Practical Round-Optimal Blind Signatures in the ROM from Standard Assumptions

Practical Round-Optimal Blind Signatures in the ROM from Standard Assumptions

Shuichi Katsumata[1], Michael Reichle[2], and Yusuke Sakai[3]

[1] PQShield Ltd. and AIST, Japan
[2] ETH Zürich, Switzerland. Work done while employed at Inria, Paris.
[3] AIST, Japan

Blind signatures serve as a foundational tool for privacy-preserving applications and have recently seen renewed interest due to new applications in blockchains and privacy-authentication tokens. With this, constructing practical *round-optimal* (i.e., signing consists of the minimum two rounds) blind signatures in the random oracle model (ROM) has been an active area of research, where several impossibility results indicate that either the ROM or a trusted setup is inherent.

In this work, we present two round-optimal blind signatures under standard assumptions in the ROM with different approaches: one achieves the smallest sum of the signature and communication sizes, while the other achieves the smallest signature size. Both of our instantiations are based on standard assumptions over asymmetric pairing groups, i.e., CDH, DDH, and/or SXDH. Our first construction is a highly optimized variant of the generic blind signature construction by Fischlin (CRYPTO'06) and has signature and communication sizes 447 B and 303 B, respectively. We progressively weaken the building blocks required by Fischlin and we result in the first blind signature where the sum of the signature and communication sizes fit below 1 KB based on standard assumptions. Our second construction is a semi-generic construction from a specific class of randomizable signature schemes that admits an *all-but-one* reduction. The signature size is only 96 B while the communication size is 2.2 KB. This matches the previously known smallest signature size while improving the communication size by several orders of magnitude. Finally, both of our constructions rely on a (non-black box) fine-grained analysis of the forking lemma that may be of independent interest.

1 Introduction

1.1 Background

Blind signature is an interactive signing protocol between a signer and a user with advanced privacy guarantees. At the end of the protocol, the user obtains a signature for his choice of message while the signer remains blind to the message she signed. To capture the standard notion of unforgeability, it is further required that a user interacting with the signer at most ℓ-times is not be able to produce valid signatures on more than ℓ distinct messages. The former and latter are coined as the *blindness* and *one-more unforgeability* properties, respectively.

Chaum introduced the notion of blind signatures [22] and showed its application to e-cash [22, 24, 48]. Since then, it has been an important building block for other applications such as anonymous credentials [16, 20], e-voting [23, 32], direct anonymous attenstation [17], and in more recent years, it has seen a renewed interest due to new applications in blockchains [57, 19] and privacy-preserving authentication tokens [55, 37].

Round-Optimality. One of the main performance measures for blind signatures is *round-optimality*, where the user and signer are required to only send one message each to complete the signing protocol. While this is an ideal feature for practical applications, unfortunately, there are a few impossibility results [45, 29, 49] on constructing round-optimal blind signatures in the plain model (i.e., without any trusted setup) from standard assumptions (*e.g.*, non-interactive assumptions and polynomial hardness). To circumvent this, cryptographers design round-optimal blind signatures by making a minimal relaxation of relying on the random oracle model (ROM) or the trusted setup model. Considering that trusted setups are a large obstacle for real-world deployment, in this work we focus on round-optimal blind signatures in the ROM under standard assumption[4]. We refer the readers to the full version for an overview on round optimal blind signatures under non-standard assumptions (*e.g.*, interactive or super polynomial hardness) or relying on stronger idealized models such as the generic group model.

Practical Round-Optimal Blind Signatures. Constructing a *practical* round-optimal blind signature has been an active area of research. In a seminal work, Fischlin [28] proposed the first generic round-optimal blind signature from standard building blocks. While the construction is simple, an efficient instantiation remained elusive since it required a non-interactive zero-knowledge (NIZK) proof for a relatively complex language.

Recently, in the lattice-setting, del Pino and Katsumata [25] showed a new lattice-tailored technique to overcome the inefficiency of Fischlin's generic construction and proposed a round-optimal blind signature with signature and communication sizes 100 KB and 850 KB.

A different approach that has recently accumulated attention is based on the work by Pointcheval [50] that bootstraps a specific class of blind signature schemes into a fully secure one (i.e., one-more unforgeable even if polynomially many concurrent signing sessions are started). This approach has been improved by Katz et al. [41] and Chairattana-Apirom et al. [21], and the very recent work by Hanzlik et al. [36] optimized this approach leading to a round-optimal blind signature based on the CDH assumption in the asymmetric pairing setting. One of their parameter settings provides a short signature size of 5 KB with a communication size 72 KB.

[4] We note that all of our results favor well even when compared with schemes in the trusted setup model.

Finally, there are two constructions in the pairing setting with a trusted setup which can be instantiated in the ROM under standard assumptions [10, 2][5]. Blazy et al. [10] exploited the randomizability of Waters signature [56] and constructed a blinded version of Waters signature consisting of mere 2 group elements, i.e. 96 B. While it achieves the shortest signature size in the literature, since the user has to prove some relation to his message in a bit-by-bit manner, the communication scales linearly in the message length. For example for 256 bit messages, it requires more than 220 KB in communication. Abe et al. [2] use structure-preserving signatures (SPS) and Groth-Ostrovsky-Sahai (GOS) proofs [35] to instantiate the Fischlin blind signature with signatures of size 5.8 KB with around 1 KB of communication.

While round-optimal blind signatures in the ROM are coming close to the practical parameter regime, the signature and communication sizes are still orders of magnitude larger compared to those relying on non-standard assumptions or strong idealized models such as blind RSA [22, 7] or blind BLS [11]. Thus, we continue the above line of research to answer the following question:

How efficient can round-optimal blind signatures in the ROM be under standard assumptions?

1.2 Contributions

We present two round-optimal blind signatures based on standard group-based assumptions in the asymmetric pairing setting. The efficiency is summarized in Table 1, along with the assumptions we rely on. The first construction has signature and communication sizes 447 B and 303 B, respectively. It has the smallest communication size among all prior schemes and is the first construction where the sum of the signature and communication sizes fit below 1 KB. The second construction has signature and communication sizes 96 B and 2.2 KB, respectively. While it has a larger communication size compared to our first construction, the signature only consists of 2 group elements, matching the previously shortest by Blazy et al. [10] while simultaneously improving their communication size by around two orders of magnitude. Both constructions have efficient partially blind variants.

For our first construction, we revisit the generic blind signature construction by Fischlin [27]. We progressively weaken the building blocks required by Fischlin and show that the blind signature can be instantiated much more efficiently in the ROM than previously thought by a careful choice of the building blocks. At a high level, we show that the generic construction remains secure even if we replace the public-key encryption scheme (PKE) and online-extractable NIZK[6] with respectively a commitment scheme and a rewinding-extractable NIZK

[5] Both [10, 2] require a trusted setup for a common reference string crs consisting of random group elements. We can remove the trusted setup by using a random oracle to sample crs.

[6] This is a type of NIZK where the extractor can extract a witness from the proofs output by the adversary in an *on-the-fly* manner.

Table 1. Comparison of Round-Optimal Blind Signatures in the ROM

Reference	Signature size	Communication size	Assumption
del Pino et al. [25]	100 KB	850 KB	DSMR, MLWE, MSIS
Blazy et al. [10]	96 B	220 KB †	SXDH, CDH
Abe et al. [2]	5.5 KB	1 KB	SXDH
Hanzlik et al. [36]‡	5 KB 9 KB	72 KB 36 KB	CDH
Ours: Section 3	447 B	303 B	SXDH
Ours: Section 4	96 B	2.2 KB	DDH, CDH

All group-based assumptions are in the asymmetric paring setting, and MLWE and MSIS denote the module version of the standard LWE and SIS, respectively. DSMR denotes the decisional small matrix ratio problem, which can be viewed as the module variant of the standard NTRU. (†): Communication of [10] scales linearly with the message size, and is given here for 256 bit messages. (‡): [36] offers tradeoffs between signature and communication sizes.

such as those offered by the standard Fiat-Shamir transform [26, 51, 8]. While these modifications may seem insignificant on the surface, it accumulates in a large saving in the concrete signature and communication sizes. Moreover, our security proof requires overcoming new technical hurdles incurred by the rewinding-extraction and relies on a fined-grained analysis of a variant of the forking lemma.

For our second construction, we revisit the idea by Blazy et al. [10] relying on randomizable signatures. However, our technique is not a simple application of their idea as their construction relies on the specific structure of the Waters signature in a non-black-box manner. Our new insight is that a specific class of signature schemes with an *all-but-one* (ABO) reduction can be used in an almost black-box manner to construct round-optimal blind signatures, where ABO reductions are standard proof techniques to prove selective security of public-key primitives (see references in [47] for examples). Interestingly, we can cast the recent blind signature by del Pino and Katsumata [25] that stated to use lattice-tailored techniques as one instantiation of our methodology.

In the instantiation of our second construction, we use the Boneh-Boyen signature [12] that comes with an ABO reduction along with an online-extractable NIZK obtained via the Fiat-Shamir transform applied to Bulletproofs [18] and a Σ-protocol for some ElGamal related statements. To the best of our knowledge, this is the first time an NIZK that internally uses Bulletproofs was proven to be online-extractable in the ROM. Prior works either showed the non-interactive version of Bulletproofs to achieve the weaker rewinding extractability [5, 4] or the stronger online simulation extractability by further assuming the algebraic group model [33]. We believe the analysis of our online extractability to be novel and may be of independent interest.

1.3 Technical Overview

We give an overview of our contributions.

Fischlin's Round-Optimal Blind Signature. We review the generic construction by Fischlin [27] as it serves as a starting point for both of our constructions. The construction relies on a PKE, a signature scheme, and an NIZK. The blind signature's verification and signing keys $(\mathsf{bvk}, \mathsf{bsk})$ are identical to those of the underlying signature scheme $(\mathsf{vk}, \mathsf{sk})$. For simplicity, we assume a perfect correct PKE with uniform random encryption keys ek and that ek is provided to all the players as an output of the random oracle. The user first sends an encryption $c \leftarrow \mathsf{PKE}(\mathsf{ek}, m; r)$ of the message m. The signer then returns a signature $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, c)$ on the ciphertext c. The user then encrypts $\widehat{c} \leftarrow \mathsf{PKE}(\mathsf{ek}, c\|r\|\sigma; \widehat{r})$ and generates an NIZK proof π of the following fact where $(c, \sigma, r, \widehat{r})$ is the witness: $\widehat{c}$ encrypts (c, r, σ) under $\widehat{r}$; c encrypts the message m under r; and σ is a valid signature on c. The user outputs the blind signature $\sigma_{\mathsf{BS}} = (\widehat{c}, \pi)$.

It is not hard to see that the scheme is blind under the $\mathsf{IND\text{-}CPA}$ security of the PKE and the zero-knowledge property of the NIZK. The one-more unforgeability proof is also straight-forward: The reduction will use the adversary $\mathcal{A}$ against the one-more unforgeability game to break the $\mathtt{euf\text{-}cma}$ of the signature scheme. The reduction first programs the random oracle so that it knows the corresponding decryption key dk of the PKE. When $\mathcal{A}$ submits c to the blind signing oracle, the reduction relays this to its signing oracle and returns $\mathcal{A}$ the signature σ it obtains. Moreover, it makes a list L of decrypted messages $m \leftarrow \mathsf{Dec}(\mathsf{dk}, c)$. When $\mathcal{A}$ outputs the forgeries $(\sigma_{\mathsf{BS},\mathsf{i}} = (\widehat{c}_i, \pi_i), m_i)_{i \in [\ell+1]}$, it searches a m_i such that $m_i \notin L$, which is guaranteed to exist since there are at most ℓ signing queries. The reduction then decrypts $(c_i, r_i, \sigma_i) \leftarrow \mathsf{Dec}(\mathsf{dk}, \widehat{c}_i)$. Since the PKE is perfectly correct and due to the soundness of the NIZK, c_i could not have been queried by $\mathcal{A}$ as otherwise $m_i \in L$, and hence, (c_i, σ_i) breaks $\mathtt{euf\text{-}cma}$ security.

Source of Inefficiency. There are two sources of inefficiency when trying to instantiate this generic construction. One is the use of a *layered* encryption: the NIZK needs to prove that c is a valid encryption of m on top of proving $\widehat{c}$ is a valid encryption of (c, r, σ). This contrived structure was required to bootstrap a sound NIZK to be *online-extractable*.[7] Specifically, the one-more unforgeability proof relied on the reduction being able to extract the (partial) witness (c_i, r_i, σ_i) in an on-the-fly manner from the outer encryption $\widehat{c}_i$ explicitly included in the blind signature. The other inefficiency stems from the heavy reliance on PKEs. As far as the correctness is concerned, the PKE seems replaceable by a computationally binding commitment scheme. This would be ideal since commitment schemes tend to be more size efficient than PKEs since decryptability is not required. However, without a PKE, it is not clear how the above proof would work.

First Construction. We explain our first construction, an optimized variant of Fischlin's generic construction.

[7] Constructing an online extractable NIZK by adding a PKE on top of a sound NIZK is a standard method.

Using Rewinding-Extractable NIZKs. The first step is to relax the online-extractable NIZK with a (single-proof) rewinding-extractable NIZK. Such an NIZK allows extracting a witness from a proof output by an adversary $\mathcal{A}$ by *rewinding* $\mathcal{A}$ on a fixed random tape. NIZKs obtained by compiling a Σ-protocol using the Fiat-Shamir transform is a representative example of an efficient rewinding-extractable NIZK. The net effect of this modification is that we can remove the layer of large encryption by $\widehat{c}$, thus making the statement simpler and allowing us to remove $\widehat{c}$ from σ_{BS}.

Let us check if this rewinding-extractable NIZK suffices in the above proof of one-more unforgeability. At first glance, the proof does not seem to work due to a subtle issue added by the rewinding extractor. Observe that the reduction now needs to simulate $\mathcal{A}$ in the *rewound execution* as well. In particular, after rewinding $\mathcal{A}$, $\mathcal{A}$ may submit a new c' to the blind signing oracle, which was not queried in the initial execution. The reduction relays this c' to its signing oracle as in the first execution to simulate the signature σ'. As before, we can argue that there exists a message m_i in the forgeries output by $\mathcal{A}$ in the *first* execution such that $m_i \notin L$, but we need to further argue that $m_i \notin L'$, where L' is the list of decrypted messages $\mathcal{A}$ submitted in the *rewound* execution. Namely, we need to argue that $m_i \notin L \cup L'$ for the reduction to break `euf-cma` security. However, a naive counting argument as done before no longer works because $|L \cup L'|$ can be large as 2ℓ, exceeding the number of forgeries output by $\mathcal{A}$, i.e., $\ell + 1$.

We can overcome this issue by taking a closer look at the internal of a particular class of rewinding-extractable NIZK. Specifically, throughout this paper, we focus on NIZKs constructed by applying the Fiat-Shamir transform on a Σ-protocol (or in more general a public-coin interactive protocol). A standard way to argue rewinding-extractability of a Fiat-Shamir NIZK is by relying on the forking lemma [51, 8], which states (informally) that if an event E happened in the first run, then it will happen in the rewound round with non-negligible probability. In the above context, we define E to be the event that the i-th message in $\mathcal{A}$'s forgeries satisfy $m_i \notin L$, where i is sampled uniformly random by the reduction at the outset of the game. Here, note that E is well-defined since the reduction can prepare the list L by decrypting $\mathcal{A}$'s signing queries. The forking lemma then guarantees that we also have $m_i \notin L'$ in the rewound execution.[8] This slightly more fine-grained analysis allows us to replace the online-extractable NIZK with a rewinding-extractable NIZK.

Issue with Using Commitments. The next step is to relax the PKE by a (computationally binding) commitment scheme. While the correctness and blindness hold without any issue, the one-more unforgeability proof seems to require a major reworking. The main reason is that without the reduction being able to decrypt $\mathcal{A}$'s signing queries c, we won't be able to define the list L. In particular, we can no longer define the event E, and hence, cannot invoke the forking lemma. Thus, we are back to the situation where we cannot argue that

[8] For the keen readers, we note that we are guaranteed to have the same i-th message in both executions since these values are fixed at the forking point due to how the Fiat-Shamir transform works.

the extracted witness (c_i, r_i, σ_i) from $\mathcal{A}$'s forgeries, is a valid forgery against the `euf-cma` security game. Even worse, $\mathcal{A}$ could potentially be breaking the computationally binding property of the commitment scheme by finding two message-randomness pairs (m_i, r_i) and (m'_i, r'_i) such that they both commit to c_i but $m_i \neq m'_i$. In such a case, extracting from a single proof does not seem sufficient since a reduction would need at least two extracted witnesses to break the binding of the commitment scheme.

To cope with the latter issue first, we extend the one-more unforgeability proof to rely on a *multi-proof* rewinding-extractable NIZK. In general, multi-proof rewinding-extractors run in exponential time in the number of proofs that it needs to extract from [53, 9]. However, in our situation, with a careful argument, we can prove that our extractor runs in strict polynomial time since $\mathcal{A}$ provides all the proofs to the extractor only at the end of the game. This is in contrast to the settings considered in [53, 9] where $\mathcal{A}$ can adaptively submit multiple proofs to the extractor throughout the game.

We note that the assumption we require has not changed: a Σ-protocol for the same relation as in the single-proof setting compiled into an NIZK via the Fiat-Shamir transform. To prove multi-proof rewinding-extractability of this Fiat-Shamir NIZK, we can no longer rely on the now standard general forking lemma by Bellare and Neven [8] that divorces the probabilistic essence of the forking lemma from any particular application context. A naive extension of the general forking lemma to the multi-forking setting will incur an exponential loss in the success probability. To provide a meaningful bound, we must take into account the extra structure offered by the Fiat-Shamir transform, and thus our analysis is akin to the more traditional forking lemma analysis by Pointcheval and Stern [51] or by Micali and Reyzin [46]. To the best of our knowledge, we provide the first formal analysis of the multi-proof rewinding-extractability of an NIZK obtained by applying the Fiat-Shamir transform to a Σ-protocol. We believe this analysis to be of independent interest.

Final Idea to Finish the Proof. Getting back to the proof of one-more unforgeability, the reduction now executes the multi-proof rewinding-extractor to extract all the witnesses $(c_i, r_i, \sigma_i)_{i \in [\ell+1]}$ from the forgeries. Relying on the binding of the commitment scheme, we are guaranteed that all the commitment c_i's are distinct. Moreover, since $\mathcal{A}$ only makes ℓ blind signature queries *in the first execution*, we further have that there exists at least one c_i in the forgeries which $\mathcal{A}$ did not submit in the first execution.

However, we are still stuck since it's unclear how to argue that this particular c_i was never queried by $\mathcal{A}$ in any of the rewound executions. Our next idea is to slightly strengthen the NIZK so that the proof π is statistically binding to a portion of the witness that contains the commitments.[9] We note that this is still strictly weaker and more efficiently instantiable compared to an online-extractable NIZK required by Fischlin's construction since we do not require the full list of

[9] At the Σ-protocol abstraction, we call this new property f-unique extraction. It is a strictly weaker property than the *unique response* property considered in the literature [27, 54].

witnesses to be efficiently extractable from the proofs in an online manner. We use this property to implicitly fix the commitments $(c_i)_{i\in[\ell+1]}$ included in the forgeries after the end of the first execution of $\mathcal{A}$. This will be the key property to completing the proof.

The last idea is for the reduction to randomize what it queries to its signing oracle. For this, we further assume the commitment scheme is randomizable, where we emphasize that this is done for ease of explanation and we do not strictly require such an assumption (see remark 1). When $\mathcal{A}$ submits a commitment c to the blind signing oracle, the reduction randomizes c to c' using some randomness rand and instead sends c' to its signing oracle. It returns the signature σ *and* rand to $\mathcal{A}$. $\mathcal{A}$ checks if c becomes randomized to c' using rand and if σ is a valid signature on c'. It then uses c' instead of c to generate the blind signature as before. The key observation is that the reduction is invoking its signing oracle with randomness outside of $\mathcal{A}$'s control. Since the commitments $(c_i)_{i\in[\ell+1]}$ were implicitly fixed at the end of the first execution, any randomized c' sampled in the subsequent rewound execution is independent of these commitments. Hence, the probability that the reduction queries c_i to the signing oracle in any of the rewound execution is negligible, thus constituting a valid forgery against the `euf-cma` security game as desired.

Instantiation. We instantiate the framework in the asymmetric pairing setting, i.e. we have groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of prime order p, some fixed generators $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$, and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$. For the commitment scheme, we choose Pedersen commitments ($\mathsf{C}_{\mathrm{Ped}}$) of the form $c = g_1^m \mathsf{pp}^r$, as $\mathsf{C}_{\mathrm{Ped}}$ is randomizable and consists of a single group element. Note that the public parameter $\mathsf{pp} \in \mathbb{G}_1$ is generated via a random oracle. We then need to choose an appropriate signature scheme that allows signing $\mathsf{C}_{\mathrm{Ped}}$ commitments. We choose SPS as all components of the scheme are group elements, in particular, the message space is $\mathbb{G}_1^\ell$, where ℓ is the message length. The most efficient choice in the standard model is [39] with signatures of size 335 Byte. Instead, we optimize KPW signatures [43] to a signature size of 223 Byte (from originally 382 Byte). Our optimized variant $\mathsf{S}_{\mathrm{KPW}}$ is no longer structure-preserving, as it consists of one element τ in $\mathbb{Z}_p$, but suffices for our applications. We refer to the full version for more details.

Note that $\mathsf{S}_{\mathrm{KPW}}$ would be an inefficient choice in the original Fischlin blind signature [28], as it requires encrypting the signature τ over $\mathbb{Z}_p$ to instantiate the online-extractable NIZK. In the pairing setting, this incurs an overhead in proof size linear in the security parameter λ [10]. The benefit of using our framework with the weaker rewinding-extractable NIZK is that we now only need to prove knowledge of τ, and thus can get away without encrypting it. Such an NIZK is possible with a single element in $\mathbb{Z}_p$ based on a Schnorr-type Σ-protocol (compiled with Fiat-Shamir). In the Σ-protocol, we further commit to group elements $(w_i)_i \in \mathbb{G}_1^n$ in the witness via ElGamal commitments (C_{EG}) of the

[10] For instance, with ElGamal, the message is encrypted in the exponent and decryption would require a discrete logarithm computation. Thus, the message is typically encrypted bit-wise which incurs an overhead of $\log_2(p)$.

form $E_i = (w_i \cdot \mathsf{pp}^{r_i}, g_1^{r_i})$, which the prover sends to the verifier in the first flow. In particular, this ensures f-unique extraction, as E_i fixes the commitment $c \in \{w_i\}_i$ statistically. Naively, this approach requires $2n$ group elements, where n is the number of group elements in the witness. Instead, we share the randomness among all commitments under different public parameters pp_i generated via a random oracle. The commitments remain secure but require only $n+1$ group elements. In particular, we set $E_i = (w_i \cdot \mathsf{pp}_i^s)$ and fix s via $S = g_1^s$. Then, we can open *all* commitments E_i in zero-knowledge with a *single* element in $\mathbb{Z}_p$, as knowledge of s is sufficient to recover the witness w_i from all E_i. Then, we compile our Σ-protocol with Fiat-Shamir to obtain a rewinding-based NIZK. We apply a well-known optimization to avoid sending some of the first flow α, and include the hash value $\beta \leftarrow \mathsf{H}(x, \alpha)$ in the proof explicitly. In total, compared to sending the witness to the verifier in the *clear*, our NIZK only has an overhead of 1 group element in $\mathbb{G}_1$ and 3 elements in $\mathbb{Z}_p$. The additional group element is S. The three additional $\mathbb{Z}_p$ elements are the hash value β, and values in the third flow required for (i) showing knowledge of s and (ii) linearizing a quadratic equation in the signature verification.

The instantiation of our framework achieves communication size of 303 Byte and signature size of 447 Byte.

Second Construction. We explain our second construction relying on randomizable signatures with an ABO reduction.

Getting Rid of NIZKs in the Signature. While the previous construction provides a small sum of signature and communication sizes, one drawback is that the blind signature has inherently a larger signature than those of the underlying signature scheme. The source of this large blind signature stems from using an NIZK to hide the underlying signature provided by the signer.

A natural approach used in the literature is to rely on techniques used to construct *randomizable* signature schemes [10, 31, 30, 42]. Informally, a randomizable signature scheme allows to publicly randomize the signature σ on a message m to a fresh signature σ'. Many standard group-based signature schemes (in the standard model and ROM) are known to satisfy this property, *e.g.*, [12, 56]. A failed attempt would be for the user to randomize the signature σ provided by the signer and output the randomized σ' as the blind signature. Clearly, this is not secure since the user is not hiding the message m, that is, σ and σ' are linkable through m thus breaking blindness. An idea to fix this would be to let the user send a commitment $c = \mathsf{Com}(m; r)$ to the signer and the signature signs the "message" c. However, unless the commitment c can be randomized consistently with σ, we would still need to rely on an NIZK to hide c. This calls for a signature scheme that is somehow compatible with commitments.

Signatures with All-But-One Reductions. Our main insight is that a specific class of signature schemes with an *all-but-one* (ABO) reduction is naturally compatible with blind signatures. An ABO reduction is a standard proof technique to prove selective security of public key primitives, *e.g.*, [13, 52, 34, 3], where a formal treatment can be found in [47]. In the context of signature schemes, this

is a proof technique that allows the reduction to embed the challenge message m^* (i.e., the signature for which the adversary forges) into the verification key. The reduction can simulate any signatures on $m \neq m^*$, and when the adversary outputs a forgery on m^*, then the reduction can break some hard problems.

Let us now specify the class of signature scheme. We assume an additive homomorphic commitment scheme, that is, $\mathsf{Com}(m;r) + \mathsf{Com}(m';r') = \mathsf{Com}(m + m'; r + r')$. We then assume a signature scheme where the signing algorithm $\mathsf{Sig}(\mathsf{sk}, m)$ can be rewritten as $\widehat{\mathsf{Sig}}(\mathsf{sk}, \mathsf{Com}(m;0) + u)$, where u is some fixed but random commitment included in the verification key. Namely, Sig first commits to the message m using no randomness, adds u to it, and proceeds with signing. Note that if $u = \mathsf{Com}(-m';r')$ for some (m', r'), then $\mathsf{Com}(m;0)+u = \mathsf{Com}(m-m';r')$. While contrived at first glance, this property is naturally satisfied by many of the signature schemes that admit an ABO reduction; the ABO reduction inherently requires embedding the challenge message m^* into the verification key in an unnoticeable manner and further implicitly requires message m submitted to the signing query to interact with the "committed" m^*. Specifically, the former hints at a need for an (implicit) commitment scheme and the later hints at the need for some operation between the commitments. Finally, to be used in the security proof, we assume there is a simulated signing algorithm $\widehat{\mathsf{SimSig}}$ along with a trapdoor td such that $\widehat{\mathsf{SimSig}}(\mathsf{td}, \mathsf{Com}(m-m';r'), m-m', r') = \widehat{\mathsf{Sig}}(\mathsf{sk}, \mathsf{Com}(m;0)+u)$ if and only if $m \neq m'$, where recall $u = \mathsf{Com}(-m';r')$. Specifically, $\widehat{\mathsf{SimSig}}$ can produce a valid signature if it knows the *non-zero* commitment message and randomness.

Let us explain the ABO reduction in slightly more detail. In the security proof, the reduction guesses (or the adversary $\mathcal{A}$ submits) a challenge message m^* that $\mathcal{A}$ will forge on. It then sets up the verification key while replacing the random commitment u to $u = \mathsf{Com}(-m^*;r^*)$ while also embedding a hard problem that it needs to solve. Due to the hiding property of the commitment scheme, this is unnoticeable from $\mathcal{A}$. Then, instead of using the real signing algorithm $\widehat{\mathsf{Sig}}$, the reduction uses the simulated signing algorithm $\widehat{\mathsf{SimSig}}$. As long as $m \neq m^*$, $\widehat{\mathsf{SimSig}}(\mathsf{td}, \mathsf{Com}(m-m^*;r^*), m-m^*, r^*)$ outputs a valid signature, and hence, can be used to simulate the signing oracle. Finally, given a forgery on m^*, the reduction is set up so that it can break a hard problem.

Turning it into a Blind Signature. To turn this into a blind signature, the key observation is that $\widehat{\mathsf{Sig}}$ is agnostic to the committed message and randomness of $\mathsf{Com}(m;0) + u$ — these are only used during the security proof when running $\widehat{\mathsf{SimSig}}$. Concretely, a user of a blind signature can generate a valid commitment $\mathsf{Com}(m;r)$, send it to the signer, and the signer can simply return $\sigma_r \leftarrow \widehat{\mathsf{Sig}}(\mathsf{sk}, \mathsf{Com}(m;r)+u)$. If the signature admits a way to map σ_r back to a normal signature σ for m, then we can further rely on the randomizability of the signature scheme to obtain a fresh signature σ' on the message m.

The proof of one-more unforgeability of this abstract blind signature construction is almost identical to the original ABO reduction with one exception. For the reduction to invoke the simulated $\widehat{\mathsf{SimSig}}$, recall it needs to know the message and randomness of the commitment $\mathsf{Com}(m;r) + u$. Hence, we modify the user

to add an *online-extractable* NIZK to prove the correctness of the commitment $\mathsf{Com}(m;r)$ so that the reduction can extract (m,r). Here, we require online-extractability rather than rewinding-extractability since otherwise, the reduction will run exponentially in the number of singing queries [53, 9]. Also, this is why the communication size becomes larger compared with our first construction. Finally, when the adversary outputs a forgery including m^*, the reduction can break a hard problem as before. Here, we note that we can simply hash the messages m with a random oracle to obtain an adaptively secure scheme using the ABO reduction.

Interestingly, while the recent lattice-based blind signature by del Pino and Katsumata [25] stated to use lattice-tailored techniques to optimize Fischlin's generic construction, the construction and the proof of one-more unforgeability follows our above template, where they use the Agrawal-Boneh-Boyen signature [3] admitting an ABO reduction. The only difference is that since lattices do not have nice randomizable signatures, they still had to rely on an NIZK for the final signature. While we focused on ABO reductions where only one challenge message m^* can be embedded in the verification key, the same idea naturally extends to all-but-*many* reductions. The blind signature by Blazy et al. [10] relying on the Waters signature can be viewed as one such instantiation. Finally, while we believe we can make the above approach formal using the ABO reduction terminology defined in [47], we focus on one class of instantiation in the main body for better readability. Nonetheless, we believe the above abstract construction will be useful when constructing round-optimal blind signatures from other assumptions.

Instantiation. We instantiate the above framework with the Boneh-Boyen signature scheme S_{BB} [12, 14]. Recall that signatures of S_{BB} on a message $m \in \mathbb{Z}_p$ are of the form $\sigma = (\mathsf{sk} \cdot (u_1^m \cdot h_1)^r, g_1^r)$, where $u_1, h_1 \in \mathbb{G}_1$ are part of the verification key, sk is the secret key and $r \leftarrow \mathbb{Z}_p$ is sampled at random. We observe that S_{BB} is compatible with the Pedersen commitment scheme $\mathsf{C}_{\mathrm{Ped}}$ with generators u_1 and g_1. Roughly, the user commits to the message m via $c = u_1^m \cdot g_1^s$[11], where $s \leftarrow \mathbb{Z}_p$ blinds the message, proves that she committed to m honestly with a proof π generated via an appropriate online-extractable NIZK Π, and sends (c, π) to the signer. The signer checks π and signs c via $(\mu_0, \mu_1) \leftarrow (\mathsf{sk} \cdot (c \cdot h_1)^r, g_1^r)$. Note that as c shares the structure u_1^m with S_{BB} signatures on message m, the user can recompute a valid signature on m via $\sigma \leftarrow (\mu_0 \cdot \mu_1^{-s}, \mu_1)$. Before presenting σ to a verifier, the user rerandomizes σ to ensure blindness. We refer to section 4 for more details.

The main challenge is constructing an efficient *online-extractable* NIZK Π for the relation $\mathsf{R}_{\mathsf{bb}} = \{(x,w) : c = u_1^m \cdot g_1^s\}$, where $x = (c, u_1, g_1)$ and $w = (m, s)$. As we require online-extraction, a simple Σ-protocol showing $c = u_1^m \cdot g_1^s$ compiled via Fiat-Shamir is no longer sufficient as in our prior instantiation, as the extractor needs to rewind the adversary in order to extract (m, s). For example, we could instantiate Π with the (online-extractable) GOS proofs but such a proof has a size of around 400 KB. Another well-known approach is to additionally encrypt

[11] In the actual construction, we further hash m by a random oracle; this effectively makes S_{BB} *adaptively* secure.

the witness (m, s) via a PKE and include the ciphertext into the relation; recall this method was used when explaining the Fischlin blind signature. The extractor can then use the secret key to decrypt the witnesses *online*. While a common choice for the PKE would be ElGamal encryption, this is insufficient since the extractor can only decrypt group elements g_1^m and g_1^s and not the witness in $\mathbb{Z}_p$ as required. To circumvent this, a common technique is to instead encrypt the binary decompositions $(m_i, s_i)_{i\in[\ell_2]}$ of m, s, respectively, with ElGamal, where $\ell_2 = \log_2(p)$. It then proves with a (non-online extractable) NIZK that $m = \sum_{i=1}^{\ell_2} m_i 2^{i-1}$ and $s = \sum_{i=1}^{\ell_2} s_i 2^{i-1}$ are valid openings of c, while also proving that m_i, s_i encrypted in the ElGamal ciphertexts are elements in $\{0, 1\}$, where the latter can be done via the equivalent identity $x \cdot (1 - x) = 0$. The extractor can now decrypt the ElGamal encryptions of m_i to $g_1^{m_i} \in \{g_1, 1_{\mathbb{G}_1}\}$ and efficiently decide whether m_i is 0 or 1. Similarly, it can recover the decomposition s_i. Unfortunately, this approach requires at least $2\ell_2$ ElGamal ciphertexts which amount to 32 KB alone. In fact, the bit-by-bit encryption of the witness is also the efficiency bottleneck of GOS proofs for $\mathbb{Z}_p$ witnesses.

We refine the above approach in multiple ways to obtain concretely efficient online-extractable NIZKs. Instead of using the binary decomposition, we observe that the extractor can still recover x from g_1^x if $x \in [0, B-1]$ is short, i.e., $B = \mathsf{poly}(\lambda)$. Thus, we let the prover encrypt the B-ary decompositions $(m_i, s_i)_{i\in[\ell]}$ of m and s, where $\ell = \log_B(p)$. For example, setting $B = 2^{32}$ allows the extractor to recover m_i via a brute-force calculation of the discrete logarithm, and the number of encryptions is reduced by a factor of 32. Concretely, we modify the prover to prove that an ElGamal ciphertext encrypts $(m_i, s_i)_{i\in[\ell]}$ such that (i) each m_i and s_i are in $[0, B-1]$, and (ii) $m = \sum_{i=1}^{\ell} m_i B^{i-1}$, $s = \sum_{i=1}^{\ell} s_i B^{i-1}$, and $c = u_1^m \cdot g_1^s$.

To instantiate our approach, we glue two different (non-online extractable) NIZKs Π_{rp} and Π_{ped} together, each being suitable to show relations (i) and (ii), respectively. For the range relation (i), we appeal to the batched variant of Bulletproofs [4] and turn it non-interactive with Fiat-Shamir. For the linear relation (ii), we use a standard NIZK with an appropriate Σ-protocol compiled with Fiat-Shamir. We further apply three optimizations to make this composition of NIZKs more efficient:

1. While Bulletproofs require committing to the decompositions $(m_i, s_i)_{i\in[\ell]}$ in Pedersen commitments, we use the shared structure of ElGamal ciphertexts and Pedersen commitments to avoid sending additional Pedersen commitments. This also makes the relation simpler since we do not have to prove consistency between the committed components in the ElGamal ciphertext and Pedersen commitment.
2. We use a more efficient discrete logarithm algorithm during extraction with runtime $\mathcal{O}(\sqrt{B})$, which allows us to choose more efficient parameters for the same level of security. This further reduces the number of encryptions by a factor 2.

3. We perform most of the proof in a more efficient elliptic curve $\widehat{\mathbb{G}}$ of same order p without pairing structure. As both the NIZKs Π_{rp} and Π_{ped} are not reliant on pairings, this reduces the size and efficiency of the NIZK considerably.

Proof of Instantiation. Finally, we analyze the security of the optimized online-extractable NIZK Π obtained by gluing Π_{rp} and Π_{ped} together. Correctness and zero-knowledge are straightforward. Also, online-extraction seems immediate on first sight. The extractor decrypts the decomposition, reconstructs the witness (m, s), and checks whether $c = u_1^m g_1^s$. To show why it works, we rely on the soundness of the range proof Π_{rp} to guarantee that the committed values are short. This allows the extractor to decrypt efficiently. Moreover, we rely on the soundness of Π_{ped} to guarantee that the decrypted values form a proper B-ary decompositions of an opening (m, s) of c. However, this high-level idea misses many subtle issues.

First, Bulletproofs are not well-established in the non-interactive setting in the ROM. While Attema et al. [5] show that special sound multi-round proof systems are knowledge sound (or rewinding-extractable) when compiled via Fiat-Shamir, Bulletproofs are only *computationally* special sound under the DLOG assumption. An easy fix for this is to relax the relation of the extracted witness. That is we use two different relations: one to be used by the prover and the other to be used by the extractor. We define an extracted witness w to be in the relaxed relation if either w is in the original relation *or* w is a DLOG solution with respect to (part of) the statement. With this relaxation, the interactive Bulletproofs becomes special sound for the relaxed relation since we can count the extracted DLOG solution as a valid witness. Observing that the result of [5] naturally translates to relaxed relations, we can conclude the non-interactive Bulletproofs to be rewinding-extractable in the ROM.

The second subtlety is more technical. For the formal proof, when the adversary submits a proof such that the online-extraction of Π fails, we must show that the adversary is breaking either the soundness of the underlying NIZKs Π_{rp} or Π_{ped}. Recall that Π_{rp} and Π_{ped} are glued together via the ElGamal ciphertext (cf. item 1). Specifically, each witness $w \in (m_i, s_i)_{i\in[\ell]}$ are encrypted as $c = (c_0, c_1) = (g^w \mathsf{pp}^r, g^r)$ with randomness $r \leftarrow \mathbb{Z}_p$, and Π_{rp} uses the partial "Pedersen part" c_0, while Π_{ped} uses the entire "ElGamal part" c. Thus one possibility for the online-extraction of Π failing is when the adversary breaks the tie between the two NIZKs by breaking the binding property of the Pedersen commitment. That is, if the adversary finds the DLOG between (g, pp), it can break the consistency between the two NIZKs in such a way that online-extraction of Π fails.

Put differently, to show that no adversary can trigger a proof for which the online-extraction of Π fails, we must show (at the minimum) that we can use such an adversary to extract a DLOG solution between (g, pp). This in particular implies that we have to *simultaneously* extract the witness w_0 of Π_{rp} containing one opening of c_0 and the witness w_1 of Π_{ped} containing the other opening of c_0 in order to break DLOG with respect to (g, pp), or equivalently to break the binding property of the Pedersen commitment. The issue with this is that we cannot conclude that both extractions succeed at the same time even if Π_{rp} and Π_{ped}

individually satisfy the standard notion of rewinding-extractability. For instance, using the standard notion of rewinding-extractability, we cannot exclude the case where the adversary sets up the proofs π_0, π_1 of $\Pi_{\mathsf{rp}}, \Pi_{\mathsf{ped}}$, respectively, in such a way that if the extractor of Π_{rp} succeeds, then the extractor of Π_{ped} fails. We thus show in a careful non-black box analysis that the extraction of both proofs succeeds at the same time with non-negligible probability. To the best of our knowledge, this is the first time an NIZK that internally uses Bulletproofs is proven to be online-extractable in the ROM. We believe that our new analysis is of independent interest.

2 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter. We use standard notations for probability, algortihms and distributions. Also, we use prime order groups $\mathbb{G}$ and pairing groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ of shared order p, with standard notation. We refer to the full version for more details. We denote with $[n]$ the set $\{1, \ldots, n\}$ for $n \in \mathbb{N}$. For any $\vec{h} = (h_1, \ldots, h_q)$ and $i \in [q]$, we denote $\vec{h}_{<i}$ as $(h_1, \ldots, h_{i-1})$ and $\vec{h}_{\geq i}$ as $(h_i, \ldots, h_q)$, where $\vec{h}_{<1}$ denotes an empty vector. Moreover, for any two vectors $\vec{h}$, $\vec{h'}$ of arbitrary length, we use $\vec{h} \| \vec{h'}$ to denote the concatenation of the two vectors. In particular, for any $i \in [q]$ and $\vec{h} \in \mathcal{H}^q$, we have $\vec{h} = \vec{h}_{<i} \| \vec{h}_{\geq i}$.

Instantiation. For our instantiations, we assume that the modulus p is of size 256 bit, and an element of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ is of size $382, 763, 4572$ bit, respectively. These are common sizes of standard BLS curves [6] with security parameter $\lambda = 128$, in particular BLS12-381 [15]. For groups that require no pairing operation, we use a curve of order p and assume that elements are of size 256 bit. We generally write $\widehat{\mathbb{G}}$ for such groups.

Assumptions. In this paper, we use the following hardness assumptions. Let $\mathbb{G}$ be an arbitrary group with generator g and $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2) \leftarrow \mathsf{PGen}(1^\lambda)$ be a pairing description.

The discrete logarithm (DLOG) assumption in $\mathbb{G}$ states that it is hard to compute the discrete logarithm x of some random $h = g^x \in \mathbb{G}$. The decisional Diffie-Hellman (DDH) assumption states that it is hard to distinguish tuples (g^a, g^b, g^{ab}) from tuples (g^a, g^b, g^c) with random $a, b, c \leftarrow \mathbb{Z}_p$. The symmetric external Diffie-Hellman (SXDH) assumption holds if the DDH assumption holds in $\mathbb{G}_1$ and in $\mathbb{G}_2$. Finally, the (asymmetric) computational Diffie-Hellman assumption states that given $(g_1^a, g_2^a, g_1^b, g_2^b)$, it is hard to computes g_1^{ab}.

Explaining Group Elements as Random Strings. Our frameworks generally require that public parameters pp (of commitment schemes) and common random strings crs (of NIZKs) are random bit strings. For readability, we allow that pp and crs contain random group elements $g \leftarrow \mathbb{G}$ for some group $\mathbb{G}$. This is without

loss of generality, as using explainable sampling, we can explain these elements as random strings. We refer to the full version for more details.

2.1 Cryptographic Primitives

We briefly recall the primitives we use throughout the article, and refer to the full version for formal definitions.

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

- $\mathsf{C.Commit}(\mathsf{pp}, m; r)$: given the public parameters $\mathsf{pp} \in \{0,1\}^{\ell_\mathsf{C}}$, message m and randomness r, computes a commitment c, and outputs the pair (c, r),

where $\mathsf{pp} \in \{0,1\}^{\ell_\mathsf{C}}$ are *uniform* public parameters, r is the randomness and c is the commitment. We do not explicitly define the opening algorithm since we can use the commitment randomness r as the decommitment (or opening) information and check if $c = \mathsf{Commit}(\mathsf{pp}, m; r)$ holds to verify that c is a valid commitment to message m.

We require the standard notions of correctness, hiding and binding. A commitment scheme is *correct* if honest commitments $c \leftarrow \mathsf{Commit}(\mathsf{pp}, m; r)$ always verify, i.e., $c = \mathsf{Commit}(\mathsf{pp}, m; r)$. 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 further say that $c = \mathsf{Commit}(\mathsf{pp}, m; r)$ is rerandomizable, if it can be rerandomized via $c' \leftarrow \mathsf{RerandCom}(\mathsf{pp}, c, \Delta r)$. We require that the new commitment c' has high min-entropy if Δr is a fresh random value, i.e., given c it is *statistically* difficult to predict c'. Also, we assume that we can recover an opening of c' via $r' \leftarrow \mathsf{RerandRand}(\mathsf{pp}, c, m, r, \Delta r)$ if an initial opening (m, r) of c and the rerandomization randomness Δr is known. That is, if compute $c' = \mathsf{RerandCom}(\mathsf{pp}, c, \Delta r)$ and $r' \leftarrow \mathsf{RerandRand}(\mathsf{pp}, c, m, r, \Delta r)$, then it holds that $c' = c''$, where $(c'', r') = \mathsf{Com}(\mathsf{pp}, m; r')$.

We note that any natural additive homomorphic commitment scheme satisfies rerandomizability if we define $\mathsf{RerandCom}(\mathsf{pp}, c, \Delta r) = c + \mathsf{Commit}(\mathsf{pp}, 0; \Delta r) = c'$. Observe that if $c = \mathsf{Commit}(\mathsf{pp}, m; r)$, the rerandomzied randomness is $r' = r + \Delta r$ since $c' = \mathsf{Commit}(\mathsf{pp}, m; r')$ by the homomorphic property. Moreover, c' has high min-entropy since $\mathsf{Commit}(\mathsf{pp}, 0)$ has high min-entropy for most natural commitment schemes. Finally, we note that while a computational variant of the high min-entropy property suffices for our generic construction, we use the statistical variant for simplicity and because our instantiation satisfies it.

Signature Schemes. We consider *deterministic* signature schemes; a scheme where the randomness of the signing algorithm is derived from the secret key and message. We can derandomize any signature scheme by using a pseudorandom function for generating the randomness used in the signing algorithm (see for example [40]). 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 vk 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 require the standard notion of correctness and `euf-cma` security. A signature scheme is *correct* if honestly generated signatures $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$ verify correctly, i.e., $\mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1$. It is *euf-cma* secure if given some vk and access to a signature oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$, it is hard to output a valid signature σ for some message m that was never queried to $\mathsf{Sign}(\mathsf{sk}, \cdot)$.

Blind Signature Scheme. We recall the definition of round-optimal blind signatures, and refer to the full version for more formal definitions of (partially) blind signatures. A blind signature scheme is a tuple of PPT algorithms $\mathsf{PBS} = (\mathsf{KeyGen}, \mathsf{User}, \mathsf{Signer}, \mathsf{Derive}, \mathsf{Verify})$ such that

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

In the following, we assume the state is kept implicit in the following for better readability. We consider the standard security notions for blind signatures [38].

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

It is *blind under malicious keys* if a malicious signer cannot distinguish whether it first signed m_0 or m_1, after engaging with a honest user in two signing sessions and being presented the obtained signatures on messages m_0, m_1 in a fixed order. Here, the honest user permutes the order of the signing sessions at random, and the verification key bvk is adversarially chosen.

It is *one-more unforgeable* if a malicious user that engages in at most Q_S signing sessions with the signer, can output at most Q_S valid distinct signature-message pairs.

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

- $\mathsf{Init}(x, w)$: given a statement $x \in \mathscr{L}_{\mathsf{R}}$, and a witness w such that $(x, w) \in \mathsf{R}$, outputs a first flow message (i.e., commitment) α and a state st, where we assume st includes x, w,

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

Here, $\mathcal{CH}$ denotes the challenge space. We call the tuple (α, β, γ) the *transcript* and say that they are *valid for* x if $\mathsf{Verify}(x, \alpha, \beta, \gamma)$ outputs 1. When the context is clear, we simply say it is valid and omit x.

We recall the standard notions of correctness, high-min entropy, honest-verifier zero-knowledge, and 2-special soundness. A Σ-protocol is *correct*, if for all $(x, w) \in \mathsf{R}$, if for any honestly generated transcripts (α, β, γ), the verifier accepts, i.e., $\mathsf{Verify}(x, \alpha, \beta, \gamma) = 1$. It has *high min-entropy* if for all $(x, w) \in \mathsf{R}$, it is statistically hard to predict a honestly generated first flow α. It is *honest-verifier zero-knowledge* (HVZK), if there exists a PPT zero-knowledge simulator Sim such that the distributions of $\mathsf{Sim}(x, \beta)$ and the honestly generated transcript with Init initialized with (x, w) are computationally indistinguishable for any $x \in \mathscr{L}_{\mathsf{R}}$, and $\beta \in \mathcal{CH}$, where the honest execution is conditioned on β being used as the challenge. Finally, it is 2-*special sound*, if there exists a *deterministic* PPT extractor Ext such that given two valid transcripts $\{(\alpha, \beta_b, \gamma_b)\}_{b \in [2]}$ for statement x with $\beta_1 \neq \beta_2$, along with x, outputs a witness w such that $(x, w) \in \mathsf{R}$.

Note that in the above, two valid transcripts for x, with the same first flow message and different challenges, imply that statement x is in $\mathscr{L}_{\mathsf{R}}$. That is, we do not guarantee x to lie in $\mathscr{L}_{\mathsf{R}}$ when invoking Ext. While subtle, this allows us to invoke Ext properly within the security proof even if the reduction cannot decide if the statement x output by the adversary indeed lies in $\mathscr{L}_{\mathsf{R}}$.

In the following, we propose a new notion of *f-unique extraction*. The notion is similar to the *unique response* property [27, 54] which requires that given an incomplete transcript (α, β), there is at most one response γ such that the transcript $\tau = (\alpha, \beta, \gamma)$ is valid. We relax this in two ways. First, we require that given a transcript τ and another challenge β', it is impossible to find two different responses γ_0, γ_1, such $w_0 \neq w_1$, where w_b is the witness extracted from τ and $\tau_b = (\alpha, \beta', \gamma_b)$. We further relax this by only requiring this property for a portion of the witness, defined by a function f, i.e., we require $f(w_0) \neq f(w_1)$ instead of $w_0 \neq w_1$.

While it may seem like an unnatural property, this is satisfied by many natural Σ-protocols. In particular, if the first flow α contains a perfectly binding commitment $c = \mathsf{Commit}(f(w); r)$ to $f(w)$, and the extractor extracts the appropriate r, then the Σ-protocol has f-unique extraction. We remark also that a statistical variant of f-unique extraction is sufficient for our purpose. We choose the definition below for simplicity and because our instantiation satisfies it. See section 3 for more details and concrete example of f-unique extraction.

Definition 1 (f-Unique Extraction). *For a (possibly non-efficient) function f, a Σ-protocol Σ has* f-unique extraction *if for any statement x, any transcript $\tau = (\alpha, \beta, \gamma)$ and challenge $\beta' \neq \beta$, there is no γ_0, γ_1, such that for $\tau_b = (\alpha, \beta', \gamma_b)$,*

we have

$$f(\mathsf{Ext}(x,\tau,\tau_0)) \neq f(\mathsf{Ext}(x,\tau,\tau_1)).$$

Non-Interactive Zero Knowledge. Given a witness w for statement x, a non-interactive zero-knowledge (NIZK) proof system allows a prover to generate a proof π that attests that she *knows* some w' such that $(w',x) \in \mathsf{R}$. Proofs π can be verified for statement x *without* revealing anything but that the statement is true. Here, we quantify "knowledge of the witness" either via *adaptive knowledge soundness* or *online-extractability*. The former informally states that if an algorithm $\mathcal{A}$ can generate a valid proof-statement pair (x,π), then there exists some extractor that when given black-box access to $\mathcal{A}$, can extract some witness w s.t. $(x,w) \in \mathsf{R}$. The latter requires that the witness w can be extracted from (x,π) "on-the-fly" without disrupting $\mathcal{A}$. In this context, we require some random oracle H on which proving and verification rely. Further, we assume that the prover and verifier are supplied with a common random string crs. As we later aim to avoid such a crs in our blind signature framework, the crs will be the output of a random oracle.

More formally, an NIZK for a relation R is a tuple of oracle-calling PPT algorithms $(\mathsf{Prove}^{\mathsf{H}}, \mathsf{Verify}^{\mathsf{H}})$ such that:

- $\mathsf{Prove}^{\mathsf{H}}(\mathsf{crs},x,w)$: receives a common random string $\mathsf{crs} \in \{0,1\}^{\ell}$, a statement x and a witness w, and outputs a proof π,
- $\mathsf{Verify}^{\mathsf{H}}(\mathsf{crs},x,\pi)$: receives a statement x and a proof π, and outputs a bit $b \in \{0,1\}$.

Here, ℓ is the length of common random strings. An NIZK is *correct* if for any $\mathsf{crs} \in \{0,1\}^{\ell}$, $(x,w) \in \mathsf{R}$, 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 $\mathsf{Sim} = (\mathsf{Sim}_{\mathsf{H}}, \mathsf{Sim}_{\pi})$ that outputs simulated proofs $\pi' \leftarrow \mathsf{Sim}_{\pi}(\mathsf{crs},x)$ that are indistinguishable from real proofs $\pi \leftarrow \mathsf{Prove}^{\mathsf{H}}(\mathsf{crs},x,w)$ that are generated with witness w such that $(x,w) \in \mathsf{R}$. Here, $\mathsf{Sim}_{\mathsf{H}}$ simulates the random oracle H for simulated proofs.

We define adaptive knowledge soundness. We remark that the soundness relation $\mathsf{R}_{\mathsf{lax}}$ can be different from the (correctness) relation R. We are typically interested in $\mathsf{R} \subseteq \mathsf{R}_{\mathsf{lax}}$ and call $\mathsf{R}_{\mathsf{lax}}$ the *relaxed* relation.

An NIZK is *adaptively knowledge sound* for relation $\mathsf{R}_{\mathsf{lax}}$ if there exists a PPT algorithm Ext such that for any $\mathsf{crs} \in \{0,1\}^{\ell}$, given oracle access to any PPT adversary $\mathcal{A}$ (with explicit random tape ρ) that makes $Q_H = \mathsf{poly}(\lambda)$ random oracle queries, then for $(x,\pi) \leftarrow \mathcal{A}^{\mathsf{H}}(\mathsf{crs};\rho)$, the extractor finds some $w \leftarrow \mathsf{Ext}(\mathsf{crs},x,\pi,\rho,\vec{h})$ with $(x,w) \in \mathsf{R}_{\mathsf{lax}}$ with probability at least $\frac{\mu(\lambda)-\mathrm{negl}(\lambda)}{\mathsf{poly}(\lambda)}$. Here, $\mu(\lambda)$ is the probability that $\mathcal{A}$ outputs valid pairs (x,π) and $\vec{h}$ are the random oracle outputs in the run of $\mathcal{A}$.

An NIZK is *online-extractable* if for all PPT adversaries $\mathcal{A}$, there exists a PPT simulator SimCRS that outputs a trapdoor td and simulated $\overline{\mathsf{crs}}$ that is indistinguishable from some random $\mathsf{crs} \leftarrow \{0,1\}^{\ell}$, and a PPT extractor Ext, such that for any $Q_H = \mathsf{poly}(\lambda)$ and PPT adversary $\mathcal{A}$ that on input $\overline{\mathsf{crs}}$

makes at most Q_H random oracle queries and outputs statement-proof pairs $\{(x_i, \pi_i)\}_{i\in[Q_S]} \leftarrow \mathcal{A}^{\mathsf{H}}(\overline{\mathsf{crs}})$, Ext outputs $w_i \leftarrow \mathsf{Ext}(\overline{\mathsf{crs}}, \mathsf{td}, x_i, \pi_i)$ such that for all i it holds that $(x_i, w_i) \in \mathsf{R}$, and all proofs verify, with probability at least $\frac{\mu(\lambda)-\mathrm{negl}(\lambda)}{\mathsf{poly}(\lambda)}$. Here, $\mu(\lambda)$ denotes the probability that the proofs output by $\mathcal{A}$ verify correctly.

3 Optimizing the Fischlin Blind Signature

In this section, we provide an optimized generic construction of blind signatures compared with the Fischlin blind signature [28]. In particular, we relax the extractable (and perfect binding) commitment and multi-online extractable NIZK used as the central building block for the Fischlin blind signature by a computationally binding commitment and a standard rewinding-based NIZK built from a Σ-protocol satisfying f-unique extraction. As we show in Section 3.3, this relaxation allows us to minimize the sum of the communication and signature size. We construct a natural partially blind variant in the full version.

3.1 Construction

Our generic construction is based on the building blocks $(\mathsf{C}, \mathsf{S}, \Sigma)$ that satisfy some specific requirements. If $(\mathsf{C}, \mathsf{S}, \Sigma)$ satisfies these requirements, then we call it $\mathsf{BS}_{\mathsf{Rnd}}$-*suitable*.

Definition 2 ($\mathsf{BS}_{\mathsf{Rnd}}$-Suitable $(\mathsf{C}, \mathsf{S}, \Sigma)$). *The tuple of schemes $(\mathsf{C}, \mathsf{S}, \Sigma)$ are called* $\mathsf{BS}_{\mathsf{Rnd}}$-suitable*, if it holds that*

- C *is a correct and hiding* rerandomizable *commitment scheme with public parameter, message, randomness, and commitment spaces $\{0,1\}^{\ell_{\mathsf{C}}}, \mathcal{C}_{\mathsf{msg}}, \mathcal{C}_{\mathsf{rnd}}$, and $\mathcal{C}_{\mathsf{com}}$, respectively, such that $\mathcal{C}_{\mathsf{msg}}$ is efficiently sampleable and $1/|\mathcal{C}_{\mathsf{msg}}| = \mathrm{negl}(\lambda)$,*
- S *is a correct and* $\mathsf{euf\text{-}cma}$ *secure* deterministic *signature scheme with message space $\mathcal{S}_{\mathsf{msg}}$ that contains $\mathcal{C}_{\mathsf{com}}$, i.e., $\mathcal{C}_{\mathsf{com}} \subseteq \mathcal{S}_{\mathsf{msg}}$ and we assume elements in $\mathcal{S}_{\mathsf{msg}}$ are efficiently checkable,*
- Σ *is a correct, HVZK, 2-special sound Σ-protocol with high min-entropy, and challenge space $\mathcal{CH}$ with $1/|\mathcal{CH}| = \mathrm{negl}(\lambda)$ for the relation*

$$\mathsf{R}_{\mathsf{rnd}} := \{x = (\mathsf{pp}, \mathsf{vk}, \overline{m}), w = (\mu, c, r) \mid \mathsf{C.Commit}(\mathsf{pp}, \overline{m}; r) = (c, r) \ \wedge \ \mathsf{S.Verify}(\mathsf{vk}, \mu, c) = 1\}.$$

We also require Σ to be f-unique extraction where $f(w) = c$, i.e., f outputs c and ignores (μ, r).

Let $(\mathsf{C}, \mathsf{S}, \Sigma)$ be $\mathsf{BS}_{\mathsf{Rnd}}$-suitable. Let $\mathsf{H}_{\mathsf{par}}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_{\beta}$ be a random oracles from $\{0,1\}^*$ into $\{0,1\}^{\ell_{\mathsf{C}}}, \mathcal{C}_{\mathsf{msg}}, \mathcal{CH}$, respectively.

Construction. We present our blind signature $\mathsf{BS}_{\mathsf{Rnd}}$. Below, we assume that the verification key *implicitly* specifies the public parameter pp for C via $\mathsf{pp} = \mathsf{H}_{\mathsf{par}}(0)$. We assume pp is provided to all of the algorithms for readability.

- $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{KeyGen}(1^\lambda)$: samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{S}.\mathsf{KeyGen}(1^\lambda)$ and outputs verification key $\mathsf{bvk} = \mathsf{vk}$ and signing key $\mathsf{bsk} = \mathsf{sk}$.
- $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{User}(\mathsf{bvk}, m)$: sets $\overline{m} \leftarrow \mathsf{H}_{\mathsf{M}}(m)$ and outputs the commitment $c \in \mathcal{C}_{\mathsf{com}}$ generated via $(c, r) \leftarrow \mathsf{C}.\mathsf{Commit}(\mathsf{pp}, \overline{m})$ as the first message and stores the randomness $\mathsf{st} = r \in \mathcal{C}_{\mathsf{rnd}}$.
- $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{Signer}(\mathsf{bsk}, c)$: checks if $c \in \mathcal{C}_{\mathsf{com}}$, samples a rerandomization randomness $\Delta r \leftarrow \mathcal{C}_{\mathsf{rnd}}$, rerandomizes the commitment c via $c' = \mathsf{C}.\mathsf{RerandCom}(\mathsf{pp}, c, \Delta r)$, signs $\mu \leftarrow \mathsf{S}.\mathsf{Sign}(\mathsf{sk}, c')$, and finally outputs the second message $\rho = (\mu, \Delta r)$.
- $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{Derive}(\mathsf{st}, \rho)$: parse $\mathsf{st} = r$, $\rho = (\mu, \Delta r)$ and checks $\Delta r \in \mathcal{C}_{\mathsf{rnd}}$. It then computes the randomized commitment $c'' = \mathsf{C}.\mathsf{RerandCom}(\mathsf{pp}, c, \Delta r)$ and randomized randomness $r' \leftarrow \mathsf{C}.\mathsf{RerandRand}(\mathsf{pp}, c, \overline{m}, r, \Delta r)$, and checks $\mathsf{S}.\mathsf{Verify}(\mathsf{vk}, c'', \mu) = 1$ and $c'' = \mathsf{C}.\mathsf{Commit}(\mathsf{pp}, \overline{m}; r')$. Finally, it outputs a signature $\sigma = \pi$, where $(\alpha, \mathsf{st}') \leftarrow \Sigma.\mathsf{Init}(x, w)$, $\beta \leftarrow \mathsf{H}_\beta(x, \alpha), \gamma \leftarrow \Sigma.\mathsf{Resp}(x, \mathsf{st}', \beta)$, $\pi = (\alpha, \beta, \gamma)$ with $x = (\mathsf{pp}, \mathsf{vk}, \overline{m})$, $w = (\mu, c'', r')$.
- $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{Verify}(\mathsf{bvk}, m, \sigma)$: parses $\sigma = \pi$ and $\pi = (\alpha, \beta, \gamma)$, sets $\overline{m} = \mathsf{H}_{\mathsf{M}}(m)$ and $x = (\mathsf{pp}, \mathsf{vk}, \overline{m})$, and outputs 1 if $\beta = \mathsf{H}_\beta(x, \alpha)$, $\Sigma.\mathsf{Verify}(x, \alpha, \beta, \gamma) = 1$, and otherwise outputs 0.

3.2 Correctness and Security

The correctness of $\mathsf{BS}_{\mathsf{Rnd}}$ follows directly from the correctness of the underlying schemes $(\mathsf{C}, \mathsf{S}, \Sigma)$. Blindness follows mainly from the HVZK property of Σ and the hiding property of C. The only thing to be aware of is that the user needs to check the validity of the rerandomized commitment c'' by computing a rerandomized randomness using the randomness r used to compute the original commitment c. In order to invoke the hiding property of C on c, we rely on the correctness of the randomization property so that the reduction no longer needs to check the validity of c''.

The main technical challenge is the proof of one-more unforgeability. The proof is given below, for an overview see Section 1. We refer to the full version for proofs of correctness and blindness.

Theorem 1. *The blind signature* $\mathsf{BS}_{\mathsf{Rnd}}$ *is correct, blind under malicious keys and one-more unforgeable if the schemes* $(\mathsf{C}, \mathsf{S}, \Sigma)$ *are* $\mathsf{BS}_{\mathsf{Rnd}}$*-suitable.*

Proof. Let $\mathcal{A}$ be a PPT adversary against one-more unforgeability. Denote by Q_S the number of signing queries, by Q_M the number of H_{M} queries, and by Q_H the number of H_β queries. Recall that we model $\mathsf{H}_{\mathsf{par}}, \mathsf{H}_{\mathsf{M}}$, and H_β as random oracles, where we assume without loss of generality that $\mathcal{A}$ never repeats queries. In the end of the interaction with $\mathcal{A}$, that is after Q_S signing queries, $\mathcal{A}$ outputs $Q_S + 1$ forgeries $\{(m_i, \sigma_i)\}_{i \in [Q_S+1]}$. We write $\sigma_i = \pi_i$ and denote by c_i the Q_S first message queries to $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{Signer}(\mathsf{bsk}, \cdot)$ issued by $\mathcal{A}$. Note that if $\mathcal{A}$ is successful, then we have $\Sigma.\mathsf{Verify}(x_i, \alpha_i, \beta_i, \gamma_i) = 1$ and $\beta_i = \mathsf{H}_\beta(x_i, \alpha_i)$ for $\overline{m}_i = \mathsf{H}_{\mathsf{M}}(m_i)$, $x_i = (\mathsf{pp}, \mathsf{vk}, \overline{m}_i)$, and $\pi_i = (\alpha_i, \beta_i, \gamma_i)$. We first slightly alter the real game and remove subtle conditions to make the later proofs easier. We denote by $\mathsf{Adv}^{\mathrm{H}_i}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in Hybrid i for $i \in \{0, 1\}$.

- Hybrid 0 is identical to the real game.
- Hybrid 1 is the same as Hybrid 0, except it aborts if there is a collision in H_{M} or H_β, or there is some (x_i, α_i) for $i \in [Q_S + 1]$ that was never queried to H_β. It suffices to upper bound the abort probability. A collision in H_{M} (resp. H_β) happens with probability at most $Q_M^2/|\mathcal{C}_{\mathsf{msg}}|$ (resp. $Q_H^2/|\mathcal{CH}|$) (which follows for example from a union bound). Moreover, the probability that some fixed β_i of $\mathcal{A}$'s output equals to $\mathsf{H}_\beta(x_i, \alpha_i)$ is exactly $1/|\mathcal{CH}|$, if (x_i, α_i) was never queried to H_β. Thus, it follows that $\mathsf{Adv}^{\mathrm{H}_0}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{H}_1}_{\mathcal{A}}(\lambda) + \frac{Q_M^2}{|\mathcal{C}_{\mathsf{msg}}|} + \frac{Q_H^2+1}{|\mathcal{CH}|} = \mathsf{Adv}^{\mathrm{H}_1}_{\mathcal{A}}(\lambda) + \mathrm{negl}(\lambda)$.

Description of Wrapper Algorithm $\mathcal{B}$. We now present a wrapper algorithm $\mathcal{B}$ that simulates the interaction between the challenger $\mathcal{G}$ and $\mathcal{A}$ in Hybrid 1. Looking ahead we apply a generalization of the standard forking lemma on $\mathcal{B}$ to extract the witnesses from all the proof (i.e. forgery) output by $\mathcal{A}$.

Notice that $\mathcal{G}$ is deterministic once the keys $(\mathsf{vk}, \mathsf{sk})$ of the (deterministic) signature scheme S, the Q_S rerandomization randomness in $\mathcal{C}_{\mathsf{rnd}}$, and the outputs of the random oracles $\mathsf{H}_{\mathsf{par}}, \mathsf{H}_{\mathsf{M}}, \mathsf{H}_\beta$ are determined. Since $\mathsf{H}_{\mathsf{par}}$ is only used to generate the public parameter pp of the commitment scheme, we assume without loss of generality that only pp is given to $\mathcal{A}$ rather than access to $\mathsf{H}_{\mathsf{par}}$. We use coin to denote all the Q_M outputs of H_{M} and the random coins used by $\mathcal{A}$. We use $\vec{h} = (\widehat{\beta}_i, \Delta r_i)_{i \in [Q_H + Q_S]} \in (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H + Q_S}$ to explicitly denote the list that will be used to simulate the outputs of H_β and rerandomziation randomness sampled by $\mathcal{G}$. Here, we note that $\vec{h}$ is deliberately defined redundantly since $\mathcal{G}$ only needs Q_H hash outputs and Q_S rerandomziation randomness, rather than $Q_H + Q_S$ of them each. We also use $\widehat{\beta} \in \mathcal{CH}$ to denote the output of H_β to distinguish between the hash value β included in $\mathcal{A}$'s forgeries. We then define $\mathcal{B}$ as an algorithm that has oracle access to $\mathsf{S.Sign}(\mathsf{sk}, \cdot)$ as follows:

$\mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{pp}, \mathsf{vk}, \vec{h}; \mathsf{coin})$: On input pp, vk, and $\vec{h} \in (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H + Q_S}$, $\mathcal{B}$ simulates the interaction between the challenger $\mathcal{G}$ and $\mathcal{A}$ in Hybrid 1. $\mathcal{B}$ invokes $\mathcal{A}$ on the randomness included in coin and simulates $\mathcal{G}$, where it runs the same code as $\mathcal{G}$ except for the following differences:

- It uses the provided pp and vk rather than generating it on its own;
- All Q_M random oracle queries to H_{M} are answered using the hash values include in coin;
- On the i-th ($i \in [Q_H]$) random oracle query to H_β, it retrieves an unused $(\widehat{\beta}_k, \Delta r_k)$ with the smallest index $k \in [Q_H + Q_S]$ and outputs $\widehat{\beta}_k$ and discards Δr_k;
- On the i-th ($i \in [Q_S]$) first message $c_i \in \mathcal{C}_{\mathsf{com}}$ from $\mathcal{A}$, it retrieves an unused $(\widehat{\beta}_k, \Delta r_k)$ with the smallest index $k \in [Q_H + Q_S]$ and discards $\widehat{\beta}_k$. It then computes $c_i' = \mathsf{C.RerandCom}(\mathsf{pp}, c_i, \Delta r_k)$, queries the signing oracle on c_i', obtains $\mu_i \leftarrow \mathsf{S.Sign}(\mathsf{sk}, c_i')$, and returns the second message $\rho_i = (\mu_i, \Delta r_k)$.

At the end of the game when $\mathcal{A}$ outputs the forgeries, $\mathcal{B}$ checks if the forgeries are valid and the added condition in Hybrid 1. If the check does not pass,

Algorithm 1 Description of the forking algorithm $\mathsf{F}_{\mathcal{B}}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{pp},\mathsf{vk})$

Pick coin for $\mathcal{B}$ at random.
$\vec{h} \leftarrow (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S}$
$\left((I_i)_{i\in[Q_S+1]}, \Lambda\right) \leftarrow \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{pp},\mathsf{vk},\vec{h};\mathsf{coin})$
if $\Lambda = \bot$ **then**
  return $\bot$ ▷ Return fail.
$D := ()$ ▷ Prepare empty list.
for $j \in [Q_S+1]$ **do**
  $(c, \mathsf{flag}) := (1, \bot)$
  while $c \in [T] \wedge \neg\mathsf{flag}$ **do**
    $\vec{h}^{(c)}_{j,\geq I_j} \leftarrow (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S-I_j+1}$
    $\vec{h}^{(c)}_{j} := \vec{h}_{<I_j} \| \vec{h}^{(c)}_{j,\geq I_j}$
    $\left((I^{(c)}_{j,i})_{i\in[Q_S+1]}, \Lambda^{(c)}_j\right) \leftarrow \mathcal{B}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{pp},\mathsf{vk},\vec{h}^{(c)}_j;\mathsf{coin})$
    if $I^{(c)}_{j,j} = I_j$ **then**
      $D = D \cup (j, I_j, \Lambda^{(c)}_j)$
      $\mathsf{flag} = \top$ ▷ Break from while loop.
    $c = c + 1$
if $|D| < Q_S + 1$ **then** ▷ Check if $\mathcal{B}$ succeeds in all Q_S+1 run.
  return $\bot$ ▷ Return fail.
return (Λ, D)

then $\mathcal{B}$ outputs $((0)_{i\in[Q_S+1]}, \bot)$, i.e., Q_S+1 zeros followed by a $\bot$. Otherwise, $\mathcal{B}$ finds the indices $I_i \in [Q_H+Q_S]$ such that $\mathsf{H}_\beta(x_i,\alpha_i) = \beta_i = \widehat{\beta}_{I_i}$ for $i \in [Q_S+1]$, which are guaranteed to exist uniquely due to the modification we made in Hybrid 1. It then sets $\Lambda = (x_i, \alpha_i, \beta_i, \gamma_i)_{i\in[Q_S+1]}$ and outputs $((I_i)_{i\in[Q_S+1]}, \Lambda)$. It can be checked that $\mathcal{B}$ perfectly simulates the view of the challenger $\mathcal{G}$ in Hybrid 1. Therefore, $\mathcal{B}$ outputs $\Lambda \neq \bot$ with probability $\mathsf{Adv}^{\mathrm{H}_1}_{\mathcal{A}}(\lambda)$.

Description of Forking Algorithm $\mathsf{F}_{\mathcal{B}}$. We now define a generalization of the standard forking algorithm F so that F keeps on rewinding $\mathcal{B}$ until some condition is satisfied. Concretely, F takes as input $(\mathsf{pp},\mathsf{vk})$, has oracle access to $\mathsf{S.Sign}(\mathsf{sk},\cdot)$, and invokes $\mathcal{B}$ internally as depicted in algorithm 1, where the number of repetition T is defined below.

We show that if $\mathcal{A}$ succeeds in breaking one-more unforgeability in Hybrid 1 with non-negligible probability, then we can set a specific number of repetition T so that the forking algorithm $\mathsf{F}_{\mathcal{B}}$ terminates in polynomial time and succeeds in outputting a non-$\bot$ with non-negligible probability. Formally, we have the following lemma.

Lemma 1. *Let* $\epsilon = \mathsf{Adv}^{H_1}_{\mathcal{A}}(\lambda)$. *Then, if we set* $T = \left(\frac{\epsilon}{(Q_H+Q_S)(Q_S+2)^2}\right)^{-1} \cdot \log(2Q_S+2)$, $\mathsf{F}_{\mathcal{B}}$ *outputs a non-*$\bot$ *with probability at least* $\frac{\epsilon}{2(Q_S+2)^2}$.

In particular, if ϵ is non-negligible, then $T = \mathsf{poly}(\lambda)$. Moreover, the running time of $\mathsf{F}_{\mathcal{B}}$ is at most (roughly) a factor $T \cdot (Q_S+1)+1$ more of $\mathcal{B}$ (or equivalently $\mathcal{A}$), so $\mathsf{F}_{\mathcal{B}}$ runs in polynomial time.

Proof. Assume $\mathcal{B}$ outputs a valid $\Lambda = (x_i, \alpha_i, \beta_i, \gamma_i)_{i\in[Q_S+1]}$ in the first execution and denote this event as E. For $i \in [Q_S+1]$, we denote the tuple $(x_i, \alpha_i, \beta_i, \gamma_i)$ as the i-th *forgery*. For any $(i,k) \in [Q_S+1] \times [Q_H+Q_S]$, we denote $\mathsf{E}_{i,k}$ as the event that forgery is associated to the k-th hash query, i.e., the k-th entry of $\vec{h} \in (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S}$ includes β_i. Here, note that $\forall i \in [Q_S+1]$, we have $\sum_{k\in[Q_H+Q_S]} \Pr[\mathsf{E}_{i,k}] = 1$. We define the set P_i as

$$P_i = \left\{ k \;\middle|\; \Pr[\mathsf{E}_{i,k} \mid \mathsf{E}] \geq \frac{1}{(Q_H+Q_S)(Q_S+2)} \right\},$$

where for any $k \in P_i$, we have $\Pr[\mathsf{E}_{i,k}] \geq \frac{\epsilon}{(Q_H+Q_S)(Q_S+2)}$. Let us define $\mathsf{E}_i^{\mathsf{good}} = \bigvee_{k\in P_i} \mathsf{E}_{i,k}$. Then, we have $\Pr\left[\mathsf{E}_i^{\mathsf{good}} \;\middle|\; \mathsf{E}\right] \geq \frac{Q_S+1}{Q_S+2}$, since there are at most $(Q_H + Q_S)$ possible values of k's not in P_i and they can only account to a probability at most $(Q_H+Q_S) \times \frac{1}{(Q_H+Q_S)(Q_S+2)} = \frac{1}{Q_S+2}$.

Next, for any $(i,k) \in [Q_S+1] \times P_i$, let us define $X_{i,k} = R_{\mathsf{coin}} \times (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{k-1}$ and $Y_{i,k} = (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S-k+1}$, where R_{coin} denotes the randomness space of coin. Here, note that $(\mathsf{x}_i, \vec{h}_{\geq k}) \in X_{i,k} \times Y_{i,k}$ can be parsed appropriately to be $(\mathsf{coin}, \vec{h})$, and defines all the inputs of $\mathcal{B}$, where we assume a fixed $(\mathsf{pp}, \mathsf{vk})$. We further define $A_{i,k} \subseteq X_{i,k} \times Y_{i,k}$ to be the set of inputs that triggers event $\mathsf{E}_{i,k}$. Then using the splitting lemma with $\alpha = \frac{Q_S+1}{Q_S+2} \cdot \frac{\epsilon}{(Q_H+Q_S)(Q_S+2)}$, there exists a set $B_{i,k} \subset X_{i,k} \times Y_{i,k}$ such that

$$B_{i,k} = \left\{ (\mathsf{x}_i, \vec{h}_{\geq k}) \in X_{i,k} \times Y_{i,k} \;\middle|\; \Pr_{\vec{h'}_{\geq k} \leftarrow Y_{i,k}} \left[(\mathsf{x}_i, \vec{h'}_{\geq k}) \in A_{i,k} \right] \geq \frac{\epsilon}{(Q_H+Q_S)(Q_S+2)^2} \right\}, \tag{1}$$

and

$$\Pr_{(\mathsf{x}_i, \vec{h'}_{\geq k}) \leftarrow X_{i,k} \times Y_{i,k}} \left[(\mathsf{x}_i, \vec{h}_{\geq k}) \in B_{i,k} \;\middle|\; (\mathsf{x}, \vec{h}_{\geq k}) \in A_{i,k} \right] \geq \frac{Q_S+1}{Q_S+2}. \tag{2}$$

We are now ready to evaluate the success probability of the forking algorithm $\mathsf{F}_{\mathcal{B}}$. With probability ϵ, $\mathcal{B}$ outputs $((I_i)_{i\in[Q_S+1]}, \Lambda)$ in the first execution on input $(\mathsf{coin}, \vec{h}) \in R_{\mathsf{coin}} \times (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S}$. Then the probability that event $\mathsf{E}_i^{\mathsf{good}}$ occurrs for all $i \in [Q_S+1]$ is at least

$$\Pr\left[\forall i \in [Q_S+1],\ \mathsf{E}_i^{\mathsf{good}} \middle| \mathsf{E}\right] \geq 1 - \sum_{i\in[Q_S+1]} \Pr\left[\neg \mathsf{E}_i^{\mathsf{good}} \middle| \mathsf{E}\right] \geq \frac{1}{Q_S+2},$$

where the first inequality follows from the union bound and the second inequality follows from $\Pr\left[\mathsf{E}_i^{\mathsf{good}} \;\middle|\; \mathsf{E}\right] \geq \frac{Q_S+1}{Q_S+2}$.

Then, from eq. (2) and following the same union bound argument, $\mathsf{F}_{\mathcal{B}}$ samples a good input such that $(\mathsf{coin}, \vec{h}) \in B_{i,I_i}$ for all $i \in [Q_S+1]$ conditioned on $\mathsf{E}_i^{\mathsf{good}}$ for all $i \in [Q_S+1]$ with probability at least $\frac{1}{(Q_S+2)}$. Therefore, by eq. (1), if $\mathsf{F}_{\mathcal{B}}$ resamples $\vec{h}_{i,\geq I_i} \in Y_{i,I_i} = (\mathcal{CH} \times \mathcal{C}_{\mathsf{rnd}})^{Q_H+Q_S-I_i+1}$ conditioned on the set B_{i,I_i}, $\mathcal{B}$ succeeds on input $(\mathsf{coin}, \vec{h}_{i,<I_i} \| \vec{h}_{i,\geq I_i})$ with probability at least $\frac{\epsilon}{(Q_H+Q_S)(Q_S+2)^2}$. Conditioning on sampling an input $(\mathsf{coin}, \vec{h}) \in B_{i,I_i}$ for all $i \in [Q_S+1]$ and noting the independence of each rewinding, the probability that $\mathcal{B}$ succeeds in all j-th rewinding for $j \in [Q_S+1]$ is at least

$$\left(1 - \left(1 - \frac{\epsilon}{(Q_H+Q_S)(Q_S+2)^2}\right)^T\right)^{Q_S+1} \geq \left(1 - \frac{1}{e^{\log(2Q_S+2)}}\right)^{Q_S+1} = \left(1 - \frac{1}{2(Q_S+1)}\right)^{Q_S+1} \geq \frac{1}{2}.$$

Collecting all the bounds, we conclude that $\mathsf{F}_{\mathcal{B}}$ succeeds with probability at least $\frac{\epsilon}{2(Q_S+2)^2}$ as desired. Moreover, the running time of $\mathsf{F}_{\mathcal{B}}$ is roughly the same as running $\mathcal{B}$ for at most $T \cdot (Q_S+1)+1$ times, where the runtime of $\mathcal{B}$ is roughly the same as the runtime of $\mathcal{A}$.

Using $\mathsf{F}_{\mathcal{B}}$ to Break Binding of C or `euf-cma` of S. We are now ready to finish the proof. Assume $\epsilon = \mathsf{Adv}^{\mathrm{H}_1}_{\mathcal{A}}(\lambda)$ is non-negligible. We use $\mathsf{F}_{\mathcal{B}}$ to extract the witnesses from the proofs output by $\mathcal{A}$ with non-negligible probability and show that such witnesses can be used to break either the binding of C or the `euf-cma` security of S. Thus establishing that $\epsilon = \mathrm{negl}(\lambda)$ by contradiction.

We define adversary $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ on both the binding property of C and the `euf-cma` property of S as follows. Initially, $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ obtains pp from the binding challenger. Further, she receives vk and oracle access to a signing oracle $\mathsf{S.Sign}(\mathsf{sk}, \cdot)$ from the `euf-cma` challenger. Then, she runs the forking algorithm $R \leftarrow \mathsf{F}_{\mathcal{B}}^{\mathsf{S.Sign}(\mathsf{sk},\cdot)}(\mathsf{pp}, \mathsf{vk})$. She checks $R \neq \bot$, and parses $R = (\Lambda, D)$, where $\Lambda = (x_i, \alpha_i, \beta_i, \gamma_i)_{i \in [Q_S+1]}$ and $D = (j, I_j, \Lambda_j)_{j \in [Q_S+1]}$. Due to lemma 1, $\mathsf{F}_{\mathcal{B}}$ runs in polynomial time and has non-negligible success probability. Below, we describe the second part of $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ and analyze its success probability conditioned on $\mathsf{F}_{\mathcal{B}}$ succeeding. (If $R = \bot$, then $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ outputs $\bot$ and aborts.)

For $j \in [Q_S+1]$, we denote by $(x'_j, \alpha'_j, \beta'_j, \gamma'_j)$ the j-th element of the tuple Λ_j. Moreover, note that the same coin and values $(\widehat{\beta}_1, \Delta r_1), \ldots (\widehat{\beta}_{I_j-1}, \Delta r_{I_j-1})$ are used for the initial run of $\mathcal{B}$ and the run of $\mathcal{B}$ where $\mathcal{B}$ outputs Λ_j. Thus, we have for all $j \in [Q_S+1]$ that $(x_i, \alpha_i) = (x'_i, \alpha'_i)$. Moreover, we have $\widehat{\beta}_{I_j} \neq \widehat{\beta}_{j,I_j}$, or equivalently $\beta_j \neq \beta'_j$ for all $j \in [Q_S+1]$ with probability at least $1 - \frac{Q_S+1}{|\mathcal{CH}|} = 1 - \mathrm{negl}(\lambda)$ since each hash outputs are sampled uniformly and independently at random. This allows $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ to invoke 2-special soundness of Σ with overwhelming probability. For all $i \in [Q_S+1]$, she runs Ext on $(x_i, (\alpha_i, \beta_i, \gamma_i), (\alpha_i, \beta'_i, \gamma'_i))$ to extract a witness $w_i = (\mu_i, c_i, r_i)$ such that $\mathsf{C.Commit}(\mathsf{pp}, \overline{m}_i; r_i) = c_i \wedge \mathsf{S.Verify}(\mathsf{vk}, \mu_i, c_i) = 1$, where $x_i = (\mathsf{pp}, \mathsf{vk}, \overline{m}_i)$.

If there exists distinct $i, j \in [Q_S + 1]$ with $c_i = c_j$, $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ sends $(\overline{m}_i, \overline{m}_j, r_i, r_j)$ to the binding security game of C. Note that due to the check in Hybrid 1, the $(\overline{m}_i)_{i\in[Q_S+1]}$ are pairwise distinct, in particular $\overline{m}_i \neq \overline{m}_j$ but $\mathsf{C.Commit}(\mathsf{pp}, \overline{m}_i; r_i) = \mathsf{C.Commit}(\mathsf{pp}, \overline{m}_j; r_j)$. However, due to the binding property of C, this can happen with only negligible probability. Thus, the extracted commitments $(c_i)_{i\in[Q_S+1]}$ must be distinct with overwhelming probability.

In such a case, there must be at least one $i^* \in [Q_S + 1]$ such that c_i^* was never queried to the signing oracle $\mathsf{S.Sign}(\mathsf{sk}, \cdot)$ *in the first execution* of $\mathcal{B}$ or equivalently of $\mathcal{A}$. This is because due to the one-more unforgeability game, $\mathcal{A}$ only queries the signing oracle Q_S times. Thus, $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ finds such i^* with the smallest index and outputs (μ_{i^*}, c_i^*) as a forgery against the `euf-cma` security of S.

It remains to show that what $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ output is a valid forgery, i.e., $\mathcal{B}$ never queried c_i^* to the signing oracle in any of the *rewound executions*. To argue this, we first show that all the extracted commitments $(c_i)_{i\in[Q_S+1]}$ are fixed *after the first execution ends* due to f-unique extraction. For any $(x_i, \tau_i := (\alpha_i, \beta_i, \gamma_i)) \in \Lambda$ defined in the first execution of $\mathcal{B}$, conditioning on $\mathsf{F}_{\mathcal{B}}$ succeeding, another valid transcript $(x_i, \tau_i' := (\alpha_i, \beta_i', \gamma_i')) \in \Lambda_i$ with $\beta_i \neq \beta_i'$ is guaranteed to exist with overwhelming probability. Due to f-unique extraction, for any such valid transcript the value $f(\mathsf{Ext}(x_i, \tau_i, \tau_i')) = c_i$ is identical, where recall f simply outputs the commitment included in the witness. Put differently, conditioning on $\mathsf{F}_{\mathcal{B}}$ succeeding, (x_i, τ_i) uniquely defines c_i with overwhelming probability. We emphasize that c_i does not need to be efficiently computable given only (x_i, τ_i); we only care if c_i is determined by (x_i, τ_i) in a statistical sense.

Now, assume $\mathcal{B}$ queried c_i^* to the signing oracle in one of the rewound executions. This means $\mathcal{A}$ outputs some c^* to $\mathcal{B}$ (or equivalently the simulated challenger $\mathcal{G}$ in Hybrid 1) and $\mathcal{B}$ computed $c_i^* = \mathsf{C.RerandCom}(\mathsf{pp}, c^*, \Delta r^*)$, where Δr^* is a fresh randomness sampled by $\mathsf{F}_{\mathcal{B}}$ to be used in the rewound execution. However, this cannot happen with all but negligible probability due to the rerandomizability of C since we have established above that Δr^* is sampled independently from c_i^*. Since there are at most $T \cdot (Q_S + 1)$ rewound executions, the probability that $\mathcal{B}$ queries c_i^* to the signing oracle during in one of the rewound execution is bounded by $T \cdot (Q_S + 1) \cdot \mathrm{negl}(\lambda) = \mathrm{negl}(\lambda)$, where we use $T = \mathsf{poly}(\lambda)$ due to lemma 1.

Thus, with overwhelming probability, what $\mathcal{A}_{\mathsf{C},\mathsf{S}}$ output is a valid forgery against the `euf-cma` security of S. However, due to the hardness of `euf-cma` security of S, this cannot happen with all but negligible probability. Combining all the arguments, we conclude that $\epsilon = \mathsf{Adv}^{\mathrm{H}_1}_{\mathcal{A}}(\lambda)$ is negligible. This completes the proof.

Remark 1 (Removing the Rerandomizability Property). As briefly noted in our technical overview, an alternative approach to using rerandomizable commitment is to let the signer (i.e., $\mathsf{BS}_{\mathsf{Rnd}}.\mathsf{Signer}$) sample a random string rand and run $\mu \leftarrow \mathsf{S.Sign}(\mathsf{sk}, c \| \mathsf{rand})$ instead of $\mu \leftarrow \mathsf{S.Sign}(\mathsf{sk}, c')$, where $c' = \mathsf{C.RerandCom}(\mathsf{pp}, c, \Delta r)$ is the rerandomized commitment. The signer then sends $\rho = (\mu, \mathsf{rand})$ as the second message instead of $\rho = (\mu, \Delta r)$. By observing that rand has an identical effect as Δr in the security proof, it can be checked that

the same proof can be used to show blindness and one-more unforgeability of this modified protocol. While this approach works for any commitment scheme, we chose not to since it requires a slightly larger NIZK proof due to the enlarged signing space of the underlying signature scheme S.

3.3 Instantiation

We describe briefly how we instantiate the schemes $(\mathsf{C}, \mathsf{S}, \Sigma)$ in the asymmetric pairing setting. More details can be found in Section 1.3 and in the full version. For C, we choose Pedersen commitments in $\mathbb{G}_1$ of the form $c = g_1^{\overline{m}} \mathsf{pp}^r$, which are naturally rerandomizable and consist of a single element in $\mathbb{G}_1$.

For the signature scheme S, we use a variant of the Kiltz-Pan-Wee (KPW) structure-preserving signature (SPS) scheme [43] in the asymmetric pairing setting. The message space of KPW is $\mathbb{G}_1^\ell$, where $\ell \in \mathbb{N}$ is the message length.

Any SPS must contain at least three group elements, and at least one in each $\mathbb{G}_2$ and in $\mathbb{G}_1$ [1]. But as the bit size of elements in $\mathbb{G}_2$ is larger than the bit size of elements in $\mathbb{G}_1$ and $\mathbb{Z}_p$, removing elements in $\mathbb{G}_2$ in the signature is desirable. For $\mathsf{BS}_{\mathsf{Rnd}}$, we do not require the full structure-preserving property of KPW, as we can design efficient Σ-protocols for signature verification, even if the signature contains elements in $\mathbb{Z}_p$.

Indeed, KPW signatures contain an element σ_4 in $\mathbb{G}_2$. We observe that we can safely replace σ_4 with its discrete logarithm τ. Further, we can omit two more elements in $\mathbb{G}_1$ for free, as they can be recomputed via τ and the remaining signature elements.

Our optimized variant is given below.

- $\mathsf{S}_{\mathrm{KPW}}.\mathsf{KeyGen}(1^\lambda)$: samples $a, b \leftarrow \mathbb{Z}_p$ and sets $\boldsymbol{A} \leftarrow (1, a)^\top$ and $\boldsymbol{B} \leftarrow (1, b)^\top$. It samples $\boldsymbol{K} \leftarrow \mathbb{Z}_p^{(\ell+1)\times 2}$, $\boldsymbol{K}_0$, $\boldsymbol{K}_1 \leftarrow \mathbb{Z}_p^{2\times 2}$ and sets $\boldsymbol{C} \leftarrow \boldsymbol{K}\boldsymbol{A}$. It sets $(\boldsymbol{C}_0, \boldsymbol{C}_1) \leftarrow (\boldsymbol{K}_0\boldsymbol{A}, \boldsymbol{K}_1\boldsymbol{A})$, $(\boldsymbol{P}_0, \boldsymbol{P}_1) \leftarrow (\boldsymbol{B}^\top\boldsymbol{K}_0, \boldsymbol{B}^\top\boldsymbol{K}_1)$, $\mathsf{vk} \leftarrow ([\boldsymbol{C}_0]_2, [\boldsymbol{C}_1]_2, [\boldsymbol{C}]_2, [\boldsymbol{A}]_2)$, and $\mathsf{sk} \leftarrow (\boldsymbol{K}, [\boldsymbol{P}_0]_1, [\boldsymbol{P}_1]_1, [\boldsymbol{B}]_1)$. It outputs $(\mathsf{vk}, \mathsf{sk})$.
- $\mathsf{S}_{\mathrm{KPW}}.\mathsf{Sign}(\mathsf{sk}, [\boldsymbol{m}]_1)$: samples $r, \tau \leftarrow \mathbb{Z}_p$ and sets $\sigma_1 \leftarrow [(1, \boldsymbol{m}^\top)\boldsymbol{K} + r(\boldsymbol{P}_0 + \tau\boldsymbol{P}_1)]_1 \in \mathbb{G}_1^2$, $\sigma_2 \leftarrow [r\boldsymbol{B}^\top]_1 \in \mathbb{G}_1^2$, and $\sigma_3 \leftarrow \tau \in \mathbb{Z}_p$. It outputs $(\sigma_1, \sigma_2, \sigma_3)$.
- $\mathsf{S}_{\mathrm{KPW}}.\mathsf{Verify}(\mathsf{vk}, [\boldsymbol{m}]_1, (\sigma_1, \sigma_2, \sigma_3))$: checks $e(\sigma_1, [\boldsymbol{A}]_2) = e([(1, \boldsymbol{m}^\top)]_1, [\boldsymbol{C}]_2) \cdot e(\sigma_2, [\boldsymbol{C}_0]_2 \cdot \tau[\boldsymbol{C}_1]_2)$.

We show that $\mathsf{S}_{\mathrm{KPW}}$ is $\mathtt{euf\text{-}cma}$ under the SXDH assumption in the full version. The proof relies on the computational core lemma of [44]. $\mathsf{S}_{\mathrm{KPW}}$ can be made deterministic via a pseudorandom function.

For an efficient instantiation of the Σ-protocol Σ, we refer to the full version. In the resulting blind signature $\mathsf{BS}_{\mathsf{Rnd}}$, the user sends 1 element in $\mathbb{G}_1$ and 1 element in $\mathbb{Z}_p$, the signer sends 4 elements in $\mathbb{G}_1$ and 1 element in $\mathbb{Z}_p$ and the final signature contains 6 elements in $\mathbb{G}_1$ and 5 elements in $\mathbb{Z}_p$. The total communication is 303 Byte and signatures are of size 447 Byte for $\lambda = 128$.

4 Blind Signatures based on Boneh-Boyen Signature

In this section, we provide a blind signature based on randomizable signatures. Compared to the optimized generic construction of the Fischlin blind signature in section 3, the resulting signature size is much smaller since it only consists of one signature of the underlying randomizable signature scheme. The construction also relies on an online-extractable NIZK which can be instantiated efficiently by carefully combining Bulletproofs and another NIZK for an ElGamal commitment (see the full version). In the full version we show how to adapt the scheme for a partially blind variant, where we modify the Boneh-Boyen signature [12, 14] in order to embed the common message into the verification key.

4.1 Construction

We focus on the asymmetric pairing setting. We note that there is also a natural variant of this scheme in the symmetric setting and we omit details. First, we recall the Boneh-Boyen signatures [12, 14] in the asymmetric setting. While this is implicit in our proof, we note the following is known to be selectively secure in the standard model under the CDH assumption:

- $\mathsf{S_{BB}.KeyGen}(1^\lambda)$: samples $\alpha, \beta, \gamma \leftarrow \mathbb{Z}_p$, and sets $u_1 = g_1^\alpha, u_2 = g_2^\alpha, h_1 = g_1^\gamma, h_2 = g_2^\gamma, v = e(g_1, g_2)^{\alpha\beta}$, and outputs $\mathsf{vk} = (u_1, u_2, h_1, h_2, v)$ and $\mathsf{sk} = g_1^{\alpha\beta}$,
- $\mathsf{S_{BB}.Sign}(\mathsf{sk}, m)$: samples $r \in \mathbb{Z}_p$ and outputs $(\sigma_1, \sigma_2) = (\mathsf{sk} \cdot (u_1^m h_1)^r, g_1^r) \in \mathbb{G}_1^2$,
- $\mathsf{S_{BB}.Verify}(\mathsf{vk}, m, (\sigma_1, \sigma_2))$: outputs 1 if $e(\sigma_1, g_2) = v \cdot e(\sigma_2, u_2^m h_2)$, and otherwise outputs 0.

Let Π be an online-extractable NIZK proof system, with random oracle $\mathsf{H_{zk}} : \{0,1\}^* \mapsto \{0,1\}^{\ell_{\mathsf{zk}}}$ and common reference string crs of length ℓ_{crs} for the relation

$$\mathsf{R_{bb}} := \{x = (c, u_1, g_1), w = (\overline{m}, s) \mid c = u_1^{\overline{m}} \cdot g_1^s\}.$$

Let $\mathsf{H_M}, \mathsf{H_{crs}}$ be a random oracles mapping into $\mathbb{Z}_p, \{0,1\}^{\ell_{\mathsf{crs}}}$ respectively.

Construction. We present our blind signatures based on $\mathsf{S_{BB}}$, where we assume that $\mathsf{crs} = \mathsf{H_{crs}}(0)$ is provided to all of the algorithms for readability.

- $\mathsf{BS_{BB}.KeyGen}(1^\lambda)$: outputs $(\mathsf{bvk}, \mathsf{bsk}) \leftarrow \mathsf{S_{BB}.KeyGen}(1^\lambda)$, where $\mathsf{bvk} = (u_1, u_2, h_1, h_2, v)$ with $u_1 = g_1^\alpha, u_2 = g_2^\alpha, h_1 = g_1^\gamma, h_2 = g_2^\gamma, v = e(g_1, g_2)^{\alpha\beta}$ and $\mathsf{bsk} = g_1^{\alpha\beta}$.
- $\mathsf{BS_{BB}.User}(\mathsf{bvk}, m)$: checks validity of the verification key bvk via $e(u_1, g_2) = e(g_1, u_2)$ and $e(h_1, g_2) = e(g_1, h_2)$, sets $\overline{m} \leftarrow \mathsf{H_M}(m)$ and computes a Pedersen commitment $c = u_1^{\overline{m}} g_1^s \in \mathbb{G}_1$ to $\overline{m}$ and a proof $\pi \leftarrow \Pi.\mathsf{Prove}^{\mathsf{H_{zk}}}(\mathsf{crs}, x, w)$, where $s \leftarrow \mathbb{Z}_p$, $x = (c, u_1, g_1)$, and $w = (\overline{m}, s)$. It outputs the first message $\rho_1 = (c, \pi)$ and stores the randomness $\mathsf{st} = s$.
- $\mathsf{BS_{BB}.Signer}(\mathsf{bsk}, \rho_1)$: parses $\rho_1 = (c, \pi)$, checks $\Pi.\mathsf{Verify}^{\mathsf{H_{zk}}}(\mathsf{crs}, x, \pi) = 1$ and outputs the second message $\rho_2 = (\rho_{2,0}, \rho_{2,1}) \leftarrow (\mathsf{sk} \cdot (c \cdot h_1)^r, g_1^r) \in \mathbb{G}_1^2$, where $r \leftarrow \mathbb{Z}_p$.

- $\mathsf{BS}_{\mathsf{BB}}.\mathsf{Derive}(\mathsf{st}, \rho_2)$: parses $\mathsf{st} = s$ and $\rho_2 = (\rho_{2,0}, \rho_{2,1})$, checks $e(\rho_{2,0}, g_2) = v \cdot e(\rho_{2,1}, u_2^{\overline{m}} g_2^s \cdot h_2)$, and outputs the signature $\sigma = (\rho_{2,0}/\rho_{2,1}^s \cdot (u_1^{\overline{m}} h_1)^{r'}, \rho_{2,1} \cdot g_1^{r'}) \in \mathbb{G}_1^2$ for $r' \leftarrow \mathbb{Z}_p$.
- $\mathsf{BS}_{\mathsf{BB}}.\mathsf{Verify}(\mathsf{bvk}, m, \sigma)$: sets $\overline{m} \leftarrow \mathsf{H}_{\mathsf{M}}(m)$ and outputs $b \leftarrow \mathsf{S}_{\mathsf{BB}}.\mathsf{Verify}(\mathsf{bvk}, \overline{m}, \sigma)$.

4.2 Correctness and Security

We prove correctness, blindness and one-more unforgeability. Correctness follows from a simple calculation. Blindness follows from the zero-knowledge property of Π, and as c statistically hides the message and σ is re-randomized. The proof follows a similar all-but-one reduction as the underlying Boneh-Boyen signature. The only difference is that we modify the Boneh-Boeyn signature which is selectively secure in the standard model, to be adaptively secure in the ROM, and to use the (multi)-online extractor to extract randomness of c submitted by the adversary. Concretely, the reduction first guesses a query $\overline{m}^* = \mathsf{H}_{\mathsf{M}}(m^*)$ and embeds a CDH challenge into vk such that it can sign all values in $\mathbb{Z}_p \setminus \{\overline{m}^*\}$. For each signing query, the reduction extracts the randomness of c from the proof π, simulates the signing of m as in the original `euf-cma` proof of S_{BB}, and finally reapplies the randomness of c to the intermediate signature. If the extracted message is $\overline{m}^*$, the reduction aborts. Here, we crucially require that Π is online-extractable. In the end, the reduction hopes to receive a valid signature on $\overline{m}^*$ with which it can solve CDH. More details can be found in section 1.3. A formal security analysis is given in the full version.

Theorem 2. *The blind signature S_{BB} is correct, blind under malicious keys under the zero-knowledge property of Π, and one-more unforgeable under the CDH assumption and the online-extractability of Π.*

4.3 Instantiation

We give a brief overview of our online-extractable NIZK Π. More details can also be found in the full version. As our online-extraction techniques are not reliant on pairings, we use an additional group $\widehat{\mathbb{G}}$ with generators $\widehat{g}_1$ and $\widehat{\mathsf{pp}}$.

Tools. In the full version, we construct a secure Σ-protocol Σ_{ped} for relation $\mathsf{R}_{\mathsf{ped}}$. In this overview, we compile it into a NIZK Π_{ped} via Fiat-Shamir. This is kept implicit in the instantiation as we cannot rely on the security of Π_{ped} in a black-box manner. The relation $\mathsf{R}_{\mathsf{ped}}$ is defined as

$$\mathsf{R}_{\mathsf{ped}} = \{(x, w) : c = u_1^m g_1^s, E_i = \widehat{g}^{e_i} \widehat{\mathsf{pp}}^{r_i}, R_i = \widehat{g}^{r_i}, \\ \prod_{i \in [\ell]} E_i^{B^{i-1}} = \widehat{g}^m \cdot \widehat{\mathsf{pp}}^{t_m}, \prod_{i \in [\ell]} E_{i+\ell}^{B^{i-1}} = \widehat{g}^s \cdot \widehat{\mathsf{pp}}^{t_s}\},$$

where $x = (c, u_1, g_1, \widehat{g}, \widehat{\mathsf{pp}}, (E_i, R_i)_{i \in [2\ell]}, B)$ and $w = (m, s, (e_i, r_i)_{i \in [2\ell]}, t_m, t_s)$. Note that the relation shows that $m = \sum_{i=1}^{\ell} e_i B^{i-1}$ and $s = \sum_{i=1}^{\ell} e_{i+\ell} B^{i-1}$

under the DLOG assumption. Also, we use a NIZK Π_{rp} with random oracle H_{rp} for the relation

$$\mathsf{R}_{\mathsf{rp}} = \{(x, w) : \ E_i = \widehat{g}^{e_i} \cdot \widehat{\mathsf{pp}}^{r_i}, e_i \in [0, B-1] \text{ for } i \in [2\ell]\},$$

We obtain Π_{rp} by applying the Fiat-Shamir transformation as described in [5] to the multi-round interactive proof system $\Sigma_{\mathsf{rp}}^{2\ell}$ with $\mathsf{crs} = (\widehat{g}, \widehat{\mathsf{pp}}, (\widehat{g}_i)_{i\in[\ell_{\mathsf{rp}}]})$ from [4] (Appendix F.2), for appropriate $\ell_{\mathsf{rp}} \in \mathbb{N}$. Denote with $\mathsf{R}_{\mathsf{dlog}} = \{(\mathsf{crs}, w^*)\}$ the relation that contains all non-trivial DLOG relations w^* for crs, i.e. computing w^* for random crs allows to solve the DLOG assumption. Using Theorem 4 of [5], we show in the full version that Π_{rp} is adaptively knowledge sound for the relaxed relation $\mathsf{R}_{\mathsf{lax}} \coloneqq \{(x, w) : \ (x, w) \in \mathsf{R}_{\mathsf{rp}} \text{ or } (\mathsf{crs}, w) \in \mathsf{R}_{\mathsf{dlog}}\}$.

Construction of Π. Equipped with the above tools, we instantiate the online-extractable NIZK Π for relation R_{bb} with $\mathsf{crs} = (\widehat{g}, \widehat{\mathsf{pp}}, (\widehat{g}_i)_{i\in[\ell_{\mathsf{rp}}]})$ and hash function $\mathsf{H}_{\mathsf{bb}} = (\mathsf{H}_{\mathsf{rp}}, \mathsf{H}_\beta)$, where H_{rp} (resp. H_{eg}) is the hash function for Π_{rp} (resp. Π_{ped}). Let $B = \mathsf{poly}(\lambda)$.

To generate a poof π for statement $x = (c, u_1, g_1)$, the prover decomposes the witness (m, s) into $m = \sum_{i=1}^{\ell} m_i B^{i-1}, s = \sum_{i=1}^{\ell} s_i B^{i-1}$, commits to the decompositions $\boldsymbol{e} = (m_1, \ldots, m_\ell, s_1, \ldots, s_\ell)$ via ElGamal in $R_i = \widehat{g}^{r_i}, E_i = \widehat{g}^{e_i}\widehat{\mathsf{pp}}_i^{r_i}$ for $i \in [2\ell]$, where $r_i \leftarrow \mathbb{Z}_p$, and sets $t_m \leftarrow \sum_{i=1}^{\ell} r_i B^{i-1}$ and $t_s \leftarrow \sum_{i=1}^{\ell} r_{i+\ell} B^{i-1}$, and finally outputs proofs $\pi = (\pi_0, \pi_1, (E_i, R_i)_{i\in[2\ell]})$, where π_0, π_1 are proofs generated appropriately via $\Pi_{\mathsf{rp}}, \Pi_{\mathsf{ped}}$, respectively.

To check validity of a proof π, the verifier checks both proofs π_0 and π_1 with appropriate statements x_0 and x_1, respectively, and outputs 1 iff both are valid.

Security. In the full version, we formally show that Π is correct, zero-knowledge under the DDH assumption and online-extractable under the DLOG assumption. Correctness follows immediately from the correctness of Π_{rp} and Π_{ped}. Also, zero-knowledge is easy to show via the hiding property of ElGamal commitments, the zero-knowledge property of Π_{rp} and Π_{ped}.

The proof for multi-proof extractability is more intricate. Roughly, the extractor embeds a trapdoor td for the commitment scheme in the crs. Then, given a statement-proof pair (x, π) with $x = (c, u_1, g_1)$ and $\pi = (\pi_0, \pi_1, (E_i, R_i)_{i\in[2\ell]})$, it decrypts the witnesses $(e_i)_i$ from the ElGamal commitment $(E_i, R_i)_i$ and tries to check if the extracted witness reconstructs to a witness in the relation R_{bb}. We expect that this is possible, as the range proof guarantees that the committed values are short and Σ_{ped} proves the linear relations in the exponents.

For the sake of exposition, below we only consider extracting from a single pair $(x, \pi) \leftarrow \mathcal{A}(\mathsf{crs})$ generated by some adversary $\mathcal{A}$. The argument generalizes to Q_S pairs in a straightforward manner. Note that (x, π) defines statement-proof pairs (x_0, π_1) for Π_{rp} and (x_1, π_1) as in verification.

Denote with Fail the event that online-extraction of (x, π) fails, and assume for the sake of contradiction that Fail occurs. We first try to extract a witness $w_0 = (e_i', r_i')_i$ from π_0 via the knowledge extractor of Π_{rp}, and a witness $w_1 = (m, s, (e_i, r_i)_i)$ from π_1 from two related transcripts obtained via rewinding $\mathcal{A}$. Here, it is important that $\mathcal{A}$ is run with the same random tape $\mathsf{coin}_{\mathcal{A}}$ for both

extractions to guarantee that the statements x_0 and x_1 share the commitments $(E_i)_i$. For now, let us assume that both extractions succeed, i.e. $(x_0, w_0) \in \mathsf{R}_{\mathsf{rp}}$ and $(x_1, w_1) \in \mathsf{R}_{\mathsf{ped}}$. Assuming the soundness of Π_{rp}, we have $e'_i \in [0, B-1]$. Moreover, assuming the soundness of Π_{ped}, the extracted $(e_i)_i$ form the B-ary decomposition of a valid opening of c. Then, under the assumption that extraction fails, we must have $e'_i \neq e_i$ for some i. However, this breaks the binding property of the Pedersen commitment implicitly defined by the ElGamal commitments. In particular, we found a DLOG relation for the tuple $(\widehat{g}, \widehat{\mathsf{pp}}_i)$. Note that while the extracted DLOG relation is a trapdoor information td the extractor uses to extract the witnesses, this will not be an issue since we do not need td to analyze the success probability of the adversary.

It remains to show that extraction of w_0 and w_1 succeeds. Recall that we assumed that the extraction of w_0 and w_1 succeeds simultaneously, even if we initially run $\mathcal{A}$ on a shared random coin. We first extract w_0 with the extractor of Π_{rp}. We can argue with adaptive knowledge soundness of Π_{rp} that with a probability of $\varepsilon = \frac{\Pr[\mathsf{Fail}]-\mathrm{negl}(\lambda)}{\mathsf{poly}(\lambda)}$, we have have that $(x_0, w_0) \in \mathsf{R}_{\mathsf{rp}}$ and Fail occurs. At this point, the randomness $\mathsf{coin}_{\mathcal{A}}$ of the adversary $\mathcal{A}$ is conditioned on successful extraction of w_0. In particular, we cannot apply adaptive knowledge soundness of Π_{ped}, as the extractor of Π_{ped} has only sufficient success probability if $\mathsf{coin}_{\mathcal{A}}$ is chosen at random.

Instead, we define a specialized forking algorithm that first runs $\mathcal{A}$ on the same randomness (and same initial random oracle choices), and then rewinds $\mathcal{A}$ to obtain related transcripts. A careful non-black box analysis of the forking algorithm, similar to [51], allows us to conclude that the algorithm succeeds in finding two related transcripts in polynomial time with probability $\varepsilon/8$.

If Fail is non-negligible, then with a probability of $\varepsilon/8$, the above adversary breaks the DLOG assumption. So indeed the event Fail occurs with at most negligible probability, i.e. the extractor of Π succeeds on valid proof-statement pairs with high probability.

Efficiency of $\mathsf{BS}_{\mathsf{BB}}$. When $\mathsf{BS}_{\mathsf{BB}}$ is instantiated with Π for $B = \mathsf{poly}(\lambda)$, the user sends 1 element in $\mathbb{G}_1$, $2\lceil \log_2(2n\ell + \ell + 4) \rceil + 4\ell + 1$ in $\widehat{\mathbb{G}}$, and $10 + 2\ell$ elements in $\mathbb{Z}_p$ to the signer. The signer sends 2 elements in $\mathbb{G}_1$, and the final signature contains 2 elements in $\mathbb{G}_1$.

We set $B = 2^{64}$ in order to have an extractor that performs roughly $\ell \cdot 2^{32}$ group operations, where $\ell = \lceil \log_B p \rceil = 4$. The total communication is 2.2 KB and signatures are of size 96 Byte for $\lambda = 128$.

5 Acknowledgments

This work was partially supported by JST CREST Grant Number JPMJCR22M1, Japan, JST AIP Acceleration Research JPMJCR22U5, Japan, JSPS KAKENHI Grant Numbers JP18K18055 and JP19H01109. Also, we would like to thank Michael Klooß for helpful discussions on the soundness of Bulletproofs, and Geoffroy Couteau for helpful discussions in early stages of this work.

References

1. Abe, M., Groth, J., Haralambiev, K., Ohkubo, M.: Optimal structure-preserving signatures in asymmetric bilinear groups. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 649–666. Springer, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_37
2. 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
3. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_28
4. 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
5. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs. In: Kiltz, E., Vaikuntanathan, V. (eds.) Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13747, pp. 113–142. Springer (2022). https://doi.org/10.1007/978-3-031-22318-1_5, https://doi.org/10.1007/978-3-031-22318-1_5
6. 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)
7. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. J. Cryptol. **16**(3), 185–215 (2003). https://doi.org/10.1007/s00145-002-0120-1, https://doi.org/10.1007/s00145-002-0120-1
8. 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
9. Bernhard, D., Fischlin, M., Warinschi, B.: Adaptive proofs of knowledge in the random oracle model. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 629–649. Springer, Heidelberg (Mar / Apr 2015). https://doi.org/10.1007/978-3-662-46447-2_28
10. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Short blind signatures. Journal of computer security **21**(5), 627–661 (2013)
11. 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
12. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_14

13. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_4
14. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology **21**(2), 149–177 (Apr 2008). https://doi.org/10.1007/s00145-007-9005-7
15. Bowe, S.: Bls12-381: New zk-snark elliptic curve construction. https://electriccoin.co/blog/new-snark-curve/ (2017), accessed: 2023-02-02
16. 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
17. Brickell, E.F., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004. pp. 132–145. ACM Press (Oct 2004). https://doi.org/10.1145/1030083.1030103
18. 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
19. Buser, M., Dowsley, R., Esgin, M.F., Gritti, C., Kermanshahi, S.K., Kuchta, V., LeGrow, J.T., Liu, J.K., Phan, R.C.W., Sakzad, A., Steinfeld, R., Yu, J.: A survey on exotic signatures for post-quantum blockchain: Challenges & research directions. ACM Comput. Surv. (2022). https://doi.org/10.1145/3572771, https://doi.org/10.1145/3572771, just accepted
20. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44987-6_7
21. 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
22. 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)
23. 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
24. 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
25. 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
26. 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

27. Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 152–168. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_10
28. 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
29. 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
30. 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
31. 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
32. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: AUSCRYPT. pp. 244–251. Springer (1992)
33. Ganesh, C., Orlandi, C., Pancholi, M., Takahashi, A., Tschudi, D.: Fiat-shamir bulletproofs are non-malleable (in the algebraic group model). In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 397–426. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_14
34. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press (Oct / Nov 2006). https://doi.org/10.1145/1180405.1180418, available as Cryptology ePrint Archive Report 2006/309
35. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for non-interactive zero-knowledge. Journal of the ACM (JACM) **59**(3), 1–35 (2012)
36. Hanzlik, L., Loss, J., Wagner, B.: Rai-choo! evolving blind signatures to the next level. To Appear at EUROCRYPT (2023)
37. Hendrickson, S., Iyengar, J., Pauly, T., Valdez, S., Wood, C.A.: Private access tokens. internet-draft draft-private-access-tokens-01 (April 2022), https://datatracker.ietf.org/doc/draft-private-access-tokens/, work in Progress
38. 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
39. Jutla, C.S., Roy, A.: Improved structure preserving signatures under standard bilinear assumptions. In: Fehr, S. (ed.) PKC 2017, Part II. LNCS, vol. 10175, pp. 183–209. Springer, Heidelberg (Mar 2017). https://doi.org/10.1007/978-3-662-54388-7_7
40. Katz, J.: Digital signatures: Background and definitions. In: Digital Signatures. Springer (2010)
41. 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

42. Khalili, M., Slamanig, D., Dakhilalian, M.: Structure-preserving signatures on equivalence classes from standard assumptions. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 63–93. Springer, Heidelberg (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_3
43. Kiltz, E., Pan, J., Wee, H.: Structure-preserving signatures from standard assumptions, revisited. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 275–295. Springer, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-48000-7_14
44. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_4
45. 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
46. Micali, S., Reyzin, L.: Improving the exact security of digital signature schemes. Journal of Cryptology **15**(1), 1–18 (Jan 2002). https://doi.org/10.1007/s00145-001-0005-8
47. 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
48. 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
49. 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
50. 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
51. 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
52. Sahai, A., Waters, B.R.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (May 2005). https://doi.org/10.1007/11426639_27
53. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 1–16. Springer, Heidelberg (May / Jun 1998). https://doi.org/10.1007/BFb0054113
54. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_10
55. Vpn by Google one, explained. https://one.google.com/about/vpn/howitworks (2022), accessed: 2023-02-02
56. Waters, B.R.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (May 2005). https://doi.org/10.1007/11426639_7
57. 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