen} for $x = (N, \ell, h, \vec{g})$ and $w = (\alpha_i, \beta_i)_{i\in[\ell]}$ with $\vec{g} = (g_1, \cdots, g_\ell)$. We denote the Hadamard product by $\circ$.

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

Theorem 7. *The Σ-protocol Σ_{gen} given in Fig. 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, we recall that Σ-protocol Σ_{gen} is used for N comes from the $\mathsf{pp} = (N, h, \vec{g})$ for the Pedersen commiment MPed in QR_N. (Since membership in QR_N cannot be efficiently tested without factorization of N, the MPed commitment is formally defined over $\mathbb{Z}_N^*$. Later, N and pp are given in the crs, and pp are generators of QR_N else we find witness of some relaxed DLOG relation *as per* Definition 7.) Thus as long as $h, g_i \neq 1$ for $i \in [\ell]$, the groups $\langle h\rangle, \langle g_i\rangle$ are of exponentially large orders in λ. Therefore, using the fact that the space $\left([0, N \cdot 2^{2\lambda}]^\ell\right)^\lambda$ is exponentially large in λ, Σ_{gen} has *high min-entropy*. Next, *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 $(\vec{\Omega}_g, \vec{\Omega}_h), \vec{\gamma}^b, \vec{\tau}_\alpha^b, \vec{\tau}_\beta^b$ where $b \in \{0,1\}$ and $\vec{\gamma}^0 \neq \vec{\gamma}^1$, a deterministic polynomial-time extractor Ext can executes as follows: First, identify an index j s.t. $\gamma_j^0 \neq \gamma_j^1$

1. For each $i \in [\ell]$, Ext sets $\alpha_i \coloneqq \frac{\vec{\tau}_{\alpha,i}^0 - \vec{\tau}_{\alpha,i}^1}{\gamma_j^0 - \gamma_j^1}$.
2. For each $i \in [\ell]$, Ext sets $\beta_i \coloneqq \frac{\vec{\tau}_{\beta,i}^0 - \vec{\tau}_{\beta,i}^1}{\gamma_j^0 - \gamma_j^1}$.
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 $\gamma^0 \neq \gamma^1$, $\gamma_j^0 - \gamma_j^1 \in \{-1, 1\}$ (and thus has an efficiently computable multiplicative inverse) and

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

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 $\vec{\tau}_{\alpha,i}, \vec{\tau}_{\beta,i} \leftarrow ([0, N \cdot 2^{2\lambda}])^\lambda$.
   - Sim computes $\vec{\Omega}_{g,i} := g_i^{\vec{\tau}_{\alpha,i}} (h^{\vec{\gamma}})^{-1}$ and $\vec{\Omega}_{h,i} := h^{\vec{\tau}_{\beta,i}} \circ (g_i^{\vec{\gamma}})^{-1}$. The commitments are defined $\vec{\Omega}_g := (\vec{\Omega}_{g,i})_{i \in [\ell]}, \vec{\Omega}_h := (\vec{\Omega}_{h,i})_{i \in [\ell]}$.
   - 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 follows a distribution statistically close to that of the honestly generated transcript with Init initialized with (x, w). First we argue the distribution of the commitments $\vec{\Omega}_g, \vec{\Omega}_h$. We use the fact that because $\langle h \rangle = \langle g_i \rangle$, it holds that $\vec{\Omega}_{g,i} := \frac{g_i^{\vec{\tau}_{\alpha,i}}}{h^{\vec{\gamma}}} \in \langle g_i \rangle$ and $\vec{\Omega}_{h,i} := \frac{h^{\vec{\tau}_{\beta,i}}}{g_i^{\vec{\gamma}}} \in \langle h \rangle$ having the statistically close distributions thanks to **Noise Flooding** recalled in Appendix A.2. Indeed, $\vec{\tau}_\alpha, \vec{\tau}_\beta \leftarrow ([0, N \cdot 2^{2\lambda}]^\ell)^\lambda$, where each i-th responses $\vec{\tau}_{\alpha,i}, \vec{\tau}_{\beta,i} \leftarrow ([0, N \cdot 2^{2\lambda}])^\lambda$, act as masks for the given values $\vec{\gamma} \circ \vec{\alpha}, \vec{\gamma} \circ \vec{\beta} \in ([0, N]^\ell)^\lambda$. By noise flooding the induced $\vec{\mu}_\alpha \leftarrow -\vec{\gamma} \circ \vec{\alpha} + \vec{\tau}_\alpha$, $\vec{\mu}_\beta \leftarrow -\vec{\gamma} \circ \vec{\beta} + \vec{\tau}_\beta$ are distributed close to uniform within distance $1/2^{2\lambda}$. Next we argue that the distribution of the *real* responses $\vec{\tau}_\alpha, \vec{\tau}_\beta$ in the real protocol are statistically close to unform over $([0, N \cdot 2^{2\lambda}]^\ell)^\lambda$. More specifically, in the real protocol

$$\vec{\tau}_\alpha \leftarrow \vec{\gamma} \circ \vec{\alpha} + \vec{\mu}_\alpha, \vec{\tau}_\beta \leftarrow \vec{\gamma} \circ \vec{\beta} + \vec{\mu}_\beta$$

where $\vec{\mu}_\alpha, \vec{\mu}_\beta \leftarrow \left([0, N \cdot 2^{2\lambda}]^\ell\right)^\lambda$ act as masks for $\vec{\gamma} \circ \vec{\alpha}, \vec{\gamma} \circ \vec{\beta} \in ([0, N]^\ell)^\lambda$. Then similarly noise flooding concludes that the induced $\vec{\tau}_\alpha, \vec{\tau}_\beta$ are distributed close to uniform within distance $1/2^{2\lambda}$. This means the way of simulating $\vec{\tau}_\alpha, \vec{\tau}_\beta \leftarrow ([0, N \cdot 2^{2\lambda}])^\lambda$ is statistically close to the real responses and 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 $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{snd}}(\lambda) \leq \frac{Q_{\mathsf{H}_{\mathsf{gen}}}}{2^\lambda}$.

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 $C \coloneqq 2^{\lambda}-1$ using Fiat-Shamir with abort. To ensure (relaxed) range membership we use techniques from [33, 32]. 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 Fig. 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 $\Sigma_{\mathsf{int}}.\mathsf{Resp}$ 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 6 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* $(1 - \frac{1}{L})^{-\ell} = \mathsf{poly}(\lambda)$*, 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.

Correctness. Note that a single run succeeds with probability $(1 - \frac{1}{L})^\ell$ because $1/L$ is the abort probability of the size check in line 10 per coordinate (cf. Appendix A.2). Thus, proof generation

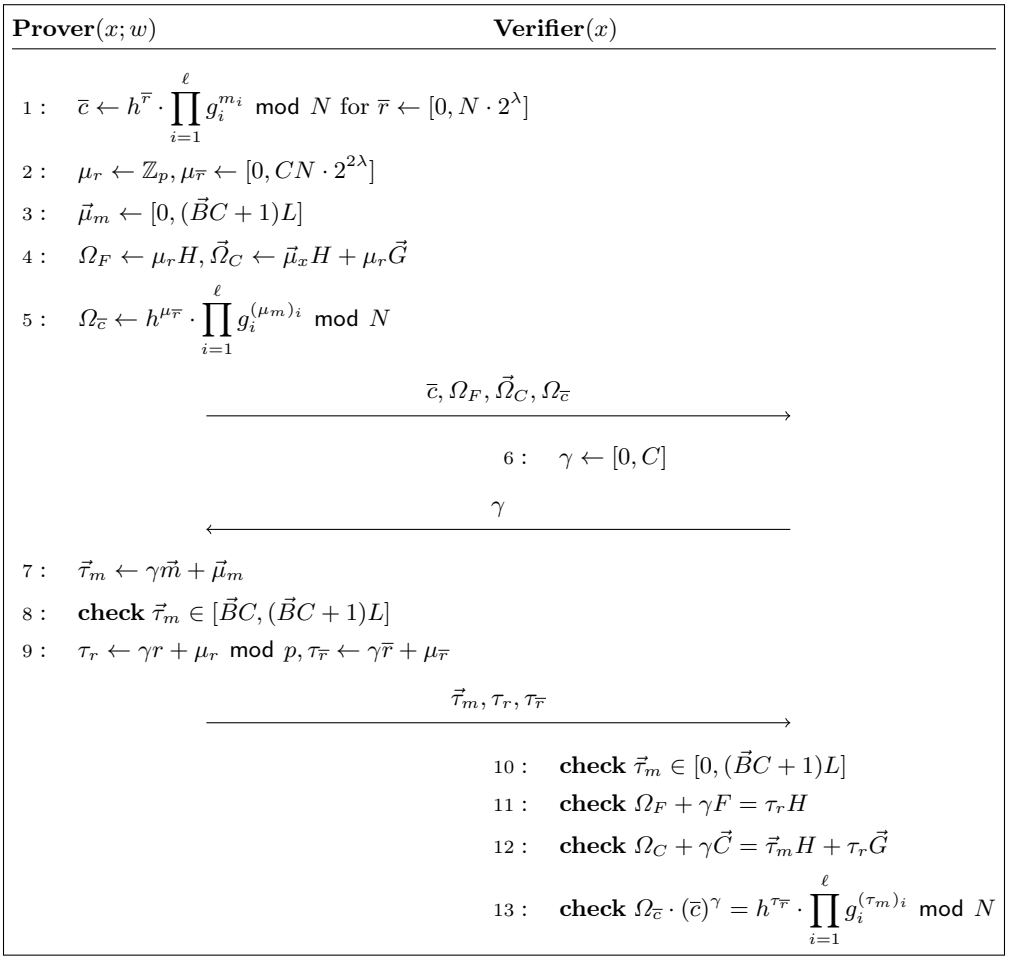

Fig. 3: Description of Σ_{int}, an efficient Σ-protocol for opening $\mathsf{C}_{\mathsf{RInt}}$. Here, $x = (\mathsf{pp}, \vec{C}, F, \mathsf{crs})$ and $w = (\vec{m}, r)$. Also, $\mathsf{crs} = (N, h, \vec{g}, \pi_{\mathsf{gen}})$ for $\vec{g} = (g_1, \cdots, g_\ell)$. If a check fails, the party aborts.

runs in time $\mathcal{O}((1 - \frac{1}{L})^{-\ell})$ in expectation. In case of no abort, the verification equations verify by construction.

Subversion zero-knowledge. This follows with standard arguments. We sketch the zero-knowledge simulator below. The simulator samples a challenge $\gamma_\Sigma \leftarrow [0, C]$ and $\vec{\tau}_m \leftarrow [\vec{B}C, (\vec{B}C+1)L]$, $\tau_r \leftarrow \mathbb{Z}_p$ and $\tau_{\bar{r}} \leftarrow [0, (\vec{B}C + 1)L]$. It is easy to check that the response follows the honest distribution (conditioned on no abort due to Appendix A.2). Next, the simulator an MPed commitment $\bar{c}$ to zero. Because for any $\mathsf{srs} \in \mathcal{SRS}$, the scheme MPed is statistically hiding (under soundness of Π_{gen}), this commitment also follows the distribution of honestly generated $\bar{c}$ with negligible statistical distance. Finally, the simulator samples $\Omega_F, \vec{\Omega}_C, \Omega_{\bar{c}}$ according to the verification equations, and sets

- $\Omega_\Sigma = (\bar{c}, \Omega_F, \vec{\Omega}_C, \Omega_{\bar{c}})$,
- $\tau_\Sigma = (\vec{\tau}_m, \tau_r, \tau_{\bar{r}})$.

Before the simulator outputs the proof $\pi = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$, it programs the random oracle $\mathsf{H}_{\mathsf{int}}$ accordingly. Observe that the underlying Σ-protocol has high min-entropy (since Ω_F is distributed uniform over $\mathbb{G}$), thus $\mathsf{H}_{\mathsf{int}}$ is not defined at input $(x_\Sigma, \bar{c}, \Omega_F, \vec{\Omega}_C, c_Z \bar{c})$ yet with overwhelming probability. As discussed above, conditioned on no abort, the proof π is identically distributed to honestly generated proofs. Repetitions due to aborts are only noticeable if the adversary observes $\mathsf{H}_{\mathsf{int}}$ queries for aborted transcripts. Due to high min-entropy, this occurs only with negligible probability.

Adaptive knowledge soundness. For soundness, we obtain two valid transcripts $tr = (\alpha, \gamma, \omega)$, $tr' = (\alpha, \gamma', \omega')$ with shared $\alpha = (\bar{c}, \Omega_F, \vec{\Omega}_C, \Omega_{\bar{c}})$ 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

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

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 $\bar{c}$. As above, we obtain

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

Recall that $\Delta\gamma \in [0, C]$ with $C = 2^\lambda - 1$. Under lemma 6, 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 $\mathsf{C}_{\mathsf{Grp}}$. A commitment of $\mathsf{C}_{\mathsf{Grp}}$ consists of a Pedersen commitment (in $\hat{\mathbb{G}}$) and a $\mathsf{C}_{\mathsf{RInt}}$ commitment. If $\mathsf{C}_{\mathsf{RInt}}$ is instantiated as in Section 5.1, it is straightforward to obtain a NIZK for opening $\mathsf{C}_{\mathsf{Grp}}$ in zero-knowledge using the techniques from Appendix C.1 (since the decomposition of s is linear). An example of this NIZK is given within the NIZK provided in Appendix E.2.

D Deferred content from Section 6

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

Lemma 1. *For $E = 2^{3\lambda}, \overline{E} = 2^{5\lambda}$, there are $\Omega(2^{2\lambda})$ primes in $\mathcal{S}_e = [\overline{E}, \overline{E} + E]$.*

Proof. 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 [58], which refines the celebrated Huxley's bound [60, 57], 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 Eq. (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

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

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 2 and Corollary 1

Lemma 2. *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.

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

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

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

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

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[28]. The proof is completed. □

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

Corollary 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, with overwhelming probability over the choice of e*

$$\begin{aligned} \psi : \mathbb{G} &\to \langle g^e \rangle \\ z &\mapsto z^e \bmod N \end{aligned}$$

is a group isomorphism.

Proof. Suppose $\langle g^e \rangle = \mathbb{G}$, we will prove that

$$\begin{aligned} \psi : \mathbb{G} &\to \langle g^e \rangle \\ z &\mapsto z^e \bmod N \end{aligned}$$

which is a group isomorphism. For $a, b \in \mathbb{G}$, $\psi(ab^{-1}) = \left(ab^{-1}\right)^e = \psi(a) \cdot \psi(b)^{-1} \bmod N$ by arithmetic in $\mathbb{Z}_N^*$. As a consequence ψ is a group homomorphism. We now show that ψ is *surjective*. Thanks to the hypothesis $\langle g^e \rangle = \mathbb{G}$, it holds that $\gcd(e, \mathsf{ord}(\mathbb{G})) = 1$, $\mathsf{ord}(\mathbb{G}) = \mathsf{ord}(\langle g^e \rangle)$, and $e^{-1} \bmod \mathsf{ord}(\langle g^e \rangle)$ is well defined. Therefore, for any $z' \in \langle g^e \rangle$, we define $d \coloneqq e^{-1} \bmod \mathsf{ord}(\langle g^e \rangle)$, and $z \coloneqq (z')^d \bmod N$. It can be verified that

$$\psi(z) = (z')^{ed} = z' \bmod N$$

as $ed \equiv 1 \bmod \mathsf{ord}(\langle g^e \rangle)$. Next, ψ is *injective* if and only if $\ker(\psi) = \{1\}$. The inclusion $\{1\} \subseteq \ker(\psi)$ is clear. Suppose for the sake of contradiction that there exists $1 \neq z \in \mathbb{G}$ so that $\psi(z) = 1$. This implies $z^e \equiv 1 \bmod N$ and thus $\mathsf{ord}(z) \mid e$. Moreover, from $1 \neq z \in \mathbb{G}$ it holds that $1 < \mathsf{ord}(z) \mid \mathsf{ord}(\mathbb{G})$ thanks to Larange. Combiningly we obtain $1 < \mathsf{ord}(z) \mid \gcd(e, \mathsf{ord}(\mathbb{G}))$, which contradicts the hypothesis that $\langle g^e \rangle = \mathbb{G}$. Therefore $\ker(\psi) = \{1\}$ and ψ is injective.

Finally, with overwhelming probability over the choice of $e \leftarrow \mathcal{S}_e$, lemma 2 concludes that $\langle g^e \rangle = \mathbb{G}$ and this finishes the proof. □

D.3 Blindness under Malicious Keys of $\mathsf{BS}_{\mathsf{fis}}$ - Proof of Theorem 3

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

Proof. We proceed by a sequence of hybrids. We denote by $\mathsf{Adv}^{\text{blind}}_{\mathcal{A},Game\ i}(\lambda)$ the probability that a PPT adversary $\mathcal{A}$ outputs 1 in *Game* i. We assume that all the **check** steps are passed during the execution. This is *not* without loss of generality, but for the ease of presentation. In Remark 6 we elaborate on the cases when some **check** steps are not passed.

Game 1: We start with the game following Definition 3 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 $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{gen}} \in \mathcal{URS}$. The blindness adversary $\mathcal{A}$ also sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}$ 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 7 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

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

and is negligible in λ.

Lemma 7. *Under the* subversion zero-knowledge *of* Π_{fis} *as well as the* subversion adaptive soundness Π_{gen}*, the games Game 2 and Game 1 are indistinguishable. For any blindness adversary* $\mathcal{A}$*, there exist PPT* $\mathcal{B}_2, \mathcal{B}_3$ *so that*

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

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 suppose that 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

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

From Corollary 1, with overwhelming probability over the choice of $e \leftarrow \mathcal{S}_e$, raising to the power of e is a bijection. Therefore, except with negligible probability, inequality (†) contradicts the fact that during Derive it is set $z' \leftarrow h \cdot h_1^a \cdot c \cdot h_2^a$ and the hypothesis that the **check** $z^e \equiv z' \bmod N$ holds. Equivalently, this current case with the inequality (†) happens only with negligible probability.

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 subversion soundness of Π_{gen}.

We provide a PPt adversary $\mathcal{B}_2$ breaking the subversion 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 $\mathsf{urs}_{\mathsf{ped}} \in \mathcal{URS}$. Then $\mathcal{B}_2$ receives $\mathsf{urs}_{\mathsf{gen}}$ from its subversion soundness challenger.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{fis}} = (\mathsf{srs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{fis}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}$ in bvk.
- Specifically, as soon as $\mathcal{A}$ outputs

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

$\mathcal{B}_2$ outputs the instance $(N, 3, g, (h, h_1, h_2))$ to its challenger against the subversion 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 subversion 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 $\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{gen}} \in \mathcal{URS}$.
- 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 $\mathsf{H}_{\mathsf{ped}}$ queries, receives (z, a) 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 Fig. 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 23, 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 23, 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

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

for PPT adversaries $\mathcal{B}_2, \mathcal{B}_3$ as described above. Assuming the *subversion zero-knowledge* of Π_{fis} as well as the *subversion adaptive soundness* of Π_{gen} against all such PPT $\mathcal{B}_2, \mathcal{B}_3$, 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

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

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 $(\mathsf{urs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}}) \leftarrow \mathsf{H}_{\mathsf{urs}}(0)$ together with honest $\mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}} \in \mathcal{URS}$. The blindness adversary $\mathcal{A}$ also sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$ along with $\mathsf{crs}_{\mathsf{fis}}, \mathsf{crs}_{\mathsf{gen}}$ 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 8 argues that this simulation of π_{ped} is indistinguishable from the real proofs. The following lemma 8 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 8. *Under the* subversion zero-knowledge *of* Π_{ped} *as well as the* subversion 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*

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

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:

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

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 subversion soundness of Π_{gen}. We provide a PPT adversary $\mathcal{B}_1$ breaking the subversion 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 $\mathsf{urs}_{\mathsf{fis}} \in \mathcal{URS}$. Then $\mathcal{B}_1$ receives $\mathsf{urs}_{\mathsf{gen}}$ from its subversion soundness challenger.
- The blindness adversary $\mathcal{A}$ sets up $\mathsf{srs}_{\mathsf{zkp}}$ for $\mathsf{zkp} \in \{\mathsf{ped}, \mathsf{fis}, \mathsf{gen}\}$. The common reference strings are defined, in particular $\mathsf{crs}_{\mathsf{ped}} = (\mathsf{srs}_{\mathsf{ped}}, \mathsf{urs}_{\mathsf{ped}})$ along with $\mathsf{crs}_{\mathsf{ped}}, \mathsf{crs}_{\mathsf{gen}}$ in bvk.
- Specifically, as soon as $\mathcal{A}$ outputs

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

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

Hence, the probability of this case is bounded by $\mathsf{Adv}^{\mathrm{snd}}_{\mathcal{B}_1,\Pi_{\mathsf{gen}}}(\lambda)$ for some PPT $\mathcal{B}_1$ against the subversion 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 $\mathsf{urs}_{\mathsf{fis}}, \mathsf{urs}_{\mathsf{gen}} \in \mathcal{URS}$.
- 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 Fig. 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 23, 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 23, 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 *subversion 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 subversion soundness of Π_{gen}. We provide a PPT adversary $\mathcal{B}_1$ breaking the subversion 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 statistical subversion soundness of Π_{gen}.

Case 2 Else, suppose that $\langle h_2 \rangle = \langle g \rangle$. By lemma 2 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 3 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 subversion 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 3 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 negligible in λ and the proof is completed. □

Remark 6. We now give more details on the argument when some checks in the blindness game are not passed. Suppose there is m_b among the blindness challenges (m_0, m_1) such that one of the checks in User or Derive does not pass.

- If the failed check is in User : The only check therein is on bvk, independent from (m_0, m_1), hence the first signing messages from User are

  $$\rho_{1,0} = \rho_{1,1} = \bot$$

  independent from coin. Consequently, for whatever second signing messages $\rho_{2,0}$ and $\rho_{2,1}$ that A outputs, we have $\sigma_0 = \sigma_1 = \bot$. Therefore, its advantage to output coin correctly is 0.
- If the failed check is in Derive : this implies that the Fischlin $\mathsf{S}_{\mathsf{fis}}.\mathsf{Verify}(\mathsf{vk}, m_b, \sigma_b) = 1$ does not pass for some $b \in \{0, 1\}$. Then our Derive algorithm outputs $\sigma_b = \bot$ and therefore $\mathsf{Verify}(\mathsf{bvk}, m_b, \sigma_b) = 0$. Thus the 'if' condition in Definition 3 is satisfied, implying both derived signatures are set

  $$\sigma_0 = \sigma_1 = \bot$$

  and vacuously do not depend on (m_0, m_1). Hence, by combining the aforementioned with the fact that the check in User passes in this current case, an argument similar to our game hops Game 4→ Game 5 → Game 6 in the blindness proof argue that the first signing messages $\rho_{1,0}$ and $\rho_{1,1}$ can be made independent from coin as well. This concludes that in this case $\mathcal{A}$'s advantage to correctly output coin stays negligible in λ (union bound, 0 from the derived signatures and negligible from the first signing messages).

Finally, it remains to argue that whether the check(s) fail(s) or not does not depend on the blindness challenger's coin. This is clear for the check in User that is only on bvk and does not involve coin. With respect to the check in Derive, conditioned that the check in User passes, the first signing messages $\rho_{1,0}$ and $\rho_{1,1}$ of our scheme (will be given to the adversary as defined in the blindness game Definition 3) can be made independent from coin in the same vein of the game hops Game 4→ Game 5 → Game 6 in the blindness proof. Therefore, the event that "the check in Derive fails" in the adversary's view, given $\rho_{1,0}$ and $\rho_{1,1}$, happens independently from coin.

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 $\zeta \neq \zeta'$.

Lemma 9. $|\mathsf{Adv}_{\mathcal{A},Game\ 1}(\lambda) - \mathsf{Adv}_{\mathcal{A},Game\ 2}(\lambda)| \leq Q_{\mathsf{H}}^2/2 \cdot 2^{2\lambda}$

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 $\zeta \neq \zeta'$.

Lemma 10. $|\mathsf{Adv}_{\mathcal{A},Game\ 2}(\lambda) - \mathsf{Adv}_{\mathcal{A},Game\ 3}(\lambda)| \leq \mathrm{negl}(\lambda)$

Proof. This follows via a birthday bound since due to lemma 1, the image of $\mathsf{H}_{\mathbb{P}}$ is of exponential size.

Game 4: In this game, we alter how the parameters for $\mathsf{C_Z}$ are set up. Namely, we use the algorithm $\Pi_{\mathsf{ped}}.\mathsf{Ext}_1$ to set up the parameters for $\mathsf{C_Z}$ 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:

Lemma 11. *There exists a PPT adversary $\mathcal{B}_1$ against the online-extractability of Π_{ped} such that $\mathsf{Adv}_{\mathcal{A},Game\ 5}(\lambda) \geq \frac{\mathsf{Adv}_{\mathcal{A},Game\ 4}(\lambda) - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_H)}$ where we plugged in $\mathsf{Adv}_{\mathcal{A},Game4}(\lambda)$ as $\mu(\lambda)$ from Definition 25 and $\mathrm{negl}, p_{\mathsf{P}}$ are as in Definition 25.*

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 7. We note that as the commitment scheme $\mathsf{C_Z}$ 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 $\mathsf{H}(m)$.

Lemma 12. $|\mathsf{Adv}_{\mathcal{A},Game5} - \mathsf{Adv}_{\mathcal{A},Game6}(\lambda)| \leq \frac{1}{2^{2\lambda}}$.

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 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

$$\mathsf{Adv}_{\mathcal{A},Game7}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game6}(\lambda)$$

Game 8: 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}.

Lemma 13. *There exists a reduction $\mathcal{B}_2$ such that* $|\mathsf{Adv}_{\mathcal{A},Game8} - \mathsf{Adv}_{\mathcal{A},Game7}| \leq \mathsf{Adv}^{\mathrm{crs}}_{\mathcal{B}_2}(\lambda)$

Proof. The reduction $\mathcal{A}_2$ receives a CRS from the CRS indistinguishability challenger.

It samples all other parts of the verification key as in Game 7 and outputs them to the adversary. It answers signing queries as in Game 7. 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 9: In Game 9 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 9 has statistical distance at most $1/2^\lambda$ from the distribution of a in Game 8.

Thus, we get that $||\Pr[\text{Game } 9 = 1] - \Pr[\text{Game } 8 = 1]|| \leq \frac{1}{2^\lambda}$.

Game 10: In Game 10, 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 $e_1, \dots e_{Q_{\mathsf{H}_{\mathbb{P}}}}$ Everything else we do as in Game 9. As the keys are distributed the same, it holds that

$$\mathsf{Adv}_{\mathcal{A},Game10}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game9}(\lambda).$$

Game 11: In Game 11, 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 outputs y', a to the adversary.

As the signatures produced by this game are identically distributed to the ones output by Game 10, we obtain that

$$\mathsf{Adv}_{\mathcal{A},Game11}(\lambda) = \mathsf{Adv}_{\mathcal{A},Game10}(\lambda).$$

Game 12: In Game 12 we switch the setup of the CRS for Π_{fis} to generating it using the simulator SimCRS for extraction.

Lemma 14.

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

Proof. We construct an adversary $\mathcal{B}_3$ against CRS indistinguishability (see Definition 24). The adversary takes as input a CRS for Π_{fis}. It sets up the rest of the verification key vk as Game 12 and simulates all oracles except for $\mathsf{H}_{\mathsf{fis}}$ as Game 12. It simulates $\mathsf{H}_{\mathsf{fis}}$ by forwarding the queries of the adversary to its own hash oracle provided by the CRS indistinguishability challenger. It aborts whenever Game 12 would abort. If the adversary outputs a valid one-more forgery, the reduction $\mathcal{B}_3$ outputs 1, otherwise 0. The claim follows from the CRS indistinguishability according to Definition 24 of Π_{fis}.

Moving Towards Breaking sRSA: We now want to use the knowledge soundness property of the NIZK Π_{fis} to obtain a signature that will be used by the final reduction to break sRSA.

Namely, after the adversary has submitted its signatures, we extract a witness from π_{fis}.

We describe below how this extraction procedure works.

We describe a "wrapper" $\mathcal{B}_4$ around $\mathcal{A}$ to extract from. This is necessary for formal reasons as the adversary $\mathcal{A}$ requires additional inputs and oracles compared to a knowledge soundness adversary, in particular $\mathcal{A}$ expects a verification key, several random oracles, as well as a signing oracle. The wrapper $\mathcal{B}_4$ will be our soundness adversary to extract from and it will provide all additional inputs and oracles to $\mathcal{A}$.

Setup The algorithm $\mathcal{B}_4$ takes as input a (simulated) CRS and has access to the random oracle $\mathsf{H}_{\mathsf{fis}}$. It generates all other parts of the verification key as Game 12, that is, it chooses the bits b, b' and runs the corresponding alternative key generation algorithm, however it replaces the CRS for Π_{fis} with the one from its input. Note that the CRS of Π_{fis} is generated completely independently of the rest of the verification key in the honest setup of KeyGen, as well as in the alternative setup algorithms that use a different secret key to sign, and hence this new verification key vk using the input CRS is identically distributed to a verification key set up by Game 12.

Online Phase. The wrapper simulates the following oracles to the adversary. It aborts whenever Game 12 would abort.

- **Signing** The wrapper answers signing queries as Game 12 using the online-extraction and the alternative signing algorithms corresponding to the bits b, b'.
- **Hash oracle $\mathsf{H}_{\mathsf{fis}}$** It simulates $\mathsf{H}_{\mathsf{fis}}$ by forwarding queries and responses to and from its own oracle $\mathsf{H}_{\mathsf{fis}}$ provided by the extractor.
- **Other Random Oracles** The other random oracles it provides itself and implements them in the same way as Game 12.

Output determination Due to the changes made in Game 6, for each message that the adversary $\mathcal{A}$ 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 wrapper 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 12, and therefore, the wrapper can efficiently identify which of the messages submitted as part of the final message-signature pairs it has not signed. Once the adversary outputs its message-signature pairs, the wrapper identifies the first message-signature pair $(m^*, (c_I^*, \pi_{\mathsf{fis}}^*))$ where it has never signed the message m^* itself. It outputs $(c_I^*, \pi_{\mathsf{fis}}^*)$

Lemma 15. *$\mathcal{B}_4$ is a valid adversary against adaptive knowledge soundness of Π_{fis} and it holds that*

$$\Pr[(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda), (x, \pi) \leftarrow \mathcal{B}_4^{\mathsf{H}_{\mathsf{fis}}}(\overline{\mathsf{crs}}; \rho) : \mathsf{Verify}^{\mathsf{H}_{\mathsf{fis}}}(\overline{\mathsf{crs}}, x, \pi) = 1] \geq \varepsilon_{\mathcal{B}_4},$$

where $\varepsilon_{\mathcal{B}_4} = \mathsf{Adv}_{\mathcal{A},Game12}(\lambda)$

Proof. As the wrapper perfectly simulates Game 12 to the adversary $\mathcal{A}$, with probability $\mathsf{Adv}_{\mathcal{A},Game12}(\lambda)$, the adversary outputs a one-more forgery. Therefore, with the same probability, the wrapper outputs a pair $(c_I^*, \pi_{\mathsf{fis}}^*)$ which is a statement-witness pair with a valid proof π_{fis}^*. The claim follows.

The extractor $\Pi_{\mathsf{fis}}.\mathsf{Ext}$ is now run on the wrapper $\mathcal{B}_4$ wrapping $\mathcal{A}$.

Lemma 16. *It holds that*

$$\Pr\left[\begin{array}{l}(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda), \\ (c_I^*, \pi_{\mathsf{fis}}^*) \leftarrow \mathcal{B}_4^{\mathsf{H}_{\mathsf{fis}}}(\overline{\mathsf{crs}}; \rho), \\ w \leftarrow \mathsf{Ext}(\overline{\mathsf{crs}}, \mathsf{td}, c_I^*, \pi_{\mathsf{fis}}^* \rho, \vec{h})\end{array} : (c_I^*, w) \in \widetilde{\mathsf{R}}_{\mathsf{fis}}\right] \geq \frac{\varepsilon_{\mathcal{B}_4}^c - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_{\mathsf{H}_{\mathsf{fis}}})}$$

Proof. This follows immediately from lemma 15 along with the adaptive knowledge soundness (see Definition 24) of Π_{fis}.

We now discuss how the witness is affected by the wrapper:

Information-theoretically, c_I^* contains a unique opening. We define the distribution

$$D_{Game12} \coloneqq \{w | c_I' \text{ is a commitment to } w\}$$

where $(m', c_I', \pi_{\mathsf{fis}}')$ is the first tuple in the output of $\mathcal{A}$ Game 12 such that Game 12 has not signed m' in a signing session.

Furthermore, we define the distribution

$$D_{\text{ext}} \coloneqq \left\{ w \middle| \begin{array}{l}(\overline{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{SimCRS}(1^\lambda), \\ (c_I^*, \pi_{\mathsf{fis}}^*) \leftarrow \mathcal{B}_4^{\mathsf{H}_{\mathsf{fis}}}(\overline{\mathsf{crs}}; \rho), \\ w \leftarrow \mathsf{Ext}(\overline{\mathsf{crs}}, \mathsf{td}, c_I^*, \pi_{\mathsf{fis}}^* \rho, \vec{h})\end{array} \right\}.$$

Lemma 17. $D_{Game12} = D_{ext}$

Proof. The witness extracted by $\Pi_{\mathsf{fis}}.\mathsf{Ext}$ is information-theoretically fixed already after one run of the wrapper and adversary because c_I is perfectly binding. As the wrapper simulates Game 12 perfectly to the adversary, the witness contained in the commitment c_I^* is therefore the unique witness the adversary would have committed to in Game 12. As there exist no other openings for the commitment, i.e., no other witnesses, this is the unique witness that the extractor can extract from $\mathcal{B}_4$ wrapping around $\mathcal{A}$.

Remark 8. We note that a similar argument would not go through if c_I was only a computationally binding commitment. In this case, there would be no unique witness that the extractor is guaranteed to extract – any witness would mean success of the extractor, and thus the witness extracted might turn out to be dependent on the behavior of the wrapper in the same way as the witness extracted in [6, 62] can depend on the behavior of the wrapper there.

Reduction simulating Game 12 We describe a reduction $\mathcal{B}_5$ that simulates Game 12 and 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 12.

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 outputs z, a.

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. 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 4.

It is easy to see that $\mathcal{B}_5$ simulates Game 12 perfectly.

We compute the probability that the reduction can use the solution output by the adversary to solve its sRSA challenge:

Lemma 18. *The probability that none of the following properties hold for the combined run of the reduction, extractor, wrapper, and adversary is at least* $\frac{1}{4Q_{\mathsf{H}_{\mathbb{P}}}} \cdot \frac{\varepsilon^{\mathsf{c}}_{\mathcal{B}_5} - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_{\mathsf{H}_{\mathsf{fis}}})}$*:*

- *extraction fails*
- $b = 0$ *and* $e^* \in \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$
- $b = 1$ *and* $e^* \notin \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$
- $b = 1$ *and* $e^* \neq e_j$ *where* j *is the index* j *from* $\mathsf{sk}_{b,b'}$.
- $b = 1$*, the previous condition doesn't apply,* j *as above. Denote by* a_j *and* $\overline{m}_j$ *the values used in the first signing session where* e_j *was used as an exponent. If no such session exists, sample* $a_j \leftarrow \mathcal{S}_a$.
  - $b' = 0$ *and* $a^* = a_j$
  - $b' = 1$ *and* $a^* + \overline{m}^* = a_j + \overline{m}_j$

Proof. We calculate the probability of the opposite event.

By lemma 16, extraction works with probability at least $\frac{\varepsilon^{\mathsf{c}}_{\mathcal{B}_5} - \mathrm{negl}(\lambda)}{p_{\mathsf{P}}(\lambda, Q_{\mathsf{H}_{\mathsf{fis}}})}$.

The witness committed to in Game 12 is independent of the choices of b, b', j. This follows as in Theorem 6.

Therefore, by lemma 17, the witness extracted is also independent. Thus the probability of $b = 1$ and $e^* \in \{e_1, \ldots, e_{Q_{\mathsf{H}_{\mathbb{P}}}}\}$ is at least $\frac{1}{2}$. The probability of $e^* = e_j$ and $b = 0$ is at least $\frac{1}{2Q_{\mathsf{H}_{\mathbb{P}}}}$, multiplied by the probability $\frac{1}{2}$ that either $b' = 1$ and $a^* = a_j$ or $b' = 0$ and $a^* + \overline{m}^* = a_j + \overline{m}_j$.

We briefly describe why solving sRSA works as in Section 4. 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 4 can be applied.

E Instantiations of NIZKs

In this section, we instantiate the remaining NIZKs required for $\mathsf{BS}_{\mathsf{fis}}$: Π_{ped} and Π_{fis}. Let us give first give a brief overview of the constructions. Both constructions follow roughly the Fiat-Shamir template: (1) Construct a Σ-protocol for the desired relation and (2) apply the Fiat-Shamir transformation to obtain a NIZK.

As before, we use the following notational conventions. In Σ-protocols, we denote by Ω the ZK commitments sent in the first flow. To improve readability, specific Ω are indexed by the relation to be shown. We denote by γ the challenge. The responses (generally consisting of a linear combination of γ, a witness w and a mask denoted by μ_w) are denoted by τ_w.

Design principles of our Σ-protocols. Throughout, we will utilize two groups types of groups: $\mathbb{Z}_N^*$ and prime-order $\mathbb{G}$. The latter is used for commitments (cf. Section 5). The technical difficulty of both instantiations is due to the need to argue over both groups (of distinct order) *consistently*. Also, we cannot reduce to computational assumptions with respect to N because it is part of the statement. Roughly, our approach is as follows:

- For relations over either $\mathbb{Z}_N^*$ or $\mathbb{G}$ alone, we employ Schnorr-style Σ-protocol techniques. If a witness w appears within relations over both $\mathbb{Z}_N^*$ and $\mathbb{G}$, the verifier checks both equations using the same response τ_w for both relations. Later, this ensures that the extracted witness satisfies the relations in both groups consistently. Because the order of $\mathbb{Z}_N^*$ and $\mathbb{G}$ is different [29], for instance, we cannot reuse a response in $\mathbb{Z}_p$ within an equation over $\mathbb{Z}_N^*$—this would lead to inconsistencies and already correctness fails.
  Instead, to compute the Σ-protocol responses, the prover masks the witness over the integers with either rejection sampling or noise flooding. Roughly, this ensures correctness when reusing the same response over both $\mathbb{Z}_N^*$ and $\mathbb{G}$.
- For relations between witnesses over the integers, we use a technique often used in Lattice-based NIZKs (*e.g.*, [15]). Roughly, we embed the desired relation in the leading coefficient f_d of a polynomial f of degree d, where γ is interpreted as variable. Here, f is defined over the integers to avoid a dependence on the order of the groups. We design f such that the desired relation holds iff $f_d = 0$ over the integers. Then, the prover proves that $f_d = 0$ over the integers by employing MPed commitments over $\mathsf{QR}_{\tilde{N}}$, where $\tilde{N}$ is a fresh RSA modulus [30]. For this, the prover commits to each coefficient *except* the leading coefficient f_d via MPed. The evaluation $f_d(\gamma)$ is computable by the verifier by design, and the verifier checks that the prover committed to $f_d(\gamma)$ by linearity of MPed. A Schwarz-Zippel argument then allows to argue that $f_d = 0$ over the integers [31] Note that MPed over $\mathsf{QR}_{\tilde{N}}$ is binding over the integers, so $f_d = 0 \in \mathbb{Z}$ independent of the order of the group(s).
- To show (relaxed) ranges within the Σ-protocol, we employ the techniques in Section 5 and Appendix C.1. That is, we commit to the witness via a relaxed integer commitment and open it in zero-knowledge by verifying range memberships of the response.
- To argue special soundness, we need to ensure that we can extract witnesses in $\mathbb{Z}$. Then, all relations are well-defined since exponentiation in $\mathbb{Z}_N^*$ and $\mathbb{G}$ is well-defined for integer exponents. For this, we proceed as in Appendix C.1. That is, the prover commits to all witnesses over $\mathbb{Z}$ in an MPed commitment over $\mathsf{QR}_{\tilde{N}}$. Due to lemma 6, we can argue that extracted values must be integers. Then, to argue that specific relations hold, we follow standard proof techniques.
  Formally, we show that the extractor finds a witness for the desired relations, or it finds a witness for a hard relation (*e.g.*, for a relaxed DLOG relation over $\mathsf{QR}_{\tilde{N}}$). Later, we argue that the latter does not occur under computational assumptions within the NIZKs.
- Another technicality is that for non-abort HVZK, we need that the witnesses lie within a specified range for masking to hide the witness statistically. Since some range checks are introduced in a modular manner later, the soundness extractor does not guarantee this at this

[29] Furthermore, the order of $\mathbb{Z}_N^*$ is unknown to both verifier and prover within this section.

[30] We stress that we cannot reuse the modulus N for the MPed commitment because soundness relies on binding of MPed (which holds under sRSA). As N is part of the statement, we cannot reduce to computational assumptions related to N within our proof. On the other hand, we later embed $\tilde{N}$ into the NIZK's crs, more precisely srs. In that case, we can reduce extraction failure to assumptions over $\tilde{N}$. Let us also remark that for subversion zero-knowledge, we also *cannot* use assumptions related to $\tilde{N}$ either since $\tilde{N}$ is non-uniform.

[31] Roughly, the check ensures that $f_d(x)$ and polynomial $f'(x) \coloneqq \sum_{i\in[0,d-1]} f_i x^i$ committed to in MPed evaluate to the same values for $x = \gamma$. If $f_d(\gamma) - f'(\gamma) = 0$ for $d+1$ different values γ, then it must hold that $f_d(x) = f'(x)$. Since the degree of f' is $d-1$, this means that $f_d = 0$. We provide more details when introducing our specific Σ-protocols below.

point [32]. These technicalities are why there are more than one relation per Σ-protocol: one for correctness and HVZK, and one for special soundness.

NIZKs. To construct both NIZKs, we construct appropriate Σ-protocols. Let us now give more intuition on how they are employed within our NIZKs.

- Π_{ped}: The first NIZK Π_{ped} is an online-extractable NIZK. We achieve online-extraction as follows. We commit to each witness in an ElGamal commitment. A trapdoor for extraction is a decryption key for ElGamal. Because we encrypt in the exponent, we need the messages to be small (*i.e.*, within $[0, B]$ for $B = \mathsf{poly}$—this also ensures that they are well-defined as integers). The latter is proven via Bulletproofs [22]. An appropriate Σ-protocol compiled with Fiat-Shamir proves that the desired relation holds for the witnesses that are encrypted via ElGamal. (Because the witnesses are in general larger than B, we decompose them in B-ary representation first.) The extractor simply decrypts the ElGamal commitments. Due to the Bulletproof, decryption is guaranteed to work and due the Σ-protocol, the decrypted values satisfy the desired relation. To show that online extraction succeeds with high probability, we rely on the techniques in [66].
- Π_{fis}: This NIZK is obtained by applying the Fiat-Shamir transformation to an appropriate Σ-protocol.

More details are given below.

E.1 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.*, [66]). 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 [22, 9]) 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 25).

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 $\ell_m = \lfloor \frac{\lambda}{\log B} - 1 \rfloor$ and $\ell_r = \lfloor \log_B(S) - 1 \rfloor$. The scheme C_{Z} is defined below.

- $\mathsf{C}_{\mathsf{Z}}.\mathsf{Setup}(1^\lambda)$: Samples $\overline{G}, \overline{H} \leftarrow \overline{\mathbb{G}}_p$ and outputs $\mathsf{pp} \leftarrow \overline{G}, \overline{H}$.
- $\mathsf{C}_{\mathsf{Z}}.\mathsf{Commit}(\mathsf{pp}, (\overline{m}, r))$: 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 $d_Z = (s_1, \ldots, s_{\ell_m + \ell_r})$.
- $\mathsf{C}_{\mathsf{Z}}.\mathsf{Verify}(\mathsf{pp}, c_Z, (\overline{m}, r), d_Z)$: 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 19. *The integer commitment scheme with bounded range C_{Z} with message space $\mathsf{C}_{\mathsf{Z}}.\mathcal{C}_{\mathsf{msg}} = [0, 2^\lambda - 1] \times [0, S]$ is correct, hiding under DDH in $\overline{\mathbb{G}}_p$ and perfectly binding (cf. Definition 6).*

[32] Also, sometimes the ranges required for HVZK and ranges guaranteed by soundness are different due to optimizations akin to Section 5.

Proof. For correctness, given c_Z, d_Z from $\mathsf{C_Z.Commit}(\mathsf{pp}, (\overline{m}, r))$ and the opening $(\overline{m}, r)$, we parse $c_Z = (E_i, S_i)_{i=1}^{\ell_m+\ell_r}$ and $d_Z = (s_i)_{i=1}^{\ell_m+\ell_r}$. The values s_i are the same that are used in $\mathsf{C_Z.Commit}$. Since $B = \mathsf{poly}(\lambda)$ is a power of two, or more generally since the super-increasing $(1, B, B^2, \ldots)$ satisfies $B^i > \sum_{j=0}^{i-1} B^j$, the B-ary decomposition of $\overline{m}$ and r is unique. Hence, in $\mathsf{C_Z.Verify}(\mathsf{pp}, c_Z, (\overline{m}, r), d_Z)$, decomposing $\overline{m} = \sum_{i=1}^{\ell_m} m_i B^{i-1}$ and $r = \sum_{i=1}^{\ell_r} r_i B^{i-1}$ gives the same vector $\vec{e} = (m_1, \ldots, m_{\ell_m}, r_1, \ldots, r_{\ell_r}) \in [0, B-1]^{\ell_m+\ell_r}$ that is used in $\mathsf{C_Z.Commit}$, satisfying the range checks of e_i for all $i \in [\ell_m + \ell_r]$. Finally, the fact that $\overline{G}, \overline{H} \leftarrow \overline{\mathbb{G}}_p$ being cyclic ensures that the equality $E_i = e_i\overline{G} + s_i\overline{H}$ holds, by using the same (e_i, s_i) that are used in $\mathsf{C_Z.Commit}(\mathsf{pp}, (\overline{m}, r))$ to set E_i.

For hiding, observe that a commitment c_Z is comprised of $\ell_m + \ell_r = \mathsf{poly}(\lambda)$ ElGamal commitments (E_i, S_i). Let us argue with a hybrid argument. Let $\mathcal{A}$ be an adversary on hiding. In the i-th hybrid, replace (E_i, S_i) with $(E_i', S_i') \leftarrow \overline{\mathbb{G}}_p$. Since $(\overline{H}, S_i, s_i\overline{H})$ form a DDH-tuple, it is straightforward to construct an adversary on DDH that distinguishes two consecutive hybrids with advantage $\varepsilon = \mathsf{Adv}_{\mathcal{A}}^{\mathrm{hide}}(\lambda)$. Since $(\ell_m + \ell_r)\varepsilon = \mathrm{negl}(\lambda)$ under DDH, the claim follows.

Let us show binding. Let $c_Z = (E_i, S_i)_{i\in[\ell_m+\ell_r]} \in \mathbb{G}^{(2\cdot\ell_m+\ell_r)}$. Observe that (E_i, S_i) fixes $e_i \bmod p$ perfectly [33]. If $\exists i : e_i \notin [0, B]$, then the commitment cannot be opened because the interval membership check in verification fails. Else, since $\mathsf{poly}(\lambda) = B < 2^\lambda \leq p$, any valid opening of (E_i, G_i) fixes unique e_i over the integers as described above. These values determine the message $(\overline{m}, r)$ uniquely within $[0, 2^\lambda - 1] \times [0, S]$ through B-ary decomposition. In conclusion, c_Z can only be opened to $(\overline{m}, r)$ fixed as described above.

Online-Extractable NIZK. We are now ready to instantiate Π_{ped}. Let $B = \mathsf{poly}(\lambda)$ and $\ell_m := \lfloor \frac{\lambda}{\log B} - 1 \rfloor, \ell_r := \lfloor \log_B(S) - 1 \rfloor$. 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 $w = (\overline{m}, r, (s_i)_{i\in[\ell_m+\ell_r]})$. Above, $\overline{m} \in [0, 2^\lambda - 1], r \in [0, S]$ and $s_i \in \mathbb{Z}_p$. Note and that the values e_i are unique given $\overline{m}$ and r via the B-ary decomposition. Moreover, by the choices of (ℓ_m, ℓ_r) it holds that

$$0 \leq \overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1} \leq \sum_{i=1}^{\ell_m} (B-1) B^{i-1} \leq \sum_{i=1}^{\ell_m} B^i < \sum_{i=0}^{\ell_m} B^i < B^{\ell_m+1} \overset{(*)}{\leq} 2^\lambda$$

where $(*)$ follows from the fact that $\ell_m = \lfloor \frac{\lambda}{\log B} - 1 \rfloor$ and $\lfloor x \rfloor \leq x$ for all $x \in \mathbb{R}$. A similar calculation on r can be obtained

$$0 \leq r < B^{\ell_r+1} \overset{(**)}{\leq} S$$

and $(**)$ follows from the fact that $\ell_r = \lfloor \log_B(S) - 1 \rfloor$.

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 [9]. 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 [66], Section 6. This approach was shown to be secure in [66].

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 6). We can also use MPed commitments to show the statements over the integers (leveraging the binding property of MPed). To ensure subversion zero-knowledge,

[33] An inefficient algorithm can recompute a unique $e_i \in \mathbb{Z}_p$ that satisfies $E_i = e_i\overline{G} + s_i\overline{H}$, where s_i is the DLOG of S_i.

we add a Π_{gen} proof (cf. Appendix C.1) 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 $\tilde{N} \in \mathbb{N}$ and let $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \ldots, \tilde{g}_{\ell_m+\ell_r}) \in (\mathbb{Z}^*_{\tilde{N}})^{1+\ell_m+\ell_r}$ denote the public parameters of an MPed commitment with message space $\mathbb{Z}^{\ell_m+\ell_r}$. (Since membership in $\mathsf{QR}_{\tilde{N}}$ cannot be efficiently tested without factorization of $\tilde{N}$, the MPed commitment is formally defined over $\mathbb{Z}^*_{\tilde{N}}$. Later, $\tilde{N}$ and $\tilde{\mathsf{pp}}$ are given in the crs.) Denote by R the relation

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

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 $w = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$. Here, $\overline{m}, r, e_i \in \mathbb{Z}$ and $s_i \in \mathbb{Z}_p$. 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 Fig. 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 [34]. For the soundness relation, we 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—in particular these relations are not ensured to hold by Σ-protocol yet). We remark that to show the decompositions, we use a standard technique that is often used in lattices to show multiplicative relations (*e.g.*, [15]). In particular, we 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 7 and $\mathsf{R}_{\mathsf{dlog}}$ denotes the relation that contains all non-trivial DLOG relations in $\tilde{\mathsf{pp}}$ (see [9] 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 $\mathsf{QR}_{\tilde{N}}$.

Our construction follows the design principles discussed at the start of this section. Let us give a brief overview. To show the statements over $\mathbb{Z}^*_N$ and $\overline{\mathbb{G}}_p$, we use use Schnorr-type Σ-protocol techniques *except* that the witnesses are masked over the integers via noise flooding [35]. This ensures that we can reuse the responses over groups of distinct order. For the relations related to the B-ary decomposition of $\overline{m}$ and r, we define two polynomials f_m and f_r of degree 1 where the leading coefficient is $f_{m,1} = \overline{m} - \sum_{i=1}^{\ell_m} e_i B^{i-1}$ and $f_{r,1} = r - \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}$, respectively. The polynomials are designed such that $f_m(\gamma)$ and $f_r(\gamma)$ can be recomputed given the Σ-protocol responses, where γ is the challenge. Notably, if f_m and f_r are constant, then the desired relations hold. 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 $f_{m,0}(\gamma) = f_m(\gamma)$ and $f_{r,0}(\gamma) = f_r(\gamma)$ using the commitment's linearity. Roughly, special soundness is argued as follows: If $f_{m,0}(\gamma) - f_m(\gamma) = 0$ for two distinct values γ, then because $f_{m,0} - f_m$ is of degree 1 but has two zeroes, it must hold that $f_{m,0} - f_m = 0$. But then, $f_{m,1} = 0$ because $f_{m,0}$ is constant. Thus, the relation holds as discussed above.

The above discussion is formalized below.

Lemma 20. *The Σ-protocol Σ_{ped} is correct, HVZK, has high min-entropy for relation $\mathsf{R}_{\Sigma_{\mathsf{ped}}}$, and is 2-special sound for relation $\widetilde{\mathsf{R}}_{\Sigma_{\mathsf{ped}}}$.*

[34] As discussed in the beginning of this section, we commit to e_i via an MPed commitment over $\tilde{N}$ to ensure that $e_i \in \mathbb{Z}$ and 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, we know that in an honest execution, we have that $e_i \in [0, B-1]$. We add this condition in the correctness and HVZK relation in order to mask over the integers with small masks. (This is required to compute the Σ-protocol responses.)

[35] We could use rejection sampling and compile Σ_{ped} with Fiat-Shamir with aborts for better efficiency. We choose noise flooding for simplicity.

Proof. For correctness, the first **check** on line 16 follows arithmetics modulo N where $c^{-\gamma} \equiv h_2^{-\gamma\overline{m}} \cdot g^{-\gamma re} \bmod N$ together with

$$h_2^{\tau_{\overline{m}}} \cdot g^{\tau_r e} \equiv h_2^{\gamma\overline{m}+\mu_{\overline{m}}} \cdot g^{\gamma re+\mu_r e} \bmod N$$

and $\Omega_c \leftarrow h_2^{\mu_{\overline{m}}} \cdot g^{\mu_r e} \bmod N$ implies $c^{-\gamma} h_2^{\tau_{\overline{m}}} \cdot g^{\tau_r e} \equiv \Omega_c \bmod N$. In a similar manner, the checks on lines 23 and 22 follow from modular arithmetics in $\mathbb{Z}_{\tilde{N}}$. More specifically, the check on line 22 passes because

$$\tilde{c}^{-\gamma} \cdot \tilde{h}^{\tau_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{\tau_{e_i}} \equiv \tilde{h}^{-\gamma\tilde{t}+\tau_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{-\gamma e_i+\tau_{e_i}} \overset{(*)}{\equiv} \tilde{h}^{\mu_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{\mu_{e_i}} \bmod \tilde{N}$$

and is congrument to $\Omega_{\tilde{c}}$ modulo $\tilde{N}$ thanks to $\Omega_{\tilde{c}} \leftarrow \tilde{h}^{\mu_{\tilde{t}}} \cdot \prod_{i=1}^{\ell_m+\ell_r} \tilde{g}_i^{\mu_{e_i}} \bmod \tilde{N}$, where $(*)$ follows from $\tau_{\tilde{t}} \leftarrow \gamma\tilde{t} + \mu_{\tilde{t}}$ and $\tau_{e_i} \leftarrow \gamma e_i + \mu_{e_i}$. To examine the check on line 23, we observe the following in $\mathbb{Z}$

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

and in the same manner, also in $\mathbb{Z}$,

$$\begin{aligned} f_r &= (\textstyle\sum_{i=1}^{\ell_r} B^{i-1} \tau_{e_{\ell_m+i}}) - \tau_r \\ &= (\textstyle\sum_{i=1}^{\ell_r} (\gamma e_{\ell_m+i} + \mu_{e_{\ell_m+i}}) B^{i-1}) - (\gamma r + \mu_r) \\ &= \gamma \underbrace{(\textstyle\sum_{i\in[\ell_r]} e_{\ell_m+i} B^{i-1} - r)}_{=0} + (\textstyle\sum_{i=1}^{\ell_r} \mu_{e_{\ell_m+i}} B^{i-1}) - \mu_r = f_{r,0} \ . \end{aligned}$$

This gives

$$\tilde{h}^{t_q} \cdot \tilde{g}_1^{f_m} \cdot \tilde{g}_2^{f_r} \equiv \tilde{h}^{t_q} \cdot \tilde{g}_1^{f_{m,0}} \cdot \tilde{g}_2^{f_{r,0}} \equiv \Omega_q \bmod \tilde{N} \ .$$

Finally, the checks on lines 18 and 19 are done in $\overline{\mathbb{G}}_p$, even though $\gamma s_i \in \mathbb{Z}$ while $\mu_{s_i} \in \mathbb{Z}_p$, the computation is in $\overline{\mathbb{G}}_p$ of order p and thus $\tau_{s_i}\overline{H} = (\gamma s_i + \mu_{s_i})\overline{H} \in \overline{\mathbb{G}}_p$ is well-defined (which would not be the case if μ_{s_i} was a congruent class of a modulus other than p). The calculations can be verified with ease

$$-\gamma E_i + \tau_{e_i}\overline{G} + \tau_{s_i}\overline{H} = -\gamma(e_i\overline{G} + s_i\overline{H}) + (\gamma e_i + \mu_{e_i})\overline{G} + (\gamma s_i + \mu_{s_i})\overline{H} = \mu_{e_i}\overline{G} + \mu_{s_i}\overline{H} = \Omega_{E_i} \in \overline{\mathbb{G}}_p$$

and $-\gamma S_i + \tau_{s_i}\overline{G} = -\gamma s_i\overline{G} + (\gamma s_i + \mu_{s_i})\overline{G} = \mu_{s_i}\overline{G} = \Omega_{S_i} \in \overline{\mathbb{G}}_p$.

For showing *high min-entropy*, we consider the first flow

$$\Omega \coloneqq (\tilde{c}, \Omega_{\tilde{c}}, \Omega_q, (\Omega_{E_i})_i, (\Omega_{S_i})_i, \Omega_c) \ .$$

The cyclic group $\overline{\mathbb{G}}_p$ has order $p \geq 2^\lambda$ and because $\mu_{s_i} \leftarrow \mathbb{Z}_p$ uniformly at random modulo p, the terms $\mu_{s_i}\overline{H}, \mu_{s_i}\overline{G}$ are uniformly distributed in $\overline{\mathbb{G}}_p$ and implying $\Omega_{E_i}, \Omega_{S_i}$ are uniformly distributed in $\overline{\mathbb{G}}_p$. The probability that an adversary correctly guesses Ω is thus at most $2^{-\lambda}$, concluding the high min-entropy property.

For *HVZK*, we observe that following the properties of noise flooding (see Supplementary materials A.2), the masking by

$$\begin{gathered} \mu_{\tilde{t}} \leftarrow [0, C\tilde{N} \cdot 2^{2\lambda}], \quad \mu_{\overline{m}} \leftarrow [0, C2^{3\lambda}], \quad \mu_r \leftarrow [0, CS \cdot 2^\lambda] \\ \mu_{e_i} \leftarrow [0, CB \cdot 2^\lambda], \quad \mu_{s_i} \leftarrow \mathbb{Z}_p \end{gathered}$$

ensures that the values

$$\begin{gathered} \tau_{\overline{m}} \leftarrow \gamma\overline{m} + \mu_{\overline{m}}, \quad \tau_r \leftarrow \gamma r + \mu_r, \quad \tau_{\tilde{t}} \leftarrow \gamma\tilde{t} + \mu_{\tilde{t}} \\ \tau_{e_i} \leftarrow \gamma e_i + \mu_{e_i}, \quad \tau_{s_i} \leftarrow \gamma s_i + \mu_{s_i} \end{gathered}$$

sent in the third flow are statistically close to uniform over their respective ranges $[0, CB \cdot 2^{\lambda}]^{\ell_m} \times \mathbb{Z}_p^{\ell_r} \times [0, C \cdot 2^{3\lambda}] \times [0, CS \cdot 2^{\lambda}] \times [0, C\tilde{N} \cdot 2^{2\lambda}] \times [0, \tilde{N} \cdot 2^{\lambda}]$. Moreover, the commitment $\tilde{c}$ is distributed close to uniform over $\langle \tilde{h} \rangle$ (with statistical distance at most $2^{-\lambda}$) because $\tilde{t}$ is drawn uniform over $[0, \tilde{N} \cdot 2^{\lambda}]$, and because $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$. The remaining values $\Omega_{\tilde{c}}, \Omega_q, (\Omega_{E_i})_i, (\Omega_{S_i})_i, \Omega_c$ are determined from the third flow and the challenge γ, by setting with respect to the verification equations

$$
\begin{aligned}
&\Omega_c = c^{-\gamma} h_2^{\tau_{\overline{m}}} \cdot g^{\tau_r e} \bmod N \\
&\forall\, i \in [\ell_m + \ell_r] : \Omega_{E_i} = -\gamma E_i + \tau_{e_i} \overline{G} + \tau_{s_i} \overline{H}; \quad \Omega_{S_i} = -\gamma S_i + \tau_{s_i} \overline{G} \\
&\Omega_{\tilde{c}} = \tilde{c}^{-\gamma} \cdot \tilde{h}^{\tau_{\tilde{t}}} \cdot \textstyle\prod_{i=1}^{\ell_m + \ell_r} \tilde{g}_i^{\tau_{e_i}} \bmod \tilde{N} \\
&\Omega_q = \tilde{h}^{t_q} \cdot \tilde{g}_1^{f_m} \cdot \tilde{g}_2^{f_r} \bmod \tilde{N} \ .
\end{aligned}
$$

Given the above discussion, the simulator receives the challenge γ and samples the responses (for the third flow) according to the above distributions. It also samples $\tilde{c}$ (with negligible distance close to) uniform over $\langle \tilde{h} \rangle$. Finally, it recomputes the remaining first flow that is determined as above and outputs the full transcript. Since the distribution is statistically close to the real protocol, the simulator suffices.

Let us finally show 2 special soundness. We want to construct a deterministic PT extractor so that, for

$$
x = (\tilde{N}, \tilde{\mathsf{pp}}, B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m + \ell_r}) \ ,
$$

given 2 valid transcripts that are indexed by $i \in [2]$ with

$$
\begin{aligned}
\text{Identical first flow:} \quad & \Omega = (\tilde{c}, \Omega_{\tilde{c}}, \Omega_q, (\Omega_{E_i})_i, (\Omega_{S_i})_i, \Omega_c) \\
\text{Pairwise distinct challenge:} \quad & \gamma_i \\
\text{Third flow:} \quad & \vec{\tau}_i = \big((\tau_{e_{k,i}})_k, (\tau_{s_{k,i}})_k, \tau_{\overline{m}_i}, \tau_{r_i}, \tau_{\tilde{t}_i}, t_{q,i}\big)
\end{aligned}
$$

the extractor extracts the witness $w = ((e_k, s_k)_{k=1}^{\ell_m + \ell_r}, \overline{m}, r)$ such that $(x, w) \in \widetilde{\mathsf{R}}_{\Sigma_{\mathsf{ped}}}$. In the following, we denote by $\Delta v_{i,j} \coloneqq \tau_{v,i} - \tau_{v,j} \in \mathbb{Z}$ for $v \in \{(e_k, s_k)_{k=1}^{\ell_m + \ell_r}, \overline{m}, r\}$ and $\Delta\gamma_{i,j} \coloneqq \gamma_i - \gamma_j \neq 0$.

By verification on line 22, we have that for $i \in [2]$, it holds that

$$
\tilde{c}^{-\gamma_i} \cdot \tilde{h}^{\tau_{\tilde{t},i}} \cdot \textstyle\prod_{k=1}^{\ell_m + \ell_r} \tilde{g}_k^{\tau_{e_k,i}} \equiv \Omega_{\tilde{c}} \bmod \tilde{N}
$$

Thus, it holds for $i \neq j \in [2]$ that

$$
\tilde{c}^{\Delta\gamma_{i,j}} \equiv \tilde{h}^{\Delta\tilde{t}_{i,j}} \cdot \textstyle\prod_{k=1}^{\ell_m + \ell_r} \tilde{g}_k^{\Delta e_{k\,i,j}} \bmod \tilde{N}
$$

Thus, either $\Delta\gamma_{i,j}$ divides $\Delta v_{i,j}$ for $v \in \{(e_i, s_i)_{i=1}^{\ell_m + \ell_r}, \overline{m}, r\}$ over $\mathbb{Z}$, or the above equation yields a witness w for $(\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$. In the latter case, the extractor outputs the witness w.

Further, the verification checks ensure that

$$
\begin{aligned}
&c^{-\gamma_1} h_2^{\tau_{\overline{m},1}} \cdot g^{\tau_{r,1} e} && = c^{-\gamma_2} h_2^{\tau_{\overline{m},2}} \cdot g^{\tau_{r,2} e} && (11) \\
&\forall\, k \in [\ell_m + \ell_r] : -\gamma_1 E_k + \tau_{e_k,1} \overline{G} + \tau_{s_k,1} \overline{H} && = -\gamma_2 E_k + \tau_{e_k,2} \overline{G} + \tau_{s_k,2} \overline{H} && (12) \\
&\forall\, k \in [\ell_m + \ell_r] : -\gamma_1 S_k + \tau_{s_k,1} \overline{G} && = -\gamma_2 S_k + \tau_{s_k,2} \overline{G} && (13)
\end{aligned}
$$

We set $e_k = \Delta e_{k1,2}/\Delta\gamma_{1,2}, s_k = \Delta s_{k1,2}/\Delta\gamma_{1,2} \bmod p, \overline{m} = \Delta\overline{m}_{1,2}/\Delta\gamma_{1,2}$, and $r = \Delta r_{1,2}/\Delta r_{1,2}$. From (11), it holds that

$$
c \equiv h_2^{\Delta\overline{m}_{1,2}/\Delta\gamma_{1,2}} \cdot g^{(\Delta r_{1,2}/\Delta\gamma_{1,2}) \cdot e} \bmod N
$$

where we note that the division $\Delta\overline{m}_{1,2}/\Delta\gamma_{1,2}$ and $\Delta r_{1,2}/\Delta\gamma_{1,2}$ are well defined over integer values $\Delta\overline{m}_{1,2}, \Delta r_{1,2}, \Delta\gamma_{1,2} \in \mathbb{Z}$, with respect to the above argument regarding verification in line 22. Consequently, setting $\overline{m} = \Delta\overline{m}_{1,2}/\Delta\gamma_{1,2}$, and $r = \Delta r_{1,2}/\Delta r_{1,2}$ satisfies $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$. Next, for each $k \in [\ell_m + \ell_r]$, (12) and (13) imply that

$$
E_k = \Delta e_{k1,2}/\Delta\gamma_{1,2}\overline{G} + \Delta s_{k1,2}/\Delta\gamma_{1,2}\overline{H} \quad \text{and} \quad S_k = \Delta s_{k,1,2}/\Delta\gamma_{1,2}\overline{G}
$$

where the inverse $1/\Delta\gamma_{1,2}$ is well defined in $\mathbb{Z}$ following a similar argument regarding verification in line 22, else we find a witness for the relaxed DLOG relation. As a result, setting $e_k = \Delta e_{k\,1,2}/\Delta\gamma_{1,2}$ and $s_k = \Delta s_{k\,1,2}/\Delta\gamma_{1,2}$ satisfies $E_k = e_k\overline{G} + s_k\overline{H}, S_k = s_k\overline{G}$ for all $k \in [\ell_m + \ell_r]$.

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 [34], Section 5.1. By definition, we have

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

Similarly, we obtain $r = \sum_{i=1}^{\ell_r} e_{\ell_m+i}B^{i-1}$. This concludes the proof.

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 $w = ((e_i, s_i)_{i=1}^{\ell_m+\ell_r}, \overline{m}, r)$.

Step 2: the range proof. Next, let Π_{rp} be the NIZK with random oracle H_{rp} from [66] (cf. Section 6.2). Note that Π_{rp} is obtained by compiling Bulletproofs [22, 9] 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 [36]. 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

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

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 [9] for more details). Note that for uniform statement, it is hard to find a witness for $\mathsf{R}_{\mathsf{dlog}}$ under the DLOG assumption. We recall well-known properties of Π_{rp} in lemma 21.

Lemma 21 ([66], 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} for the relation

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

The construction is similar to the NIZK in Section 6.3 [66] except that our Σ-protocol is more involved. We first define the srs space. Recall that Σ_{ped} relies on a fresh modulus $\tilde{N}$ and MPed parameters $\tilde{\mathsf{pp}}$. These are provided in the srs. To ensure subversion zero-knowledge, we follow the approach in Appendix C.1. That is, we add a NIZK to prove that shows that $\tilde{\mathsf{pp}}$ is setup in a hiding manner and set

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

The proof π_{gen} ensures that the preconditions with respect to $\tilde{\mathsf{pp}}$ for the HVZK relation of Σ_{ped} are ensured for malicious srs, and thus ensure 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_γ be a random oracle mapping into $[0, C]$. The random oracle of Π_{ped} is $\mathsf{H}_{\mathsf{ped}} = (\mathsf{H}_{\mathsf{rp}}, \mathsf{H}_\gamma)$. 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. Roughly, the prover decomposes $\overline{m}$ and r via B-ary decomposition, commits to e_i in ElGamal commitments (E_i, S_i), and then proves that $e_i \in [0, B]$ via Π_{rp} and that the committed values satisfy $\widetilde{\mathsf{R}}_{\Sigma_{\mathsf{ped}}}$ via Fiat-Shamir compiled Σ_{ped}.

- $\Pi_{\mathsf{ped}}.\mathsf{GenSRS}(1^\lambda)$: 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 $\mathsf{srs} = (\tilde{N}, \tilde{\mathsf{pp}}, \pi_{\mathsf{gen}})$.
- $\Pi_{\mathsf{ped}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{ped}}}(\mathsf{crs}, x, w)$: Parses $x = (B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$ and $w = (\overline{m}, r, (s_i)_{i\in[\ell_m+\ell_r]})$. Decomposes $\overline{m}$ and r into $(m_i)_i$ and $(r_i)_i$ via B-ary decomposition, respectively. Note that $(E_i, S_i) = (e_i\overline{G} + s_i\overline{H}, s_i\overline{G})$. Then, computes

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

[36] 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.

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

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

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 $\pi = (\pi_0, \pi_1)$.

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

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

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 6 (*i.e.*, the NIZK is correct, subversion zero-knowledge, and partially online-extractable). Correctness and subversion zero-knowledge follow from the discussion above. 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 $\mathsf{srs} \leftarrow \Pi_{\mathsf{ped}}.\mathsf{GenSRS}(1^\lambda)$ 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 [66]. 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. In more detail, let us show that $(x_0, w_0) \in \mathsf{R}_{\mathsf{rp}}$ and $(x_1, w_1) \in \mathsf{R}_{\mathsf{sub}}$. Then, correctness of both Π_{rp} and Σ_{ped} yield correctness of Π_{ped}. First, observe that $e_i \in [0, B-1]$. Also, $E_i = e_i\overline{G} + s_i\overline{H}$ holds since $(x, w) \in \mathsf{R}_{\Sigma_{\mathsf{ped}}}$. Thus, $(x_0, w_0) \in \mathsf{R}_{\mathsf{rp}}$. Similarly, $(x, w) \in \mathsf{R}_{\Sigma_{\mathsf{ped}}}$ yields that $c \equiv h_2^{\overline{m}} \cdot g^{re} \bmod N$, $S_i = s_i\overline{G}$. By construction, it holds that $e_i \in [0, B-1]$ and $\overline{m} = \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1}$. Further, we have that $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$ by definition of $\mathsf{MPed.Setup}$. Thus, $(x_1, w_1) \in \mathsf{R}_{\mathsf{sub}}$ as desired.

Subversion zero-knowledge follows similarly to Theorem 9. Notably, we have $(x_0, w_0) \in \mathsf{R}_{\mathsf{sub}}$ by design—this follows as above and since π_{gen} in the adversarial srs ensures that $\langle \tilde{h} \rangle = \langle \tilde{g}_i \rangle$ under soundness of Π_{gen} with overwhelming probability. The simulator then simply simulates π_1 via HVZK of Σ_{ped} and by programming H_γ accordingly. (The latter is possible due to high min-entropy of Σ_{ped}.) Similarly, π_0 is simulated via the zero-knowledge simulator of Π_{rp}. This is possible since $(x_1, w_1) \in \mathsf{R}_{\mathsf{rp}}$ which again, follows as above.

Online-extraction requires some care, but the proof is similar to the proof of Theorem 20, [66] (taking into account that we embed the trapdoor into the statement $x_0 = (\overline{G}, \overline{H})$ instead of the crs). That is, 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) to $e_i = \mathsf{DLOG}_{\overline{G}}(E_i')$ via a discrete logarithm 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 $E_i = E_i' + \mathsf{td} \cdot S_i = e_i\overline{G} + s_i\overline{H}$.
– 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 $x_\Sigma = (\tilde{N}, \tilde{\mathsf{pp}}, B, \overline{G}, \overline{H}, N, e, h_2, g, c, (E_i, S_i)_{i=1}^{\ell_m+\ell_r})$. We then use the procedure from Theorem 20, [66] to obtain 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 [37].

Under the DLOG assumption, we have $((\overline{G}, \overline{H}, \mathsf{urs}_{\mathsf{rp}}), w_{\mathsf{rp}}) \in \mathsf{R}_{\mathsf{dlog}}$ with at most negligible probability. 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

$$E_i = e_i'\overline{G} + s_i'G' \text{ and } e_i' \in [0, B-1]. \tag{14}$$

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

$$\begin{aligned} c &\equiv h_2^{\overline{m}} \cdot g^{re} \bmod N \\ E_i &= e_i\overline{G} + s_i\overline{H}, S_i = s_i\overline{G} \\ \overline{m} &= \sum_{i=1}^{\ell_m} e_i B^{i-1}, r = \sum_{i=1}^{\ell_r} e_{\ell_m+i} B^{i-1} \end{aligned} \tag{15}$$

Notably, we have $e_i = e_i'$ under DLOG as otherwise, we can compute a non-trivial DLOG relation between $\overline{H}$ and $\overline{G}$. Finally, observe that Eqs. (14) and (15) with $e_i = e_i'$ imply that extraction of Π_{ped} via Ext succeeds, *i.e.*, Fail_i does *not* occur. Thus, $\Pr[\mathsf{Fail}_i] = \mathrm{negl}(\lambda)$. This concludes the proof.

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_γ.

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 $\mathsf{C_Z}$ 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.

[37] Roughly, the procedure extracts a witness for x_{rp} via the knowledge extractor of Π_{rp} and two related transcripts for x_Σ via forking. Notably, the argument is agnostic to the concrete Σ-protocol that is being used. [66] proves that the procedure succeeds in polynomial time with probability close to $\Pr[\mathsf{Fail}_i]$. We refer to [66] for more details.

E.2 Instantiation of Π_{fis}

In this section, we instantiate 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. 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 $L \in \mathbb{N}$ (cf. Section 5). As discussed in Section 6.1, 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 assume without loss of generality that $G_1, G_2, H \neq 0$. Denote by $\vec{G} \coloneqq (G_1, G_2)$. We can rewrite the relation $\mathsf{R}_{\mathsf{fis}}$ as follows

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

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 $\widetilde{\mathsf{R}}_{\mathsf{fis}}$

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

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

Step 0: Preparing a ZK-friendly relation for $\mathsf{R}_{\mathsf{fis}}$. First, we derive a more Σ-protocol friendly relation that implies $\mathsf{R}_{\mathsf{fis}}$. Assume that $y \in \langle h_1 \rangle$ [38]. 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 5.2) over the group $\langle h_1 \rangle$. We recall that $\mathsf{C}_{\mathsf{Grp}}$ is an ElGamal commitment of the form $c_N = y \cdot h_1^s \in \langle h_1 \rangle$, where the integer randomness s is fixed via an Π_{int} commitment over a prime-order group. Then, with $\omega = e \cdot s \in \mathbb{Z}$, we have that

$$y^e \equiv c_N^e \cdot h_1^{-es} \equiv c_N^e \cdot h_1^{-\omega} \bmod N$$

and consequently, we have an equivalence

$$y^e \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \Leftrightarrow c_N^e \cdot h_1^{-\omega} \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \tag{16}$$

in $\mathbb{Z}_N^*$. More specifically, it suffices to show

$$c_N^e \cdot h_1^{-\omega} \equiv h \cdot h_1^a \cdot h_2^{a+\overline{m}} \bmod N \ \wedge \ \omega = e \cdot s \ \wedge \ \mathsf{C}_{\mathsf{Grp}}.\mathsf{Verify}(\mathsf{pp}, c_N, y, s) = 1$$

and we can resort to well-known techniques for quadratic equations over $\mathbb{Z}$. Also, we follow the technique to open c_N in zero-knowledge outlined in Section 5. Recall that $\mathsf{C}_{\mathsf{Grp}}$ is defined over a prime-order group. In order to use the same group $\mathbb{G}$ for the relaxed integer commitment of $\mathsf{C}_{\mathsf{Grp}}$ and the commitment c_I of (e, a) defined above, we split the integer randomness s of $\mathsf{C}_{\mathsf{Grp}}$ into ℓ_s values $s_i \in [0, 2^{3\lambda} - 1]$ via $2^{3\lambda}$-ary decomposition, *i.e.*, $s = \sum_{i=1}^{\ell_s} s_i \cdot 2^{3\lambda(i-1)}$. We recall to Section 5.2 for this splitting of s into ℓ_s values in a vector $\vec{s} = (s_i)_{i \in [\ell_s]}$, while remarking that the domain of s is $[0, N \cdot 2^\lambda]$, since the order of $\langle h_1 \rangle$ is upper bounded by N. Specifically, we set $\ell_s = \lceil \frac{\log(N \cdot 2^\lambda)}{3\lambda} \rceil$ and $\vec{B'} = (2^{3\lambda}, \cdots, 2^{3\lambda}) \in \mathbb{N}^{\ell_s}$. Then, we use a 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 $\vec{s}$ satisfying $s_i \in [0, 2^{3\lambda} - 1]$ for all $i \in [\ell_s]$. Below, we denote by $\vec{G'} \coloneqq (G'_1, \cdots, G'_{\ell_s})$.

We modify the relation in two other ways:

1. Instead of using witness e, we use $\overline{e} = e - \overline{E}$ for some even $\overline{E} \in \mathbb{Z}$ and adapt the relations accordingly. As $\overline{e}$ is shorter, this reduces the proof size. Also, note that $e \equiv 1 \bmod 2$ is equivalent to showing that $\overline{e} = 2e' + 1$ over $\mathbb{Z}$ because $\overline{E}$ is even.

[38] This is guaranteed by the correctness and HVZK relation.

2. Let $\tilde{N} \in \mathbb{N}$. We add public parameters $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{\tilde{\ell}}) \in (\mathbb{Z}^*_{\tilde{N}})^{\tilde{\ell}+1}$ for MPed to the statement, where $\tilde{\ell} = 4 + \ell_s$. In the Σ-protocol, the prover commits to the witnesses over $\mathbb{Z}$ via MPed. As in previous NIZKs, this allows to ensure that the extracted values are integers in the soundness proof.

We are now ready to describe the Σ-protocol-friendly relation. First, let us define the relation R as follows. (R serves as basis for both the correctness and HVZK relation, and the soundness relation.)

$$\mathsf{R} := \Big\{(x, w) \mid c_N^{\overline{e}+\overline{E}} \cdot h_1^{-\omega} \cdot h_1^{-s\cdot\overline{E}} \equiv h \cdot h_1^{a} \cdot h_2^{a+\overline{m}} \bmod N, \tag{17}$$

$$\omega = \overline{e} \cdot s, \overline{e} = 2e' + 1, \tag{18}$$

$$\vec{C} = (\overline{e} \cdot H, a \cdot H) + r\vec{G}, F = rH, \tag{19}$$

$$\vec{C}' = \vec{s}H' + r'\vec{G}', F' = r'H, s = \textstyle\sum_{i\in[\ell_s]} s_i 2^{3\lambda(i-1)}, c_N \equiv y \cdot h_1^{s} \bmod N\Big\}\ , \tag{20}$$

where $x = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}'_I, c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C}', F'), w = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$. Let us remark that Eqs. (17) and (18) correspond to verifying that the signature (e, a, y) is valid with respect to $\mathsf{S}_{\mathsf{fis}}$ verification (except the range membership), where $e = \overline{e} + \overline{E}$. Also, y is committed in a $\mathsf{C}_{\mathsf{Grp}}$ commitment $(c_N, \vec{C}', F')$ and $(\overline{e}, a)$ is committed in a $\mathsf{C}_{\mathsf{RInt}}$ commitment $(\vec{C}, F)$, as specified by Eqs. (19) and (20), respectively, except range checks are omitted. Then, adding the range and subgroup checks, gives the correctness and HVZK relation R_{Σ} as follows:

$$\mathsf{R}_{\Sigma} := \{(x, w) \mid (x, w) \in \mathsf{R}, y \in \langle h_1\rangle, (\overline{e}, a) \in [0, \vec{B}], (s_1, \ldots, s_{\ell_s}) \in [0, \vec{B}'], \langle\tilde{h}\rangle = \langle\tilde{g}_i\rangle\}. \tag{21}$$

Note that R_{Σ} is obtained by applying the modifications discussed above to $\mathsf{R}_{\mathsf{fis}}$. Similarly, applying the above modifications to the soundness relation $\widetilde{\mathsf{R}}_{\mathsf{fis}}$, we obtain relation

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

where $\mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$ is defined in Definition 7 and $\mathsf{R}_{\mathsf{dlog}}$ denotes the relation that contains all non-trivial DLOG relations in $\tilde{\mathsf{pp}}$ (cf. Appendix E.1). Later, we show that a Σ-protocol for these relations allows to construct a NIZK for $\mathsf{R}_{\mathsf{fis}}$ via Fiat-Shamir under sRSA.

Step 1: the Σ-protocol. We now construct a Σ-protocol for relations $\mathsf{R}_{\Sigma}, \widetilde{\mathsf{R}}_{\Sigma}$ defined in the previous paragraph. Set $C = 2^{\lambda} - 1$ which defines the challenge space $[0, C]$. The Σ-protocol $\mathsf{\Sigma}_{\mathsf{fis}}$ is given in Fig. 5. We briefly discuss the construction. (Note that we follow the design guidelines described in the beginning of Appendix E.)

- The Eq. (19) and the (relaxed) range membership for $(\overline{e}, a)$ correspond to opening a $\mathsf{C}_{\mathsf{RInt}}$ commitment in zero-knowledge. We proceed as discussed in Appendix C.1 for these equations (using MPed with parameters $\tilde{\mathsf{pp}}$ to argue over the integers).
- The Eq. (20) and the (relaxed) range membership for $(s_i)_{i\in[\ell_s]}$ correspond to opening a $\mathsf{C}_{\mathsf{Grp}}$ commitment in zero-knowledge. We proceed as discussed in Appendix C.1 for these equations (again, using MPed with parameters $\tilde{\mathsf{pp}}$ to argue over the integers).
- We show Eq. (17) with Schnorr-style Σ-protocol techniques.
- To show Eq. (18), we define degree 2 polynomials, where the leading coefficient is 0 iff both relations hold. 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 $\overline{e} = 2e' + 1$. The prover shows that indeed $f_{\omega,2} = f_{e,2} = 0$ by committing to the non-zero coefficients in a MPed commitment. Then, the verifier recomputes $f_\omega(\gamma)$ and $f_e(\gamma)$ (which is possible by construction) and verifies that $f_\omega(\gamma) = f_{\omega,1}\gamma + f_{\omega,0}$ and $f_e(\gamma) = f_{e,1}\gamma + f_{e,0}$ by linearity of MPed over $\mathbb{Z}^*_{\tilde{N}}$.
  Then, 3 special soundness is argued as follows. If $f_e(\gamma) = f_{\omega,1}\gamma + f_{\omega,0}$ holds for three 3 distinct challenge γ, then the polynomial $f_{\mathsf{diff}} := f_e(x) - f_{e,1}x + f_{e,0}$ has three zeroes. Since the polynomial f_{diff} is of degree 2 with 3 zeroes, it must be the zero polynomial. In particular, the leading coefficient of f_e is zero and thus, $\overline{e} = 2e' + 1$. The equation $\omega = \overline{e} \cdot s$ is argued similarly.

– The integer witnesses are committed in a MPed commitment $\tilde{c}$ (with public parameters $\tilde{\mathsf{pp}}$). This allows to argue that extracted values are integers.

Lemma 22. *The Σ-protocol Σ_{fis} for relation R_Σ is correct with abort probability $1-(1-\frac{1}{L})^{\ell_s+3}$, is non-abort HVZK, has high min-entropy, and is 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. For *correctness*, we examine each of the checks. The range checks on lines 19 and 20 for $(\tau_{\overline{e}}, \tau_a, \vec{\tau}_s, \tau_{e'})$ are saisfied thanks to the checks by prover before sending the responses (on lines 17 and 18). The check on line 21 on the commitment Ω_f is satisfied by

$$
\begin{aligned}
& c_N^{-\tau_{\overline{e}}} \cdot h_1^{\tau_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\tau_{s_i}2^{3\lambda(i-1)}} \cdot h^{\gamma} \cdot h_1^{\tau_a} \cdot h_2^{\tau_a+\gamma\overline{m}} \\
& \equiv c_N^{-\gamma\overline{e}-\mu_{\overline{e}}} \cdot h_1^{\gamma\omega+\mu_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}(\gamma s_i+\mu_{s_i})2^{3\lambda(i-1)}} \cdot h^{\gamma} \cdot h_1^{\gamma a+\mu_a} \cdot h_2^{\gamma a+\mu_a+\gamma\overline{m}} \bmod N \\
& \equiv c_N^{-\mu_{\overline{e}}} \cdot h_1^{\mu_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\mu_{s_i}2^{3\lambda(i-1)}} \cdot h_1^{\mu_a} \cdot h_2^{\mu_a} \cdot \left(c_N^{-\gamma\overline{e}} \cdot h_1^{\gamma\omega} \cdot h_1^{\gamma\overline{E}\sum_{i\in[\ell_s]}s_i2^{3\lambda(i-1)}} \cdot h^{\gamma} \cdot h_1^{\gamma a} \cdot h_2^{\gamma a+\gamma\overline{m}} \right) \bmod N \\
& \overset{(*)}{\equiv} c_N^{-\mu_{\overline{e}}} \cdot h_1^{\mu_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\mu_{s_i}2^{3\lambda(i-1)}} \cdot h_1^{\mu_a} \cdot h_2^{\mu_a} \cdot \left(c_N^{-\gamma\overline{e}} \cdot h_1^{\gamma\omega} \cdot h_1^{\overline{E}s} \cdot h^{\gamma} \cdot h_1^{\gamma a} \cdot h_2^{\gamma a+\gamma\overline{m}} \right) \bmod N \\
& \overset{(**)}{\equiv} c_N^{-\mu_{\overline{e}}} \cdot h_1^{\mu_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\mu_{s_i}2^{3\lambda(i-1)}} \cdot h_1^{\mu_a} \cdot h_2^{\mu_a} \bmod N \\
& \equiv \Omega_f \bmod N
\end{aligned}
$$

where we make use of the response calculations $\tau_e \leftarrow \gamma e + \mu_e, \tau_a \leftarrow \gamma a + \mu_a, \tau_\omega \leftarrow \gamma\omega + \mu_\omega$ over $\mathbb{Z}$, $(*)$ follows from the unique $2^{3\lambda(i-1)}$-decomposition of $s = \sum_{i\in[\ell_s]} s_i 2^{3\lambda(i-1)}$, and $(**)$ follows from Equation (17). Next, the check on line 22 is satisfied *as per* the following computation over $\mathbb{G}$

$$
\begin{aligned}
& (\tau_{\overline{e}}, \tau_a)H + \tau_r\vec{G} - \gamma\vec{C} \\
& = (\gamma\overline{e}+\mu_{\overline{e}}, \gamma a+\mu_a)H + (\gamma r+\mu_r)\vec{G} - \gamma((\overline{e}\cdot H, a\cdot H) + r\vec{G}) && (22) \\
& = (\mu_{\overline{e}}, \mu_a)H + \mu_r\vec{G} \\
& = \vec{\Omega}_C, && (23) \\
& \tau_r H - \gamma F \\
& = (\gamma r+\mu_r)H - \gamma r H \\
& = \Omega_F && (24)
\end{aligned}
$$

where the first equality (22) follows from the definition (19) of $\vec{C} = (\overline{e}H, aH) + r\cdot\vec{G}$ in relation R_Σ, and equalities (23, 24) follow from the definitions by prover at line 6.

We now examine the nextt check on line 23. In the same vein of computation

$$
\begin{aligned}
& \vec{\tau}_s H' + \tau_{r'}\vec{G}' - \gamma\vec{C}' \\
& = (\gamma\vec{s}+\vec{\mu}_s)H' + (\gamma r'+\mu_{r'})\vec{G}' - \gamma(\vec{s}\cdot H' + r'\vec{G}') \\
& = \vec{\mu}_s H' + \mu_{r'}\vec{G}' \\
& = \vec{\Omega}_{C'}, \\
& \tau_{r'}H' - \gamma F' \\
& = (\gamma r'+\mu_{r'})H' - \gamma r' H' \\
& = \Omega_{F'} \ ,
\end{aligned}
$$

where the opening $\vec{\tau}_s \leftarrow \gamma\vec{s}+\vec{\mu}_s$ and the opening $\tau_{r'} \leftarrow \gamma r' + \mu_{r'}$ are both over $\mathbb{Z}$ from lines (16, 13), for which the group exponentiation is well defined in the cyclic group $\mathbb{G}$. Regarding the check on line 24, we apply the specification of the opening $\tau_{\tilde{t}} \leftarrow \gamma\tilde{t} + \mu_{\tilde{t}}$, together with those of $\tau_{\overline{e}} \leftarrow \gamma\overline{e} + \mu_{\overline{e}}, \tau_a \leftarrow \gamma a + \mu_a, \tau_{e'} \leftarrow \gamma e' + \mu_{e'}$ as well as $\vec{\tau}_s \leftarrow \gamma\vec{s} + \vec{\mu}_s$ by the prover on lines (14, 15, 16),

respectively:

$$
\begin{aligned}
&\tilde{c}^{-\gamma} \cdot h^{\tau_{\tilde{t}}} \cdot \tilde{g}_1^{\tau_{\overline{e}}} \tilde{g}_2^{\tau_a} \tilde{g}_3^{\tau_{e'}} \textstyle\prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\tau_{s_i}} \cdot \tilde{g}_{4+\ell_s}^{\tau_\omega} \\
&\equiv \tilde{c}^{-\gamma} \cdot h^{\gamma\tilde{t}+\mu_{\tilde{t}}} \cdot \tilde{g}_1^{\gamma\overline{e}+\mu_{\overline{e}}} \tilde{g}_2^{\gamma a+\mu_a} \tilde{g}_3^{\gamma e'+\mu_{e'}} \textstyle\prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\gamma s_i+\mu_{s_i}} \cdot \tilde{g}_{4+\ell_s}^{\gamma\omega+\mu_\omega} \\
&\overset{(*)}{\equiv} \left(h^{\tilde{t}} \cdot \tilde{g}_1^{\overline{e}} \tilde{g}_2^{a} \tilde{g}_3^{e'} \textstyle\prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{s_i} \cdot \tilde{g}_{4+\ell_s}^{\omega}\right)^{-\gamma} \cdot h^{\gamma\tilde{t}+\mu_{\tilde{t}}} \cdot \tilde{g}_1^{\gamma\overline{e}+\mu_{\overline{e}}} \tilde{g}_2^{\gamma a+\mu_a} \tilde{g}_3^{\gamma e'+\mu_{e'}} \textstyle\prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\gamma s_i+\mu_{s_i}} \tilde{g}_{4+\ell_s}^{\gamma\omega+\mu_\omega} \\
&\equiv \left(h^{\mu_{\tilde{t}}} \cdot \tilde{g}_1^{\mu_{\overline{e}}} \tilde{g}_2^{\mu_a} \tilde{g}_3^{\mu_{e'}} \textstyle\prod_{i\in[\ell_s]} \tilde{g}_{3+i}^{\mu_{s_i}} \cdot \tilde{g}_{4+\ell_s}^{\mu_\omega}\right) \\
&\equiv \Omega_{\tilde{c}} \bmod \tilde{N}
\end{aligned}
$$

where $(*)$ and the last equality follow from the definition of $\tilde{c}$ and the definition of $\Omega_{\tilde{c}}$ by prover on line 10, respectively. Lastly, the check on line 26 is satisfied by

$$
\begin{aligned}
\tilde{c}_q^{\gamma} \cdot \Omega_q &\equiv \left(\tilde{h}^{t_q} \cdot \tilde{g}_1^{f_{\omega,1}} \cdot \tilde{g}_2^{f_{e,1}}\right)^{\gamma} \cdot \tilde{h}^{\mu_{t_q}} \cdot \tilde{g}_1^{f_{\omega,0}} \cdot \tilde{g}_2^{f_{e,0}} && (25)\\
&\equiv \tilde{h}^{\gamma t_q+\mu_{t_q}} \cdot \tilde{g}_1^{\gamma\cdot(\mu_\omega-\sum_{i\in[\ell_s]}(\mu_{s_i}e+s_i\mu_e)2^{3\lambda(i-1)})-\sum_{i\in[\ell_s]}\mu_{s_i}\mu_e 2^{3\lambda(i-1)}} \cdot \tilde{g}_2^{\mu_e-2\gamma\mu_{e'}} && (26)\\
&\equiv \tilde{h}^{\tau_{t_q}} \cdot \tilde{g}_1^{f_\omega} \cdot \tilde{g}_2^{f_e} && (27)
\end{aligned}
$$

where equation (25) follows from the definition of $\tilde{c}_q$ and Ω_q by prover on line 10, equation (26) comes from $f_{\omega,0} \leftarrow -\sum_{i\in[\ell_s]} \mu_{s_i}\mu_{\overline{e}} 2^{3\lambda(i-1)}$ as well as

$$
f_{\omega,1} \leftarrow \mu_\omega - \textstyle\sum_{i\in[\ell_s]}(\mu_{s_i}e+s_i\mu_e)2^{3\lambda(i-1)} + \sum_{i\in[\ell_s]} \mu_{s_i}\overline{E}2^{3\lambda(i-1)} = \mu_\omega - \sum_{i\in[\ell_s]}(\mu_{s_i}\overline{e}+s_i\mu_{\overline{e}})2^{3\lambda(i-1)} \ ,
$$

where we are using the fact that $\overline{e} = e - \overline{E}$. Moreover, $f_{e,0} \leftarrow \mu_{\overline{e}}, f_{e,1} \leftarrow -2\mu_{e'}$ on lines (8, 9) by prover, and the last equation (27) results from $\tau_{t_q} \leftarrow \gamma t_q + \mu_{t_q} \in \mathbb{Z}$ together with the below calculation

$$
\begin{aligned}
f_\omega &= \gamma \cdot \tau_\omega - \left(\sum_{i\in[\ell_s]} \tau_{s_i} 2^{3\lambda(i-1)}\right) \cdot \tau_{\overline{e}} \\
&= \gamma \cdot (\mu_\omega + \gamma\omega) - \left(\sum_{i\in[\ell_s]} (\mu_{s_i} + \gamma s_i) \cdot 2^{3\lambda(i-1)}\right) \cdot (\mu_{\overline{e}} + \gamma\overline{e}) \\
&= \gamma \cdot \left(\mu_\omega - \sum_{i\in[\ell_s]} (\mu_{s_i}\overline{e} + s_i\mu_{\overline{e}})2^{3\lambda(i-1)}\right) - \sum_{i\in[\ell_s]} \mu_{s_i} 2^{3\lambda(i-1)}\mu_{\overline{e}} + \gamma^2 \cdot (\omega - s\overline{e}) \\
&= \gamma \cdot \left(\mu_\omega - \sum_{i\in[\ell_s]} (\mu_{s_i}\overline{e} + s_i\mu_{\overline{e}})2^{3\lambda(i-1)}\right) - \sum_{i\in[\ell_s]} \mu_{s_i} 2^{3\lambda(i-1)}\mu_{\overline{e}} && (28)\\
f_e &= \gamma \cdot (\tau_{\overline{e}} - (2 \cdot \tau_{e'} + \gamma)) \\
&= \gamma \cdot (\gamma\overline{e} + \mu_{\overline{e}} - 2 \cdot \mu_{e'} - 2\gamma e' - \gamma) \\
&= \gamma \cdot (\mu_{\overline{e}} - 2 \cdot \mu_{e'}) && (29)
\end{aligned}
$$

in which equation (28) follows from the condition (18) that $\omega = \overline{e}s$ in R_Σ, while equation (29) also follows from $\overline{e} = 2e' + 1$ in condition (18) of R_Σ.

For showing that the Σ-protocol has *high min-entropy*, we consider the first flow

$$
\Omega := (\tilde{c}, \tilde{c}_q, \vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}, \Omega_q).
$$

Recall that $\mathbb{G}$ is of order $p > 2^{2\lambda}$. Observe that $\Omega_F = \mu_r H$ is distributed uniform over $\mathbb{G}$, since $H \neq 0$ and μ_r is sampled uniform over $\mathbb{Z}_p$. Thus, an adversary can predict Ω with probability at most $2^{-2\lambda}$, and the Σ-protocol has high min-entropy.

For *non-abort HVZK*, we need to show that the transcript can be simulated indistinguishably conditioned that the prover does not abort and the challenge γ of an honest verifier is known in

advance. We observe that following properties of noise flooding (recall Supplementary Materials A.2), the masking by

$$\begin{aligned}\mu_{\tilde{t}}, \mu_{t_q} &\leftarrow [0, CN \cdot 2^{2\lambda}]\\ (\mu_{\overline{e}}, \mu_a) \leftarrow [0, (\vec{B}C+1)L], \mu_{e'} &\leftarrow [0, (B_1 C+1)L]\\ \vec{\mu}_s &\leftarrow [0, (\vec{B}'C+1)L]\\ \mu_\omega &\leftarrow [0, CB_1 N \cdot 2^{2\lambda}]\end{aligned}$$

ensures that the following values

$$\begin{aligned}\tau_{\tilde{t}} \leftarrow \gamma\tilde{t} + \mu_{\tilde{t}}, \tau_{t_q} &\leftarrow \gamma t_q + \mu_{t_q}\\ \tau_{\overline{e}} \leftarrow \gamma\overline{e} + \mu_{\overline{e}}, \tau_a \leftarrow \gamma a + \mu_a, \tau_{e'} &\leftarrow \gamma e' + \mu_{e'}\\ \vec{\tau}_s \leftarrow \gamma\vec{s} + \vec{\mu}_s, \tau_\omega &\leftarrow \gamma\omega + \mu_\omega\end{aligned}$$

sent in the third flow are distributed uniform over the corresponding interval in the prover's range conditioned on no abort. Similarly, $\tau_r \leftarrow \gamma r + \mu_r, \tau_{r'} \leftarrow \gamma r' + \mu_{r'}$ are distributed uniform since $\mu_r, \mu_{r'}$ are sampled uniformly at random from $\mathbb{Z}_p$. Also, $\tilde{c}$ and $\tilde{c}_q$ are distributed close to uniform over $\langle\tilde{h}\rangle$ (with statistical distance at most $2^{-\lambda}$) because $\tilde{t}$ and t_q are drawn uniform over $[0, N \cdot 2^\lambda]$, and because $\langle\tilde{h}\rangle = \langle\tilde{g}_i\rangle$. (Note that the order of $\langle\tilde{h}\rangle \subseteq \mathbb{Z}_N^*$ is at most N.) The remaining values $\vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}$ and Ω_q are determined from the third flow and the challenge γ, by setting with respect to the verification equations

$$\begin{aligned}\Omega_f &= c_N^{-\tau_{\overline{e}}} \cdot h_1^{\tau_\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\tau_{s_i}2^{3\lambda(i-1)}} \cdot h^\gamma \cdot h_1^{\tau_a} \cdot h_2^{\tau_a+\gamma\overline{m}} \bmod N\\ \vec{\Omega}_C &= (\tau_{\overline{e}}, \tau_a)H + \tau_r\vec{G} - \gamma\vec{C}, \quad \Omega_F = \tau_r H - \gamma F\\ \vec{\Omega}_{C'} &= \vec{\tau}_s H' + \tau_{r'}\vec{G}' - \gamma\vec{C}', \quad \Omega_{F'} = \tau_{r'}H' - \gamma F'\\ \Omega_{\tilde{c}} &= \tilde{c}^{-\gamma} \cdot h^{\tau_{\tilde{t}}} \cdot \tilde{g}_1^{\tau_{\overline{e}}}\tilde{g}_2^{\tau_a}\tilde{g}_3^{\tau_{e'}}\prod_{i\in[\ell_s]}\tilde{g}_{3+i}^{\tau_{s_i}} \bmod \tilde{N}\\ \Omega_q &= \tilde{c}_q^{-\gamma} \cdot \tilde{h}^{\tau_{t_q}} \cdot \tilde{g}_1^{f_\omega} \cdot \tilde{g}_2^{f_e} \bmod \tilde{N}\end{aligned}$$

where $f_\omega = \gamma\cdot\tau_\omega - ((\sum_{i\in[\ell_s]}\tau_{s_i}2^{3\lambda(i-1)})\cdot(\tau_{\overline{e}}))$, $f_e = \gamma(\tau_{\overline{e}} - (2\cdot\tau_{e'}+\gamma))$. Given the above discussion, it is straightforward to provide an appropriate simulator. Roughly, the simulator receives the challenge γ and samples the responses (for the third flow) according to the above distributions. It also samples $\tilde{c}$ and $\tilde{c}_q$ (with negligible distance close to) uniform over $\langle\tilde{h}\rangle$. Finally, it recomputes the remaining first flow that is determined as above and outputs the full transcript. Since the distribution is statistically close to the real protocol (conditioned on no abort), the simulator suffices.

For 3-*special soundness*, let us construct a deterministic PT extractor so that, for

$$x = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}_I', c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C}', F') \ ,$$

given 3 valid related transcripts that are indexed by $i \in [3]$

$$\begin{aligned}\text{Identical first flow:} \quad & \Omega = (\tilde{c}, \tilde{c}_q, \Omega_f, \vec{\Omega}_C, \Omega_F, \vec{\Omega}_{C'}, \Omega_{F'}, \Omega_{\tilde{c}}, \Omega_q)\\ \text{Pairwise distinct challenge:} \quad & \gamma_i\\ \text{Third flow:} \quad & \vec{\tau}_i = (\tau_{r,i}, \tau_{r',i}, \tau_{\tilde{t},i}, \tau_{t_q,i}, \tau_{\overline{e},i}, \tau_{a,i}, \tau_{e',i}, \vec{\tau}_{s,i}, \tau_{\omega,i})\end{aligned}$$

the extractor extracts a witness w such that $(x, w) \in \widetilde{\mathsf{R}}_{\mathbf{\Sigma}}$. In the following, we denote by $\Delta v_{i,j} \coloneqq \tau_{v,i} - \tau_{v,j} \in \mathbb{Z}$ for $v \in \{r, r', \tilde{t}, t_q, \bar{e}, a, e', s, \omega\}$ and $\Delta\gamma_{i,j} \coloneqq \gamma_i - \gamma_j$.

By verification, we have that for $i \in [3]$, it holds that

$$\tilde{c}^{-\gamma_i} \cdot h^{\tau_{\tilde{t},i}} \cdot \tilde{g}_1^{\tau_{\overline{e},i}}\tilde{g}_2^{\tau_{a,i}}\tilde{g}_3^{\tau_{e',i}}\prod_{k\in[\ell_s]}\tilde{g}_{3+k}^{\tau_{s_k,i}} \cdot \tilde{g}_{4+\ell_s}^{\tau_\omega} \equiv \Omega_{\tilde{c}} \bmod \tilde{N}$$

Thus, it holds for $i \neq j \in [3]$ that

$$\tilde{c}^{\Delta\gamma_{i,j}} \equiv h^{\Delta\tilde{t}_{i,j}} \cdot \tilde{g}_1^{\Delta\overline{e}_{i,j}}\tilde{g}_2^{\Delta a_{i,j}}\tilde{g}_3^{\Delta e'_{i,j}}\prod_{k\in[\ell_s]}\tilde{g}_{3+k}^{\Delta s_{k\,i,j}} \cdot \tilde{g}_{4+\ell_s}^{\Delta\omega_{i,j}} \bmod \tilde{N}$$

Thus, either $\Delta\gamma_{i,j}$ divides $\Delta v_{i,j}$ for $v \in \{\tilde{t}, \bar{e}, a, e', s, \omega\}$ over $\mathbb{Z}$, or the above equation yields a witness w for $(\tilde{\mathsf{pp}}, w) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$. In the latter case, the extractor outputs the witness w. In the former case, the extractor defines $r \coloneqq \Delta r_{1,2}/\Delta\gamma_{1,2} \bmod p$, $r' \coloneqq \Delta r'_{1,2}/\Delta\gamma_{1,2} \bmod p$ over $\mathbb{Z}_p$, as well $\bar{e} \coloneqq \Delta\bar{e}_{1,2}/\Delta\gamma_{1,2}, a \coloneqq \Delta a_{1,2}/\Delta\gamma_{1,2}, s_i \coloneqq \Delta(s_i)_{1,2}/\Delta\gamma_{1,2}$, and $\omega \coloneqq \Delta(\omega)_{1,2}/\Delta\gamma_{1,2}$ over $\mathbb{Z}$ (which is now well-defined).

Further, the verification checks ensure that

$$
\begin{aligned}
&(\tau_{\bar{e},1}, \tau_{a,1})H + \tau_{r,1}\vec{G} - \gamma_1\vec{C} &&= (\tau_{\bar{e},2}, \tau_{a,2})H + \tau_{r,2}\vec{G} - \gamma_2\vec{C} &&= (\tau_{\bar{e},3}, \tau_{a,3})H + \tau_{r,3}\vec{G} - \gamma_3\vec{C} && (30)\\
&\tau_{r,1}H - \gamma_1 F &&= \tau_{r,2}H - \gamma_2 F &&= \tau_{r,3}H - \gamma_3 F && (31)\\
&\vec{\tau}_{s,1}H' + \tau_{r',1}\vec{G}' - \gamma_1\vec{C}' &&= \vec{\tau}_{s,2}H' + \tau_{r',2}\vec{G}' - \gamma_2\vec{C}' &&= \vec{\tau}_{s,3}H' + \tau_{r',3}\vec{G}' - \gamma_3\vec{C}' && (32)\\
&\tau_{r',1}H' - \gamma_1 F' &&= \tau_{r',2}H' - \gamma_2 F' &&= \tau_{r',3}H' - \gamma_3 F' && (33)
\end{aligned}
$$

We first use the identities (31, 33) to obtain

$$
\begin{aligned}
F &= \frac{\Delta r_{1,2}}{\Delta\gamma_{1,2}}H = rH\\
F' &= \frac{\Delta r'_{1,2}}{\Delta\gamma_{1,2}}H' = r'H
\end{aligned}
$$

as correct representations of F and F'. Then, the identity (30) gives

$$
\begin{aligned}
&(\tau_{\bar{e},1}, \tau_{a,1})H + \tau_{r,1}\vec{G} - \gamma_1\vec{C} && = (\tau_{\bar{e},2}, \tau_{a,2})H + \tau_{r,2}\vec{G} - \gamma_2\vec{C}\\
\implies &(\tau_{\bar{e},1} - \tau_{\bar{e},2}, \tau_{a,1} - \tau_{a,2})H + (\tau_{r,1} - \tau_{r,2})\vec{G} && = (\gamma_1 - \gamma_2)\vec{C}\\
\implies &\vec{C} && = \left(\frac{\Delta\bar{e}_{1,2}}{\Delta\gamma_{1,2}}, \frac{\Delta a_{1,2}}{\Delta\gamma_{1,2}}\right)H + \frac{\Delta r_{1,2}}{\Delta\gamma_{1,2}}\vec{G}\\
\implies &\vec{C} && = (\bar{e}, a)\,H + r\vec{G}.
\end{aligned}
$$

Thus, the extractor obtains a correct opening for $\vec{C}$ where the range is ensured $(\bar{e}, a) \in [-\vec{B}T, \vec{B}T]$ thanks to check on line 19 and $0 \neq \Delta\gamma_{1,2} \in (0, C]$ without loss of generality. The same calculation based on (32) can be carried out for $\vec{C}'$ as well, yielding

$$
\vec{C}' = \vec{s}H' + r'\vec{G}'.
$$

The obtained opening is ensured to be correct (in the sense that $\vec{s} \in [-\vec{B}'T, \vec{B}'T]$) thanks to passing the check on line 20 and $0 \neq \Delta\gamma_{1,2} \in (0, C]$ without loss of generality.

Next, we look at the check on line 21, where the three valid transcripts give

$$
\begin{aligned}
& c_N^{-\tau_{\bar{e},1}} \cdot h_1^{\tau_{\omega,1}} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\tau_{1,s_i}2^{3\lambda(i-1)}} \cdot h^{\gamma_1} \cdot h_1^{\tau_{a,1}} \cdot h_2^{\tau_{a,1}+\gamma_1\overline{m}}\\
\equiv\ & c_N^{-\tau_{\bar{e},2}} \cdot h_1^{\tau_{\omega,2}} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\tau_{2,s_i}2^{3\lambda(i-1)}} \cdot h^{\gamma_2} \cdot h_1^{\tau_{a,2}} \cdot h_2^{\tau_{a,2}+\gamma_2\overline{m}}\\
\equiv\ & c_N^{-\tau_{\bar{e},3}} \cdot h_1^{\tau_{\omega,3}} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\tau_{3,s_i}2^{3\lambda(i-1)}} \cdot h^{\gamma_3} \cdot h_1^{\tau_{a,3}} \cdot h_2^{\tau_{a,3}+\gamma_3\overline{m}}\\
\equiv\ & \Omega_f \bmod N\ .
\end{aligned}
$$

In particular, this implies

$$
c_N^{-\Delta\bar{e}_{1,2}} \cdot h_1^{\Delta\omega_{1,2}} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\Delta s_{i(1,2)}2^{3\lambda(i-1)}} \equiv h^{\Delta\gamma_{1,2}} \cdot h_1^{\Delta a_{1,2}} \cdot h_2^{\Delta a_{1,2}+\Delta\gamma_{1,2}\overline{m}} \bmod N. \tag{34}
$$

As noted above, $\Delta\gamma_{1,2}$ divides all the values in the exponent over $\mathbb{Z}$, and thus, equation (34) implies that:

$$
\begin{aligned}
& c_N^{-\Delta\bar{e}_{1,2}/\Delta\gamma_{1,2}} \cdot h_1^{\Delta\omega_{1,2}/\Delta\gamma_{1,2}} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}\frac{\Delta s_i(1,2)}{\Delta\gamma_{1,2}}2^{3\lambda(i-1)}} \equiv h \cdot h_1^{\Delta a_{1,2}/\Delta\gamma_{1,2}} \cdot h_2^{\Delta a_{1,2}/\Delta\gamma_{1,2}+\overline{m}} \bmod N\\
\implies & c_N^{-\bar{e}} \cdot h_1^{\omega} \cdot h_1^{\overline{E}\sum_{i\in[\ell_s]}s_i 2^{3\lambda(i-1)}} \equiv h \cdot h_1^{a} \cdot h_2^{a+\overline{m}} \bmod N
\end{aligned}
$$

leading to a witness for equation (17) in $\widetilde{\mathsf{R}}_{\Sigma}$. Also, the extractor sets y such that $y \equiv c_N \cdot h_1^{-s}$, where $s = \sum_{i\in[\ell_s]} s_i 2^{3\lambda(i-1)}$ and finally, $w = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$. It remains to show that $\omega = \overline{e} \cdot s, \overline{e} = 2e' + 1$. Roughly, this follows due to the check in line 26 since we there are three related transcripts. This follows similarly to the equations over $\mathbb{Z}$ in Lemma 20, 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).

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.1, *i.e.*,

$$\mathcal{SRS} = \big\{(\tilde{N}, \tilde{\mathsf{pp}}, \pi_{\mathsf{gen}}) \mid \tilde{N} \in \mathbb{N}, \tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{4+\ell_s}) \in (\mathbb{Z}_N^*)^{5+\ell_s},$$
$$\Pi_{\mathsf{gen}}.\mathsf{Verify}^{\mathsf{H}_{\mathsf{gen}}}(x_{\mathsf{gen}}, \pi_{\mathsf{gen}}) = 1, x_{\mathsf{gen}} = (\tilde{N}, 3 + \ell_s, \tilde{h}, (\tilde{g}_1, \cdots, \tilde{g}_{\ell_m+\ell_r}))\big\}.$$

Let $\mathcal{URS} = \mathbb{G}^{1+\ell_s}$. Note that $\mathsf{urs} = \mathsf{pp}'_I$ specifies the public parameters for a second $\mathsf{C}_{\mathsf{RInt}}$ commitment (in addition to the pp_I within the statement). This is the commitment for $\mathsf{C}_{\mathsf{Grp}}$. We denote by $\mathsf{H}_{\mathsf{fis}} = (\mathsf{H}_{\mathsf{gen}}, \mathsf{H}_{\gamma})$ random oracle of Π_{fis}. (Here, $\mathsf{H}_{\mathsf{gen}}$ corresponds to the random oracle for Π_{gen} and H_{γ} maps into $[0, C]$.) Below, we have $\mathsf{urs} \in \mathcal{URS}$ and $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ for some $\mathsf{srs} \in \mathcal{SRS}$. Also, let $x = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, \vec{C}, F)$ and $w = (e, a, y, r)$. (Note that $(\vec{C}, F)$ corresponds to an $\mathsf{C}_{\mathsf{RInt}}$ commitment to $(a, e - \bar{E})$ with opening r.) The scheme is given below.

- $\Pi_{\mathsf{fis}}.\mathsf{GenSRS}(1^\lambda)$: Samples $\mathsf{pp}_{\mathsf{MPed}} = (\tilde{N}, \tilde{\mathsf{pp}}) \leftarrow \mathsf{MPed.Setup}(1^\lambda)$ with $\tilde{\mathsf{pp}} = (\tilde{h}, \tilde{g}_1, \cdots, \tilde{g}_{4+\ell_s})$. 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 $\mathsf{srs} = (\tilde{N}, \tilde{\mathsf{pp}}, \pi_{\mathsf{gen}})$.
- $\Pi_{\mathsf{fis}}.\mathsf{Prove}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}, x, w)$: First, commits to y via $\mathsf{C}_{\mathsf{Grp}}$. That is, 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 $c_N = y \cdot h_1^s, \vec{C}' = \vec{s}H' + r'\vec{G}'$ and $F' = r'H$. Then, compiles Σ_{fis} into a proof π via

$$\begin{aligned}
&(\Omega_\Sigma, \mathsf{st}) \leftarrow \Sigma_{\mathsf{fis}}.\mathsf{Init}(x_\Sigma, w_\Sigma),\\
&\gamma_\Sigma \leftarrow \mathsf{H}_\gamma(x_\Sigma, \Omega_\Sigma),\\
&\tau_\Sigma \leftarrow \Sigma_{\mathsf{fis}}.\mathsf{Resp}(x_\Sigma, \mathsf{st}, \gamma_\Sigma),\\
&\pi_\Sigma \leftarrow (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma),
\end{aligned}$$

  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 $w_\Sigma = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$, where $\omega = (e - \overline{E}) \cdot s$. 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

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

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

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

Proof. First, correctness follows by correctness of Σ_{fis}. (It is easy to check that $(x_\Sigma, w_\Sigma) \in \mathsf{R}_{\mathsf{fis}}$ by design.)

Subversion ZK follows as in previous NIZKs (cf. Theorems 9 and 10) and since $\mathsf{C}_{\mathsf{Grp}}$ is hiding. In more detail, observe that π_{gen} in the adversarial srs ensures that $\langle\tilde{h}\rangle = \langle\tilde{g}_i\rangle$ under soundness of Π_{gen} with overwhelming probability. Since thus $(x_\Sigma, w_\Sigma) \in \mathsf{R}_{\mathsf{fis}}$ in the real proof generation, the simulator simply simulates π_Σ via non-abort HVZK of Σ_{fis} and by programming H_γ accordingly. (The latter is possible due to high min-entropy of Σ_{fis}.) Also, the simulator outputs a fresh $\mathsf{C}_{\mathsf{Grp}}$ commitment $(c_N, \vec{C}', F')$ to $1 \in \langle h_1\rangle$ (instead of a commitment to y). This is justified by the hiding property of $\mathsf{C}_{\mathsf{Grp}}$. A simple hybrid argument allows to show that the above simulator suffices for subversion zero-knowledge.

Finally, we show adaptive knowledge soundness (cf. Definition 24). Roughly, this follows by forking the adversary twice to obtain three related transcripts. Then, 3 special soundness of Σ_{fis} ensures that a witness can be recomputed. In more detail, SimCRS simply outputs $\mathsf{crs} = (\mathsf{srs}, \mathsf{urs})$ with $\mathsf{srs} \leftarrow \mathsf{GenCRS}(1^\lambda)$ and $\mathsf{urs} \leftarrow \mathcal{URS}$. Since crs follows the real distribution, CRS indistinguishability holds. (Note that $\mathsf{td} = \bot$.) To define the extractor Ext with oracle access to some prover $\mathcal{A}$ making Q H_γ queries, let us make some preparations. First, assume that

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

Denote by $(x,\pi) \leftarrow \mathcal{A}^{\mathsf{H}_\gamma}(\mathsf{crs};\rho)$ the statement-proof pair output by $\mathcal{A}$ on input $(\mathsf{crs};\rho)$, where the oracle queries to H_γ are answered via a vector $\vec{h} \in [0,C]^Q$ and $\mathsf{H}_{\mathsf{gen}}$ queries are answered via a vector $\vec{h'}$. Parse $x = (\mathsf{pp}_I, N, h_1, h_2, h, \overline{m}, \vec{C}, F), \pi = (\pi_\Sigma, c_N, \vec{C'}, F')$ and $\pi_\Sigma = (\Omega_\Sigma, \gamma_\Sigma, \tau_\Sigma)$. Denote $x_\Sigma = (c_N, \tilde{N}, \tilde{\mathsf{pp}}, \mathsf{pp}_I, \mathsf{pp}'_I, c, N, h_1, h_2, h, \overline{m}, \vec{C}, F, \vec{C'}, F')$, where $(\tilde{\mathsf{pp}}, \mathsf{pp}'_I)$ are specified by the crs. Define by i the index of the first oracle query $q_i = (x_\Sigma, \Omega_\Sigma)$ to H_γ made by $\mathcal{A}$. If no such query exists, then $i = \bot$. Note that if (x,π) verifies, then except with probability $1/(C+1)$ such a query exists because $\mathsf{H}_\gamma(x_\Sigma, \Omega_\Sigma) = \gamma_\Sigma$ must hold. Denote by E the event that $\mathsf{Verify}^{\mathsf{H}_{\mathsf{fis}}}(\mathsf{crs}, x, \pi) = 1$ and $i \neq \bot$. By the above discussion, we have

$$\Pr[E] \geq \mu(\lambda) - 1/(C+1).$$

For fixed $(\mathsf{crs}, \rho, \vec{h'})$, let us define a function $F_{\mathsf{Ext}} : [0,C]^Q \mapsto [Q]$ by $F_{\mathsf{Ext}}(\vec{h}) = i - 1$ if E occurs and $F_{\mathsf{Ext}}(\vec{h}) = 1$ otherwise.

We are now ready to define the extractor Ext. First, Ext samples randomness ρ for $\mathcal{A}$ and a tuple $\vec{h}_1 \in [0,C]^Q$ which corresponds to the H_γ outputs to $\mathcal{A}$'s queries. Also, samples the $\mathsf{H}_{\mathsf{gen}}$ outputs $\vec{h'}$ which remain unaltered in the following. Then, Ext runs $\mathcal{A}$ on input $(\mathsf{crs};\rho)$ answering the $\mathsf{H}_{\mathsf{fis}}$ queries via $\vec{h}_1$ and $\vec{h'}$. If E occurs, then Ext samples 2 other vectors $\vec{h}_2$ and $\vec{h}_3$ from $[0,C]$ at random conditioned on $F_{\mathsf{Ext}}(\vec{h}_1) = F_{\mathsf{Ext}}(\vec{h}_2) = F_{\mathsf{Ext}}(\vec{h}_3)$. Then, Ext runs $\mathcal{A}$ two more times on input $(\mathsf{crs};\rho)$ by answering the H_γ queries $\vec{h}_2$ and $\vec{h}_3$ (and the $\mathsf{H}_{\mathsf{gen}}$ queries via $\vec{h'}$). If E occurs for all $\vec{h}_k$, $k \in [3]$, parse $\mathcal{A}$'s output for each run as above but indexed by k. By construction, we have that $(x_\Sigma, \Omega_\Sigma) \coloneqq (x_{\Sigma,1}, \Omega_{\Sigma,1}) = (x_{\Sigma,2}, \Omega_{\Sigma,2}) = (x_{\Sigma,3}, \Omega_{\Sigma,3})$ [39]. If the challenges $(\gamma_{\Sigma,k})_{k\in[3]}$ are distinct, then Ext applies the extractor of Σ_{fis} to the three related transcripts $(\Omega_{\Sigma,k}, \gamma_{\Sigma,k}, \tau_{\Sigma,k})_{k\in[3]}$ for statement x_Σ. This yields a witness w_Σ such that $(w_\Sigma, x_\Sigma) \in \widetilde{\mathsf{R}}_{\mathbf{\Sigma}}$. Parse $w_\Sigma = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$. Finally, Ext outputs $w = (e, a, y, r)$. If the challenges are not distinct or E does not occur for $i \in [k]$, Ext outputs $\bot$.

It remains to show that Ext outputs a witness w such that $(x,w) \in \widetilde{\mathsf{R}}_{\mathsf{fis}}$ with sufficient probability. For convenience, let us recall the definition of $\widetilde{\mathsf{R}}_{\mathsf{fis}}$ below.

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

First, let us analyze the probability that Ext outputs some witness $w \neq \bot$. By lemma 4, the event E occurs for all runs of $\mathcal{A}$ with probability at least $\mu(\lambda)^3/Q^2$. Also, with probability $1 - 3/(C+1)$, the challenges $(\gamma_{\Sigma,k})_k$ are distinct. Thus, we have

$$\Pr[(x_\Sigma, w_\Sigma) \in \widetilde{\mathsf{R}}_{\mathbf{\Sigma}}] \geq \mu(\lambda)^3/q^2 - 3/(C+1) = \mu(\lambda)^3/Q^2 - \mathrm{negl}(\lambda),$$

Further, under the sRSA assumption, it holds that

$$\Pr[(\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}}) \text{ or } (\tilde{\mathsf{pp}}, w_\Sigma) \in \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})] = \mathrm{negl}(\lambda).$$

To see this, recall that finding a witness for either of the above relations is hard under sRSA for random $\tilde{\mathsf{pp}} \in \mathsf{QR}_{\tilde{N}}^{\ell_s+5}$ and RSA modulus $\tilde{N}$. Since $(\tilde{N}, \tilde{\mathsf{pp}})$ is part of the srs, it is possible to embed a hard instance into srs. It is straightforward to construct appropriate adversaries on sRSA and we omit details. Thus, we have

$$\Pr[(x_\Sigma, w_\Sigma) \in \widetilde{\mathsf{R}}_{\mathbf{\Sigma}} \wedge (\tilde{\mathsf{pp}}, w_\Sigma) \notin \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}}), (\tilde{\mathsf{pp}}, w_\Sigma) \notin \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})] \geq \mu(\lambda)^3/Q^2 - \mathrm{negl}(\lambda),$$

[39] Observe that for fixed $\vec{h} \in [0,C]^Q$, the output of $\mathcal{A}$ is deterministic. Because each run, the H_γ queries are answered identically until query $q_{i,k} = (x_{\Sigma,k}, \Omega_{\Sigma,k})$, the input $q_{i,k}$ to this query is identical for all $k \in [3]$.

By definition of $\widetilde{\mathsf{R}}_{\Sigma}$, if $(w_{\Sigma}, x_{\Sigma}) \in \widetilde{\mathsf{R}}_{\Sigma}$ but $(\tilde{\mathsf{pp}}, w_{\Sigma}) \notin \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$ and $(\tilde{\mathsf{pp}}, w_{\Sigma}) \notin \mathsf{R}_{C,\tilde{\ell}}(\tilde{\mathsf{pp}})$, the following holds:

$$
\begin{aligned}
&c_N^{\overline{e}+\overline{E}} \cdot h_1^{-\omega} \cdot h_1^{-s\cdot\overline{E}} \equiv h \cdot h_1^{a} \cdot h_2^{a+\overline{m}} \bmod N, \omega = \overline{e}\cdot s, \overline{e} = 2e' + 1, e = \overline{e} + \overline{E} \\
&\vec{C} = (\overline{e}\cdot H, a\cdot H) + r\vec{G}, F = rH \\
&\vec{C}' = \vec{s}H' + r'\vec{G}', F' = r'H, s = \textstyle\sum_{i\in[\ell_s]} s_i 2^{3\lambda(i-1)}, c_N \equiv y \cdot h_1^{s} \bmod N, \\
&(\overline{e}, a) \in [-\vec{B}T, \vec{B}T], (s_1, \ldots, s_{\ell_s}) \in [-\vec{B}'T, \vec{B}'T]
\end{aligned}
$$

It follows that $(x, w) \in \widetilde{\mathsf{R}}_{\mathsf{fis}}$. To see this, observe that all equations but $y^e \equiv h \cdot h_1^{a} \cdot h_2^{a+\overline{m}} \bmod N$ follow immediately. The latter follows because

$$
\begin{aligned}
&c_N^{\overline{e}+\overline{E}} \cdot h_1^{-\omega} \cdot h_1^{-s\cdot\overline{E}} \\
\equiv &(y \cdot h_1^{s})^{e} \cdot h_1^{-\overline{e}\cdot s} \cdot h_1^{-s\cdot\overline{E}} \\
\equiv &y^e \cdot (h_1^{e\cdot s} \cdot h_1^{-s(\overline{e}\cdot\overline{E})}) \equiv y^e \bmod N.
\end{aligned}
$$

This concludes the proof.

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

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 $w = (e, a, y, r, r', \omega, (s_1, \cdots, s_{\ell_s}))$.