Blind signatures from Zero-knowledge arguments

Paulo L. Barreto[1], Gustavo H. M. Zanon[2]

[1] University of Washington | Tacoma, USA.
pbarreto@uw.edu
[2] University of São Paulo, Brazil.
ghmzanon@gmail.com

Abstract. We propose a novel methodology to obtain Blind signatures that is fundamentally based on the idea of hiding part of the underlying plain signatures under a Zero-knowledge argument of knowledge of the whole signature (hence the shorthand, BZ). Our proposal is necessarily non-black-box and stated in the random oracle model. We illustrate the technique by describing two instantiations: a classical setting based on the traditional discrete logarithm assumption, and a post-quantum setting based on the commutative supersingular isogeny Diffie-Hellman (CSIDH) assumption.

Keywords: Blind signatures, Zero-knowledge arguments.

1 Introduction

Secure blind signatures [10], despite their usefulness, are notoriously hard to obtain in a truly secure fashion. Many proposals in the literature have been directly broken [6] or shown to at least contain flaws in their analysis [18]. To make things worse, many impossibility results have been discovered along the years, e.g. [4,14,19] to cite just a few, severely restricting the kind of construction that could still be obtained. Yet, secure constructions are known [17], and although they might not be viewed as particularly efficient, they give hope that further schemes are possible. New blind signature schemes are likely to have a worthwhile impact, both in a conventional setting and in a post-quantum [2] scenario as well.

Contributions: We propose a novel methodology to obtain blind signatures from certain security assumptions that is based, on the one hand, on the idea of replacing part of the usual plain signatures by a non-interactive zero-knowledge argument for that (now hidden) part, and on the other hand, on certain specific operations the underlying primitive must support that guide the instantiation of blind signatures on top of that primitive. Given that we seek to obtain Blind signatures from Zero-knowledge proofs, we call it the BZ methodology.

Since three-round blind signatures in the standard model are known to be unattainable [14], our proposal is constrained to the random oracle model instead. Black-box constructions are also known to be impossible even in the random oracle model [19], for which reason our proposal is *non-black-box*, and part of the analysis must be deferred to the actual instantiations, albeit in a systematic way. Moreover, our proposal relies on the overall structure of Schnorr-style signatures, and again impossibility theorems are known that rule out large classes of such schemes [4]. We circumvent this by avoiding the conditions of those theorems and reducing to a stronger security assumption than the natural one (e.g. OMDL instead of DL), and the zero-knowledge arguments of knowledge are pivotal to achieve it.

We instantiate the *BZ* methodology for the traditional discrete logarithm assumption and for the commutative supersingular isogeny Diffie-Hellman, or CSIDH [9], assumption, and argue their formal security in terms of perfect blindness and the notion of one-more MitM unforgeability introduced in [17].

Organization: The remainder of this paper is organized as follows. Section 2 introduces notation and basic algebraic concepts. Section 3 describes the proposed *BZ* methodology. Section 4 recaps the standard security notions concerning blind signatures, tailoring them to *BZ* schemes. Sections 5 and 6 specify two instantiations of the *BZ* construction, one based on the conventional discrete logarithm assumption and one based on the commutative supersingular isogeny Diffie-Hellman (CSIDH) assumption. Section 7 discusses parameters and compares the instantiations above against related blind signature constructions. We conclude the main text in Section 8.

Supplementary material: In supplement A we explore plausible ways to mount a ROS-style attack against *BZ* and *IZ* schemes, and show how and why they fail. We also provide more details on the signing equations of the proposed $BZ[\mathsf{DL}]$ and $BZ[\mathsf{CSI}]$ schemes in supplement B.

2 Preliminaries

2.1 Notation

Throughout the paper we write $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z}$, $\mathbb{Z}_n^* := \mathbb{Z}_n \setminus \{0\}$. For an Abelian group $\mathbb{G}$ written multiplicatively with neutral element $\mathbf{1}$, we write $\mathbb{G}^* := \mathbb{G} \setminus \{\mathbf{1}\}$. For any finite set S, $x \overset{\$}{\leftarrow} S$ denotes the uniformly random sampling of an element $x \in S$.

For any positive integer n, $\mathcal{S}_n$ denotes the symmetric group (i.e. the set of all permutations) of the set $\{1 \dots n\}$, and for any $k \mid n$, $\mathcal{M}_k^n$ denotes the set of sequences from $\{0 \dots k-1\}^n$ where each of the k possible distinct values occurs exactly n/k times.

Matrices taken from $S^{T \times t}$ (for some set S) will be written with row indices as superscripts and column indices as subscripts, e.g. $e_{1 \dots t}^{1 \dots T}$.

Given any vector or matrix $e_{1 \dots n}$, $e^{1 \dots m}$, or $e_{1 \dots n}^{1 \dots m}$, and given any permutations $\pi \in \mathcal{S}_n$ and $\rho \in \mathcal{S}_m$, we write $\pi(e_{1 \dots n}) := e_{\pi(1 \dots n)} := (e_{\pi(1)} \dots e_{\pi(n)})$, $\rho(e^{1 \dots m}) := e^{\rho(1 \dots m)} := (e^{\rho(1)} \dots e^{\rho(m)})$, $\pi_{\downarrow}(e_{1 \dots n}^{1 \dots m}) := e_{\pi(1 \dots n)}^{1 \dots m}$, $\rho^{\uparrow}(e_{1 \dots n}^{1 \dots m}) := e_{1 \dots n}^{\rho(1 \dots m)}$, and $\rho^{\uparrow}\pi_{\downarrow}(e_{1 \dots n}^{1 \dots m}) = \pi_{\downarrow}\rho^{\uparrow}(e_{1 \dots n}^{1 \dots m}) := e_{\pi(1 \dots n)}^{\rho(1 \dots m)}$. NB: permutations and simple vectors/matrix arithmetic operations commute, e.g. $\pi(e_{1\dots n}) \pm \pi(e'_{1\dots n}) = \pi(e_{1\dots n} \pm e'_{1\dots n})$.

Given sequences $e_{0\dots k-1}$ and $e^{0\dots k-1}$ and an element $c_{1\dots n} \in \mathcal{M}_k^n$, we write $e_{\downarrow c_{1\dots n}} := (e_{c_1} \dots e_{c_n})$ and $e^{\uparrow c_{1\dots n}} := (e^{c_1} \dots e^{c_n})$, and similarly for matrices where both upper and lower indices are present.

Supersingular curves and isogenies: Given a prime $p = 3 \pmod 4$, the set of supersingular elliptic curves (in Montgomery form) over $\mathbb{F}_p$ is denoted $\mathcal{E}\ell\ell_p := \{E_A/\mathbb{F}_p : y^2 = x^3 + Ax^2 + x \mid A \in \mathbb{F}_p \wedge \#E(\mathbb{F}_p) = p + 1\}$. The neutral element in the group of points of such a curve E is denoted O_E (or simply O when the curve is clear from the context).

Two elliptic curves E and E' are called *isogenous* if there is a group homomorphism $\phi : E \to E'$ with $\phi(O_E) = O_{E'}$. That homomorphism is then called an *isogeny*.

The ring of endomorphisms $\mathrm{End}(E)$ of a curve E is the set of all isogenies from E to itself. The ring of $\mathbb{F}_p$-rational endomorphisms of E is denoted $\mathrm{End}_{\mathbb{F}_p}(E)$. For supersingular curves over $\mathbb{F}_p$, $\mathrm{End}(E)$ is an order in a quaternion algebra whereas $\mathcal{O} := \mathrm{End}_{\mathbb{F}_p}(E)$ is an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. The ideal class group of $\mathcal{O}$, denoted $\mathrm{Cl}(\mathcal{O})$, is the quotient of the group of fractional invertible ideals in $\mathcal{O}$ by the principal fractional invertible ideals. It is known that $\mathrm{Cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{E}\ell\ell_p$ through the mapping $\mathcal{O} \times \mathcal{E}\ell\ell_p \to \mathcal{E}\ell\ell_p$ defined as $\mathfrak{a} \star E := E/S_{\mathfrak{a}}$, where $S_{\mathfrak{a}} := \bigcap_{\alpha \in \mathfrak{a}} \ker(\alpha)$ and $E/S_{\mathfrak{a}}$ is the image of the isogeny defined on E with kernel $S_{\mathfrak{a}}$.

We henceforth assume that $\mathrm{Cl}(\mathcal{O})$ is cyclic of order $N := \#\mathrm{Cl}(\mathcal{O})$. Let $\mathfrak{g}$ be a fixed generator of $\mathrm{Cl}(\mathcal{O})$ and let $[\cdot] : \mathbb{Z}_N \times \mathcal{E}\ell\ell_p \to \mathcal{E}\ell\ell_p$ be the mapping $(s, E) \mapsto [s]E := \mathfrak{g}^s \star E$. For each $s \in \mathbb{Z}_N$, $[s] : \mathcal{E}\ell\ell_p \to \mathcal{E}\ell\ell_p$ denotes the specific action $E \mapsto [s]E$.

Given $A \in \mathcal{E}\ell\ell_p$, $U_{1\dots n}^{1\dots m} \in \mathcal{E}\ell\ell_p^{m\times n}$, and $\delta_{1\dots n}^{1\dots m} \in (\mathbb{Z}_N^*)^{m\times n}$, we denote $[\delta_{1\dots n}^{1\dots m}]A := \big([\delta_1^1]A \dots [\delta_n^1]A, \dots, [\delta_1^m]A \dots [\delta_n^m]A\big)$ and $[\delta_{1\dots n}^{1\dots m}]U_{1\dots n}^{1\dots m} := \big([\delta_1^1]U_1^1 \dots [\delta_n^1]U_n^1, \dots, [\delta_1^m]U_1^m \dots [\delta_n^m]U_n^m\big)$.

Notice that, for any permutations $\pi \in \mathcal{S}_n$ and $\rho \in \mathcal{S}_m$, it holds that $\pi_\downarrow\big([\delta_{1\dots n}^{1\dots m}]U_{1\dots n}^{1\dots m}\big) = \big[\pi_\downarrow(\delta_{1\dots n}^{1\dots m})\big]\,\pi_\downarrow(U_{1\dots n}^{1\dots m})$ and $\rho^\uparrow\big([\delta_{1\dots n}^{1\dots m}]U_{1\dots n}^{1\dots m}\big) = \big[\rho^\uparrow(\delta_{1\dots n}^{1\dots m})\big]\,\rho^\uparrow(U_{1\dots n}^{1\dots m})$.

Protocols: We hereby adopt the term "meddler-in-the-middle" (MitM) as a gender-neutral alternative to the standard "man-in-the-middle" label in formal security games.

We employ a simple color code to facilitate keeping track of analogous or related quantities upon reading the text or comparing the instantiations, e.g. key pairs are colored blue, generators and starting curves are colored red, primary commitments are colored green, and so on.

Transcript components from protocol sessions are indicated with a circumflex, e.g. $\hat{c}$.

Blinding elements in signatures are indicated with Greek letters (except σ, reserved to denote the signatures themselves).

3 The *BZ* methodology

As we mentioned in the Introduction, our proposal builds on the basic structure of the Schnorr blind signature scheme. We now recap that scheme and then describe the general *BZ* ideas.

3.1 The Schnorr signature scheme

Let $\mathbb{G}$ be an Abelian group of prime order n where the discrete logarithm problem is assumed to be hard, and let $\mathcal{G} : \mathbb{G} \to \mathbb{Z}_n$ and $\mathcal{H} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_n$ be random oracles.

The Schnorr identification scheme [21] for a group generator $g \in \mathbb{G}$ and a key pair $(x \in \mathbb{Z}_n, y := g^x \in \mathbb{G})$ is a challenge-response protocol where, given a commitment $u \in \mathbb{G}$ and a challenge $c \in \mathbb{Z}_n$, the response $z \in \mathbb{Z}_n$ satisfies $u = g^z y^c$. It is made into a signature scheme by replacing the random challenge by the output from a random oracle call that includes[3] the message $m \in \{0,1\}^*$, namely $c := \mathcal{H}(u, m)$, yielding a short transcript $(c, z) \in \mathbb{Z}_n^2$ satisfying $c = \mathcal{H}(g^z y^c, m)$.

Since verification requires computing $u \leftarrow g^z y^c$, one might be tempted to speed it up by changing the transcript to $(c, h := g^z)$ directly instead, but this is of course insecure since one could fake a valid transcript by picking u at random, then computing the forged transcript $(c \leftarrow \mathcal{H}(u, m), h \leftarrow u/y^c)$. So it is simply not enough to offer (c, h) for verification even if $c = \mathcal{H}(hy^c, m)$ as expected: knowledge of z is absolutely required for the verification to make sense (notice that z is actually unknown to the faker in the above forgery, only g^z is obtained).

Yet, there is an alternative to exhibiting the actual z, namely, one could offer a zero-knowledge argument of its knowledge instead, even in a succinct and non-interactive way (that is, one could offer a SNARK). Indeed, the pair $(z, h = g^z)$ has precisely the same nature of the signing key pair, an idea previously used to obtain identity-based signatures [15]. Thus, in principle it could be used to produce a one-time signature to play the role of a SNARK. To that end, just prepare a pair (d, w) such that $d = \mathcal{G}(g^w h^d)$.

This idea is clearly less efficient than showing a Schnorr transcript (c, z) since the full signature is now (c, h, d, w) or at least (u, d, w), but it paves the way to obtaining blind signatures systematically.

3.2 The general *BZ* construction

We will closely follow [17], adapting their definitions on demand.

Definition 1 (Canonical Three-move Blind-signature-from-Zero-knowledge Scheme). *A* canonical three-move *B*lind-signature-from-*Z*ero-knowledge scheme *BZ is a tuple of algorithms* $BZ = (BZ.PG, BZ.KG, BZ.S = (BZ.S_1, BZ.S_2), BZ.U = (BZ.U_1, BZ.U_2), BZ.Ver)$ *where:*

[3] Technically the generator g and the public key y should be included in the hash as well, but we follow a common convention (adopted e.g. in [17]) and omit them for brevity and clarity.

Fig. 1. The canonical BZ scheme

Signer $BZ.S(sk)$		User $BZ.U(pk, m)$
$(\hat{u}, \hat{v}, \mathsf{st}_S) \xleftarrow{\$} BZ.S_1(sk)$	$\xrightarrow{\hat{u},\hat{v}}$	
	$\xleftarrow{\hat{c},\hat{d}}$	$(\hat{c}, \hat{d}, \mathsf{st}_U) \xleftarrow{\$} BZ.U_1(pk, \hat{u}, \hat{v}, m)$
$\hat{w} \leftarrow BZ.S_2(\mathsf{st}_S, \hat{c}, \hat{d})$	$\xrightarrow{\hat{w}}$	$\sigma \leftarrow BZ.U_2(\mathsf{st}_U, \hat{w})$

- *The randomized parameter generation algorithm $BZ.PG$ takes a security parameter 1^κ as input and returns system parameters par.*
- *The randomized key generation algorithm $BZ.KG$ takes par as input and creates a key pair (pk, sk). In turn, the public key defines challenge spaces $(\mathcal{C}, \mathcal{D}) := \mathcal{CD}(pk)$, and is known to all parties.*
- *The first, randomized signer algorithm $BZ.S_1$ takes as input the secret key sk and returns commitments $(\hat{u}, \hat{v})$ and the signer's state st_S.*
- *The first, randomized user algorithm $BZ.U_1$ takes as input the public key pk, commitments $(\hat{u}, \hat{v})$, and a message m, and returns challenges $(\hat{c} \in \mathcal{C}, \hat{d} \in \mathcal{D})$ and the user's state st_U.*
- *The second, deterministic signer algorithm $BZ.S_2$ takes as input the signer's state st_S, which includes the secret key sk and the commitments $(\hat{u}, \hat{v})$, together with challenges $(\hat{c} \in \mathcal{C}, \hat{d} \in \mathcal{D})$, and returns the response $\hat{w}$.*
- *The second, deterministic user algorithm $BZ.U_2$ takes as input the user's state st_U, which includes the public key pk, the commitments $(\hat{u}, \hat{v})$, the message m, and the challenges $(\hat{c} \in \mathcal{C}, \hat{d} \in \mathcal{D})$, together with response $\hat{w}$, and returns a signature σ, or $\perp$ in case of failure.*
- *The deterministic verification algorithm $BZ.Ver$ takes as input the public key pk, a signature σ, and a message m, and returns 1 to indicate acceptance or 0 to indicate rejection (if $\sigma = \perp$ the output is always 0).*

The core *BZ* ideas and properties: The BZ methodology adopts two core ideas to obtain blind signatures.

- The first core idea is to derive a one-time key pair $(\hat{h}, z)$ from the z component of a traditional Schnorr transcript $(\hat{u}, \hat{c}, z)$, and replace z by a SNARK of its knowledge, that is, by a commitment-challenge-response triple $(\hat{v}, \hat{d}, \hat{w})$ verifiable under $\hat{h}$. For instance, in a discrete logarithm setting the one-time key pair would be $(\hat{h} := g^z, z)$, and the triple $(\hat{v}, \hat{d}, \hat{w})$ would satisfy $\hat{v} = g^{\hat{w}} \hat{h}^{\hat{d}}$. This effectively introduces one more unknown variable (the randomness involved in the SNARK creation) without increasing the net number of linear equations that are not protected in exponents.
- The second core idea is how the blinding applies in a way that is compatible with the first idea above. Specifically, we seek to blind commitments using (multiplicative) *permutations* and (additive) *displacements* in the space of

possible commitments, and then blind challenges with the *inverse* permutations as used to blind the commitments. The nature of those 'permutations' and 'displacements' depends on the specific primitive/hard problem underlying each instantiation of the methodology. This mimics the way that Schnorr signatures are created, in the sense that hash values are transformed by means of a multiplicative permutation (namely, multiplication by the private key) and an additive displacement (namely, adding the random nonce). It also contrasts with more common exclusively additive blinding mechanisms as used in blind Schnorr signatures and variants, whereby commitments are blinded by adding a linear combination of the public key and the generator, and challenges are blinded by adding one of the coefficients of that linear combination.

We stress that the BZ methodology is *non-black-box*, in the sense that the actual permutation and displacement operations make direct use if the internals of the underlying primitive, and thus part of the analysis is necessarily deferred to the actual instantiations.

We wish the instantiations of the BZ scheme to satisfy the following three basic informal properties, that will be formalized later:

- *Perfect correctness:* for any signature σ that genuinely results from $BZ.U_2$ after a protocol session with a key pair (pk, sk) and a message m, it must hold that $BZ.Ver(pk, \sigma, m) = 1$;
- *Perfect blindness:* if $BZ.S$ chooses two messages (m_0, m_1), establishes two sessions with $BZ.U$ with transcripts (T_0, T_1), and ends up observing the two signatures (σ_0, σ_1), then $BZ.S$ cannot tell whether T_0 corresponds to m_0 (and T_1 corresponds to m_1), or T_0 corresponds to m_1 (and T_1 corresponds to m_0) any better than by random guessing;
- *One-more unforgeability:* if $BZ.U$ interacts with $BZ.S$ in ℓ protocol sessions, then $BZ.U$ obtains no more than ℓ blind signatures (namely, only from those sessions that reach completion).

We first formalize the latter two notions (blindness in Section 4.1, one-more unforgeability in Section 4.2), then prove all of these properties for two instantiations of the BZ construction, namely:

- $BZ[\mathsf{DL}]$, whose security stems from the Discrete Logarithm assumption and Schnorr signatures [21] (DL for short);
- $BZ[\mathsf{CSI}]$, whose security stems from the Commutative Supersingular Isogeny Diffie-Hellman assumption [9] and CSI-FiSh signatures [7] (CSI for short).

For these schemes the standard MitM security (defined in Remark 1) is unattainable, which suggests that security against impersonation under concurrent attacks (IMP-CA), defined in Section 4.3, may be the strongest security notion that can be achieved within the BZ methodology[4].

[4] The notion of one-more MitM unforgeability [17, Definition 4.2] is stronger than IMP-CA, but attaining it, if possible at all, remains elusive.

Fig. 2. The canonical *IZ* scheme

Prover $IZ.P(sk)$		Verifier $IZ.U(pk, m)$
$(\hat{u}, \hat{v}, \mathsf{st}_P) \xleftarrow{\$} IZ.P_1(sk)$	$\xrightarrow{\hat{u},\hat{v}}$	
	$\xleftarrow{\hat{c},\hat{d}}$	$\hat{c} \xleftarrow{\$} \mathcal{C}$, $\hat{d} \xleftarrow{\$} \mathcal{D}$
$\hat{w} \leftarrow IZ.P_2(\mathsf{st}_P, \hat{c}, \hat{d})$	$\xrightarrow{\hat{w}}$	$b \leftarrow IZ.Ver(pk, \hat{u}, \hat{v}, \hat{c}, \hat{d}, \hat{w})$

The proofs of the several properties will be tailored to those instantiations by the very nature of the underlying algebraic properties: perfect correctness in Theorems 1 and 6, perfect blindness in Theorems 2 and 7, unattainability of standard MitM security in Theorems 3 and 8, reduction of one-more unforgeability to impersonation under concurrent attacks security in Theorems 4 and 9, reduction of impersonation under concurrent attacks security to the one-more discrete logarithm or one-more commutative supersingular isogeny in Theorems 5 and 10.

3.3 The underlying *IZ* identification scheme

The unforgeability properties of *BZ* instantiations will be related to an underlying identification scheme, *IZ*, which we now define. Again, we closely follow [17], adapting their definitions on demand.

Definition 2 (Canonical Three-move Identification-from-Zero-knowledge Scheme). *A* canonical three-move *I*dentification-from-*Z*ero-knowledge scheme *IZ is a tuple of algorithms* $IZ = (IZ.PG, IZ.KG, IZ.P = (IZ.P_1, IZ.P_2), IZ.Ver)$ *where:*

- *The randomized parameter generation algorithm $IZ.PG$ takes a security parameter 1^κ as input and returns system parameters par.*
- *The randomized key generation algorithm $IZ.KG$ takes par as input and creates a key pair (pk, sk). In turn, the public key defines challenge spaces $(\mathcal{C}, \mathcal{D}) := \mathcal{CD}(pk)$, and is known to all parties.*
- *The first, randomized signer algorithm $IZ.P_1$ takes as input the secret key sk and returns commitments $(\hat{u}, \hat{v})$ and the prover's state st_P.*
- *The second, deterministic prover algorithm $IZ.P_2$ takes as input the prover's state st_P, which includes the secret key sk and the commitments $(\hat{u}, \hat{v})$, together with challenges $(\hat{c} \in \mathcal{C}, \hat{d} \in \mathcal{D})$, and returns the response $\hat{w}$.*
- *The deterministic verification algorithm $IZ.Ver$ takes as input the public key pk, commitments $(\hat{u}, \hat{v})$, challenges $(\hat{c} \in \mathcal{C}, \hat{d} \in \mathcal{D})$, and response $\hat{w}$, and returns 1 to indicate acceptance or 0 to indicate rejection.*

4 Security notions

4.1 Blindness

We adapt our definition of BZ blindness from [17, Definition 5.3]. Consider the following *blindness game* with its accompanying oracles Init, U_1, and U_2:

Game 1 $\mathbf{Blind}_{BZ}$

```
$par \leftarrow BZ.PG(1^\kappa)$
$b \xleftarrow{\$} \{0,1\}, \quad b_0 \leftarrow b, \quad b_1 \leftarrow 1-b$
$(pk, sk) \xleftarrow{\$} BZ.KG(par)$   ▷ honest signer model
$sess_0 \leftarrow sess_1 \leftarrow \mathsf{none}$
$b' \xleftarrow{\$} \mathcal{A}^{\mathrm{Init},\mathrm{U}_1,\mathrm{U}_2}(pk, sk)$
return $\big(b = b'\big)\ ?\ 1\ :\ 0$
```

Oracle 1 $\mathrm{Init}(\tilde{m}_0, \tilde{m}_1)$

```
if $sess_0 \neq \mathsf{none}$ or $sess_1 \neq \mathsf{none}$ : return $\bot$   ▷ call Init only once
$m_0 \leftarrow \tilde{m}_0, \quad m_1 \leftarrow \tilde{m}_1$
$sess_0 \leftarrow sess_1 \leftarrow \mathsf{init}$
```

Oracle 2 $\mathrm{U}_1(i, \hat{u}, \hat{v})$

```
if $i \notin \{0,1\}$ or $sess_i \neq \mathsf{init}$ : return $\bot$
$sess_i \leftarrow \mathsf{open}$
$u_i \leftarrow \hat{u}, \quad v_i \leftarrow \hat{v}$
$(c_i, d_i, \mathsf{st}_{i,U}) \leftarrow BZ.U_1(pk, u_i, v_i, m_{b_i})$
return $i, c_i, d_i$
```

Oracle 3 $\mathrm{U}_2(i, \hat{w})$

```
if $i \notin \{0,1\}$ or $sess_i \neq \mathsf{open}$ : return $\bot$
$sess_i \leftarrow \mathsf{closed}$
$w_i \leftarrow \hat{w}$
$\sigma_{b_i} \leftarrow BZ.U_2(\mathsf{st}_{i,U}, w_i)$
if $sess_0 = sess_1 = \mathsf{closed}$ :
    if $\sigma_0 = \bot$ or $\sigma_1 = \bot$ : return $(\bot, \bot)$
    return $(\sigma_0, \sigma_1)$
return $i, \mathsf{closed}$
```

The *advantage* of an adversary $\mathcal{A}$ in game $\mathbf{Blind}_{BZ}$ is defined as $\mathbf{Adv}^{\mathbf{Blind}}_{BZ}(\mathcal{A}) := \Big|\Pr[\mathbf{Blind}^{\mathcal{A}}_{BZ} \Rightarrow 1] - 1/2\Big|$.

Let BZ be a canonical three-move blind-signature-from-zero-knowledge scheme. We say that BZ is *perfectly blind* if for all (even unbounded) adversaries $\mathbf{Adv}^{\mathbf{Blind}}_{BZ}(\mathcal{A}) = 0$.

4.2 One-more unforgeability

We adapt our definition of BZ one-more unforgeability from [17, Definition 5.4]. Consider the following *one-more unforgeability game* with its accompanying oracles S_1 and S_2:

Game 2 $\mathbf{OMUF}_{BZ}$

```
$par \leftarrow BZ.PG(1^{\kappa})$
$(pk, sk) \xleftarrow{\$} BZ.KG(par)$   ▷ honest signer model
$sid \leftarrow 0$   ▷ no session yet
$(m_1, \sigma_1), \ldots, (m_{\ell(\mathcal{A})}, \sigma_{\ell(\mathcal{A})}) \xleftarrow{\$} \mathcal{A}^{\mathrm{S}_1, \mathrm{S}_2, \mathrm{GH}}(pk)$
if $\exists i \neq j : m_i = m_j$ or $\exists k \in \{1 \ldots \ell(\mathcal{A})\} : BZ.Ver(pk, \sigma_k, m_k) = 0$ : return 0
$\mathrm{Q}_{\mathrm{S}_1}(\mathcal{A}) \leftarrow \#\{1 \leq k \leq sid \mid sess_k = \mathsf{open}\}$   ▷ abandoned sessions
$\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) \leftarrow \#\{1 \leq k \leq sid \mid sess_k = \mathsf{closed}\}$
return $\big(\ell(\mathcal{A}) \geq \mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) + 1\big)$ ? 1 : 0
```

Oracle 4 $\mathrm{S}_1(\cdot)$

```
$sid \mathrel{+}= 1$
$sess_{sid} \leftarrow \mathsf{open}$
$(\hat{u}_{sid}, \hat{v}_{sid}, \mathsf{st}_{sid,S}) \xleftarrow{\$} BZ.S_1(sk)$
return $sid, \hat{u}_{sid}, \hat{v}_{sid}$
```

Oracle 5 $\mathrm{S}_2(sid, \hat{c}, \hat{d})$

```
if $sess_{sid} \neq \mathsf{open}$ : return $\perp$
$sess_{sid} \leftarrow \mathsf{closed}$
$\hat{w}_{sid} \leftarrow BZ.S_2(\mathsf{st}_{sid,S}, \hat{c}, \hat{d})$
return $\hat{w}_{sid}$
```

The *advantage* of an adversary $\mathcal{A}$ in game $\mathbf{OMUF}_{BZ}$ is defined as $\mathbf{Adv}^{\mathbf{OMUF}}_{BZ}(\mathcal{A}) := \Pr[\mathbf{OMUF}^{\mathcal{A}}_{BZ} \Rightarrow 1]$ and denote its running time as $\mathbf{Time}^{\mathbf{OMUF}}_{BZ}(\mathcal{A})$.

Let BZ be a canonical three-move blind-signature-from-zero-knowledge scheme. We say that BZ is $(\epsilon, \tau, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF secure in the random oracle model if for all adversaries $\mathcal{A}$ satisfying $\mathbf{Time}^{\mathbf{OMUF}}_{BZ}(\mathcal{A}) \leq \tau$, $\mathrm{Q}_{\mathrm{S}_1}(\mathcal{A}) \leq \mathrm{Q}_{\mathrm{S}_1}$, $\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) \leq \mathrm{Q}_{\mathrm{S}_2}$, it holds that $\mathbf{Adv}^{\mathbf{OMUF}}_{BZ}(\mathcal{A}) \leq \epsilon$, where Q_{GH} is the maximum between the number of queries to the joint GH oracle that models a pair of hash functions[5] $\mathcal{G}$ and $\mathcal{H}$. An adversary $\mathcal{A}$ that satisfies these constraints is said to *break* the $(\epsilon, \tau, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF security of BZ if $\mathbf{Adv}^{\mathbf{OMUF}}_{BZ}(\mathcal{A}) > \epsilon$.

Remark 1. In the *standard* MitM experiment, the winning condition is relaxed so that there must only *exist* a successful session with the verifier sporting a transcript that does not result from a closed session with the prover.

4.3 (Non-)impersonation under concurrent attacks

Now we present the definition of unforgeability we adopt for IZ from [5], namely, security against *impersonation under concurrent attacks*, or IMP-CA for short.

An IMP-CA adversary is a pair $\mathcal{A} := (\mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}, \mathcal{P}^{\mathrm{C}_1,\mathrm{C}_2})$ of randomized polynomial-time algorithms, the *cheating verifier* $\mathcal{V}$ and *cheating prover* $\mathcal{P}$. The IMP-CA game that $\mathcal{A}$ plays has two phases.

In the first phase, $\mathcal{V}$ is initialized with the public key pk and a random tape, and then interacts with the prover P via the oracle $\mathrm{P}_1(\cdot)$, which opens a new independent session with the prover and returns the session identifier pid and a commitment pair (u_{pid}, v_{pid}), and the oracle $\mathrm{P}_2(pid, c_{pid}, d_{pid})$, which closes session pid and returns a response w_{pid} to the challenge pair (c_{pid}, d_{pid}), or $\perp$ if pid is not open. These interactions can be arbitrarily interleaved. Eventually, $\mathcal{V}$ outputs some state information $\mathbf{st}_{\mathcal{V}}$ and stops, ending the first phase.

In the second phase of the game, the cheating prover $\mathcal{P}$ has its own state $\mathbf{st}_{\mathcal{P}}$ initialized with $\mathbf{st}_{\mathcal{V}}$, while the verifier V gets the public key pk and a fresh

[5] The use of two hash functions in practice enables an optimization where one challenge is omitted from the signature and derived on demand from existing information.

random tape, and then interacts with $\mathcal{P}$ via the oracles $\mathrm{C}_1(\cdot)$, which takes no input and returns a commitment pair (u, v), and $\mathrm{C}_2(c, d)$, which returns the response w corresponding to that commitment pair and the given challenge pair. These oracles make use of corresponding procedures, respectively $\mathcal{P}.\mathrm{F}_1(\mathbf{st}_\mathcal{P})$ and $\mathcal{P}.\mathrm{F}_2(\mathbf{st}_\mathcal{P}, c, d)$, that extract the required information from $\mathbf{st}_\mathcal{P}$ and update that state accordingly, mimicking Algorithms $\mathit{IZ}.P_1$ and $\mathit{IZ}.P_2$. $\mathcal{P}$ returns the transcript of the most favorable interaction with V (where presumably V accepts). Adversary $\mathcal{A}$ wins if V indeed accepts in this interaction.

This is summarized in game $\mathbf{IMP\text{-}CA}_{\mathit{IZ}}$ and its associated oracles P_1, P_2, C_1, and C_2.

Game 3 $\mathbf{IMP\text{-}CA}_{\mathit{IZ}}$

```
▷ 1st phase:
$par \leftarrow \mathit{IZ}.PG(1^\kappa)$,   $(pk, sk) \xleftarrow{\$} \mathit{IZ}.KG(par)$   ▷ honest signer model
$pid \leftarrow vid \leftarrow 0$   ▷ no session yet
$\mathbf{st}_\mathcal{V} \xleftarrow{\$} \mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}(par, pk)$,   $\mathbf{st}_\mathcal{P} \leftarrow \mathbf{st}_\mathcal{V}$
$(\hat{u}, \hat{v}, \hat{c}, \hat{d}, \hat{w}) \leftarrow \mathcal{P}^{\mathrm{C}_1,\mathrm{C}_2}(\mathbf{st}_\mathcal{P})$
$b \leftarrow \mathit{IZ}.Ver(pk, \hat{u}, \hat{v}, \hat{c}, \hat{d}, \hat{w})$
return $b$
```

Oracle 6 $\mathrm{P}_1(\cdot)$

```
$pid \mathrel{+}= 1$
$pSess_{pid} \leftarrow$ open
$(\hat{u}_{pid}, \hat{v}_{pid}, \mathsf{st}_{pid,P}) \xleftarrow{\$} \mathit{IZ}.P_1(sk)$
return $pid, \hat{u}_{pid}, \hat{v}_{pid}$
```

Oracle 7 $\mathrm{P}_2(pid, \hat{c}, \hat{d})$

```
if $pSess_{pid} \neq$ open : return $\bot$
$pSess_{pid} \leftarrow$ closed
$\hat{w}_{pid} \leftarrow \mathit{IZ}.P_2(\mathsf{st}_{pid,P}, \hat{c}, \hat{d})$
return $\hat{w}_{pid}$
```

Oracle 8 $\mathrm{C}_1(\cdot)$

```
$vid \mathrel{+}= 1$
$vSess_{vid} \leftarrow$ open
$(\hat{u}_{vid}, \hat{v}_{vid}) \xleftarrow{\$} \mathrm{F}_1(\mathbf{st}_\mathcal{P})$   ▷ NB: updates $\mathbf{st}_\mathcal{P}$
return $vid, \hat{u}_{vid}, \hat{v}_{vid}$
```

Oracle 9 $\mathrm{C}_2(vid, \hat{c}, \hat{d})$

```
if $pSess_{vid} \neq$ open : return $\bot$
$vSess_{vid} \leftarrow$ closed
$\hat{w}_{vid} \leftarrow \mathrm{F}_2(\mathbf{st}_\mathcal{P}, \hat{c}, \hat{d})$   ▷ NB: updates $\mathbf{st}_\mathcal{P}$
return $\hat{w}_{vid}$
```

The *advantage* of an adversary $\mathcal{A}$ in game $\mathbf{IMP\text{-}CA}_{\mathit{IZ}}$ is defined as $\mathbf{Adv}_{\mathit{IZ}}^{\mathbf{IMP\text{-}CA}}(\mathcal{A}, \kappa) := \Pr[\mathbf{IMP\text{-}CA}_{\mathit{IZ}}^{\mathcal{A}} \Rightarrow 1]$, where the probability is taken over the coins of $\mathit{IZ}.PG$, the coins of $\mathcal{V}$, the coins of the prover sessions, and the coins of V. Denote its running time as $\mathbf{Time}_{\mathit{IZ}}^{\mathbf{IMP\text{-}CA}}(\mathcal{A})$. Let IZ be a canonical three-move identification-from-zero-knowledge scheme. We say that IZ is (ϵ, τ)-IMP-CA secure[6], or $(\epsilon, \tau, \mathrm{Q}_{\mathrm{P}_1}, \mathrm{Q}_{\mathrm{P}_2}, \mathrm{Q}_V)$-IMP-CA secure when one wants to specify explicitly the numbers $\mathrm{Q}_{\mathrm{P}_1}$, $\mathrm{Q}_{\mathrm{P}_2}$, Q_V of calls to P_1, calls to P_2, and interactions with the verifier V respectively, if for all adversaries $\mathcal{A}$ satisfying $\mathbf{Time}_{\mathit{IZ}}^{\mathbf{IMP\text{-}CA}}(\mathcal{A}) \leq \tau$ (and performing no more than the stated numbers of oracle calls), it holds that $\mathbf{Adv}_{\mathit{IZ}}^{\mathbf{IMP\text{-}CA}}(\mathcal{A}, \kappa) \leq \epsilon$. An adversary $\mathcal{A}$ that satisfies these constraints is said to *break* the (ϵ, τ)-IMP-CA security of IZ if $\mathbf{Adv}_{\mathit{IZ}}^{\mathbf{IMP\text{-}CA}}(\mathcal{A}, \kappa) > \epsilon$.

[6] The original definition from [5] further requires $\tau \in O(\mathrm{poly}(\kappa))$ and $\epsilon \in O(\mathrm{poly}(\kappa))$.

4.4 One-more discrete logarithm and one-more class group action

We recap the one-more discrete logarithm (OMDL) assumption as defined in [5], then introduce a natural adaptation to the commutative supersingular isogeny setting, the one-more class group action (OMGA) assumption.

OMDL: An *OMDL adversary* $\mathcal{I}$ is a randomized, polynomial-time algorithm that gets input a prime number n and a generator g of an Abelian group $\mathbb{G}$ or order n, and has access to two oracles: the *discrete logarithm solving oracle* $\mathcal{I}.Sol(g \in \mathbb{G}, h \in \mathbb{G})$ that computes $z \in \mathbb{Z}_n$ such that $g^z = h$, and the *group element challenge oracle* $\mathcal{I}.Ch(\cdot)$ that returns a uniformly random challenge point $h \in \mathbb{G}$ each time it is called.

Adversary $\mathcal{D}$ plays a game where a *discrete logarithm parameter generator* algorithm $PG(1^\kappa)$ is executed to get $par := (n, g)$, then $\mathcal{I}(par)$ is executed with its oracles. Let $h_1 \dots h_{\ell+1} \in \mathbb{G}$ denote the challenges returned by $\mathcal{I}$'s challenge oracle. Adversary $\mathcal{I}$ wins if it outputs a sequence $z_1 \dots z_{\ell+1} \in \mathbb{Z}_n$ satisfying $h_i = g^{z_i}$ after making ℓ calls (or less) to its $\mathcal{I}.Sol$ oracle. Adversary $\mathcal{I}$'s OMDL advantage in winning this game, denoted $\mathbf{Adv}^{\mathbf{OMDL}}_{PG}(\mathcal{I}, \kappa)$, is the probability that $\mathcal{I}$ wins, taken over the coins of $PG(1^\kappa)$, the coins of $\mathcal{I}$ itself, and the coins used by the challenge oracle $\mathcal{I}.Ch(\cdot)$ across its calls. We say that $PG(1^\kappa)$ is *OMDL-secure* if $\mathbf{Adv}^{\mathbf{OMDL}}_{PG}(\mathcal{I}, \kappa)$ is negligible for any OMDL adversary $\mathcal{I}$ of time complexity $O(\mathrm{poly}(\kappa))$.

OMGA: An *OMGA adversary* $\mathcal{I}$ is a randomized, polynomial-time algorithm that gets input an integer N and a supersingular curve A_0 from a set $\mathcal{E}\ell\ell_p$ where the order of the class group action is N, and has access to two oracles: the *class group action solving oracle* $\mathcal{I}.Sol(A \in \mathcal{E}\ell\ell_p^{\mathsf{r}' \times \mathsf{c}}, Q \in \mathcal{E}\ell\ell_p^{\mathsf{r} \times \mathsf{c}})$ that computes a class group action representative $a \in \mathbb{Z}_N^{\mathsf{r} \times \mathsf{c}}$ such that $Q = [a]A$ for the desired matrix dimensions r', r, and c, and the *supersingular curve challenge oracle* $\mathcal{I}.Ch(\mathsf{r}, \mathsf{c})$ that returns uniformly random challenge curves $Q \in \mathcal{E}\ell\ell_p^{\mathsf{r} \times \mathsf{c}}$ each time it is called.

Adversary $\mathcal{I}$ plays a game where a *commutative supersingular isogeny parameter generator* algorithm $PG(1^\kappa)$ is executed to get $par := (N, A_0, C, t, C', T)$, then $\mathcal{I}(par)$ is executed with its oracles. Let $Q_i \in \mathcal{E}\ell\ell_p^{\mathsf{r}_i \times \mathsf{c}_i}$, $1 \leq i \leq \ell + 1$, denote the challenges returned by $\mathcal{I}$'s challenge oracle. $\mathcal{I}$ wins if it outputs a sequence $a_i \in \mathbb{Z}_N^{\mathsf{r}_i \times \mathsf{c}_i}$, $1 \leq i \leq \ell + 1$, satisfying $Q_i = [a_i]A_0$ after making ℓ calls (or less), involving no more than the corresponding numbers of rows and columns, to its $\mathcal{I}.Sol$ oracle. Adversary $\mathcal{I}$'s OMGA advantage in winning this game, denoted $\mathbf{Adv}^{\mathbf{OMGA}}_{IZ}(\mathcal{I}, \kappa)$, is the probability that $\mathcal{I}$ wins, taken over the coins of $PG(1^\kappa)$, the coins of $\mathcal{I}$ itself, and the coins used by the challenge oracle $\mathcal{I}.Ch(\cdot)$ across its calls. We say that $PG(1^\kappa)$ is *OMGA-secure* if $\mathbf{Adv}^{\mathbf{OMGA}}_{IZ}(\mathcal{I}, \kappa)$ is negligible for any OMGA adversary $\mathcal{I}$ of time complexity $O(\mathrm{poly}(\kappa))$.

5 The $BZ[\mathsf{DL}]$ and $IZ[\mathsf{DL}]$ schemes

We now define the $BZ[\mathsf{DL}]$ blind signature scheme and its associated identification scheme $IZ[\mathsf{DL}]$. $BZ[\mathsf{DL}]$ consists of the following algorithms:

Algorithm 1 $BZ[\mathsf{DL}].PG(1^\kappa)$

```
Select a secure Abelian group $\mathbb{G}$ of prime order $n \approx 2^{2\kappa}$, a generator $g \in \mathbb{G}$, and secure hash
    functions $\mathcal{H} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_n^*$ and $\mathcal{G} : \mathbb{G} \to \mathbb{Z}_n^*$.
return $par := (n, g, \mathcal{G}, \mathcal{H})$
```

Algorithm 2 $BZ[\mathsf{DL}].KG(par)$

```
$(n, g, \mathcal{G}, \mathcal{H}) \leftarrow par$
$x \xleftarrow{\$} \mathbb{Z}_n^*, \quad y \leftarrow g^x$
$sk \leftarrow (x, par), \quad pk \leftarrow (y, par)$
return $(pk, sk)$
```

Algorithm 3 $BZ[\mathsf{DL}].S_1(sk)$

```
$(x, par) \leftarrow sk, \quad (n, g, \mathcal{G}, \mathcal{H}) \leftarrow par$
$r \xleftarrow{\$} \mathbb{Z}_n^*, \quad s \xleftarrow{\$} \mathbb{Z}_n^*$
$\hat{u} \leftarrow g^r, \quad \hat{v} \leftarrow g^s, \quad \mathsf{st}_S \leftarrow (sk, r, s)$
return $(\hat{u} \in \mathbb{G}^*, \hat{v} \in \mathbb{G}^*, \mathsf{st}_S)$
```

Algorithm 4 $BZ[\mathsf{DL}].S_2(\mathsf{st}_S, \hat{c} \in \mathbb{Z}_n^*, \hat{d} \in \mathbb{Z}_n^*)$

```
$(sk, r, s) \leftarrow \mathsf{st}_S, \quad (x, par) \leftarrow sk,$
    $(n, g, \mathcal{G}, \mathcal{H}) \leftarrow par$
$z \leftarrow r - \hat{c}x \pmod{n}$    $\triangleright$ $\hat{h} = g^z$ implicitly
$\hat{w} \leftarrow s - \hat{d}z \pmod{n}$
return $\hat{w} \in \mathbb{Z}_n^*$
```

Algorithm 5 $BZ[\mathsf{DL}].U_1(pk, \hat{u} \in \mathbb{G}^*, \hat{v} \in \mathbb{G}^*, m)$

```
$(y, par) \leftarrow pk, \quad (n, g, \mathcal{G}, \mathcal{H}) \leftarrow par$
$\pi \xleftarrow{\$} \mathbb{Z}_n^*, \quad \delta \xleftarrow{\$} \mathbb{Z}_n^*, \quad \rho \xleftarrow{\$} \mathbb{Z}_n^*, \quad \varepsilon \xleftarrow{\$} \mathbb{Z}_n^*$
$u \leftarrow \hat{u}^\pi g^\delta, \quad v \leftarrow \hat{v}^{\rho\pi} g^\varepsilon$
$c \leftarrow \mathcal{H}(u, m), \quad d \leftarrow \mathcal{G}(v)$
$\hat{c} \leftarrow c/\pi \pmod{n}, \quad \hat{d} \leftarrow d/\rho \pmod{n}$
$\mathsf{st}_U \leftarrow (pk, u, \hat{u}, \hat{v}, \pi, \delta, \rho, \varepsilon, d, \hat{c}, \hat{d})$
return $(\hat{c} \in \mathbb{Z}_n^*, \hat{d} \in \mathbb{Z}_n^*, \mathsf{st}_U)$
```

Algorithm 6 $BZ[\mathsf{DL}].U_2(\mathsf{st}_U, \hat{w} \in \mathbb{Z}_n^*)$

```
$(pk, u, \hat{u}, \hat{v}, \pi, \delta, \rho, \varepsilon, d, \hat{c}, \hat{d}) \leftarrow \mathsf{st}_U,$
    $(y, par) \leftarrow pk, \quad (n, g, \mathcal{G}, \mathcal{H}) \leftarrow par$
$\hat{h} \leftarrow \hat{u} y^{-\hat{c}}$
$\triangleright$ check that the signer is honest:
if $\hat{v} \neq g^{\hat{w}} \hat{h}^{\hat{d}}$ : return $\bot$
$w \leftarrow \rho\pi\hat{w} - d\delta + \varepsilon \pmod{n}$
return $\sigma := (u, d, w) \in \mathbb{G}^* \times (\mathbb{Z}_n^*)^2$
```

Algorithm 7 $BZ[\mathsf{DL}].Ver(pk, \sigma \in \mathbb{G}^* \times (\mathbb{Z}_n^*)^2, m)$

```
if $\sigma = \bot$ : return 0
$(y, par) \leftarrow pk, \quad (n, g, \mathcal{G}, \mathcal{H}) \leftarrow par, \quad (u, d, w) \leftarrow \sigma$
$c \leftarrow \mathcal{H}(u, m), \quad h \leftarrow u y^{-c}, \quad v \leftarrow g^w h^d$
return $(d = \mathcal{G}(v))$ ? 1 : 0
```

For $IZ[\mathsf{DL}]$, algorithms $IZ.PG$ and $IZ.KG$ are, respectively, identical to $BZ.PG$ and $BZ.KG$, except that the hash functions are ignored and omitted. The remaining algorithms are as follows.

Algorithm 8 $IZ[\mathsf{DL}].P_1(sk)$

```
$(x, par) \leftarrow sk, \quad (n, g) \leftarrow par$
$r \xleftarrow{\$} \mathbb{Z}_n^*, \quad s \xleftarrow{\$} \mathbb{Z}_n^*$
$\hat{u} \leftarrow g^r, \quad \hat{v} \leftarrow g^s, \quad \mathsf{st}_P \leftarrow (sk, r, s)$
return $(\hat{u} \in \mathbb{G}^*, \hat{v} \in \mathbb{G}^*, \mathsf{st}_P)$
```

Algorithm 9 $IZ[\mathsf{DL}].P_2(\mathsf{st}_P, \hat{c} \in \mathbb{Z}_n^*, \hat{d} \in \mathbb{Z}_n^*)$

```
$(sk, r, s) \leftarrow \mathsf{st}_P, \quad (x, par) \leftarrow sk,$
    $(n, g) \leftarrow par$
$z \leftarrow r - \hat{c}x \pmod{n}$    $\triangleright$ $\hat{h} = g^z$ implicitly
$\hat{w} \leftarrow s - \hat{d}z \pmod{n}$
return $\hat{w} \in \mathbb{Z}_n^*$
```

Algorithm 10 $IZ[\mathsf{DL}].Ver(pk, \hat{u} \in \mathbb{G}^*, \hat{v} \in \mathbb{G}^*, \hat{c} \in \mathbb{Z}_n^*, \hat{d} \in \mathbb{Z}_n^*, \hat{w} \in \mathbb{Z}_n^*)$

```
$(y, par) \leftarrow pk, \quad (n, g) \leftarrow par$
$\hat{h} \leftarrow \hat{u} y^{-\hat{c}}$
return $\left(\hat{v} = g^{\hat{w}} \hat{h}^{\hat{d}}\right)$ ? 1 : 0
```

5.1 Perfect correctness of *BZ*[DL]

Theorem 1. *The proposed $BZ[\mathsf{DL}]$ protocol is perfectly correct.*

Proof. For $\forall par \in BZ[\mathsf{DL}].PG(1^\kappa), \forall(pk, sk) \in BZ[\mathsf{DL}].KG(par), \forall m \in \{0,1\}^*$, let $\sigma := (u, d, w)$ be a genuine signature for m, properly obtained from the protocol.

Suppose by contradiction that $BZ[\mathsf{DL}].Ver(pk, \sigma, m) = 0$. This can only happen (at step 4 of $BZ[\mathsf{DL}].Ver$) if $d \neq \mathcal{G}(v)$, which means that either d is malformed (a contradiction, from the way it is created in step 4 of $BZ[\mathsf{DL}].U_1$), or v is malformed. Because $v = g^w h^d$, $h = ug^{-c}$, and $c = \mathcal{H}(u, m)$, the latter means that either w is malformed or u is malformed, since c matches its definition in step 4 of $BZ[\mathsf{DL}].U_1$. In turn, this means that either $\hat{u} \neq g^z y^{\hat{c}}$, given the way u is defined from $\hat{u}$ in step 3 of $BZ[\mathsf{DL}].U_1$, or $\hat{v} \neq g^{\hat{w}} \hat{h}^{\hat{d}}$, given the way w is defined from $\hat{w}$ in step 4 of $BZ[\mathsf{DL}].U_2$, respectively. But this would have caused $BZ[\mathsf{DL}].U_2$ to abort at its step 3 (supported by step 2), and σ would not have been created, which is contradiction.

Therefore $BZ[\mathsf{DL}].Ver$ must end with $BZ[\mathsf{DL}].Ver(pk, \sigma, m) = 1$ as expected, and $BZ[\mathsf{DL}]$ is thus perfectly correct as claimed. □

5.2 Perfect blindness of *BZ*[DL]

Theorem 2. *The proposed $BZ[\mathsf{DL}]$ protocol is perfectly blind.*

Proof. Let $\mathcal{A}$ be an adversary playing the $\mathbf{Blind}^{\mathcal{A}}_{BZ}$ game. Assume w.l.o.g. that $\mathcal{A}$'s randomness is fixed and $\mathcal{A}$ always finishes both sessions and receives valid signatures (σ_0, σ_1). $\mathcal{A}$'s view after its execution is $(y, m_0, m_1, T_0, T_1, \sigma_0, \sigma_1)$ where $T_i := (\hat{u}_i, \hat{v}_i, \hat{c}_i, \hat{d}_i, \hat{w}_i)$ are the transcripts that $\mathcal{A}$ observes from the i-th signing session and $\sigma := (u_i, d_i, w_i)$, $i \in \{0, 1\}$.

Since $\mathcal{A}$'s randomness is fixed, the only randomness under consideration is that in $BZ[\mathsf{DL}].U_1$ ($BZ[\mathsf{DL}].U_2$ introduces no further randomness), that is, $(\pi_b, \delta_b, \rho_b, \varepsilon_b)$ for each $b \in \{0, 1\}$.

Suppose also w.l.o.g. that $\mathcal{A}$ takes $T_i := (\hat{u}_i, \hat{v}_i, \hat{c}_i, \hat{d}_i, \hat{w}_i)$ to refer to m_b, i.e. $\hat{c}_i = \pi_b^{-1}\mathcal{H}(\hat{u}_i^{\pi_b} g^{\delta_b}, m_b)$ and $\hat{d}_i = \rho_b^{-1}\mathcal{G}(\hat{v}_i^{\rho_b \pi_b} g^{\varepsilon_b})$, *for some specific choice* of $b \in \{0, 1\}$ that $\mathcal{A}$ makes.

NB: $\hat{w}_i$ does not introduce new, independent constraints, since the signatures are presumed valid and this quantity is deterministically obtained from the other ones.

To prove the theorem, we argue that $\exists(\pi_b, \delta_b, \rho_b, \varepsilon_b) \in (\mathbb{Z}_n^*)^4$, taken always from the same distribution independently from b, such that $\hat{c}_i = \pi_b^{-1}\mathcal{H}(\hat{u}_i^{\pi_b} g^{\delta_b}, m_b)$ and $\hat{d}_i = \rho_b^{-1}\mathcal{G}(\hat{v}_i^{\rho_b \pi_b} g^{\varepsilon_b})$ for *either* choice of $b \in \{0, 1\}$. In other words, we will show that both transcripts are equally consistent with, and suggest no preference for, either $b = 0$ or $b = 1$.

Consider the effect of taking $\mu_b \leftarrow g^{\zeta_b}$ and $\nu_b \leftarrow g^{\chi_b}$, for some uniformly distributed $\zeta_b, \chi_b \in \mathbb{Z}_n^*$, and computing $\gamma_b \leftarrow \mathcal{H}(\mu_b, m_b)$ and $\lambda_b \leftarrow \mathcal{G}(\nu_b)$, for *either* choice of $b \in \{0, 1\}$. We want to be able to write $\hat{c}_i = \pi_b^{-1}\gamma_b$ and $\hat{d}_i =$

$\rho_b^{-1}\lambda_b$, so we can just take $\pi_b \leftarrow \gamma_b/\hat{c}_i$ and $\rho_b \leftarrow \lambda_b/\hat{d}_i$: since the outputs from $\mathcal{H}$ and $\mathcal{G}$ are uniformly distributed over $\mathbb{Z}_n^*$, the distributions of π_b and ρ_b defined this way are uniform over $\mathbb{Z}_n^*$ as well.

Now we also want $\mu_b = g^{\zeta_b} = u_i = \hat{u}_i^{\pi_b} g^{\delta_b} = g^{\pi_b r_i + \delta_b}$ and $\nu_b = g^{\chi_b} = v_i = \hat{v}_i^{\rho_b \pi_b} g^{\varepsilon_b} = g^{\rho_b \pi_b s_i + \varepsilon_b}$, revealing the constraints $\zeta_b = \pi_b r_i + \delta_b \pmod n$ and $\chi_b = \rho_b \pi_b s_i + \varepsilon_b \pmod n$, whereby we can take $\delta_b \leftarrow \zeta_b - \pi_b r_i \pmod n$ and $\varepsilon_b \leftarrow \chi_b - \rho_b \pi_b s_i \pmod n$, respectively: as long as ζ_b and χ_b are uniformly distributed over $\mathbb{Z}_n^*$, so are δ_b and ε_b, provided that $\delta_b \neq 0$ and $\varepsilon_b \neq 0$.

NB: only the signer knows r_i and s_i, but the point is to show that *some* suitable tuple $(\pi_b, \delta_b, \rho_b, \varepsilon_b)$ exists at all for either choice of b.

In other words, whichever choice of $b \in \{0, 1\}$ the adversary makes to try and associate a message m_b of their choice (and its corresponding signature σ_b) to a transcript T_0 or T_1, it is always possible to find a suitable tuple $(\pi_b, \delta_b, \rho_b, \varepsilon_b) \in (\mathbb{Z}_n^*)^4$ suggesting that choice m_b, without establishing any preference for either one. Therefore, the $BZ[\mathsf{DL}]$ scheme is perfectly blind, as claimed. □

5.3 Unattainability of standard MitM security for $\boldsymbol{IZ}[\mathsf{DL}]$

Theorem 3. *Standard meddler-in-the-middle (MitM) security is unachievable for the underlying identification scheme* $IZ[\mathsf{DL}]$.

Proof. A MitM adversary $\mathcal{A}$:

1. chooses $\alpha \xleftarrow{\$} \mathbb{Z}_n^*$;
2. intercepts $(\hat{u}, \hat{v})$ from the prover and sends $(\tilde{u} := \hat{u} g^\alpha, \tilde{v} := \hat{v})$ to the verifier;
3. forwards $(\tilde{c} := \hat{c}, \tilde{d} := \hat{d})$ from the verifier to the prover;
4. intercepts the $\hat{w}$ from the prover and completes the attack by setting $\tilde{w} := \hat{w} - \alpha\hat{d} \pmod n$.

Now $(\tilde{u}, \tilde{v}, \tilde{c}, \tilde{d}, \tilde{w}) \neq (\hat{u}, \hat{v}, \hat{c}, \hat{d}, \hat{w})$, but since $\tilde{h} = \tilde{u} y^{-\tilde{c}} = (\hat{u} g^\alpha) y^{-\hat{c}} = (\hat{u} y^{-\hat{c}}) g^\alpha = \hat{h} g^\alpha$, this tuple still satisfies $\tilde{v} = \hat{v} = g^{\hat{w}} \hat{h}^{\hat{d}} = g^{\hat{w} - \alpha\hat{d}} \hat{h}^{\hat{d}} g^{\alpha\hat{d}} = g^{\tilde{w}} (\hat{h} g^\alpha)^{\tilde{d}} = g^{\tilde{w}} \tilde{h}^{\tilde{d}}$, and $\mathcal{A}$ wins the standard MitM experiment with advantage 1. □

5.4 Breaking $\boldsymbol{IZ}[\mathsf{DL}]$ OMUF implies breaking IMP-CA

We now show that, given an adversary $\mathcal{A}$ that breaks the one-more unforgeability (OMUF) of $BZ[\mathsf{DL}]$, one can build an adversary $\mathcal{B}$ that breaks the impersonation against concurrent attacks (IMP-CA) security of the underlying $IZ[\mathsf{DL}]$ scheme.

For convenience, in this section we suppress an optimization that is implicit in the protocol, namely, the omission of c from the signature contents. This value is known by the user in line 4 of Algorithm $BZ[\mathsf{DL}].U_1(pk, \hat{u}, \hat{v}, m)$ and could be passed to Algorithm $BZ[\mathsf{DL}].U_2(\mathsf{st}_U, \hat{w})$ via st_U, but is not included explicitly in σ for being redundant for verification, since it can be recovered from existing information in line 3 of Algorithm $BZ[\mathsf{DL}].\mathit{Ver}(pk, \sigma, m)$. Yet, the following description becomes simpler if c is taken to be readily available as part of the (extended) signature σ'.

Adversary 1 $\mathcal{B}^{\mathrm{P}_1,\mathrm{P}_2,\mathrm{C}_1,\mathrm{C}_2,\mathrm{Ver}}(pk)$

```
1: $((m_1, \sigma'_1), \ldots, (m_{\ell(\mathcal{A})}, \sigma'_{\ell(\mathcal{A})})) \xleftarrow{\$} \mathcal{A}^{\mathrm{S}_1,\mathrm{S}_2,\mathrm{GH}}(pk)$
2: for $i \in \{1 \ldots \ell(\mathcal{A})\}$ :
3:     $(c'_i, u'_i, d'_i, w'_i) \leftarrow \sigma'_i$    ▷ NB: $c'_i$ explicitly attached to $\sigma'_i$
4:     $h'_i \leftarrow u'_i / y^{c'_i}$,   $v'_i \leftarrow g^{w'_i} {h'_i}^{d'_i}$
5:     $\mathrm{GH}(u'_i, v'_i, m_i)$
6:     $vid \leftarrow vSess_{u'_i, v'_i, m_i}$
7:     $b_i \leftarrow \mathrm{Ver}(vid, w'_i)$
```

Procedure 1 $\mathrm{S}_1(\cdot)$

```
1: $(pid, \hat{u}_{pid}, \hat{v}_{pid}) \xleftarrow{\$} \mathrm{P}_1(\cdot)$
2: return $pid, \hat{u}_{pid}, \hat{v}_{pid}$
```

Procedure 2 $\mathrm{S}_2(pid, \hat{c}, \hat{d})$

```
1: $\hat{w}_{pid} \leftarrow \mathrm{P}_2(pid, \hat{c}, \hat{d})$
2: return $\hat{w}_{pid}$
```

Procedure 3 $\mathrm{GH}(u', v', m)$

```
1: if $\mathcal{GH}[u', v', m] \neq \perp$ : return $\mathcal{H}[u', v', m]$
2: $(vid, c', d') \xleftarrow{\$} \mathrm{C}_2(u', v')$,   $\mathcal{GH}[u', v', m] \leftarrow (c', d')$
3: $vSess_{u', v' m} \leftarrow vid$
4: return $\mathcal{GH}[u', v', m]$
```

The following theorem, which adapts [18, Theorem 5.6] to the three-move scheme-from-zero-knowledge setting, establishes that adversary $\mathcal{B}$ does indeed achieve the aforementioned goal:

Theorem 4. *Let $\mathcal{H} : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_n^*$ and $\mathcal{G} : \mathbb{G} \to \mathbb{Z}_n^*$ be hash functions, and let $BZ[\mathsf{DL}][\mathcal{G}, \mathcal{H}]$ and $IZ[\mathsf{DL}]$ be instantiations of the proposed blind signature scheme and its underlying identification scheme on top of the* DL. *If $IZ[\mathsf{DL}]$ is $(\epsilon', \tau', \mathrm{Q}_{\mathrm{P}_1}, \mathrm{Q}_{\mathrm{P}_2}, \mathrm{Q}_V)$-IMP-CA secure, then $BZ[\mathsf{DL}][\mathcal{G}, \mathcal{H}]$ is $(\epsilon, \tau, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF secure in the random oracle model, where $\epsilon = \epsilon'$, $\tau = \tau'$, $\mathrm{Q}_{\mathrm{S}_1} = \mathrm{Q}_{\mathrm{P}_1}$, $\mathrm{Q}_{\mathrm{S}_2} = \mathrm{Q}_{\mathrm{P}_2}$, and $\mathrm{Q}_{\mathrm{GH}} = \mathrm{Q}_V - \mathrm{Q}_{\mathrm{S}_2} - 1$.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the $(\epsilon, t, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF of $BZ[\mathsf{DL}][\mathcal{G}, \mathcal{H}]$ in the random oracle model. Adversary $\mathcal{B}$, as we constructed it, runs the $\mathbf{IMP\text{-}CA}_{IZ}$ experiment and perfectly simulates $\mathcal{A}$'s oracles S_1, S_2, and GH via its own oracles P_1, P_2, C_1, and C_2. Note that $\mathcal{B}$ calls P_2 at most $\mathrm{Q}_{\mathrm{P}_2} = \mathrm{Q}_{\mathrm{S}_2}$ times over the course of its simulation, and moreover, $\mathrm{Q}_{\mathrm{P}_2}(\mathcal{B}) = \mathrm{Q}_{\mathrm{S}_2}(\mathcal{A})$. We show that $\mathcal{B}$ breaks the $(\epsilon', t', \mathrm{Q}_{\mathrm{P}_1}, \mathrm{Q}_{\mathrm{P}_2}, \mathrm{Q}_V)$-IMP-CA security of $IZ[\mathsf{DL}]$.

Suppose that $\mathcal{A}$ is successful, i.e. it outputs $\ell(\mathcal{A}) \geq \mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) + 1 = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B}) + 1$ valid signatures on distinct messages and the number of closed sessions with the signer is at most $\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B})$. Since all messages m_i are distinct, each signature corresponds to a distinct session vid_i with the oracle C_2 via the relation $\mathrm{GH}(u', v', m_i) = \mathrm{C}_2(u', v')$ from line 2 of Procedure GH. Since also $\sigma'_i = (c'_i, u'_i, d'_i, w'_i,)$ is a valid signature on m_i, we know that $\mathrm{GH}(u'_i, g^{w'_i} {h'_i}^{d'_i}, m_i) = \mathrm{GH}(u'_i, v'_i, m_i) = \mathrm{C}_2(u'_i, v'_i)$ with $h'_i = u'_i / y^{c'_i}$. Therefore, $\mathcal{B}$ can make a successful query to oracle $Ver(vid, w'_i)$, on the last line resulting in $b_i = 1$ for every valid signature. Since overall $\mathcal{B}$ makes $\ell(\mathcal{B}) = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B}) + 1$ successful queries to Ver, $\mathcal{B}$ wins $\mathbf{IMP\text{-}CA}_{IZ}$ whenever $\mathcal{B}$ wins $\mathbf{OMUF}_{BZ}$. This proves $\epsilon' \geq \epsilon$.

Moreover, the number $\mathrm{Q}_{\mathrm{S}_1}(\mathcal{A})$ of abandoned sessions in the $\mathbf{OMUF}_{BZ}$ experiment equals the number $\mathrm{Q}_{\mathrm{P}_1}(\mathcal{B})$ of abandoned sessions in the $\mathbf{IMP\text{-}CA}_{IZ}$ experiment and the number $\mathrm{Q}_V(\mathcal{B})$ of calls to the oracle C_2 is bounded by Q_{GH} (for the simulation of GH) plus additional $\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) + 1$ calls to each of them on line 5 (the latter is necessary in case $\mathcal{A}$ guesses the output of C_2 on some points). Finally, the running times of $\mathcal{A}$ and $\mathcal{B}$ are roughly the same, $\tau' \approx \tau$. □

5.5 Breaking *IZ*[DL] IMP-CA security implies solving OMDL

Theorem 5. *Let $IZ[\mathsf{DL}] = (IZ.PG, IZ.KG, IZ.P = (IZ.P_1, IZ.P_2), IZ.Ver)$ be the instantiation of IZ with discrete logarithm parameter generator $IZ[\mathsf{DL}].PG$, and let $l(\kappa) := \max\{\lg|\mathcal{C}|, \lg|\mathcal{D}|\}$ where $\mathcal{C}$ and $|\mathcal{D}$ are the challenge spaces (i.e. $l(\kappa)$ is the maximum challenge bitlength). Let $\mathcal{B}^{\mathrm{P}_1,\mathrm{P}_2,\mathrm{C}_1,\mathrm{C}_2,\mathrm{Ver}} := (\mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}, \mathcal{P}^{\mathrm{C}_1,\mathrm{C}_2})$ be an IMP-CA adversary of time complexity $\tau(\kappa)$ attacking $IZ[\mathsf{DL}]$. Then there exists an OMDL adversary $\mathcal{I}$ attacking $IZ[\mathsf{DL}].PG$ such that, for every κ, $\mathbf{Adv}^{\mathbf{IMP\text{-}CA}}_{IZ}(\mathcal{A},\kappa) \le 2^{-l(\kappa)} + \sqrt{\mathbf{Adv}^{\mathbf{OMDL}}_{IZ.PG}(\mathcal{I},\kappa)}$. Furthermore, the time complexity of $\mathcal{I}$ is $\tau'(\kappa) = 2\tau(\kappa) + O(\kappa^3 + (l(\kappa) + \ell(\kappa)) \cdot \kappa^2)$, where $\ell(\kappa)$ is the number of prover sessions with which $\mathcal{V}$ interacts.*

Proof. We closely follow the proof [5, Theorem 5.1]. We assume w.l.o.g. that $\mathcal{V}$ never repeats a request. Fix $\kappa \in \mathbb{N}$ and let (n, g) be an output of $IZ[\mathsf{DL}].PG$ running on input 1^κ. Adversary $\mathcal{I}$ has access to a discrete logarithm solving oracle $\mathcal{I}.Sol(g, q)$ that returns $a \in \mathbb{Z}_n$ such that $q = g^a$, and a group element challenge oracle $\mathcal{I}.Ch(\cdot)$ that takes no inputs and returns a random challenge point $q \in \mathbb{G}$ each time it is invoked. $\mathcal{I}$ calls the $\mathcal{I}.Ch(\cdot)$ oracle $2\ell + 1$ times and tries to compute the corresponding discrete logarithms, while making only 2ℓ queries to $\mathcal{I}.Sol(\cdot\,,\cdot)$. $\mathcal{I}$ simulates an interaction between $\mathcal{B}$ and the prover sessions (that is, $\mathcal{I}$ executes $\mathcal{V}$, which is only allowed to query its P_1 and P_2 oracles). To do so, $\mathcal{I}$ first queries its $\mathcal{I}.Ch(\cdot)$ oracle, obtaining a random group element $q_0 \xleftarrow{\$} \mathcal{I}.Ch(\cdot) \in \mathbb{G}$, and uses it to create a public key pk for the IMP-CA adversary $\mathcal{B}$. It then runs $\mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}$ and answers its requests.

In response to a $\mathrm{P}_1(\cdot)$ request, $\mathcal{I}$ opens a new prover session pid, makes queries $q_{pid,0} \xleftarrow{\$} \mathcal{I}.Ch(\cdot)$ and $q_{pid,1} \xleftarrow{\$} \mathcal{I}.Ch(\cdot)$, and returns the answer $(pid, q_{pid,0}, q_{pid,1})$ to $\mathcal{V}$. This mimics the behavior of a genuine prover session pid, which would start from a fresh random tape and the public key pk, sample commitments u_{pid}, v_{pid}, and return (pid, u_{pid}, v_{pid}) to $\mathcal{V}$.

$\mathcal{I}$ does not possess the secret key corresponding to q_0, which the genuine prover sessions would use to respond to $\mathcal{V}$'s $\mathrm{P}_2(pid, c \in \mathcal{C}, d \in \mathcal{D})$ requests, where $\max\{\lg|\mathcal{C}|, \lg|\mathcal{D}|\} = l(\kappa)$, but it compensates with its access to the $\mathcal{I}.Sol(\cdot\,,\cdot)$ oracle to answer these requests. Specifically, in response to request $\mathrm{P}_2(pid, c_{pid}, d_{pid})$, $\mathcal{I}$ makes the queries $z_{pid} \leftarrow \mathcal{I}.Sol(g, q_{pid,0}\, q_0^{-c_{pid}})$ and $w_{pid} \leftarrow \mathcal{I}.Sol(g, q_{pid,1}(q_{pid,0}\, q_0^{-c_{pid}})^{-d_{pid}})$, and returns the answer w_{pid}. This is exactly the expected response from the prover session pid, as one can see with the correspondence $y \leftrightarrow q_0$, $u_{pid} \leftrightarrow q_{pid,0}$, $v_{pid} \leftrightarrow q_{pid,1}$. Hence $\mathcal{I}$ simulates the prover's behavior perfectly.

Since $\mathrm{Q}_{\mathrm{P}_2} = \ell(\kappa)$ is the number of prover sessions that $\mathcal{V}$ opens, interacts with, and closes, when $\mathcal{V}$ stops $\mathcal{I}$ has made $2\ell(\kappa)$ queries to its discrete logarithm oracle, hence $\mathcal{I}$ needs to find the discrete logarithm of $2\ell(\kappa)+1$ challenge points to win the game. $\mathcal{I}$ will first attempt to extract the discrete logarithm a_0 of the challenge q_0 from $\mathcal{P}$, which is initialized with the output $\mathbf{st}_{\mathcal{V}}$ of $\mathcal{V}$, and it will then use a_0 to compute the discrete logarithm of each of the other $2\ell(\kappa)$ challenge points.

To that end, $\mathcal{I}$ obtains the commitment pair $(u, v) \overset{\$}{\leftarrow} \mathcal{P}.\mathrm{F}_1(\mathbf{st}_{\mathcal{P}})$, samples a uniformly random challenge pair $(c \overset{\$}{\leftarrow} \mathcal{C}, d \overset{\$}{\leftarrow} \mathcal{D})$, runs $w \overset{\$}{\leftarrow} \mathcal{P}.\mathrm{F}_2(\mathbf{st}_{\mathcal{P}}, c, d)$ to obtain the response to the challenge pair, and evaluates the verification predicate $b \leftarrow IZ[\mathsf{DL}].Ver(pk, u, v, c, d, w)$. $\mathcal{I}$ then selects another challenge pair $(c' \overset{\$}{\leftarrow} \mathcal{C}, d' \leftarrow d)$, where c' is chosen uniformly at random but the same d is kept as before, rewinds $\mathcal{P}$ to the same starting state $\mathbf{st}_{\mathcal{P}} \leftarrow \mathbf{st}_{\mathcal{V}}$ and reruns it to eventually obtain its response $w' \overset{\$}{\leftarrow} \mathcal{P}.\mathrm{F}_2(\mathbf{st}_{\mathcal{P}}, c', d')$ to the new challenge pair, then evaluates the verification predicate $b' \leftarrow IZ[\mathsf{DL}].Ver(pk, u, v, c', d', w')$.

If the verification predicates evaluate to $b = b' = 1$, both tuples (u, v, c, d, w) and (u, v, c', d', w') must be valid for public key $q_0 = g^{a_0}$. That means $v = g^w h^d = g^{w'} h'^{d'}$ where $h = uq_0^{-c}$ and $h' = uq_0^{-c'}$, that is, $h' = hq_0^{c-c'}$ and $g^{w'} = g^w h^d (h')^{-d'} = g^w h^d (hq_0^{c-c'})^{-d'} = g^w h^{d-d'} q_0^{d'(c'-c)} = g^{w+z(d-d')+a_0 d'(c'-c)}$ where $h = g^z$, hence it must hold that $w' = w + (d - d')z + d'(c' - c)a_0 \pmod{n}$. If furthermore $c' \neq c$ but $d' = d$, then $w' = w + d(c' - c)a_0 \pmod{n}$, whereby $\mathcal{I}$ extracts the discrete logarithm of q_0 as $a_0 \leftarrow (w' - w)(d\,(c' - c))^{-1} \pmod{n}$.

Now we know that $z_{pid} = a_{pid,0} - c_{pid}\, a_0 \pmod{n}$ where $g^{a_{pid,0}} = q_{pid,0}$, and $w_{pid} = a_{pid,1} - d_{pid}\, z_{pid} \pmod{n}$ where $g^{a_{pid,1}} = q_{pid,1}$, hence $\mathcal{I}$ can recover all the $2\ell(\kappa)$ discrete logarithms $a_{pid,0} \leftarrow z_{pid} + c_{pid}\, a_0 \pmod{n}$ and $a_{pid,1} \leftarrow w_{pid} + d_{pid}\, z_{pid} \pmod{n}$, $1 \leq pid \leq \ell(\kappa)$.

If the verification predicates do not evaluate to 1 on both occasions, or the challenges c and c' coincide, then $\mathcal{I}$ fails. That is, $\mathcal{I}$ wins if, and only if, $b = b' = 1$ and $c \neq c'$. We proceed to relate the probability of this event with the IMP-CA advantage of adversary $\mathcal{B}$.

We observe that pk has the same distribution as in the two-phase $\mathbf{IMP\text{-}CA}_{IZ}$ game. Since $\mathcal{I}$ simulates the environment provided to $\mathcal{V}$ in that game perfectly, $\mathcal{V}$ behaves as it does when performing a concurrent attack against IZ, and $\mathcal{P}$ is given state information with the same distribution as in that case. Therefore, the probability that $b = 1$ is exactly $\mathbf{Adv}_{IZ}^{\mathbf{IMP\text{-}CA}}(\mathcal{B}, \kappa)$.

Let $\mathsf{acc}(\mathbf{st}_{\mathcal{V}}, pk)$ denote the probability that $b = 1$ (taken over the choice of challenge c) when the public key created by $\mathcal{I}$ is pk and the output from $\mathcal{V}$ is $\mathbf{st}_{\mathcal{V}}$. Let $\mathsf{res}(\mathbf{st}_{\mathcal{V}}, pk)$ denote the probability that $b = 1$, $b' = 1$, and $c \neq c'$ (taken over the choice of challenges c and c') when the public key created by $\mathcal{I}$ is pk and the output from $\mathcal{V}$ is $\mathbf{st}_{\mathcal{V}}$. Then, if $E[\bullet]$ denotes the expectation of random variable $\bullet$ over the choice of pk and $\mathbf{st}_{\mathcal{V}}$, the probability that $b = 1$ is $E[\mathsf{acc}(\mathbf{st}_{\mathcal{V}}, pk)]$, and the probability that $\mathcal{I}$ wins is $E[\mathsf{res}(\mathbf{st}_{\mathcal{V}}, pk)]$. The Bellare-Palacio Reset Lemma [5, Lemma 3.1] applied to cheating prover $\mathcal{P}$ with input $\mathbf{st}_{\mathcal{P}} = \mathbf{st}_{\mathcal{V}}$ and verifier V with input pk, the latter being implemented by $\mathcal{I}$, implies $\mathsf{acc}(\mathbf{st}_{\mathcal{V}}, pk) \leq 2^{-l(\kappa)} + \sqrt{\mathsf{res}(\mathbf{st}_{\mathcal{V}}, pk)}$.

We obtain the claimed relationship as follows:

$$\begin{aligned}\mathbf{Adv}^{\mathbf{IMP\text{-}CA}}_{IZ}(\mathcal{B},\kappa) &= E[\mathsf{acc}(\mathbf{st}_{\mathcal{V}},pk)]\\ &\le E\left[2^{-l(\kappa)}+\sqrt{\mathsf{res}(\mathbf{st}_{\mathcal{V}},pk)}\right]\\ &= 2^{-l(\kappa)}+E\left[\sqrt{\mathsf{res}(\mathbf{st}_{\mathcal{V}},pk)}\right]\\ &\le 2^{-l(\kappa)}+\sqrt{E[\mathsf{res}(\mathbf{st}_{\mathcal{V}},pk)]}\\ &= 2^{-l(\kappa)}+\sqrt{\mathbf{Adv}^{\mathbf{OMDL}}_{IZ.PG}(\mathcal{I},\kappa)}.\end{aligned}$$

To complete the proof, it remains to justify the claim about the time complexity of adversary $\mathcal{I}$. Consider the OMDL game from Section 4.4, which defines $\mathbf{Adv}^{\mathbf{OMDL}}_{PG}(\mathcal{D},\kappa)$. The cost of all the steps of this game before the execution of adversary $\mathcal{I}$'s final check that $c' \neq c$ and $b = b' = 1$ is dominated by at most $2\tau(\kappa)$ for the whole interaction between the prover P and the cheating verifier $\mathcal{V}$, plus the cost of evaluating the verification predicate twice to obtain $b \leftarrow IZ.Ver(pk,u,v,c,d,w)$ and $b' \leftarrow IZ.Ver(pk,u,v,c',d',w')$, plus the cost of retrieving a_0 and all of the $a_{pid,0}$ and $a_{pid,1}$.

Each evaluation of the verification predicate involves computing one exponentiation of a $\lg|\mathcal{C}|$-bit exponent, one exponentiation of a $\lg|\mathcal{D}|$-bit exponent, and one exponentiation of a $\lg(n)$-bit exponent (disregarding the cost of other operations, which are much cheaper by comparison). Since $\lg|\mathcal{C}| \le l(\kappa)$, $\lg|\mathcal{D}| \le l(\kappa)$, and $\lg(n) \approx 2\kappa$ assuming elliptic curve groups, overall the cost is thus $O(l(\kappa)+\kappa)$ group operations, each of which has complexity $O(\kappa^2)$ for a total complexity $O((l(\kappa)+\kappa)\cdot\kappa^2)$. Finally, the cost of retrieving a_0 is one inversion modulo n, or $O((\lg n)^2) = O(\kappa^2)$, and the cost of retrieving all of the $2\ell(\kappa)$ remaining discrete logarithms is $O(\ell(\kappa)\cdot\kappa^2)$.

Therefore, overall the time complexity of $\mathcal{I}$ is $\tau'(\kappa) := 2\tau(\kappa)+O((l(\kappa)+\kappa)\cdot\kappa^2+\kappa^2+\ell(\kappa)\cdot\kappa^2) = 2\tau(\kappa)+O(\kappa^3+(l(\kappa)+\ell(\kappa))\cdot\kappa^2)$. □

Corollary 1. *If the discrete logarithm parameter generator $IZ[\mathsf{DL}].PG$ is OMDL-secure and the challenge bitlength satisfies $l(\kappa) \in \omega(\log(\kappa))$ to preclude exhaustively guessing the verifier's challenges, then the correspondingly parametrized $IZ[\mathsf{DL}]$ is IMP-CA-secure.*

6 The $BZ[\mathsf{CSI}]$ and $IZ[\mathsf{CSI}]$ schemes

For this instantiation we adopt the convention that variables denoting curves will be uppercase, all others lowercase.

Here we introduce the idea of using more than one commitment and a corresponding number of responses per challenge, enabling the use of multikey CSI-FiSh for the whole scheme. In Section 7.1 we will discuss an entirely different way to obtain potentially better BZ parameters with multikey CSI-FiSh, but applying that scheme with blind signatures (as opposed to plain signatures) remains elusive. $BZ[\mathsf{CSI}]$ consists of the following algorithms:

Algorithm 11 $BZ[\mathsf{CSI}].PG(1^\kappa)$

```
Select a prime $p = 4\prod_k \ell_k - 1$, let $\mathrm{Cl}(\mathcal{O})$ be the ideal class group over $\mathbb{F}_p$ and let $N := \#\mathrm{Cl}(\mathcal{O})$.
Select $A_0 \in \mathcal{E}\ell\ell_p$
Select the number $C \geq 2$ of curves per public key (including $A_0$) and the number $t$ of Fiat-
  Shamir [13] iterations such that $C \mid t$ and $t!/(t/C)!^C \geq 2^\kappa$.
Select the number $C' \geq 2$ of curves per public key (including $A_0$) and the number $T$ of Fiat-
  Shamir iterations such that $C' \mid T$ and $T!/(T/C')!^{C'} \geq 2^\kappa$.
Select secure hash functions $\mathcal{H} : \mathcal{E}\ell\ell_p^{(C'-1)\times t} \times \{0,1\}^* \to \mathcal{M}_C^t$ and $\mathcal{G} : \mathcal{E}\ell\ell_p^{T\times t} \to \mathcal{M}_{C'}^T$.
return $par := (N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H})$.
```

Algorithm 12 $BZ[\mathsf{CSI}].KG(par)$

```
$(N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par$
$x_0 \leftarrow 0,\quad x_{1\dots C-1} \xleftarrow{\$} (\mathbb{Z}_N^*)^{C-1},\quad A_0 \leftarrow A_0,\quad A_{1\dots C-1} \leftarrow [x_{1\dots C-1}]A_0$
$sk \leftarrow (x_{0\dots C-1}, A_{0\dots C-1}, par),\quad pk \leftarrow (A_{0\dots C-1}, par)$
return $(pk, sk)$
```

Algorithm 13 $BZ[\mathsf{CSI}].S_1(sk)$

```
$(x_{0\dots C-1}, A_{0\dots C-1}, par) \leftarrow sk,\quad (N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par$
$r_{1\dots t}^{1\dots C'-1} \xleftarrow{\$} (\mathbb{Z}_N^*)^{(C'-1)\times t},\quad s_{1\dots t}^{1\dots T} \xleftarrow{\$} (\mathbb{Z}_N^*)^{T\times t}$
$\hat{U}_{1\dots t}^{1\dots C'-1} \leftarrow [r_{1\dots t}^{1\dots C'-1}]A_0,\quad \hat{V}_{1\dots t}^{1\dots T} \leftarrow [s_{1\dots t}^{1\dots T}]A_0,\quad \mathsf{st}_S \leftarrow (sk, r_{1\dots t}^{1\dots C'-1}, s_{1\dots t}^{1\dots T})$
return $(\hat{U}_{1\dots t}^{1\dots C'-1} \in \mathcal{E}\ell\ell_p^{(C'-1)\times t}, \hat{V}_{1\dots t}^{1\dots T} \in \mathcal{E}\ell\ell_p^{T\times t}, \mathsf{st}_S)$
```

Algorithm 14 $BZ[\mathsf{CSI}].U_1(pk, \hat{U}_{1\dots t}^{1\dots C'-1} \in \mathcal{E}\ell\ell_p^{(C'-1)\times t}, \hat{V}_{1\dots t}^{1\dots T} \in \mathcal{E}\ell\ell_p^{T\times t}, m)$

```
$(A_{0\dots C-1}, par) \leftarrow pk,\quad (N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par$
$\pi \xleftarrow{\$} \mathcal{S}_t,\quad \delta_{1\dots t}^{(0)} \leftarrow 0,\ \delta_{1\dots t}^{1\dots C'-1} \xleftarrow{\$} (\mathbb{Z}_N^*)^{(C'-1)\times t},\quad \rho \xleftarrow{\$} \mathcal{S}_T,\quad \varepsilon_{1\dots t}^{1\dots T} \xleftarrow{\$} (\mathbb{Z}_N^*)^{T\times t}$
$U_{1\dots t}^{1\dots C'-1} \leftarrow [\delta_{1\dots t}^{1\dots C'-1}]\pi(\hat{U}_{1\dots t}^{1\dots C'-1}),\quad V_{1\dots t}^{1\dots T} \leftarrow [\varepsilon_{1\dots t}^{1\dots T}]\rho^{\uparrow}\pi_{\downarrow}(\hat{V}_{1\dots t}^{1\dots T})$
$c_{1\dots t} \leftarrow \mathcal{H}(U_{1\dots t}^{1\dots C'-1}, m),\quad d^{1\dots T} \leftarrow \mathcal{G}(V_{1\dots t}^{1\dots T}),\quad \hat{c}_{1\dots t} \leftarrow \pi^{-1}(c_{1\dots t}),\quad \hat{d}^{1\dots T} \leftarrow \rho^{-1}(d^{1\dots T})$
$\mathsf{st}_U \leftarrow (pk, U_{1\dots t}^{1\dots C'-1}, \hat{U}_{1\dots t}^{1\dots C'-1}, \hat{V}_{1\dots t}^{1\dots T}, \pi, \delta_{1\dots t}^{1\dots C'-1}, \rho, \varepsilon_{1\dots t}^{1\dots T}, d^{1\dots T}, \hat{c}_{1\dots t}, \hat{d}^{1\dots T})$
return $(\hat{c}_{1\dots t} \in \mathcal{M}_C^t, \hat{d}^{1\dots T} \in \mathcal{M}_{C'}^T, \mathsf{st}_U)$
```

Algorithm 15 $BZ[\mathsf{CSI}].S_2(\mathsf{st}_S, \hat{c}_{1\dots t} \in \mathcal{M}_C^t, \hat{d}^{1\dots T} \in \mathcal{M}_{C'}^T)$

```
$(sk, r_{1\dots t}^{1\dots C'-1}, s_{1\dots t}^{1\dots T}) \leftarrow \mathsf{st}_S,\ (x_{0\dots C-1}, A_{0\dots C-1}, par) \leftarrow sk,$
  $(N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par$
$s'^{1\dots T}_{1\dots t} \leftarrow s_{1\dots t}^{1\dots T} - x_{\downarrow\hat{c}_{1\dots t}} \pmod N$   ▷ $\therefore \hat{V}_{1\dots t}^{1\dots T} = [s'^{1\dots T}_{1\dots t}]A_{\downarrow\hat{c}_{1\dots t}}$
$z_{1\dots t}^{1\dots C'-1} \leftarrow r_{1\dots t}^{1\dots C'-1} - x_{\downarrow\hat{c}_{1\dots t}} \pmod N$   ▷ $\therefore \hat{U}_{1\dots t}^{1\dots C'-1} = [z_{1\dots t}^{1\dots C'-1}]A_{\downarrow\hat{c}_{1\dots t}}$
$z_{1\dots t}^{0} \leftarrow 0$   ▷ $\therefore \hat{H}_{1\dots t}^{0\dots C'-1} := (A_{\downarrow\hat{c}_{1\dots t}} \parallel \hat{U}_{1\dots t}^{1\dots C'-1}) = [z_{1\dots t}^{0\dots C'-1}]A_{\downarrow\hat{c}_{1\dots t}}$
$\hat{w}_{1\dots t}^{1\dots T} \leftarrow s'^{1\dots T}_{1\dots t} - z_{1\dots t}^{\uparrow\hat{d}^{1\dots T}} \pmod N$   ▷ $\therefore \hat{V}_{1\dots t}^{1\dots T} = [\hat{w}_{1\dots t}^{1\dots T}]\hat{H}_{1\dots t}^{\uparrow\hat{d}^{1\dots T}}$
return $\hat{w}_{1\dots t}^{1\dots T} \in (\mathbb{Z}_N^*)^{T\times t}$
```

Algorithm 16 $BZ[\mathsf{CSI}].U_2(\mathsf{st}_U, \hat{w}_{1\dots t}^{1\dots T} \in (\mathbb{Z}_N^*)^{T\times t})$

```
$(pk, U_{1\dots t}^{1\dots C'-1}, \hat{U}_{1\dots t}^{1\dots C'-1}, \hat{V}_{1\dots t}^{1\dots T}, \pi, \delta_{1\dots t}^{1\dots C'-1}, \rho, \varepsilon_{1\dots t}^{1\dots T}, d^{1\dots T}, \hat{c}_{1\dots t}, \hat{d}^{1\dots T}) \leftarrow \mathsf{st}_U,$
  $(A_{0\dots C-1}, par) \leftarrow pk,\quad (N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par$
$\hat{H}_{1\dots t}^{0\dots C'-1} \leftarrow (A_{\downarrow\hat{c}_{1\dots t}} \parallel \hat{U}_{1\dots t}^{1\dots C'-1})$   ▷ $\therefore \hat{H}_{1\dots t}^{0\dots C'-1} = [z_{1\dots t}^{0\dots C'-1}]A_{\downarrow\hat{c}_{1\dots t}}$
if $\hat{V}_{1\dots t}^{1\dots T} \neq [\hat{w}_{1\dots t}^{1\dots T}]\hat{H}_{1\dots t}^{\uparrow\hat{d}^{1\dots T}}$ : return $\perp$   ▷ check that the signer is honest
$w_{1\dots t}^{1\dots T} \leftarrow \rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\dots t}^{1\dots T}) - \delta_{1\dots t}^{\uparrow d^{1\dots T}} + \varepsilon_{1\dots t}^{1\dots T} \pmod N$
return $\sigma := (U_{1\dots t}^{1\dots C'-1}, d^{1\dots T}, w_{1\dots t}^{1\dots T}) \in \mathcal{E}\ell\ell_p^{(C'-1)\times t} \times \in \mathcal{M}_{C'}^T \times (\mathbb{Z}_N^*)^{T\times t}$
```

Algorithm 17 $BZ[\mathsf{CSI}].\mathit{Ver}(pk, \sigma \in \mathcal{E}\ell\ell_p^{(C'-1)\times t} \times \in \mathcal{M}_{C'}^{T} \times (\mathbb{Z}_N^*)^{T\times t}, m)$

```
1: if $\sigma = \perp$ : return 0
2: $(A_{0\dots C-1}, par) \leftarrow pk, \quad (N, A_0, C, t, C', T, \mathcal{G}, \mathcal{H}) \leftarrow par, \quad (U_{1\dots t}^{1\dots C'-1}, d^{1\dots T}, w_{1\dots t}^{1\dots T}) \leftarrow \sigma$
3: $c_{1\dots t} \leftarrow \mathcal{H}(U_{1\dots t}^{1\dots C'-1}, m), \quad H_{1\dots t}^{0\dots C'-1} \leftarrow (A_{\downarrow c_{1\dots t}} \parallel U_{1\dots t}^{1\dots C'-1}), \quad V_{1\dots t}^{1\dots T} \leftarrow [w_{1\dots t}^{1\dots T}] H_{1\dots t}^{\uparrow d^{1\dots T}}$
4: return $\left(d^{1\dots T} = \mathcal{G}(V_{1\dots t}^{1\dots T})\right)$ ? 1 : 0
```

For $IZ[\mathsf{CSI}]$, algorithms $IZ.PG$ and $IZ.KG$ are, respectively, identical to $BZ.PG$ and $BZ.KG$, except that the hash functions are ignored and omitted. The remaining algorithms are as follows.

Algorithm 18 $IZ[\mathsf{CSI}].P_1(sk)$

```
1: $(x_{0\dots C-1}, A_{0\dots C-1}, par) \leftarrow sk, \quad (N, A_0, C, t, C', T) \leftarrow par$
2: $r_{1\dots t}^{1\dots C'-1} \xleftarrow{\$} (\mathbb{Z}_N^*)^{(C'-1)\times t}, \quad s_{1\dots t}^{1\dots T} \xleftarrow{\$} (\mathbb{Z}_N^*)^{T\times t}$
3: $\hat{U}_{1\dots t}^{1\dots C'-1} \leftarrow [r_{1\dots t}^{1\dots C'-1}] A_0, \quad \hat{V}_{1\dots t}^{1\dots T} \leftarrow [s_{1\dots t}^{1\dots T}] A_0, \quad \mathsf{st}_S \leftarrow (sk, r_{1\dots t}^{1\dots C'-1}, s_{1\dots t}^{1\dots T})$
4: return $(\hat{U}_{1\dots t}^{1\dots C'-1} \in \mathcal{E}\ell\ell_p^{(C'-1)\times t}, \hat{V}_{1\dots t}^{1\dots T} \in \mathcal{E}\ell\ell_p^{T\times t}, \mathsf{st}_S)$
```

Algorithm 19 $IZ[\mathsf{CSI}].P_2(\mathsf{st}_P, \hat{c}_{1\dots t} \in \mathcal{M}_C^t, \hat{d}^{1\dots T} \in \mathcal{M}_{C'}^{T})$

```
1: $(sk, r_{1\dots t}^{1\dots C'-1}, s_{1\dots t}^{1\dots T}) \leftarrow \mathsf{st}_S, \quad (x_{0\dots C-1}, A_{0\dots C-1}, par) \leftarrow sk, \quad (N, A_0, C, t, C', T) \leftarrow par$
2: $s'^{1\dots T}_{1\dots t} \leftarrow s_{1\dots t}^{1\dots T} - x_{\downarrow \hat{c}_{1\dots t}} \pmod{N} \quad \triangleright \therefore \hat{V}_{1\dots t}^{1\dots T} = [s'^{1\dots T}_{1\dots t}] A_{\downarrow \hat{c}_{1\dots t}}$
3: $z_{1\dots t}^{1\dots C'-1} \leftarrow r_{1\dots t}^{1\dots C'-1} - x_{\downarrow \hat{c}_{1\dots t}} \pmod{N} \in (\mathbb{Z}_N^*)^t \quad \triangleright \therefore \hat{U}_{1\dots t}^{1\dots C'-1} = [z_{1\dots t}^{1\dots C'-1}] A_{\downarrow \hat{c}_{1\dots t}}$
4: $z_{1\dots t}^{0} \leftarrow 0 \quad \triangleright \therefore \hat{H}_{1\dots t}^{0\dots C'-1} := (A_{\downarrow \hat{c}_{1\dots t}} \parallel \hat{U}_{1\dots t}^{1\dots C'-1}) = [z_{1\dots t}^{0\dots C'-1}] A_{\downarrow \hat{c}_{1\dots t}}$
5: $\hat{w}_{1\dots t}^{1\dots T} \leftarrow s'^{1\dots T}_{1\dots t} - z_{1\dots t}^{\uparrow \hat{d}^{1\dots T}} \pmod{N} \quad \triangleright \therefore \hat{V}_{1\dots t}^{1\dots T} = [\hat{w}_{1\dots t}^{1\dots T}] \hat{H}_{1\dots t}^{\uparrow \hat{d}^{1\dots T}}$
6: return $\hat{w}_{1\dots t}^{1\dots T} \in (\mathbb{Z}_N^*)^{T\times t}$
```

Algorithm 20 $IZ[\mathsf{CSI}].\mathit{Ver}(pk, \hat{U}_{1\dots t}^{1\dots C'-1} \in \mathcal{E}\ell\ell_p^{(C'-1)\times t}, \hat{V}_{1\dots t}^{1\dots T} \in \mathcal{E}\ell\ell_p^{T\times t}, \hat{c}_{1\dots t} \in \mathcal{M}_C^t, \hat{d}^{1\dots T} \in \mathcal{M}_{C'}^{T}, \hat{w}_{1\dots t}^{1\dots T} \in (\mathbb{Z}_N^*)^{T\times t})$

```
1: $(A_{0\dots C-1}, par) \leftarrow pk, \quad (N, A_0, C, t, C', T) \leftarrow par$
2: $\hat{H}_{1\dots t}^{0\dots C'-1} \leftarrow (A_{\downarrow \hat{c}_{1\dots t}} \parallel \hat{U}_{1\dots t}^{1\dots C'-1})$
3: return $\left(\hat{V}_{1\dots t}^{1\dots T} = [\hat{w}_{1\dots t}^{1\dots T}] \hat{H}_{1\dots t}^{\uparrow \hat{d}^{1\dots T}}\right)$ ? 1 : 0
```

6.1 Perfect correctness of $BZ[\mathsf{CSI}]$

Theorem 6. *The proposed $BZ[\mathsf{CSI}]$ protocol is perfectly correct.*

Proof. For $\forall par \in BZ[\mathsf{CSI}].PG(1^\kappa)$, $\forall (pk, sk) \in BZ[\mathsf{CSI}].KG(par)$, $\forall m \in \{0,1\}^*$, let $\sigma := (U_{1\dots t}^{1\dots C'-1}, d^{1\dots T}, w_{1\dots t}^{1\dots T})$ be a genuine signature for m, properly obtained from the protocol.

Suppose by contradiction that $BZ[\mathsf{CSI}].\mathit{Ver}(pk, \sigma, m) = 0$. This can only happen (at step 4 of $BZ[\mathsf{CSI}].\mathit{Ver}$) if $d^{1\dots T} \neq \mathcal{G}(V_{1\dots t}^{1\dots T})$, which means that either $d^{1\dots T}$ is malformed (a contradiction, from the way it is created in step 4 of $BZ[\mathsf{CSI}].U_1$), or $V_{1\dots t}^{1\dots T}$ is malformed. Because $V_{1\dots t}^{1\dots T} = [w_{1\dots t}^{1\dots T}]H_{1\dots t}^{d^{1\dots T}}$, $H_{1\dots t}^{0\dots C'-1} = (A_{\downarrow c_{1\dots t}} \parallel U_{1\dots t}^{1\dots C'-1})$, and $c_{1\dots t} = \mathcal{H}(U_{1\dots t}^{1\dots C'-1}, m)$, the latter means that either $w_{1\dots t}^{1\dots T}$ is malformed or $U_{1\dots t}^{1\dots C'-1}$ is malformed, since $c_{1\dots t}$ matches its definition in step 4 of $BZ[\mathsf{CSI}].U_1$. In turn, this means that either $\hat{H}_{1\dots t}^{0\dots C'-1} := (A_{\downarrow \hat{c}_{1\dots t}} \parallel \hat{U}_{1\dots t}^{1\dots C'-1}) \neq [z_{1\dots t}^{0\dots C'-1}]A_{\downarrow \hat{c}_{1\dots t}}$, given the way $U_{1\dots t}^{1\dots C'-1}$ is defined from $\hat{U}_{1\dots t}^{1\dots C'-1}$ in step 3 of $BZ[\mathsf{CSI}].U_1$, or $\hat{V}_{1\dots t}^{1\dots T} \neq [\hat{w}_{1\dots t}^{1\dots T}]\hat{H}_{1\dots t}^{\uparrow \hat{d}^{1\dots T}}$, given the way $w_{1\dots t}^{1\dots T}$ is defined from $\hat{w}_{1\dots t}^{1\dots T}$ in step 4 of $BZ[\mathsf{CSI}].U_2$, respectively. But this would have caused $BZ[\mathsf{CSI}].U_2$ to abort at its step 3 (supported by step 2), and σ would not have been created, which is contradiction.

Therefore $BZ[\mathsf{CSI}].\mathit{Ver}$ must end with $BZ[\mathsf{CSI}].\mathit{Ver}(pk, \sigma, m) = 1$ as expected, and $BZ[\mathsf{CSI}]$ is thus perfectly correct as claimed. □

6.2 Perfect blindness of $BZ[\mathsf{CSI}]$

Theorem 7. *The proposed* $BZ[\mathsf{CSI}]$ *protocol is perfectly blind.*

Proof. For clarity, indices in the present context will only denote sessions or choices, not vector or matrix components. Let $\mathcal{A}$ be an adversary playing the $\mathbf{Blind}_{BZ}^{\mathcal{A}}$ game. Assume w.l.o.g. that $\mathcal{A}$'s randomness is fixed and $\mathcal{A}$ always finishes both sessions and receives valid signatures (σ_0, σ_1). $\mathcal{A}$'s view after its execution is $(A_{0\dots C-1}, m_0, m_1, T_0, T_1, \sigma_0, \sigma_1)$ where $T_i := (\hat{U}_i, \hat{V}_i, \hat{c}_i, \hat{d}_i, \hat{w}_i)$ are the transcripts that $\mathcal{A}$ observes from the i-th signing session and $\sigma := (U_i, d_i, w_i)$, $i \in \{0, 1\}$.

Since $\mathcal{A}$'s randomness is fixed, the only randomness under consideration is that in $BZ[\mathsf{CSI}].U_1$ ($BZ[\mathsf{CSI}].U_2$ introduces no further randomness), that is, $(\pi_b, \delta_b, \rho_b, \varepsilon_b)$ for each $b \in \{0, 1\}$. Suppose also w.l.o.g. that $\mathcal{A}$ takes $T_i := (\hat{U}_i, \hat{V}_i, \hat{c}_i, \hat{d}_i, \hat{w}_i)$ to refer to m_b, i.e. $\hat{c}_i = \pi_b^{-1}(\mathcal{H}([\delta_b]\pi_b(\hat{U}_i), m_b))$ and $\hat{d}_i = \rho_b^{-1}(\mathcal{G}([\varepsilon_b]\rho_b^{\uparrow}\pi_{b\downarrow}(\hat{V}_i))$, *for some specific choice* of $b \in \{0, 1\}$ that $\mathcal{A}$ makes. NB: $\hat{w}_i$ does not introduce new, independent constraints, since the signatures are presumed valid and these quantities are deterministically obtained from the other ones.

To prove the theorem, we argue that $\exists(\pi_b, \delta_b, \rho_b, \varepsilon_b) \in \mathcal{S}_t \times (\mathbb{Z}_N^*)^{(C'-1)\times t} \times \mathcal{S}_T \times (\mathbb{Z}_N^*)^{T\times t}$, taken always from the same distribution independently from b, such that $\hat{c}_i = \pi_b^{-1}(\mathcal{H}([\delta_b]\pi_b(\hat{U}_i), m_b))$ and $\hat{d}_i = \rho_b^{-1}(\mathcal{G}([\varepsilon_b]\rho_b^{\uparrow}\pi_{b\downarrow}(\hat{V}_i))$ for *either* choice of $b \in \{0, 1\}$. In other words, we will show that both transcripts are equally consistent with, and suggest no preference for, either $b = 0$ or $b = 1$.

Consider the effect of taking $\varUpsilon_b \leftarrow [\zeta_b]A_0$ and $\varOmega_b \leftarrow [\chi_b]A_{\downarrow c_i}$, for some uniformly distributed $\zeta_b \in (\mathbb{Z}_N^*)^{(C'-1)\times t}$, $\chi_b \in (\mathbb{Z}_N^*)^{T\times t}$, and computing $\gamma_b \leftarrow \mathcal{H}(\varUpsilon_b, m_b)$ and $\lambda_b \leftarrow \mathcal{G}(\varOmega_b)$, for *either* choice of $b \in \{0, 1\}$. We want to be able to write $\hat{c}_i = \pi_b^{-1}(\gamma_b)$ and $\hat{d}_i = \rho_b^{-1}(\lambda_b)$, so we can just take π_b and ρ_b to be any permutations compatible with these relations. For instance, if μ_i and ν_b respectively permute $\hat{c}_i$ and γ_b to the unique sorted sequence $(0\dots 0, \dots, C - 1\dots C - 1)_t$ of length t, i.e. $\mu_i(\hat{c}_i) = (0\dots 0, \dots, C - 1\dots C - 1)_t = \nu_b(\gamma_b)$, while η_i and θ_b respectively permute $\hat{d}_i$ and λ_b to the unique sorted sequence

$(0\ldots0,\ldots,C'-1\ldots C'-1)_T$ of length T, i.e. $\eta_i(\hat{d}_i) = (0\ldots0,\ldots,C'-1\ldots C'-1)_T = \theta_b(\lambda_b)$, just take $\pi_b := \nu_b^{-1} \circ \mu_i$ and $\rho_b := \theta_b^{-1} \circ \eta_i$. Since the output from $\mathcal{H}$ is uniformly distributed over $\mathcal{M}_C^t$ and the output from $\mathcal{G}$ is uniformly distributed over $\mathcal{M}_{C'}^T$, so are the distribution of ν_b over $\mathcal{S}_t$ and the distribution of θ_b over $\mathcal{S}_T$, and therefore the distributions of π_b and ρ_b defined this way are uniform as well.

Now we also want $\Upsilon_b = [\zeta_b]A_0 = U_i = [\delta_b]\pi_b(\hat{U}_i) = [\pi_b(r_i) + \delta_b]A_0$, revealing the constraint $\zeta_b = \pi_b(r_i) + \delta_b \pmod N$, and $\Omega_b = [\chi_b]A_{\downarrow c_i} = V_i = [\varepsilon_b]\rho_b^{\uparrow}\pi_{b\downarrow}(\hat{V}_i) = [\varepsilon_b]\rho_b^{\uparrow}\pi_{b\downarrow}([s_i]A_{\downarrow\hat{c}_i}) = [\varepsilon_b][\rho_b^{\uparrow}\pi_{b\downarrow}(s_i)]\rho_b^{\uparrow}(A_{\downarrow\pi_{b\downarrow}(\hat{c}_i)}) = [\rho_b^{\uparrow}\pi_{b\downarrow}(s_i) + \varepsilon_b]A_{\downarrow c_i}$, revealing the constraint $\chi_b = \rho_b^{\uparrow}\pi_{b\downarrow}(s_i) + \varepsilon_b \pmod N$, whereby we can take $\delta_b \leftarrow \zeta_b - \pi_b(r_i) \pmod N$ and $\varepsilon_b \leftarrow \chi_b - \rho_b^{\uparrow}\pi_{b\downarrow}(s_i) \pmod N$, respectively: as long as ζ_b and χ_b are uniformly distributed over $(\mathbb{Z}_N^*)^{(C'-1)\times t}$ and $(\mathbb{Z}_N^*)^{T\times t}$ respectively, so are δ_b and ε_b, provided that $\delta_b \neq 0$ and $\varepsilon_b \neq 0$.

NB: only the signer knows r_i and s_i, but the point is to show that *some* suitable tuple $(\pi_b, \delta_b, \rho_b, \varepsilon_b)$ exists at all for either choice of b.

In other words, whichever choice of $b \in \{0,1\}$ the adversary makes to try and associate a message m_b of their choice (and its corresponding signature σ_b) to a transcript T_0 or T_1, it is always possible to find a suitable tuple $(\pi_b, \delta_b, \rho_b, \varepsilon_b) \in \mathcal{S}_t \times (\mathbb{Z}_N^*)^{(C'-1)\times t} \times \mathcal{S}_T \times (\mathbb{Z}_N^*)^{T\times t}$ suggesting that choice m_b, without establishing any preference for either one. Therefore, the $BZ[\mathsf{CSI}]$ scheme is perfectly blind, as claimed. □

6.3 Unattainability of standard MitM security for *IZ*[CSI]

Theorem 8. *Standard meddler-in-the-middle (MitM) security is unachievable for the underlying identification scheme* $IZ[\mathsf{CSI}]$.

Proof. A MitM adversary $\mathcal{A}$:

1. sets $\alpha_{1\ldots t}^{(0)} \leftarrow 0$ and chooses $\alpha_{1\ldots t}^{1\ldots C'-1} \xleftarrow{\$} \mathbb{Z}_N^*$;
2. intercepts $(\hat{U}_{1\ldots t}^{1\ldots C'-1}, \hat{V}_{1\ldots t}^{1\ldots T})$ from the prover and sends $(\tilde{U}_{1\ldots t}^{1\ldots C'-1} := [\alpha_{1\ldots t}^{1\ldots C'-1}]\hat{U}_{1\ldots t}^{1\ldots C'-1}, \tilde{V}_{1\ldots t}^{1\ldots T} := \hat{V}_{1\ldots t}^{1\ldots T})$ to the verifier;
3. forwards $(\tilde{c}_{1\ldots t} := \hat{c}_{1\ldots t}, \tilde{d}^{1\ldots T} := \hat{d}^{1\ldots T})$ from the verifier to the prover;
4. intercepts $\hat{w}_{1\ldots t}^{1\ldots T}$ from the prover and completes the attack by setting $\tilde{w}_{1\ldots t}^{1\ldots T} := \hat{w}_{1\ldots t}^{1\ldots T} - \alpha_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}} \pmod N$.

Now $(\tilde{U}_{1\ldots t}^{1\ldots C'-1}, \tilde{V}_{1\ldots t}^{1\ldots T}, \tilde{c}_{1\ldots t}, \tilde{d}^{1\ldots T}, \tilde{w}_{1\ldots t}^{1\ldots T}) \neq (\hat{U}_{1\ldots t}^{1\ldots C'-1}, \hat{V}_{1\ldots t}^{1\ldots T}, \hat{c}_{1\ldots t}, \hat{d}^{1\ldots T}, \hat{w}_{1\ldots t}^{1\ldots T})$, but since $\tilde{H}_{1\ldots t}^{0\ldots C'-1} = [\alpha_{1\ldots t}^{0\ldots C'-1}]\hat{H}_{1\ldots t}^{0\ldots C'-1}$, it still satisfies $\tilde{V}_{1\ldots t}^{1\ldots T} = \hat{V}_{1\ldots t}^{1\ldots T} = [\hat{w}_{1\ldots t}^{1\ldots T}]\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}} = [\hat{w}_{1\ldots t}^{1\ldots T} - \alpha_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}}][\alpha_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}}]\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}} = [\tilde{w}_{1\ldots t}^{1\ldots T}]\tilde{H}_{1\ldots t}^{\uparrow\tilde{d}^{1\ldots T}}$, and $\mathcal{A}$ wins the standard MitM experiment with advantage 1. □

6.4 Breaking *IZ*[CSI] OMUF implies breaking IMP-CA

We now show that, given an adversary $\mathcal{A}$ that breaks the one-more unforgeability (OMUF) of $BZ[\mathsf{CSI}]$, one can build an adversary $\mathcal{B}$ that breaks the impersonation under concurrent attacks (IMP-CA) security of the underlying $IZ[\mathsf{CSI}]$ scheme.

For clarity, in this section we will omit all indices except session identifiers and indices indicating adversary signatures.

Furthermore, for convenience we suppress an optimization that is implicit in the protocol, namely, the omission of c from the signature contents. This value is known by the user in line 4 of Algorithm $BZ[\mathsf{CSI}].U_1(pk, \hat{U}, \hat{V}, m)$ and could be passed to Algorithm $BZ[\mathsf{CSI}].U_2(\mathsf{st}_U, \hat{w})$ via st_U, but is not included explicitly in σ for being redundant for verification, since it can be recovered from existing information in line 3 of Algorithm $BZ[\mathsf{CSI}].Ver(pk, \sigma, m)$. Yet, the following description becomes simpler if c is taken to be readily available as part of the (extended) signature σ'.

Adversary 2 $\mathcal{B}^{\mathrm{P}_1,\mathrm{P}_2,\mathrm{C}_1,\mathrm{C}_2,\mathrm{Ver}}(pk)$

```
$((m_1, \sigma_1), \ldots, (m_{\ell(\mathcal{A})}, \sigma_{\ell(\mathcal{A})})) \xleftarrow{\$} \mathcal{A}^{\mathrm{S}_1,\mathrm{S}_2,\mathrm{GH}}(pk)$
for $i \in \{1 \ldots \ell(\mathcal{A})\}$ :
    $(c'_i, U'_i, d'_i, w'_i) \leftarrow \sigma_i$
    $H'_i \leftarrow (A_{\downarrow c'_i} \parallel U'_i)$,   $V'_i \leftarrow [w'_i] H_i'^{\uparrow d'_i}$
    $\mathrm{GH}(U'_i, V'_i, m_i)$
    $vid \leftarrow vSess_{U'_i, V'_i, m_i}$
    $b_i \leftarrow \mathrm{Ver}(vid, w'_i)$
```

Procedure 4 $\mathrm{S}_1(\cdot)$

```
$(pid, \hat{U}_{pid}, \hat{V}_{pid}) \xleftarrow{\$} \mathrm{P}_1(\cdot)$
return $pid, \hat{U}_{pid}, \hat{V}_{pid}$
```

Procedure 5 $\mathrm{S}_2(pid, \hat{c}, \hat{d})$

```
$\hat{w}_{pid} \leftarrow \mathrm{P}_2(pid, \hat{c}, \hat{d})$
return $\hat{w}_{pid}$
```

Procedure 6 $\mathrm{GH}(U', V', m)$

```
if $\mathcal{GH}[U', V', m] \neq \bot$ : return $\mathcal{H}[U', V', m]$
$(vid, c', d') \xleftarrow{\$} \mathrm{C}_2(u', v')$,   $\mathcal{GH}[U', V', m] \leftarrow (c', d')$
$vSess_{U', V', m} \leftarrow vid$
return $\mathcal{GH}[U', V', m]$
```

The following theorem, which adapts [18, Theorem 5.6] to the three-move scheme-from-zero-knowledge setting, establishes that adversary $\mathcal{B}$ does indeed achieve the aforementioned goal:

Theorem 9. *Let $\mathcal{H} : \mathcal{E}\ell\ell_p^{(C'-1)\times t} \times \{0,1\}^* \to \mathcal{M}_C^t$ and $\mathcal{G} : \mathcal{E}\ell\ell_p^{T\times t} \to \mathcal{M}_{C'}^T$ be hash functions, and let $BZ[\mathsf{CSI}][\mathcal{G}, \mathcal{H}]$ and $IZ[\mathsf{CSI}]$ be instantiations of the proposed blind signature scheme and its underlying identification scheme on top of the* CSI. *If $IZ[\mathsf{CSI}]$ is $(\epsilon', \tau', \mathrm{Q}_{\mathrm{P}_1}, \mathrm{Q}_{\mathrm{P}_2}, \mathrm{Q}_V)$-IMP-CA secure, then $BZ[\mathsf{CSI}][\mathcal{G}, \mathcal{H}]$ is $(\epsilon, \tau, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF secure in the random oracle model, where $\epsilon = \epsilon'$, $\tau = \tau'$, $\mathrm{Q}_{\mathrm{S}_1} = \mathrm{Q}_{\mathrm{P}_1}$, $\mathrm{Q}_{\mathrm{S}_2} = \mathrm{Q}_{\mathrm{P}_2}$, and $\mathrm{Q}_{\mathrm{GH}} = \mathrm{Q}_V - \mathrm{Q}_{\mathrm{S}_2} - 1$.*

Proof. Let $\mathcal{A}$ be an adversary that breaks the $(\epsilon, \tau, \mathrm{Q}_{\mathrm{S}_1}, \mathrm{Q}_{\mathrm{S}_2}, \mathrm{Q}_{\mathrm{GH}})$-OMUF of $BZ[\mathsf{CSI}][\mathcal{G}, \mathcal{H}]$ in the random oracle model. Adversary $\mathcal{B}$, as we constructed it, runs the $\mathbf{IMP\text{-}CA}_{IZ}$ experiment and perfectly simulates $\mathcal{A}$'s oracles S_1, S_2, and GH via its own oracles P_1, P_2, C_1, and C_2. Note that $\mathcal{B}$ calls P_2 at most $\mathrm{Q}_{\mathrm{P}_2} = \mathrm{Q}_{\mathrm{S}_2}$ times over the course of its simulation, and moreover, $\mathrm{Q}_{\mathrm{P}_2}(\mathcal{B}) = \mathrm{Q}_{\mathrm{S}_2}(\mathcal{A})$. We show that $\mathcal{B}$ breaks the $(\epsilon', \tau', \mathrm{Q}_{\mathrm{P}_1}, \mathrm{Q}_{\mathrm{P}_2}, \mathrm{Q}_V)$-IMP-CA security of $\mathit{IZ}[\mathsf{DL}]$.

Suppose that $\mathcal{A}$ is successful, i.e. it outputs $\ell(\mathcal{A}) \geq \mathrm{Q}_{\mathrm{S}_2}(\mathcal{A})+1 = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B})+1$ valid signatures on distinct messages and the number of closed sessions with the signer is at most $\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B})$. Since all messages m_i are distinct, each signature corresponds to a distinct session vid_i with the oracle C_2 via the relation $\mathrm{GH}(U', V', m_i) = \mathrm{C}_2(U', V')$ from line 2 of Procedure GH. Since also $\sigma'_i = (c'_i, U'_i, d'_i, w'_i,)$ is a valid signature on m_i, we know that $\mathrm{GH}(U'_i, [w'_i]H_i'^{\uparrow d'_i}, m_i) = \mathrm{GH}(u'_i, v'_i, m_i) = \mathrm{C}_2(U'_i, V'_i)$ with $H'_i := (A_{\downarrow c'_i} \,\|\, U'_i)$. Therefore, $\mathcal{B}$ can make a successful query to oracle $\mathit{Ver}(\mathit{vid}, w'_i)$, on the last line resulting in $b_i = 1$ for every valid signature. Since overall $\mathcal{B}$ makes $\ell(\mathcal{B}) = \mathrm{Q}_{\mathrm{P}_2}(\mathcal{B})+1$ successful queries to Ver, $\mathcal{B}$ wins $\mathbf{IMP\text{-}CA}_{IZ}$ whenever $\mathcal{B}$ wins $\mathbf{OMUF}_{BZ}$. This proves $\epsilon' \geq \epsilon$.

Moreover, the number $\mathrm{Q}_{\mathrm{S}_1}(\mathcal{A})$ of abandoned sessions in the $\mathbf{OMUF}_{BZ}$ experiment equals the number $\mathrm{Q}_{\mathrm{P}_1}(\mathcal{B})$ of abandoned sessions in the $\mathbf{IMP\text{-}CA}_{IZ}$ experiment and the number $\mathrm{Q}_V(\mathcal{B})$ of calls to the oracle C_2 is bounded by Q_{GH} (for the simulation of GH) plus additional $\mathrm{Q}_{\mathrm{S}_2}(\mathcal{A}) + 1$ calls to each of them on line 5 (the latter is necessary in case $\mathcal{A}$ guesses the output of C_2 on some points). Finally, the running times of $\mathcal{A}$ and $\mathcal{B}$ are roughly the same, $\tau' \approx \tau$. □

6.5 Breaking *IZ*[CSI] IMP-CA security implies solving OMGA

Theorem 10. *Let $\mathit{IZ}[\mathsf{CSI}] = (\mathit{IZ.PG}, \mathit{IZ.KG}, \mathit{IZ.P} = (\mathit{IZ.P}_1, \mathit{IZ.P}_2), \mathit{IZ.Ver})$ be the instantiation of IZ with commutative supersingular isogeny parameter generator $\mathit{IZ}[\mathsf{CSI}].\mathit{PG}$, and let $l(\kappa) := \max\{\lg|\mathcal{C}|, \lg|\mathcal{D}|\}$ where $\mathcal{C}$ and $|\mathcal{D}$ are the challenge spaces (i.e. $l(\kappa)$ is the maximum challenge bitlength). Let $\mathcal{A} := (\mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}, \mathcal{P}^{\mathrm{C}_1,\mathrm{C}_2})$ be an IMP-CA adversary of time complexity $\tau(\kappa)$ attacking $\mathit{IZ}[\mathsf{CSI}]$. Then there exists an OMGA adversary $\mathcal{I}$ attacking $\mathit{IZ}[\mathsf{CSI}].\mathit{PG}$ such that, for every κ, $\mathbf{Adv}^{\mathbf{IMP\text{-}CA}}_{IZ}(\mathcal{A}, \kappa) \leq 2^{-l(\kappa)} + \sqrt{\mathbf{Adv}^{\mathbf{OMGA}}_{IZ.PG}(\mathcal{I}, \kappa)}$. Furthermore, the time complexity of $\mathcal{I}$ is $\tau'(\kappa) = 2\tau(\kappa) + O(\kappa^2 \cdot (\alpha(\kappa) + \ell(\kappa)))$, where $\ell(\kappa)$ is the number of prover sessions with which $\mathcal{V}$ interacts and where $\alpha(\kappa) \in O(\mathrm{poly}(\kappa))$ is the cost of computing a class group action.*

Proof. Again, we closely follow the proof [5, Theorem 5.1]. We assume w.l.o.g. that $\mathcal{V}$ never repeats a request. Fix $\kappa \in \mathbb{N}$ and let (N, A_0, C, t, C', T) be an output of $\mathit{IZ}[\mathsf{CSI}].\mathit{PG}$ running on input 1^κ. Adversary $\mathcal{I}$ has access to a class group action solving oracle $\mathcal{I}.\mathit{Sol}(A \in \mathcal{E}\ell\ell_p^{\mathsf{r}' \times \mathsf{c}}, Q \in \mathcal{E}\ell\ell_p^{\mathsf{r} \times \mathsf{c}})$ that returns $a \in \mathbb{Z}_N^{\mathsf{r} \times \mathsf{c}}$ such that $Q = [a]A$, and a supersingular curve challenge oracle $\mathcal{I}.\mathit{Ch}(\mathsf{r}, \mathsf{c})$ that takes the desired numbers of rows (r) and columns (c) as inputs and returns random challenge curves $Q \in \mathcal{E}\ell\ell_p^{\mathsf{r} \times \mathsf{c}}$ each time it is invoked. $\mathcal{I}$ calls the $\mathcal{I}.\mathit{Ch}(\cdot, \cdot)$ oracle $2\ell + 1$ times for a total of $\ell\mathsf{rc} + \mathsf{r}'$ challenge curves (for some r, c, and r'), and tries to compute the corresponding class group actions, while making only 2ℓ queries to $\mathcal{I}.\mathit{Sol}(\cdot, \cdot)$ for a total of $\ell\mathsf{rc}$ solutions. $\mathcal{I}$ simulates an interaction between $\mathcal{B}$ and the prover sessions (that is, $\mathcal{I}$ executes $\mathcal{V}$, which is only allowed to query its P_1 and P_2 oracles). To do so, $\mathcal{I}$ first queries its $\mathcal{I}.Ch(\cdot\,,\cdot)$ oracle, obtaining $C-1$ random supersingular curves $Q_0 \xleftarrow{\$} \mathcal{I}.Ch(1, C-1) \in \mathcal{E}\ell\ell_p^{C-1}$, and uses them to create a public key pk for the IMP-CA adversary $\mathcal{B}$. It then runs $\mathcal{V}^{\mathrm{P}_1,\mathrm{P}_2}$ and answers its requests.

In response to a $\mathrm{P}_1(\cdot)$ request, $\mathcal{I}$ opens a new prover session pid, makes queries $Q_{pid,0} \xleftarrow{\$} \mathcal{I}.Ch(C'-1, t) \in \mathcal{E}\ell\ell_p^{(C'-1)\times t}$ and $Q_{pid,1} \xleftarrow{\$} \mathcal{I}.Ch(T, t) \in \mathcal{E}\ell\ell_p^{T\times t}$, and returns the answer $(pid, Q_{pid,0}, Q_{pid,1})$ to $\mathcal{V}$. This mimics the behavior of a genuine prover session pid, which would start from a fresh random tape and the public key pk, sample commitments U_{pid}, V_{pid}, and return (pid, U_{pid}, V_{pid}) to $\mathcal{V}$.

$\mathcal{I}$ does not know the secret key $a_0 \in \mathbb{Z}_N^{C-1}$ corresponding to Q_0, which the genuine prover sessions would use to respond to $\mathcal{V}$'s $\mathrm{P}_2(pid, c \in \mathcal{C}, d \in \mathcal{D})$ requests, but it compensates with its access to the $\mathcal{I}.Sol(\cdot\,,\cdot)$ oracle to answer these requests. Specifically, in response to request $\mathrm{P}_2(pid, c_{pid}, d_{pid})$, $\mathcal{I}$ makes the queries $z_{pid} \leftarrow \mathcal{I}.Sol((Q_0)_{\downarrow c_{pid}}, Q_{pid,0}) \in \mathbb{Z}_N^{(C'-1)\times t}$ and $w_{pid} \leftarrow \mathcal{I}.Sol(((Q_0)_{\downarrow c_{pid}} \,\|\, Q_{pid,0})^{\uparrow d_{pid}}, Q_{pid,1}) \in \mathbb{Z}_N^{T\times t}$, and returns the answer w_{pid}. This is exactly the expected response from the prover session pid, as one can see with the correspondence $A \leftrightarrow Q_0$, $U_{pid} \leftrightarrow Q_{pid,0}$, $V_{pid} \leftrightarrow Q_{pid,1}$. Hence $\mathcal{I}$ simulates the prover's behavior perfectly.

Since $\mathrm{Q}_{\mathrm{P}_2} = \ell(\kappa)$ is the number of prover sessions that $\mathcal{V}$ opens, interacts with, and closes, when $\mathcal{V}$ stops $\mathcal{I}$ has made $2\ell(\kappa)$ queries to its class group action oracle for a total of $(C'-1+T)t\cdot\ell(\kappa)$ curves queried from $\mathcal{I}.Ch(\cdot\,,\cdot)$ and the same amount of class group actions queried from $\mathcal{I}.Sol(\cdot\,,\cdot)$. Now $\mathcal{I}$ needs to find the class group action of at least one more challenge curve to win the game. In fact, it will find all the class group actions for all $C-1$ from the first query to the $\mathcal{I}.Ch(\cdot\,,\cdot)$ oracle. Specifically, $\mathcal{I}$ will first attempt to extract the class group action a_0 of the challenge Q_0 from $\mathcal{P}$, which is initialized with the output $\mathbf{st}_{\mathcal{V}}$ of $\mathcal{V}$, and it will then use a_0 to compute the class group action of each of the other challenge curves.

To that end, $\mathcal{I}$ obtains the commitment pair $(U, V) \xleftarrow{\$} \mathcal{P}.\mathrm{F}_1(\mathbf{st}_{\mathcal{P}}) \in \mathcal{E}\ell\ell_p^{(C'-1)\times t} \times \mathcal{E}\ell\ell_p^{T\times t}$, samples a uniformly random challenge pair $(c \xleftarrow{\$} \mathcal{C}, d \xleftarrow{\$} \mathcal{D})$, runs $w \xleftarrow{\$} \mathcal{P}.\mathrm{F}_2(\mathbf{st}_{\mathcal{P}}, c, d) \in (\mathbb{Z}_N^*)^{T\times t}$ to obtain the response to the challenge pair, and evaluates the verification predicate $b \leftarrow IZ[\mathsf{CSI}].Ver(pk, U, V, c, d, w)$. $\mathcal{I}$ then selects another challenge pair $(c' \xleftarrow{\$} \mathcal{C}, d' \leftarrow d)$ where c' is chosen uniformly at random but the same $d' = d$ is kept as before, rewinds $\mathcal{P}$ to the same starting state $\mathbf{st}_{\mathcal{P}} \leftarrow \mathbf{st}_{\mathcal{V}}$, and reruns it to eventually obtain its response $w' \xleftarrow{\$} \mathcal{P}.\mathrm{F}_2(\mathbf{st}_{\mathcal{P}}, c', d') \in (\mathbb{Z}_N^*)^{T\times t}$ to the new challenge pair, then evaluates the verification predicate $b' \leftarrow IZ[\mathsf{CSI}].Ver(pk, U, V, c', d', w')$.

If the verification predicates evaluate to $b = b' = 1$, both tuples (U, V, c, d, w) and (U, V, c', d', w') must be valid for public key $Q_0 = [a_0]A_0$. That means $V = [w]H^{\uparrow d} = [w']H'^{\uparrow d'}$ where $H' = [z']A_{\downarrow c'}$ and $H = [z]A_{\downarrow c}$, and hence $H^{\uparrow d} = [z^{\uparrow d}]A_{\downarrow c} = [z^{\uparrow d} + x_{\downarrow c}]A_0$ and $H'^{\uparrow d'} = [z'^{\uparrow d'}]A_{\downarrow c'} = [z'^{\uparrow d'} + x_{\downarrow c'}]A_0$ whereby $[w][z^{\uparrow d} + x_{\downarrow c}]A_0 = [w'][z'^{\uparrow d'} + x_{\downarrow c'}]A_0$, that is, $w + z^{\uparrow d} + x_{\downarrow c} = w' + z'^{\uparrow d'} + x_{\downarrow c'}$ (mod N), so it must hold that $x_{\downarrow c} - x_{\downarrow c'} = w' - w + z'^{\uparrow d'} - z^{\uparrow d}$ (mod N). If furthermore $c' \neq c$ but $d' = d$, then taking only the indices k such that $d^k = 0$

yields $x_{\downarrow c} - x_{\downarrow c'} = w' - w \pmod{N}$, that is, a system of t linear equations in the $C-1$ unknowns $x_{1\ldots C-1}$, which reveals the whole private key for $t \geq C-1$ (which is the case, from the parameter condition $C \mid t$) and a suitable choice of c_j (e.g. any index j such that $c_j \neq 0$ and $c'_j = 0$ reveals $x_{\downarrow c_j} = w'_j - w_j \pmod{N}$ directly).

Now we know that $z_{pid} = a_{pid,0} - (a_0)_{\downarrow c_{pid}} \pmod{N}$ where $[a_{pid,0}]A_0 = Q_{pid,0}$, and $w_{pid} = a_{pid,1} - z_{pid}^{d_{pid}} \pmod{N}$ where $[a_{pid,1}]A_0 = Q_{pid,1}$, hence $\mathcal{I}$ can recover all the $(C'-1+T)t\cdot\ell(\kappa)$ class group actions $a_{pid,0} \leftarrow z_{pid} + (a_0)_{\downarrow c_{pid}}$ $\pmod{N}$ and $a_{pid,1} \leftarrow w_{pid} + z_{pid}^{d_{pid}} \pmod{N}$, $1 \leq pid \leq \ell(\kappa)$, as desired.

If the verification predicates do not evaluate to 1 on both occasions, or the challenges c and c' coincide, then $\mathcal{I}$ fails. That is, $\mathcal{I}$ wins if, and only if, $b = b' = 1$ and $c \neq c'$.

Relating the probability of this event with the IMP-CA advantage of adversary $\mathcal{B}$ proceedings exactly as in Theorem 5, and leads to precisely the same result, which is the claimed relation $\mathbf{Adv}_{IZ}^{\mathbf{IMP\text{-}CA}}(\mathcal{A},\kappa) \leq 2^{-l(\kappa)} + \sqrt{\mathbf{Adv}_{IZ.PG}^{\mathbf{OMGA}}(\mathcal{I},\kappa)}$.

To complete the proof, it remains to justify the claim about the time complexity of adversary $\mathcal{I}$. Consider the OMGA game from Section 4.4, which defines $\mathbf{Adv}_{PG}^{\mathbf{OMGA}}(\mathcal{C},\kappa)$. The cost of all the steps of this game before the execution of adversary $\mathcal{I}$'s final check that $c' \neq c$ and $b = b' = 1$ is dominated by at most $2\tau(\kappa)$ for the whole interaction between the prover P and the cheating verifier $\mathcal{V}$, plus the cost of evaluating the verification predicate twice to obtain $b \leftarrow \mathit{IZ}.\mathit{Ver}(pk, U, V, c, d, w)$ and $b' \leftarrow \mathit{IZ}.\mathit{Ver}(pk, U, V, c', d', w')$, plus the cost of retrieving a_0 and all of the $a_{pid,0}$ and $a_{pid,1}$.

Each evaluation of the verification predicate involves computing Tt class group actions with integer representatives of bitlength $\lg N \in O(\mathrm{poly}(\kappa))$ (disregarding the cost of other operations, which are much cheaper by comparison). From the conditions $|\mathcal{C}| = |\mathcal{M}_C^t| = t!/(t/C)!^C \geq 2^\kappa$ and $|\mathcal{D}| = |\mathcal{M}_{C'}^T| = T!/(T/C')!^{C'} \geq 2^\kappa$ it follows, on the one hand, that $l(\kappa) = \max\{\lg|\mathcal{C}|, \lg|\mathcal{D}|\} \geq \kappa$, which immediately precludes exhaustively guessing the verifier's challenges since that would require $l(\kappa) \in O(\mathrm{polylog}(\kappa))$, and on the other hand, that $\lg C$, $\lg t$, $\lg C'$, $\lg T \in O(W(\kappa))$ where $W(\cdot)$ is Lambert's function, and hence $Tt \in O(\kappa^2/\log^2(\kappa)) \subseteq O(\kappa^2)$, so the total cost is $O(\kappa^2 \cdot \alpha(\kappa))$. Finally, the cost of retrieving a_0 is that of solving a sparse linear system (each equation involving only two unknowns) of size $(C'-1) \times (C'-1)$, or $O(\kappa^2)$, and the cost of retrieving all of the $(C'-1+T)t \cdot \ell(\kappa) \in O(\kappa^2 \cdot \ell(\kappa))$ remaining class group actions is $O(\ell(\kappa)\cdot\kappa^2)$. Therefore, overall the time complexity of $\mathcal{I}$ is $\tau'(\kappa) := 2\tau(\kappa) + O(\kappa^2\cdot\alpha(\kappa) + \kappa^2 + \ell(\kappa)\cdot\kappa^2) = 2\tau(\kappa) + O(\kappa^2\cdot(\alpha(\kappa)+\ell(\kappa)))$. □

Corollary 2. *If the commutative supersingular isogeny parameter generator* $\mathit{IZ}[\textsf{CSI}].\mathit{PG}$ *is OMGA-secure, then the correspondingly parametrized* $\mathit{IZ}[\textsf{CSI}]$ *is IMP-CA-secure.*

7 Concrete parameters

$BZ[\mathsf{DL}]$ is a conventional discrete-logarithm scheme, and can be instantiated on top of standard elliptic curves at the desired security level, e.g. any of the curves recommended in Draft NIST SP 800-186 [12].

We compare $BZ[\mathsf{DL}]$ with the Tessaro-Zhu BS_3 scheme [23], a recent blind signature proposal based on the DL assumption. In that scheme, public keys have the form $(X, Z) \in \mathbb{G}^2$ and thus occupy $2 \lg p$ bits each, while signatures have the form $(c, s, y, t) \in \mathbb{Z}_n^* \times \mathbb{Z}_n \times \mathbb{Z}_n^* \times \mathbb{Z}_n$ and thus occupy $4 \lg n$ bits each. The signer must perform 3 exponentiations, the user must perform $4+4 = 8$ exponentiations (and one inversion in $\mathbb{Z}_n^*$), and the verifier must perform 4 exponentiations.

In comparison, in the $BZ[\mathsf{DL}]$ scheme public keys have the form $y \in \mathbb{G}$ and thus occupy $\lg p$ bits each, while signatures have the form $(h, d, w) \in \mathbb{G}^* \times (\mathbb{Z}_n^*)^2$ and thus occupy $\lg p + 2 \lg n \approx 3 \lg n$ bits each. The signer must perform 2 exponentiations, the user must perform $4 + 3 = 7$ exponentiations, and the verifier must perform 3 exponentiations.

For $BZ[\mathsf{CSI}]$ the choice of parameters is less immediate. Although the complexity of the best classical attack against CSIDH is fully exponential, the CSIDH problem is solvable in quantum subexponential time [11]. This requires adopting large class groups, and computing the order of such groups is known to be a classically (though not quantumly) hard problem, related to computing discrete logarithms in imaginary quadratic orders [8]. The largest class group whose order is known exactly seems to be that of CSIDH-512 [7], and obtaining new records transcends the scope of this work. Yet, one can estimate what the cost of $BZ[\mathsf{CSI}]$ will be in practice when the corresponding class group order is available. A given choice of C, t, C', T, and the underlying field $\mathbb{F}_p$, which also determines the class group order N, yields signatures of size $|\sigma| = \lceil (C' - 1)t \lg p \rceil + \lceil T \lg C' \rceil + \lceil Tt \lg N \rceil$ bits, and public keys of size $|pk| = \lceil (C' - 1) \lg p \rceil$. The main source of processing time complexity is the number of class group actions that must be computed. Given parameters as above, the signer must compute $(C' - 1)t + Tt$ actions, the user $(C' - 1)t + Tt + Tt$ actions, and the verifier Tt actions.

The usual 128-bit security level requires adopting CSIDH-4096 where $\lg p \approx 4096$, $\lg N \approx 2048$. The best choice of parameters seems to be $C = 35$, $t = 35$, $C' = 11$, $T = 44$, for signatures of size 560 KiB and public keys of size 17 KiB. With this choice the signer must compute 1890 actions, the user must compute 3430 actions, and the verifier must compute 1540 actions. The processing time is thus bound to be quite long, and the $BZ[\mathsf{CSI}]$ scheme is limited to niche applications.

The bandwidth requirements are clearly much larger than quantum-susceptible schemes (like $BZ[\mathsf{DL}]$ itself) and again limited to niche applications, but they are not as high as it may look for a post-quantum proposal. For instance, these space requirements are substantially smaller than the Hauck-Kiltz-Loss-Nguyen lattice-based blind signature scheme [18] yields signatures of size 7.73 MiB and public keys of size 444 KiB.

These results are summarized on Table 1, where 'exp' stands for number of group element exponentiations and 'act' stands for number of class group actions.

Table 1. Blind signature comparison

scheme	$\|pk\|$	$\|\sigma\|$	signer	user	verifier
[23]	512 bits	1024 bits	3 exp	8 exp	4 exp
$BZ[\mathsf{DL}]$	256 bits	768 bits	2 exp	7 exp	3 exp
[18]	444 KiB	7.73 MiB	NA	NA	NA
$BZ[\mathsf{CSI}]$	17 KiB	560 KiB	1890 act	3430 act	1540 act

7.1 Smaller signatures and public keys?

There is a simple way to obtain better parameters for the underlying *plain* signatures constructed according to the BZ methodology (that is, without the blinding mechanism). It consists of grouping together the commitments associated to the same challenge value, that is, viewing all $C'' := 1 + t/C$ curves $\hat{H}''^{(0\ldots C'')} := (A_b \,\|\, [\hat{U}_j^{(\cdot)} \mid 1 \le j \le t \wedge \hat{c}_j = b])$ as constituting the same public multikey from the starting curve A_b. In that case, a single commitment per challenge is needed, the above matrices become single-row vectors and the $(\cdot)$ superscript in $\hat{U}_{1\ldots t}^{(\cdot)}$ becomes superfluous. The best choice of parameters becomes $C = 4$, $t = 72$, $C'' = 19$, $T = 38$, for signatures of size 74.1 KiB and public keys of size 1.5 KiB. If this arrangement worked for blind signatures, the signer would have to compute 224 actions, the user would have to compute 376 actions, and the verifier would have to compute 152 actions, a substantial improvement over the above parameter choice.

The obstacle for the above idea is that, since the actual constitution of the $\hat{H}''^{(0\ldots C'')}$ keys (and the corresponding assignment of $\hat{V}_{1\ldots t}^{1\ldots T}$ commitments) is determined by the challenges, the π and ρ permutations would have to be chosen to keep track of the permuted constitution of those multikeys *before the challenges are even made*. Recognizing the right commitments that constitute the same each multikey is thus hindered by an apparent circularity issue. Whether this can be circumvented is unclear at this time, and overcoming it is left as an open problem.

8 Conclusion

We have proposed a novel methodology to help designing blind signature schemes in both the classical and the post-quantum setting, shorthanded BZ for its reliance on zero-knowledge arguments. To showcase its potential, we have described two instantiations, one based on the conventional discrete logarithm assumption and one based on the CSIDH assumption. The results are arguably competitive in practice with existing proposals based on other assumptions.

Being non-black-box, the *BZ* methodology may or may not be applicable to a given underlying primitive. In particular, at this time it is unclear whether it can be applied efficiently to hardness assumptions like lattices and codes despite many structural similarities with the actual instantiations covered herein (for instance, we could not find any sensible permutation and displacement operations that may be compatible with the notions of short vectors or low Hamming-weight error patterns that are commonplace in those setting). We leave such possibilities (or impossibility proofs) as open problems.

References

1. M. Abe and T. Okamoto. Provably secure partially blind signatures. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, pages 271–286, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. DOI:10.1007/3-540-44598-6_17.
2. G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, R Perlner, A. Robinson, D. Smith-Tone, and L. Yi-Kai. *NIST IR 8413: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process.* NIST, July 2022. https://csrc.nist.gov/publications/detail/nistir/8413/final.
3. F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS)*, pages 1087–1098, New York, NY, USA, 2013. Association for Computing Machinery. DOI:10.1145/2508859.2516687.
4. F. Baldimtsi and A. Lysyanskaya. On the security of one-witness blind signature schemes. In K. Sako and P. Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 82–99, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. DOI:10.1007/978-3-642-42045-0_5.
5. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, pages 162–177, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. DOI:10.1007/3-540-45708-9_11.
6. F. Benhamouda, T. Lepoint, J. Loss, M. Orrù, and M. Raykova. On the (in)security of ROS. In A. Canteaut and F.-X. Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 33–53, Cham, 2021. Springer International Publishing. DOI:10.1007/978-3-030-77870-5_2.
7. W. Beullens, T. Kleinjung, and F. Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology – ASIACRYPT 2019*, pages 227–247, Cham, 2019. Springer International Publishing. DOI:10.1007/978-3-030-34578-5_9.
8. J.-F. Biasse. Improvements in the computation of ideal class groups of imaginary quadratic number fields. *Advances in Mathematics of Communications*, 4(2):141–154, 2010. DOI:10.3934/amc.2010.4.141.
9. W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. CSIDH: An efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427, Cham, 2018. Springer International Publishing. DOI:10.1007/978-3-030-03332-3_15.
10. D. Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US. DOI:10.1007/978-1-4757-0602-4_18.

11. J. Chávez-Saab, J.-J. Chi-Domínguez, S. Jaques, and F. Rodríguez-Henríquez. The SQALE of CSIDH: Sublinear Vélu quantum-resistant isogeny action with low exponents. *Journal of Cryptographic Engineering*, early online access, 2021. DOI:10.1007/s13389-021-00271-w.
12. L. Chen, D. Moody, A. Regenscheid, and K. Randall. *Draft NIST Special Publication 800-186: Recommendations for Discrete Logarithm-Based Cryptography: Elliptic Curve Domain Parameters*. NIST, October 2019. https://csrc.nist.gov/publications/detail/sp/800-186/draft.
13. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology – CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer. DOI:10.1007/3-540-47721-7_12.
14. M. Fischlin and D. Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 197–215, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. DOI:10.1007/978-3-642-13190-5_10.
15. D. Galindo and F. Garcia. A Schnorr-like lightweight identity-based signature scheme. In *Progress in Cryptology – AFRICACRYPT 2009*, pages 135–148, Berlin, Heidelberg, 2009. Springer. DOI:10.1007/978-3-642-02384-2_9.
16. R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20:51–83, 2007. DOI:10.1007/s00145-006-0347-3.
17. E. Hauck, E. Kiltz, and J. Loss. A modular treatment of blind signatures from identification schemes. In Y. Ishai and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 345–375, Cham, 2019. Springer International Publishing. DOI:10.1007/978-3-030-17659-4_12.
18. E. Hauck, E. Kiltz, J. Loss, and N. K. Nguyen. Lattice-based blind signatures, revisited. In *Advances in Cryptology – CRYPTO 2020*, pages 500–529, Cham, 2020. Springer International Publishing. 10.1007/978-3-030-56880-1_18.
19. J. Katz, D. Schröder, and A. Yerukhimovich. Impossibility of blind signatures from one-way permutations. In Yuval Ishai, editor, *Theory of Cryptography*, pages 615–629, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. DOI:10.1007/978-3-642-19571-6_37.
20. G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille. Simple Schnorr multi-signatures with applications to Bitcoin. *Designs, Codes and Cryptography*, 87:2139–2164, 2019. 10.1007/s10623-019-00608-x.
21. C. P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology – CRYPTO '89 Proceedings*, pages 239–252, New York, NY, 1990. Springer New York. DOI:10.1007/0-387-34805-0_22.
22. E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford. Keeping authorities "honest or bust" with decentralized witness cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 526–545, 2016. DOI:10.1109/SP.2016.38.
23. S. Tessaro and C. Zhu. Short pairing-free blind signatures with exponential security. Cryptology ePrint Archive, Paper 2022/047, 2022. https://eprint.iacr.org/2022/047.
24. D. Wagner. A generalized birthday problem. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, pages 288–303, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. DOI:10.1007/3-540-45708-9_11.
25. Alexandros Zacharakis, Panagiotis Grontas, and Aris Pagourtzis. Conditional blind signatures. Cryptology ePrint Archive, Paper 2017/682, 2017. https://eprint.iacr.org/2017/682.

Supplementary material

A The (in)effectiveness of the ROS attack

The ROS attack [6] and its generalized version are powerful cryptanalytical tools, capable of breaking blind Schnorr signatures at the 128-bit security level in $\approx$ 20s on a single PC. These attacks similarly break many other blind signature schemes in probabilistic polynomial time given $O(\log n)$ valid signatures: Okamoto-Schnorr, Syta *et al.*'s CoSi scheme [22], Maxwell *et al.*'s two-round MuSig blind schemes [20]; the threshold scheme by Gennaro *et al.* [16]; the Abe-Okamoto [1] and Anonymous Credentials Light [3] partially blind schemes; the conditional blind scheme by Zacharakis *et al.* [25]; and others.

We now investigate the applicability of such attacks against the *BZ* scheme. Spoiler: surprisingly, if the hash function is fixed-point resistant, the *BZ* scheme appears impervious to the ROS and generalized ROS attacks.

For the basic ROS attack, the goal is to construct a probabilistic polynomial-time adversary $\mathcal{A}$ that is able to produce (with overwhelming probability) $\ell + 1$ blind signatures after opening $\ell \geq \lceil \log n \rceil$ parallel sessions with the signer. We also investigate the more powerful generalized ROS attack which enhances the basic attack with Wagner's subexponential k-list attack [24].

In the remainder of this Appendix we adopt the additive elliptic curve arithmetic notation, so as to keep the presentation closer to the original attack from [6].

A.1 Basic attack against blind Schnorr

We begin by recapitulating the basic attack against blind Schnorr signatures. Let G be a generator of the group $\mathbb{G}$ of prime order n, and let the signer's key pair be $(x \in \mathbb{Z}_n^*, X := xG \in \mathbb{G}^*)$.

$\mathcal{A}$ chooses $\ell+1$ arbitrary messages $m_i \in \{0,1\}^*$ and samples $2(\ell+1)$ blinding elements $(\alpha_{i,b}, \beta_{i,b}) \xleftarrow{\$} (\mathbb{Z}_n^*)^2$, $1 \leq i \leq \ell+1$, $b \in \{0,1\}$.

$\mathcal{A}$ requests ℓ parallel commitments $\hat{U}_i \in \mathbb{G}^*$ from the signer, and computes 2ℓ blind pre-commitments $U_{i,b} \leftarrow \hat{U}_i + \alpha_{i,b}G + \beta i, bX$ and 2ℓ pre-challenges $c_{i,b} \leftarrow \mathcal{H}(U_{i,b}, m_i) \in \mathbb{Z}_n^*$, $1 \leq i \leq \ell$, $b \in \{0,1\}$. Assume $c_{i,0} \neq c_{i,1}$.

Define the multivariate affine polynomial $\boldsymbol{\tau} \in \mathbb{Z}_n[x_1 \dots x_\ell]$:

$$\boldsymbol{\tau}(x_1 \dots x_\ell) := \sum_{j=1}^{\ell} 2^{j-1} \cdot \frac{x_j - c_{j,0}}{c_{j,1} - c_{j,0}} = \sum_{j=1}^{\ell} \tau_j x_j + \tau_0.$$

The polynomial so defined satisfies $\boldsymbol{\tau}(c_{1,b_1} \dots c_{\ell,b_\ell}) = \sum_{j=1}^{\ell} 2^{j-1} b_j$ for all $(b_1 \dots b_\ell) \in \{0,1\}^\ell$, that is, any binary sequence of length ℓ can be expressed as the output from $\boldsymbol{\tau}$ evaluated on a selection of the $c_{1,b} \dots c_{\ell,b}$ spelled out by that binary sequence.

With that polynomial, $\mathcal{A}$ sets $U_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{U}_j$ (NB: omitting the τ_0 term), computes $c_{\ell+1} \leftarrow \mathcal{H}(U_{\ell+1}, m_{\ell+1})$, and obtains the binary decomposition $\sum_{j=1}^{\ell} 2^{j-1} b_j := c_{\ell+1} + \tau_0$, thereby specifying the actual $(b_1 \ldots b_\ell)$.

$\mathcal{A}$ blinds the challenges $\hat{c}_i \leftarrow c_{i,b_i} + \beta_{i,b_i} \pmod{n}$ and sends them to the signer, obtaining back the responses $\hat{z}_i \in \mathbb{Z}_n^*$, $1 \leq i \leq \ell$.

Now $\mathcal{A}$ defines $z_i \leftarrow \hat{z}_i + \alpha_{i,b_i} \pmod{n}$, $1 \leq i \leq \ell$, and $z_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{z}_j \pmod{n}$.

In summary, out of ℓ sessions $\mathcal{A}$ obtains $\ell + 1$ blind signatures:

$$(U_i, z_i) = \begin{cases} (\hat{U}_i + \alpha_{i,b_i} G + \beta_{i,b} X,\ \hat{z}_i + \alpha_{i,b_i} \pmod{n}), & 1 \leq i \leq \ell, \\ (\sum_{j=1}^{\ell} \tau_j \hat{U}_j,\ \sum_{j=1}^{\ell} \tau_j \hat{z}_j \pmod{n}), & i = \ell + 1. \end{cases}$$

A.2 Basic attack attempt against *BZ*[DL]

One could now try and adapt this attack to the particular way *BZ* blinds the signatures.

$\mathcal{A}$ chooses $\ell+1$ arbitrary messages $m_i \in \{0,1\}^*$ and samples $4(\ell+1)$ blinding elements $(\pi_{i,b}, \delta_{i,b}, \rho_{i,b}, \varepsilon_{i,b}) \xleftarrow{\$} (\mathbb{Z}_n^*)^4$, $1 \leq i \leq \ell + 1$, $b \in \{0,1\}$. $\mathcal{A}$ requests ℓ parallel commitments $(\hat{U}_i \in \mathbb{G}^*, \hat{V}_i \in \mathbb{G}^*)$ from the signer and computes 2ℓ blind pre-commitments $U_{i,b} \leftarrow \pi_{i,b} \hat{U}_i + \delta_{i,b} G$, 2ℓ pre-challenges $c_{i,b} \leftarrow \mathcal{H}(U_{i,b}, m_i)$, and 2ℓ pre-challenges $d_{i,b} \leftarrow \mathcal{H}(V_{i,b})$.

Now $\mathcal{A}$ chooses whether to mount the attempted ROS attack based on the primary commitments $\hat{U}_i$ or the secondary commitments $\hat{V}_i$.

When basing the attack on the *primary* commitments, $\mathcal{A}$ obtains the $\boldsymbol{\tau}$ polynomial, assuming $c_{i,0} \neq c_{i,1}$, $1 \leq i \leq \ell$, $b \in \{0,1\}$. $\mathcal{A}$ sets $U_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{U}_j$ (NB: omitting the τ_0 term), computes $c_{\ell+1} := \mathcal{H}(U_{\ell+1}, m_{\ell+1})$, and obtains the binary decomposition $\sum_{j=1}^{\ell} 2^{j-1} b_j := c_{\ell+1} + \tau_0$, thus specifying the actual $(b_1 \ldots b_\ell)$. $\mathcal{A}$ blinds the challenges $\hat{c}_i \leftarrow c_{i,b_i} / \pi_{i,b_i} \pmod{n}$ and, assuming the $\hat{d}_i$ are somehow computed as well (discussed below), sends them to the signer, who then computes but does not reveal the pre-responses $z_i \in \mathbb{Z}_n^*$, $1 \leq i \leq \ell$. This fixes the implicit one-more pre-response $z_{\ell+1} := \sum_{j=1}^{\ell} \tau_j z_j \pmod{n}$. The signer returns $\hat{w}_i = s_i - \hat{d}_i z_i \pmod{n}$, so from $\hat{H}_i := \hat{U}_i - \hat{c}_i X = z_i G$ $\mathcal{A}$ can only compute $H_i \leftarrow \pi_{i,b_i} \hat{H}_i + \delta_{i,b_i} G = z_i G$, $1 \leq i \leq \ell$, and $H_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{H}_j = z_{\ell+1} G$ with the implicit z_i and $z_{\ell+1}$.

Still, $\mathcal{A}$ needs to decide on how to complete the first ℓ blind signatures with d_i, w_i, and the blinding elements ρ_{i,b_i}, ε_{i,b_i}, and also find suitable $d_{\ell+1}$ and $w_{\ell+1}$ to complete the extra signature. The obvious way to obtain the first ℓ blind signatures is to set $\hat{d}_i \leftarrow d_i / \rho_{i,b_i} \pmod{n}$ and $w_i \leftarrow \rho_{i,b_i} \pi_{i,b_i} \hat{w}_i - \hat{d}_i \delta_{i,b_i} + \varepsilon_{i,b_i} \pmod{n}$, but regardless of how $\mathcal{A}$ chooses to do it, computing $d_{\ell+1}$ and $w_{\ell+1}$ incurs an issue. Specifically, this requires $w_{\ell+1} = s_{\ell+1} - d_{\ell+1} z_{\ell+1} \pmod{n}$ for the already fixed private signing key $z_{\ell+1} := \sum_{j=1}^{\ell} \tau_j z_j \pmod{n}$, and $s_{\ell+1}$ must depend on the corresponding but implicit (i.e. unknown to $\mathcal{A}$) s_i such that $\hat{V}_i = s_i G$. Since $\hat{w}_i = s_i - \hat{d}_i z_i \pmod{n}$, i.e. $\hat{w}_i / \hat{d}_i = s_i / \hat{d}_i - z_i \pmod{n}$, then $\sum_{j=1}^{\ell} \tau_j \hat{w}_j / \hat{d}_j = \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j - \sum_{j=1}^{\ell} \tau_j z_j = \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j - z_{\ell+1} \pmod{n}$, so it must hold that $d_{\ell+1} \sum_{j=1}^{\ell} \tau_j \hat{w}_j / \hat{d}_j = d_{\ell+1} \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j - d_{\ell+1} z_{\ell+1} \pmod{n}$. Hence $\mathcal{A}$ must take $w_{\ell+1} \leftarrow d_{\ell+1}(\sum_{j=1}^{\ell} \tau_j \hat{w}_j / \hat{d}_j + \xi) \pmod{n}$, allowing here for whatever fudge term ξ might be necessary to relate the actual choice of $w_{\ell+1}$ to this expression involving the $\hat{w}_i$, and thus implicitly set $s_{\ell+1} = d_{\ell+1}(\sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j + \xi) \pmod{n}$ and $V_{\ell+1} = s_{\ell+1} G = d_{\ell+1}(\sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j + \xi) G = d_{\ell+1} E$ where $E := \sum_{j=1}^{\ell} (\tau_j / \hat{d}_j) \hat{V}_j + \xi G$.

But $d_{\ell+1} = \mathcal{G}(V_{\ell+1}) = \mathcal{G}(d_{\ell+1} E)$, so $d_{\ell+1}$ must be a fixed point of the function $\mathcal{F}(x) := \mathcal{G}(xE)$. There is no known way to obtain such a $d_{\ell+1}$ efficiently (faster than collision finding) when $\mathcal{G}$ is modeled as a random oracle[7]. Some attack variations are plausible, e.g. including the $\hat{d}_i$ in the definition of the $\boldsymbol{\tau}$ polynomial, but in the end they face the same issue of needing to find a fixed point of a random oracle.

Suppose, then, that $\mathcal{A}$ bases the attack on the *secondary* commitments instead, obtaining the following modified $\boldsymbol{\psi}$ polynomial:

$$\boldsymbol{\psi}(x_1 \ldots x_\ell) := \sum_{j=1}^{\ell} 2^{j-1} \cdot \frac{x_j - d_{j,0}}{d_{j,1} - d_{j,0}} = \sum_{j=1}^{\ell} \psi_j x_j + \psi_0.$$

assuming $d_{i,0} \neq d_{i,1}$, $1 \leq i \leq \ell$, $b \in \{0,1\}$. $\mathcal{A}$ sets $V_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \psi_j \hat{V}_j$ (NB: omitting the ψ_0 term), computes $d_{\ell+1} := \mathcal{G}(V_{\ell+1})$, and obtains the binary decomposition $\sum_{j=1}^{\ell} 2^{j-1} b_j := d_{\ell+1} + \psi_0$, thus specifying the actual $(b_1 \ldots b_\ell)$. $\mathcal{A}$ blinds the challenges $\hat{c}_i \leftarrow c_{i,b_i} / \pi_{i,b_i} \pmod{n}$ and $\hat{d}_i \leftarrow d_{i,b_i} / \rho_{i,b_i} \pmod{n}$, and sends them to the signer, who responds with $\hat{H}_i$ and $\hat{w}_i$, $1 \leq i \leq \ell$. This fixes the one-more pre-response $w_{\ell+1} := \sum_{j=1}^{\ell} \psi_j \hat{w}_j \pmod{n}$.

But the signing key is no longer the same for all signatures: each secondary signature $(\hat{d}_i, \hat{w}_j)$ corresponds to a different signing key, so $\mathcal{A}$ must somehow make sure that the linear combination thus obtained will satisfy the verification equation. Indeed, $\hat{V}_i = \hat{w}_i G + \hat{d}_i \hat{H}_i \Rightarrow \sum_{j=1}^{\ell} \psi_j \hat{V}_j = \sum_{j=1}^{\ell} \psi_j \hat{w}_j G + \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{H}_j \Rightarrow V_{\ell+1} = w_{\ell+1} G + d_{\ell+1} \left(\frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{H}_j \right)$, revealing that $H_{\ell+1} = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{H}_j$ and hence $z_{\ell+1} = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j z_j$. Now $\hat{U}_i = z_i G + \hat{c}_i X \Rightarrow \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{U}_j = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j z_j G + \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{c}_j X$, revealing that $U_{\ell+1} = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{U}_j$ and $c_{\ell+1} = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{c}_j$ so that $U_{\ell+1} = z_{\ell+1} G + c_{\ell+1} X$ as expected. But it must also hold that $c_{\ell+1} = \mathcal{H}(U_{\ell+1}, m_{\ell+1})$, so from $c_{\ell+1} = \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell-1} \psi_j \hat{d}_j \hat{c}_j + \frac{1}{d_{\ell+1}} \psi_\ell \hat{d}_\ell \hat{c}_\ell$ (NB: any other index $1 \leq k < \ell$ could equivalently have been brought to focus here instead of ℓ) it follows that $\frac{\hat{c}_\ell}{\pi_{\ell,b_\ell}} = \frac{d_{\ell+1}}{\psi_\ell \hat{d}_\ell} \left(\mathcal{H} \left(\frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell} \psi_j \hat{d}_j \hat{U}_j, m_{\ell+1} \right) - \frac{1}{d_{\ell+1}} \sum_{j=1}^{\ell-1} \psi_j \hat{d}_j \hat{c}_j \right)$. And yet, π_{ℓ,b_ℓ} is fixed beforehand since it is required to compute V_ℓ and hence d_ℓ, which in turn is required to compute ψ_ℓ, and these must all already exist

[7] There exist specific hash functions that are not fixed-point resistant, e.g. plain free-start Davies-Meyer. However, most hash constructions like Miyaguchi-Preneel and cryptographic sponges are fixed-point resistant.

before the above hash is computed. This circularity is essentially similar to, but even more involved than, the issue with fixed points that $\mathcal{A}$ faces when trying to mount the attack based on the primary commitments, and just as intractable.

In summary, at this time no efficient method is known to obtain the required $d_{\ell+1}$ and $w_{\ell+1}$, and the ROS attack fails against $BZ[\mathsf{DL}]$.

Remark 2. For $BZ[\mathsf{CSI}]$ it is not clear even how to begin mounting the ROS attack, since the hash values are not taken from $\mathbb{Z}_n$ as the attack seems to require.

A.3 Basic attack attempt against $\boldsymbol{IZ}[\mathsf{DL}]$

The above discussion relies on properties of hash functions, so at first glance it would seem that the underlying $IZ[\mathsf{DL}]$ scheme might still be susceptible to a ROS-style attack. We now argue this is not the case.

Adapting the attack to the identification scheme, $\mathcal{A}$ samples $2(\ell+1)$ blinding elements $\delta_{i,b} \xleftarrow{\$} \mathbb{Z}_n^*$, $1 \le i \le \ell+1$, $b \in \{0,1\}$, requests ℓ pairs of parallel commitments $(\hat{U}_i, \hat{V}_i) \in (\mathbb{G}^*)^2$ from the prover, and computes 2ℓ pairs of blind parallel pre-commitments $U_{i,b} \leftarrow \hat{U}_i + \delta_{i,b}G$, $V_{i,b} \leftarrow \langle \mathit{something} \rangle$, where we leave the $\langle \mathit{something} \rangle$ expression open as it will have no impact on the discussion, regardless of how $\mathcal{A}$ decides to define it. Then $\mathcal{A}$ requests 2ℓ pre-challenges pairs $(c_{i,b}, d_{i,b}) \in (\mathbb{Z}_n^*)^2$ from the verifier, corresponding to the pre-commitment pairs $(U_{i,b}, V_{i,b})$. Assuming $c_{i,0} \ne c_{i,1}$, $1 \le i \le \ell$, $b \in \{0,1\}$, $\mathcal{A}$ obtains the $\boldsymbol{\tau}$ polynomial, sets $U_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{U}_j$, $V_{\ell+1} \leftarrow \langle \mathit{something} \rangle$, and requests one more challenge pair $(c_{\ell+1}, d_{\ell+1}) \in (\mathbb{Z}_n^*)^2$ from the verifier, corresponding to that commitment pair. $\mathcal{A}$ sends the challenges $(\hat{c}_i := c_{i,b_i}, \hat{d}_i := \langle \mathit{something} \rangle)$ to the prover, obtaining back $\hat{w}_i = s_i - \hat{d}_i z_i \pmod n$ for some (not revealed) pre-responses $z_i \in \mathbb{Z}_n^*$, $1 \le i \le \ell$. From these, $\mathcal{A}$ can only compute $\hat{H}_i := \hat{U}_i - \hat{c}_i X = z_i G$, $H_i \leftarrow \hat{H}_i + \delta_{i,b_i} G = z_i G$, $1 \le i \le \ell$, and $H_{\ell+1} \leftarrow \sum_{j=1}^{\ell} \tau_j \hat{H}_j = z_{\ell+1} G$, but still faces the same issue of ZK-proving knowledge of the z_i without knowing them. Specifically, this requires $w_{\ell+1} = s_{\ell+1} - d_{\ell+1} z_{\ell+1} \pmod n$ for the already fixed (but unknown) private signing key $z_{\ell+1} := \sum_{j=1}^{\ell} \tau_j z_j \pmod n$ and challenge $d_{\ell+1}$, and $s_{\ell+1}$ must depend on the corresponding (but also unknown) s_i such that $\hat{V}_i = s_i G$. Since $\hat{w}_i = s_i - \hat{d}_i z_i \pmod n$, it must hold that $d_{\ell+1} \sum_{j=1}^{\ell} \tau_j \hat{w}_j / \hat{d}_j = d_{\ell+1} \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j - d_{\ell+1} z_{\ell+1} \pmod n$. Hence $\mathcal{A}$ must take $w_{\ell+1} \leftarrow d_{\ell+1} \sum_{j=1}^{\ell} \tau_j \hat{w}_j / \hat{d}_j \pmod n$ and thus implicitly set $s_{\ell+1} \leftarrow d_{\ell+1} \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j \pmod n$ while also somehow ensuring $V_{\ell+1} = s_{\ell+1} G$.

And here we see the circularity problem: it appears in the relation $s_{\ell+1} \leftarrow d_{\ell+1} \sum_{j=1}^{\ell} \tau_j s_j / \hat{d}_j \pmod n$, as $\mathcal{A}$ must commit to $V_{\ell+1}$ (and hence somehow choose $s_{\ell+1}$) *before* obtaining the challenge $d_{\ell+1}$ from the verifier, and *without* ever knowing any of the s_j (so there is little if any hope of tweaking the $\hat{d}_j$ to force this relation to hold).

In summary, mounting an effective ROS-style attack against $IZ[\mathsf{DL}]$ faces an obstacle that seems as insurmountable as the ROS attack against $BZ[\mathsf{DL}]$ despite the absence of hashing functions, since here the protocol is interactive.

A.4 Generalized ROS attack against *BZ*?

Would the generalized ROS attack fare any better? At first glance this seems unlikely, given the above discussion on basing the attack on either the primary or the secondary commitments. And yet, one is entitled to wonder whether some different construction for the $\boldsymbol{\psi}$ polynomial might yield better results, especially taking into account that the $\hat{w}_i$ signature components are not hidden in an exponent as the z_i components are.

But even if this were the case, consider the following result:

Theorem 11 (Generalized ROS attack [6]). *Let* $L, w \geq 0$ *be integers. Under Wagner's conjecture, if* $\ell \geq \max\{2^w - 1, \lceil 2^w - 1 + \lambda - (w+1)L \rceil\}$*, then there exists an adversary that runs in expected time* $O(2^{w+L})$ *and solves the ROS problem relative to the parameter generation algorithm and dimension* ℓ*, where* $\lambda := \lceil \lg n \rceil$.

Since a single signature $(\hat{d}_i, \hat{w}_i)$ corresponds to each 'key pair' $(z_i, \hat{H}_i)$, the effective constraint for the SNARKs is $\ell = 1$, whereby $\max\{2^w - 1, \lceil 2^w - 1 + \lambda - (w+1)L \rceil\} \leq 1$. That means $w = 0$ or $w = 1$. For the former, $\lceil \lambda - L \rceil = \lambda - L \leq 1$, that is, $L \geq \lambda - 1$, and the cost is $O(2^{\lambda-1}) = O(2^\lambda)$. For the latter, $\lceil 1 + \lambda - 2L \rceil = 1 + \lambda - 2L \leq 1$, that is, $L \geq \lambda/2$, and the cost is $O(2^{1+\lambda/2}) = O(2^{\lambda/2})$.

So the most favorable situation for an attacker is the latter, but it coincides with the usual birthday limit. Therefore, the generalized ROS attack is just as ineffective as the plain attack.

B The *BZ*[DL] and *BZ*[CSI] signing formulas

We summarize the derivation of step 4 of Algorithm $BZ[\mathsf{DL}].U_2$ and step 4 of Algorithm $BZ[\mathsf{CSI}].U_2$, since these may perhaps not be immediately clear.

***BZ*[DL]:** Step 3 of Algorithm $BZ[\mathsf{DL}].U_2$ only ensures $\hat{v} = g^{\hat{w}}\hat{h}^{\hat{d}}$ with $\hat{h} := \hat{u}y^{-\hat{c}}$, while Algorithm $BZ[\mathsf{DL}].\mathit{Ver}$ requires $v = g^w h^d$ with $h := uy^{-c}$ instead, where $u = \hat{u}^\pi g^\delta$ and $v = \hat{v}^{\rho\pi} g^\varepsilon$ as defined in defined in step 3 of Algorithm $BZ[\mathsf{DL}].U_1$. Thus $\hat{v} = g^{\hat{w}}\hat{h}^{\hat{d}} \Rightarrow \hat{v}^{\rho\pi} g^\varepsilon = (g^{\hat{w}}\hat{h}^{\hat{d}})^{\rho\pi} g^\varepsilon \Rightarrow v = g^{\rho\pi\hat{w}}\hat{h}^{\rho\pi\hat{d}} g^\varepsilon = g^{\rho\pi\hat{w}+\varepsilon}\hat{h}^{\pi d}$. But $\hat{h} = \hat{u}y^{-\hat{c}} \Rightarrow \hat{h}^\pi = \hat{u}^\pi y^{-c} = ug^{-\delta}y^{-c} = hg^{-\delta}$. Hence $v = g^{\rho\pi\hat{w}+\varepsilon}\hat{h}^{\pi d} = g^{\rho\pi\hat{w}+\varepsilon}(hg^{-\delta})^d = g^{\rho\pi\hat{w}-\delta+\varepsilon}h^d = g^w h^d$, wherefore we obtain the signing formula $w = \rho\pi\hat{w} - d\delta + \varepsilon \pmod{n}$ adopted in step 4 of Algorithm $BZ[\mathsf{DL}].U_2$.

***BZ*[CSI]:** Step 3 of Algorithm $BZ[\mathsf{CSI}].U_2$ only ensures $\hat{V}_{1\ldots t}^{1\ldots T} = [\hat{w}_{1\ldots t}^{1\ldots T}]\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}}$ with $\hat{H}_{1\ldots t}^{0\ldots C'-1} := (A_{\downarrow\hat{c}_{1\ldots t}} \,\|\, \hat{U}_{1\ldots t}^{1\ldots C'-1})$, while Algorithm $BZ[\mathsf{CSI}].\mathit{Ver}$ requires $V_{1\ldots t}^{1\ldots T} = [w_{1\ldots t}^{1\ldots T}]H_{1\ldots t}^{\uparrow d^{1\ldots T}}$ with $H_{1\ldots t}^{0\ldots C'-1} := (A_{\downarrow c_{1\ldots t}} \,\|\, U_{1\ldots t}^{1\ldots C'-1})$ instead, where $U_{1\ldots t}^{1\ldots C'-1} = [\delta_{1\ldots t}^{1\ldots C'-1}]\pi(\hat{U}_{1\ldots t}^{1\ldots C'-1})$ and $V_{1\ldots t}^{1\ldots T} = [\varepsilon_{1\ldots t}^{1\ldots T}]\rho^\uparrow\pi_\downarrow(\hat{V}_{1\ldots t}^{1\ldots T})$ as defined in step 3 of Algorithm $BZ[\mathsf{CSI}].U_1$. Thus $\hat{V}_{1\ldots t}^{1\ldots T} = [\hat{w}_{1\ldots t}^{1\ldots T}]\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}} \Rightarrow [\varepsilon_{1\ldots t}^{1\ldots T}]\rho^\uparrow\pi_\downarrow(\hat{V}_{1\ldots t}^{1\ldots T}) = [\varepsilon_{1\ldots t}^{1\ldots T}]\rho^\uparrow\pi_\downarrow([\hat{w}_{1\ldots t}^{1\ldots T}]\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}}) \Rightarrow V_{1\ldots t}^{1\ldots T} =$

$[\varepsilon_{1\ldots t}^{1\ldots T}][\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T})]\rho^{\uparrow}\pi_{\downarrow}(\hat{H}_{1\ldots t}^{\uparrow\hat{d}^{1\ldots T}}) = [\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) + \varepsilon_{1\ldots t}^{1\ldots T}]\pi_{\downarrow}(\hat{H}_{1\ldots t}^{\rho(\uparrow\hat{d}^{1\ldots T})}) = [\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) + \varepsilon_{1\ldots t}^{1\ldots T}]\pi_{\downarrow}(\hat{H}_{1\ldots t}^{\uparrow d^{1\ldots T}})$. But $\hat{H}_{1\ldots t}^{0\ldots C'-1} = (A_{\downarrow\hat{c}_{1\ldots t}} \,\|\, \hat{U}_{1\ldots t}^{1\ldots C'-1}) \Rightarrow \pi_{\downarrow}(\hat{H}_{1\ldots t}^{0\ldots C'-1}) = (A_{\pi(\downarrow\hat{c}_{1\ldots t})} \,\|\, \pi_{\downarrow}(\hat{U}_{1\ldots t}^{1\ldots C'-1})) = (A_{\downarrow c_{1\ldots t}} \,\|\, \pi_{\downarrow}(\hat{U}_{1\ldots t}^{1\ldots C'-1})) = ([-\delta_{1\ldots t}^{0}]A_{\downarrow c_{1\ldots t}} \,\|\, [-\delta_{1\ldots t}^{1\ldots C'-1}]U_{1\ldots t}^{1\ldots C'-1}) = [-\delta_{1\ldots t}^{0\ldots C'-1}]H_{1\ldots t}^{0\ldots C'-1}$. Hence $V_{1\ldots t}^{1\ldots T} = [\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) + \varepsilon_{1\ldots t}^{1\ldots T}]\pi_{\downarrow}(\hat{H}_{1\ldots t}^{\uparrow d^{1\ldots T}}) = [\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) + \varepsilon_{1\ldots t}^{1\ldots T}][-\delta_{1\ldots t}^{\uparrow d^{1\ldots T}}]H_{1\ldots t}^{\uparrow d^{1\ldots T}} = [\rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) - \delta_{1\ldots t}^{\uparrow d^{1\ldots T}} + \varepsilon_{1\ldots t}^{1\ldots T}]H_{1\ldots t}^{\uparrow d^{1\ldots T}} = [w_{1\ldots t}^{1\ldots T}]H_{1\ldots t}^{\uparrow d^{1\ldots T}}$, wherefore we obtain the signing formula $w_{1\ldots t}^{1\ldots T} = \rho^{\uparrow}\pi_{\downarrow}(\hat{w}_{1\ldots t}^{1\ldots T}) - \delta_{1\ldots t}^{\uparrow d^{1\ldots T}} + \varepsilon_{1\ldots t}^{1\ldots T} \pmod N$ adopted in step 4 of Algorithm $BZ[\mathsf{CSI}].U_2$.