Introducing two ROS attack variants: breaking one-more unforgeability of BZ blind signatures

Bruno M. F. Ricardo [1], Lucas C. Cardoso [2], Leonardo Kimura [2], Marcos A. Simplicio Junior [2], Paulo L. Barreto [3]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (USP) – São Carlos, SP – Brazil

[2]Laboratório de Arquitetura e Redes de Computadores
Universidade de São Paulo (USP) – São Paulo, SP – Brazil.

[3]School of Engineering and Technology
University of Washington – Tacoma, U.S.A.

{brun0matheus, lucas.cupertino, leonardo.kimura}@usp.br

mjunior@larc.usp.br, pbarreto@uw.edu

Abstract. *In 2023, Barreto and Zanon proposed a three-round Schnorr-like blind signature scheme, leveraging zero-knowledge proofs to produce one-time signatures as an intermediate step of the protocol. The resulting scheme, called BZ, is proven secure in the discrete-logarithm setting under the one-more discrete logarithm assumption with (allegedly) resistance to the Random inhomogeneities in a Overdetermined Solvable system of linear equations modulo a prime number p attack, commonly referred to as ROS attack. The authors argue that the scheme is resistant against a ROS-based attack by building an adversary whose success depends on extracting the discrete logarithm of the intermediate signing key. In this paper, however, we describe a distinct ROS attack on the BZ scheme, in which a probabilistic polynomial-time attacker can bypass the zero-knowledge proof step to break the one-more unforgeability of the scheme. We also built a BZ variant that, by using one secure hash function instead of two, can prevent this particular attack. Unfortunately, though, we show yet another ROS attack that leverages the BZ scheme's structure to break the one-more unforgeability principle again, thus revealing that this variant is also vulnerable. These results indicate that, like other Schnorr-based strategies, it is hard to build a secure blind signature scheme using BZ's underlying structure.*

Keywords*: Blind signature. Schnorr. ROS. Zero-knowledge proofs. Cryptanalysis.*

1. Introduction

Blind signatures are cryptographic schemes that allow a user to obtain a signature on a message in such a way that the signer does not find out the actual contents of the message. Several applications, including electronic voting [Fuchsbauer and Wolf 2024, Fujioka et al. 1993] and e-cash [Chaum 1983], build upon the two main security properties provided by these schemes: *perfect blindness* and *one-more unforgeability*. The

first means that it should be computationally hard for a signer, after seeing n message-signature pairs, to correlate any pair to the specific interaction that originated it. Meanwhile, the second states that the user cannot output $n + 1$ signatures after having only n interactions with the signer.

EXISTING BLIND SIGNATURE SCHEMES. There are many blind signature schemes in the literature, each relying on different computational problems and presenting different challenges. Blind RSA [Lysyanskaya 2023, Denis et al. 2023] builds upon the one-more RSA assumption, yielding large signatures and key sizes for modern security levels [Chaum 1983, Coron 2000]. The blind ECDSA scheme proposed by [Qin et al. 2021] relies on a combination of additive homomorphic encryption and non-interactive zero-knowledge arguments. Blind BLS and other pairing-based blind signature schemes use pairing-friendly curves, which are slower than standard curves [Boldyreva 2002, Hanzlik et al. 2023]. There are also proposals for boosting techniques [Katz et al. 2021], which can improve the efficiency of some blind signature schemes while ensuring that they remain secure for (polynomially many) concurrent executions of the signing protocol. Finally, post-quantum alternatives are still a work in progress due to performance issues and flaws identified in their corresponding security proofs [Hauck et al. 2020, Katsumata et al. 2024].

BLIND SCHNORR SIGNATURE. In terms of efficiency, blind signature schemes based on regular Schnorr signatures [Schnorr 1990] are among the most prominent constructions [Schnorr 2001]. Unfortunately, however, the security of Blind Schnorr signatures commonly relies on the hardness of the Random inhomogeneities in an Overdetermined Solvable system of linear equations (commonly referred to as ROS) problem. Until the early 2000s, this assumption was considered reasonable for practical purposes, since the best-known attack, using a k-dimensional generalization of the birthday problem, could solve the ROS problem in sub-exponential time [Wagner 2002]. In 2021, though, Benhamouda et al. proposed a polynomial-time algorithm to solve the ROS problem modulo a prime number p in $\ell > \log p$ dimensions [Benhamouda et al. 2021], showing that malicious users could forge one-more signature using only poly-logarithm concurrent sessions. This led to renewed research interest in the topic, aiming to build Blind Schnorr variants that could resist to ROS attacks while still providing fast computations, small key sizes, and short signatures.

SCHNORR BZ. One of the Schnorr-based blind signature schemes recently proposed to resist ROS attacks, which is the focus of this manuscript, is the so-called *Schnorr Blind Signature Zero-Knowledge* (BZ) scheme, described in [Barreto et al. 2023, Barreto and Zanon 2023]. It consists of a three-move protocol where the usual, plain Schnorr signatures are replaced by non-interactive zero-knowledge arguments of knowledge arising from natural properties of Schnorr-based protocols. The security of BZ relies on the *one-more discrete logarithm assumption* [Bellare et al. 2003], hereby referred to as (OMDL), which is a stronger notion when compared to the discrete logarithm assumption. Further, the authors discuss how to mount possible ROS-based attacks against BZ [Barreto and Zanon 2023, Supplementary Material A], showing that they fail unless the attacker can solve a discrete logarithm. From those observations, the authors claim that the resulting BZ construction should be resistant against ROS-like attacks.

OUR CONTRIBUTION. In this paper, our contribution is threefold:

i) We describe a novel ROS-based attack that, albeit possibly of independent interest, is capable of violating the *one-more unforgeability* of the BZ scheme, bypassing the zero-knowledge step of the protocol;
ii) In an attempt to solve the issue, we present a variant of the BZ scheme, hereby referred to as *modified*–BZ (or *m*–BZ for short), which indeed resists to our first attack;
iii) Going further, we show yet another variant of ROS-based attack on *m*–BZ Scheme, which once again breaks its *one-more unforgeability* property, showing that it is hard to build a secure blind signature scheme using BZ's underlying structure.

PAPER OUTLINE. The rest of this paper is organized as follows. Section 2 introduces the mathematical notation and color scheme used throughout the paper. Section 3 outlines the required background on Schnorr signatures, the ROS problem, and the BZ scheme. Section 4 describes our first ROS-based attack, as well as the *m*–BZ variant built to avoid this attack. Section 5 presents the second ROS-based attack on the modified BZ scheme, for which we could not find a fix. Section 6 concludes the paper.

2. Premilinaries and Notation

SETS. For any finite set S, $x \leftarrow\!\$\ S$ denotes the uniformly random sampling of an element $x \in S$, while $|S|$ is the number of elements in the set. We use $[n]$ to mean the integers from 1 to n, and $[m : n]$ to represent the integers from m to $n \geq m$. We write $\{0,1\}^\infty$ to denote the set of all binary strings, and $\{0,1\}^*$ for the set of all binary strings with arbitrary finite size. Bit values are represented as $b \in \{0,1\}$.

ALGEBRA. We denote the ring of integers modulo p by $\mathbb{Z}_p$, while its subset of strictly positive values is denoted $\mathbb{Z}_p^* := \mathbb{Z}_p \backslash \{0\}$. All arithmetic operations between elements of $\mathbb{Z}_p$ are done modulo p, so we omit the $(\bmod\ p)$ notation for conciseness. We use additive notation for all operations in an Abelian group $\mathbb{G} = \langle G \rangle$ with prime order p (i.e., $|\mathbb{G}| = p$), where G is a generator of $\mathbb{G}$. Group elements are represented with uppercase letters (e.g., $G \in \mathbb{G}$), while non-group elements are represented as lowercase letters (e.g., $g \notin \mathbb{G}$). We use bold font style to represent vectors, such as $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$. Finally, we also use $\langle \mathbf{u}, \mathbf{v} \rangle$ to denote the usual inner-product between vectors $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_p^n$, i.e., $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{1 \leq i \leq n} u_i v_i$.

BLIND SIGNATURES. As further detailed in Section 3.1, the two main actors interacting in a blind protocol are a user U and a signer S. When describing their actions, we use a sub-index (generally 1 or 2) to represent which step they are executing during the protocol. For example, the first algorithm executed by the signer during the Sch protocol is referred to as $\mathsf{Sch.S}_1(\mathsf{param})$ where param is a set of inputs.

COLOR SCHEME. For easier reference, we use the same color scheme of [Barreto and Zanon 2023]: key pairs are colored in blue; generators in red; primary commitments in green; secondary commitments in magenta; primary challenges in purple; secondary challenges in brown; primary answers in turquoise; and secondary answers in wine-red. Transcript components from protocol sessions are indicated with a bar (e.g., $\overline{c}$), and blinding factors are indicated with Greek letters (except σ, which represents a signature).

3. Background

In this section, we present the theoretical background upon which the manuscript is built. We start by presenting Schnorr signatures in both original and blind versions; then, we present the ROS problem and the polynomial-time attack proposed by [Benhamouda et al. 2021] to solve it. Lastly, we describe the scheme proposed by Barreto and Zanon [Barreto and Zanon 2023], highlighting some aspects of its structure and cryptographic assumptions upon which its security depends.

3.1. Schnorr Signature Variants

ORIGINAL SCHNORR SIGNATURE. The original Schnorr signature scheme [Schnorr 1990] is widely used in modern cryptographic solutions. Examples include cryptocurrencies (e.g., Bitcoin recently adopted Schnorr signatures for curve secp256k1 [Wuille et al. 2020a] and Taproot [Wuille et al. 2020b]), threshold signatures [Bellare et al. 2022], identity-based schemes suited for resource-constrained environments [Galindo and Garcia 2009], implicit certification protocols for vehicular networks [Barreto et al. 2020], among others. The security of Schnorr signatures relies on the hardness of the discrete logarithm in prime order groups $\mathbb{G}$ such that $|\mathbb{G}| = p$ and on the strength of the underlying hash function $\mathcal{H} : \mathbb{G} \times \{0,1\}^* \rightarrow \mathbb{Z}_p^*$.

In the Schnorr construction, a signer uses its private key x to produce a signature σ over (the hash of) a message m by: (*i*) sampling a random element u from $\mathbb{Z}_p^*$; (*ii*) computing the commitment $U := uG$; (*iii*) calculating the digest $c := \mathcal{H}(U, \mathsf{m})$; and finally (*iv*) computing $z := u + cx \pmod p$. The output of the signature process is $\sigma = (z, c) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$. The verifier uses the signer's public key $X := xG$ and the message m to verify if σ is a valid signature. This is done by: (*i*) parsing σ as (z, c); (*ii*) computing $U' := zG - cX$; and then (*iii*) outputting the boolean value of $\mathcal{H}(U', \mathsf{m}) \stackrel{?}{=} c$. The challenge c is typically replaced by U in the signature. In this case, $\sigma := (z, U)$, and the verification process consists in parsing σ as $(z, U) \in \mathbb{Z}_p^* \times \mathbb{G}^*$ and evaluating the boolean value of $zG \stackrel{?}{=} U + cX, c := \mathcal{H}(U, \mathsf{m})$.

SCHNORR BLIND SIGNATURES. The Schnorr blind signature scheme [Schnorr 2001] derives from the plain scheme: essentially, it takes advantage of the latter's linearity to build a three-move protocol, as illustrated in Figure 1. The procedure is as follows: (*i*) the signer S uses the private key x to produce a primary commitment $\overline{U} := uG$ along with a state variable $st_{\mathsf{S}} := (x, u)$, and sends $\overline{U}$ to the user U; (*ii*) U blinds $\overline{U}$ by combining two randomly sampled blinding factors, named $\alpha, \beta \leftarrow\!\!\$\ (\mathbb{Z}_p^*)^2$, respectively with G and X, thus obtaining $U := \overline{U} + \alpha G + \beta X$; (*iii*) U also computes $c := \mathcal{H}(U, \mathsf{m})$ and uses β to calculate $\overline{c} := c + \beta \bmod p$, saving the state variable $st_{\mathsf{U}} := (X, c, U, \alpha, \beta)$ and sending $\overline{c}$ back to S; (*iv*) S uses st_{S} and $\overline{c}$ as inputs to produce $\overline{z} := u + \overline{c}x \bmod p$, and send it back to U; (*v*) lastly, the user produces the signature $\sigma = (z, c, U)$, $z := \overline{z} + \alpha \bmod p$.

As one may notice, the verification process consists in returning the boolean value of $zG \stackrel{?}{=} U + cX,\ c := \mathcal{H}(U, \mathsf{m})$, which is done identically to the original signature:

$$\begin{aligned} zG &= \overline{z}G + \alpha G \Rightarrow \\ &= \overline{U} + \overline{c}X + \alpha G = U - \alpha G - \beta X + cX + \beta X + \alpha G \Rightarrow \\ &= U + cX \end{aligned}$$

The security of the Schnorr blind signature scheme relies on the intractability of the ROS problem, which is detailed in the following section.

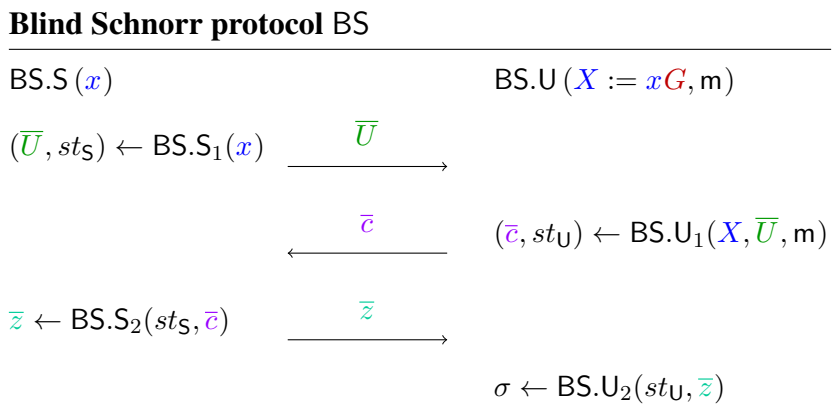

Figure 1. Three-move blind signature protocol derived from Schnorr signatures scheme.

3.2. ROS Problem

The *Random inhomogeneities in a Overdetermined Solvable system of linear equations* problem is broadly studied in [Schnorr 2001]. This is done as part of the security assumptions required to overcome the susceptibility to attacks against the one-more unforgeability property of the Schnorr blind signature scheme. The ℓ–dimension ROS problem is stated as follows[1]:

Definition 1 ([Benhamouda et al. 2021]) *Given a prime p, dimension ℓ and access to a random oracle $\mathsf{H}_{\mathsf{ros}} : \mathbb{Z}_p^\ell \to \mathbb{Z}_p$, find $\ell + 1$ vectors $\mathbf{v}_i, i \in [\ell + 1]$, and one vector $\mathbf{c}$:*

$$\mathsf{H}_{\mathsf{ros}}(\mathbf{v}_i) = \langle \mathbf{c}, \mathbf{v}_i \rangle, \mathbf{v}_i, \mathbf{c} \in \mathbb{Z}_p^\ell, i \in [\ell + 1]$$

The first positive result presenting a solution with sub-exponential time complexity to the ROS problem was due to Wagner [Wagner 2002]. This was a consequence of his work about the *Generalized Birthday Paradox to ℓ dimensions*, defined as:

Definition 2 *Let $i \in [\ell]$ such that $\mathfrak{L}_i$ corresponds to a list of random elements. Find $x_i \in \mathfrak{L}_i,\ i \in [\ell]$ such that $x_1 + \ldots + x_\ell \equiv 0 \bmod p$.*

Wagner's algorithm solved the problem in $O\left(\ell \cdot 2^{\lceil \log p \rceil / (1 + \lfloor \log p \rfloor)}\right)$ time complexity. At the time, it had major consequences to cryptanalysis, especially to schemes using the hardness of the ROS problem as part of their security assumptions. Later, Benhemouda et al. expanded Wagner's approach and proposed a polynomial-time algorithm to solve the ROS problem [Benhamouda et al. 2021], thus proving the insecurity of many Schnorr-based blind signature schemes. One of their main results is stated as follows:

Theorem 1 ([Benhamouda et al. 2021]) *Let $\mathsf{PGen}(1^\lambda)$ be a parameter generation algorithm that, given as input the security parameter λ in unary, outputs an odd prime of (bit) length $\lambda = \lceil \log p \rceil$. If $\ell \geq \lambda$, then there exists an adversary that runs in polynomial time and solves the ROS problem relative to PGen with dimension ℓ.*

[1]Note that, in Definition 1, we use our own notation applied to an alternative formulation of ROS from [Hauck et al. 2019, Fig. 3], where linear functions are used instead.

The idea of the proof is to build an attacker capable of constructing a polynomial whose coefficients depend on the information extracted from ℓ opened sessions. Then, it leverages the recursive nature of the ROS problem to obtain the target values $\mathbf{c}$ and $\mathbf{v}_{\ell+1}$ from Definition 1, since the index i lies in the interval $[\ell + 1]$. It is worth mentioning that an attack combining Wagner's k-list algorithm and Theorem 1 is also addressed in [Benhamouda et al. 2021] when discussing attacks for $\ell < \log p$. We refer the reader to [Benhamouda et al. 2021] for a detailed discussion.

3.3. Barreto and Zanon Blind signature Scheme

Seeking to produce a three-move Schnorr-based blind signature scheme resilient to ROS attacks, Barreto and Zanon [Barreto and Zanon 2023] adapted the scheme by providing a *zero-knowledge proof of knowledge* (*ZKPoK*) of the z component, instead of providing its actual value in the blind signature. More precisely, the scheme uses the z part of a Schnorr signature $\sigma = (z, U)$ as a private key, and then produces a second signature that can be used as a proof. This technique is similar to the one used in [Galindo and Garcia 2009], in the context of identity-based scenarios, and formally proved a *succinct non-interactive argument of knowledge* (SNARK) by Barreto et al. [Barreto et al. 2022]. The resulting blind signature scheme is called BZ, and its three-move protocol is shown in Figure 2.

Blind Schnorr Zero-Knowledge protocol BZ

$\mathsf{BZ.S}(x)$		$\mathsf{BZ.U}(X, \mathsf{m})$
$(\overline{U}, \overline{V}, st_{\mathsf{S}}) \leftarrow \mathsf{BZ.S}_1(x)$	$\xrightarrow{\overline{U}, \overline{V}}$	
	$\xleftarrow{\overline{c}, \overline{d}}$	$(\overline{c}, \overline{d}, st_{\mathsf{U}}) \leftarrow\!\!\$\ \mathsf{BZ.U}_1(X, \overline{U}, \overline{V}, \mathsf{m})$
$\overline{w} \leftarrow \mathsf{BZ.S}_2(st_{\mathsf{S}}, \overline{c}, \overline{d})$	$\xrightarrow{\overline{w}}$	
		$\sigma \leftarrow \mathsf{BZ.U}_2(st_{\mathsf{U}}, \overline{w})$

Figure 2. Three-move blind signature scheme proposed by Barreto and Zanon.

The BZ scheme is built on a secure abelian group $\mathbb{G}$ of prime order $p \approx 2\lambda$ generated by $G = \langle \mathbb{G} \rangle$, and two distinct hash functions defined as $\mathcal{H} : \mathbb{G} \times \{0,1\}^* \mapsto \mathbb{Z}_p^*$ and $\mathcal{G} : \mathbb{G} \mapsto \mathbb{Z}_p^*$. It leverages the signing algorithm of BS to provide a *primary* answer z to a *primary* challenge c and *primary* commitment U. Then, instead of returning z, it uses this value as a private key and returns a proof of knowledge consisting of a *secondary* answer w to a *secondary* challenge d and *secondary* commitment V. It is worth mentioning that the second process is the same as in the primary challenge, but with different values. Figure 3 presents the complete specification of the BZ blind signature scheme. Compared to the original description from [Barreto and Zanon 2023], the only visual difference is that we do not explicitly show z and h, but use their values directly for conciseness.

The security of the BZ scheme is based on the *one-more discrete-logarithm assumption* (OMDL), stated as follows:

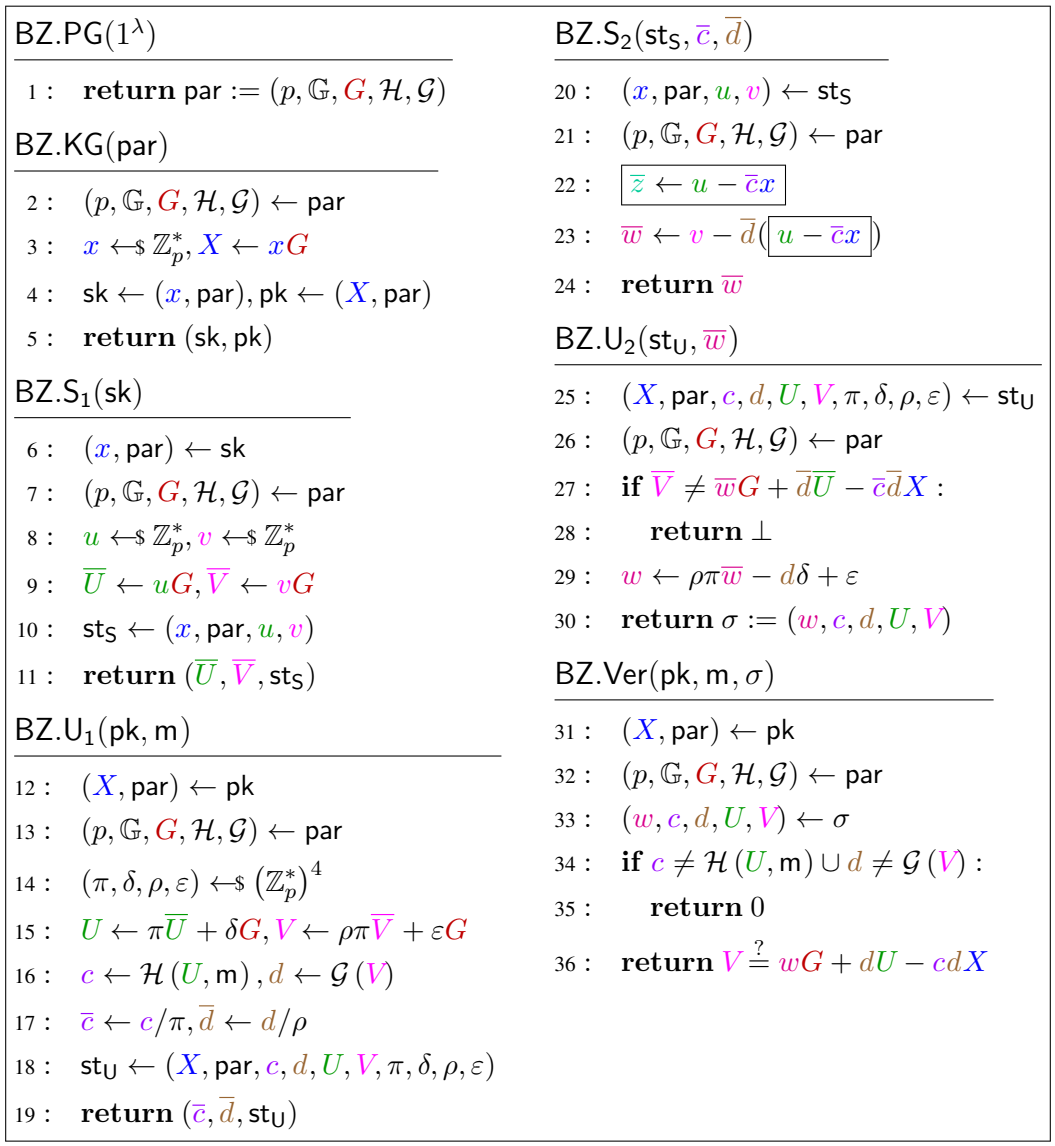

Figure 3. The complete BZ scheme. We use boxes to highlight the definition of $\overline{z}$ in line 22 and its usage in line 23 of BZ.S$_2$ (st$_S, \overline{c}, \overline{d}$).

Definition 3 (One-more Discrete Logarithm Assumption) *Let $\mathbb{G}$ be an abelian group of prime order p generated by G and $\mathcal{A}$ a probabilistic polynomial-time algorithm. Let O_{DL} be an oracle that returns the discrete logarithm of any group element submitted by the adversary $\mathcal{A}$. Upon receiving the challenges X_i, $i \in [\ell]$, it is hard for $\mathcal{A}$ to extract the discrete logarithm of all X_i and the one-more discrete logarithm $x_{\ell+1} := \log_G X_{\ell+1}$.*

The idea behind the security proof presented in [Barreto and Zanon 2023] is to build a reduction that breaks the *impersonation under concurrent attacks* (IMP-CA) of the *one-more unforgeability assumption*. At some point, the attacker must extract the discrete logarithm z, i.e., a zero-knowledge proof of knowledge of the primary signature. The proof appears to have some gaps, though, as noticed by [Chairattana-Apirom et al. 2024]: their main criticism refers to the obtained concurrent security since a Schnorr identification scheme that is IMP-CA secure does not yield a secure blind signature. De-

spite those concerns in [Chairattana-Apirom et al. 2024], to our knowledge no practical attack against BZ has been proposed in the literature. On the other hand, [Barreto and Zanon 2023] does show how to instantiate variants of ROS-based attacks against the scheme, and why they fail.

4. Attacking BZ

In this section, we propose a new ROS-based attack against BZ, and show its correctness. Subsequently, we show a possible fix to the original BZ scheme, producing m–BZ – a *modified* version of the original scheme that uses one (instead of two) hash functions to produce a blind signature. Our attack uses an strategy similar to the one discussed in [Benhamouda et al. 2021], i.e., we make use of information extracted from ℓ concurrent sessions to build a probabilistic polynomial-time adversary $\mathcal{A}$ able to forge the $(\ell+1)$–th blind signature on some chosen message.

ATTACK DESCRIPTION. The attacker $\mathcal{A}$ starts by opening $\ell > \log p$ sessions with the server to receive the commitments $\overline{U}_i$ and $\overline{V}_i$, $i \in [\ell]$. The attacker then randomly samples $\pi_i, \rho_i, \varepsilon_i$ from $\mathbb{Z}_p^*$ and computes V_i as follows:

$$V_i := \rho_i \pi_i \overline{V}_i + \varepsilon_i G \tag{1}$$

The *secondary* challenges are calculated according to the specification: $d_i := \mathcal{G}(V_i)$ and $\overline{d}_i := d_i/\rho_i$. Now, the adversary needs to generate two different values for the primary challenges c_i , as in [Benhamouda et al. 2021]. For that, it has to also use two different U_i, since $c_i = \mathcal{H}(U_i, \mathsf{m})$. Hence, for each bit $b \in \{0,1\}$, $\mathcal{A}$ generates different $\delta_{i,b}$ randomly sampled over $\mathbb{Z}_p^*$ and calculates $U_{i,b}$ as follows:

$$U_{i,b} := \pi_i \overline{U}_i + \delta_{i,b} G \tag{2}$$

$\mathcal{A}$ then computes the *primary* challenges $c_{i,b} := \mathcal{H}(U_{i,b}, \mathsf{m})$ followed by $\overline{c}_{i,b} := c_{i,b}/\pi_i$, as in the original scheme. The adversary defines $\mathbf{b} \in \{0,1\}^\ell$ as the chosen vector of bit challenges to receive the answers $\overline{w}_i$, $i \in [\ell]$, satisfying:

$$\overline{w}_i G + \overline{d}_i \overline{U}_i - \overline{c}_{i,b_i} \overline{d}_i X = \overline{V}_i, \ i \in [\ell]$$

For each $i \in [\ell]$, $\mathcal{A}$ can iterate the equation p_i times to obtain a linear combination of equations, that is:

$$\sum_{1 \le i \le \ell} p_i \left(\overline{w}_i G + \overline{d}_i \overline{U}_i - \overline{c}_{i,b_i} \overline{d}_i X \right) = \sum_{1 \le i \le \ell} p_i \overline{V}_i$$

With a slight abuse of notation, this can be rewritten as the following inner product equation by leveraging the fact that the considered group $\mathbb{G}$ is abelian and written additively:

$$\langle \mathbf{p}, \overline{\mathbf{w}} \rangle G + \langle \mathbf{p}, \overline{d}_i \overline{U}_i \rangle - \langle \mathbf{p}, \overline{c}_{i,b_i} \overline{d}_i \rangle X = \langle \mathbf{p}, \overline{\mathbf{V}} \rangle \tag{3}$$

ATTACK RATIONALE. The essence of the attack consists in manipulating $\langle \mathbf{p}, \overline{c}_{i,b_i}\overline{d}_i \rangle$, i.e., choosing $\mathbf{p}$ in such a way that, for any $y \in \mathbb{Z}_p^*$, the value of $\langle \mathbf{p}, \overline{c}_{i,b_i}\overline{d}_i \rangle$ will be equal to y when evaluated on specific values $(b_1, \ldots, b_\ell) =: \mathbf{b} \in \{0,1\}^\ell$. To succeed, the attacker must find $\mathbf{p} \in \mathbb{Z}_p^\ell$ such that, for all $y \in \mathbb{Z}_p^*$, there is a $\mathbf{x} := \left(\overline{c}_{1,b_1}\overline{d}_1, \ldots, \overline{c}_{\ell,b_\ell}\overline{d}_\ell\right) \in \mathbb{Z}_p^\ell$ satisfying $\langle \mathbf{p}, \mathbf{x} \rangle = y$. $\mathcal{A}$ can find $\mathbf{p}$ by defining the polynomial $\mathcal{P}(\mathbf{x})$ as follows:

$$\mathcal{P}(\mathbf{x}) := \sum_{1 \le i \le \ell} 2^{i-1} \frac{\left(x_i - \overline{c}_{i,0}\overline{d}_i\right)}{\overline{c}_{i,1}\overline{d}_i - \overline{c}_{i,0}\overline{d}_i}$$

Note that $\overline{c}_{i,0} = \overline{c}_{i,1}$ with negligible probability in λ. We can then rewrite this equation in terms of $\langle \mathbf{p}, \mathbf{x} \rangle$ and an additional factor p_0 by separating the numerator of $\mathcal{P}(\mathbf{x})$:

$$\mathcal{P}(\mathbf{x}) = \sum_{1 \le i \le \ell} p_i x_i + p_0 = \langle \mathbf{p}, \mathbf{x} \rangle + p_0 \tag{4}$$

where

$$p_i := \frac{2^{i-1}}{\overline{c}_{i,1}\overline{d}_i - \overline{c}_{i,0}\overline{d}_i}, p_0 := \sum_{1 \le i \le \ell} 2^{i-1} \frac{-\overline{c}_{i,0}\overline{d}_i}{\overline{c}_{i,1}\overline{d}_i - \overline{c}_{i,0}\overline{d}_i} \tag{5}$$

With this construction, $\mathcal{P}(\overline{c}_{1,b_1}\overline{d}_1, \ldots, \overline{c}_{\ell,b_\ell}\overline{d}_\ell) = \sum_{i=1}^{\ell} 2^{i-1} b_i$, i.e., $\mathcal{P}(\overline{c}_{1,b_1}\overline{d}_1, \ldots, \overline{c}_{\ell,b_\ell}\overline{d}_\ell)$ is equal to the binary representation of $\mathbf{b} \in \{0,1\}^\ell$. In order to find $\mathbf{x}$ such that $\langle \mathbf{p}, \mathbf{x} \rangle = y$, the attacker picks $\mathbf{b}$ as the binary representation of $y + p_0$, and sets $\mathbf{x} = (\overline{c}_{1,b_1}\overline{d}_1, \ldots, \overline{c}_{\ell,b_\ell}\overline{d}_\ell)$. This works because:

$$\begin{aligned} \langle \mathbf{p}, \overline{c}_{i,b_i}\overline{d}_i \rangle &= \mathcal{P}(\overline{c}_{1,b_1}\overline{d}_1, \ldots, \overline{c}_{\ell,b_\ell}\overline{d}_\ell) - p_0 \\ &= (y + p_0) - p_0 \\ &= y \end{aligned}$$

After this, $\overline{c}_{i,b_i}$, $\overline{d}_i$, $i \in [\ell]$ and $\mathbf{p}$ are well defined. Now, $\mathcal{A}$ uses Equation 3 to produce the commitments and challenges of the forged signature. For this purpose, $\mathcal{A}$ computes the parameters in the specified order:

$$\begin{cases} V^\star & := \langle \mathbf{p}, \overline{\mathbf{V}} \rangle \\ d^\star & := \mathcal{G}\left(V^\star\right) \\ U^\star & := \dfrac{\langle \mathbf{p}, \overline{d}_i \overline{U}_i \rangle}{d^\star} \\ c^\star & := \mathcal{H}\left(U^\star, \mathsf{m}^\star\right) \end{cases} \tag{6}$$

Finally, the attacker chooses $\mathbf{b}$ as the binary representation of $c^\star d^\star + p_0$ such that $\langle \mathbf{p}, \overline{c}_{i,b_i}\overline{d}_i \rangle = c^\star d^\star$ and sends the challenges accordingly (i.e., sets $\delta_i = \delta_{i,b_i}$, $U_i = U_{i,b_i}$, $c_i = c_{i,b_i}$, $\overline{c}_i = \overline{c}_{i,b_i}$ and sends $\overline{c}_i$, $\overline{d}_i$). After receiving answers $\overline{\mathbf{w}} = (\overline{w}_1, \ldots, \overline{w}_\ell)$, the adversary can output the legitimate signatures:

$$\sigma_i = (\rho_i \pi_i \overline{w}_i - \delta_i d_i + \epsilon, c_i, d_i, U_i, V_i), \quad \text{for } i \in [\ell]$$

Those are known to be valid due to the correctness of the BZ. Finally, $\mathcal{A}$ obtains the forged signature by calculating $w^\star := \langle \mathbf{p}, \overline{\mathbf{w}} \rangle$ and outputting $\sigma^\star = (w^\star, c^\star, d^\star, U^\star, V^\star)$. Its validity can be verified from the definitions listed in (6):

$$\begin{aligned} V^\star &= \langle \mathbf{p}, \overline{\mathbf{V}} \rangle \Rightarrow \\ &= \langle \mathbf{p}, \overline{\mathbf{w}} \rangle G + \langle \mathbf{p}, \overline{d}_i \overline{U}_i \rangle - \langle \mathbf{p}, \overline{c}_{i,b_i} \overline{d}_i \rangle X \overset{(6)}{\Longrightarrow} \\ &= w^\star G + d^\star U^\star - c^\star d^\star X \end{aligned}$$

This is the verification equation for the forged signature $\sigma^\star$, which corresponds to a valid signature on message $\mathsf{m}^\star = \mathsf{m}_{\ell+1}$ $\square$.

EXPERIMENTAL RESULTS. The results of Table 1 show the average time in seconds to break one-more unforgeability of BZ instantiated over different secure elliptic curves [Randall 2023]. Our experimental setup consists of common off-the-shelf equipment (an Intel Core i3-4005U @ 1.70GHz with 4GB of RAM) to simulate the attack using SageMath.

Elliptic Curve	Expected security level	Time (s)
P-224	112	5.4
P-256	128	6.5
P-384	192	16.6
P-521	256	43.3

Table 1. Time measurements of attack on BZ instantiated over different secure elliptic curves.

4.1. Suggested fix: designing the *m*–BZ scheme

One possible approach to neutralize the described ROS attack consists in using a single secure hash function, introducing a dependence between the challenges. In this case, both challenges c and d would be obtained simultaneously, i.e., $(c, d) := \mathcal{H}(U, V, \mathsf{m})$, $\mathcal{H} : \mathbb{G}^2 \times \{0,1\}^* \mapsto \mathbb{Z}_p^*$. Line 16 of algorithm $\mathsf{BZ.U}_1(\mathsf{pk}, \mathsf{m})$ would be replaced by $(c, d) := \mathcal{H}(U, V, \mathsf{m})$ after the primary and secondary commitments are calculated: $U := \pi\overline{U} + \delta G$ and $V := \rho\pi\overline{V} + \varepsilon G$.

This modification renders the described attack impossible because it creates a circularity between $d^\star$, $U^\star$ and $c^\star$ in Equation 6. Consequently, the attacker would be unable to perform the first steps of the attack, which involve calculating a *secondary* challenge $\overline{d}_i$ followed by the generation of two different *primary* challenges $\overline{c}_{i,b_i}$, $b_i \in \{0,1\}$.

5. Attack on *m*–BZ

Although the *m*–BZ construction prevents the attack described in Section 4, unfortunately it is still vulnerable to a second probabilistic polynomial-time attack that breaks its one-more unforgeability. The core of the attack consists in solving the problem of generating blinded challenges c_b with the same d. Essentially, this means that, given a message m and commitments $\overline{U}$ and $\overline{V}$, the attacker $\mathcal{A}$ must generates *only* one $\overline{d}$ and two possible $\overline{c}_b$, $b \in \{0,1\}$. If $\mathcal{A}$ sends the challenges $(\overline{c}_b, \overline{d})$ to the signer and receives an answer $\overline{w}_b$, $\mathcal{A}$ can compute a signature $\sigma_b := (w_b, c_b, d_b, U_b, V_b)$.

ATTACK'S DESCRIPTION. The attack works as follows. After sending the challenges, $\mathcal{A}$ receives an answer $\overline{w}_b$ satisfying the verification equation:

$$\overline{w}_b G + \overline{d}\overline{U} - \overline{c}_b \overline{d} X = \overline{V}$$

This equation is equivalent to:

$$\overline{w}_b G - \overline{c}_b \overline{d} X = \overline{V} - \overline{d}\overline{U} \tag{7}$$

The attacker sets the secondary commitment to $V_b = \overline{V} - \overline{d}\overline{U}$, so the primary commitment U_b can be chosen freely. In particular, it can be set to a known discrete logarithm, i.e., $U_b := u_b G,\ u_b \leftarrow\!\!\$\ \mathbb{Z}_p^*$, with u_b generated by the attacker. $\mathcal{A}$ generates two random nonces u_b and sets $U_b = u_b G$. With the two commitments defined, c_b and d_b are calculated as usual, i.e., $(c_b, d_b) := \mathcal{H}(U_b, V_b, \mathsf{m})$. Then, $\mathcal{A}$ can manipulate Equation 7 using $V_b = \overline{V} - \overline{d}\overline{U}$:

$$\begin{aligned} V_b &= \overline{w}_b G - \overline{c}_b \overline{d} X \Rightarrow \\ &= \overline{w}_b G + (d_b u_b G - d_b u_b G) - \overline{c}_b \overline{d} X \Rightarrow \\ &= (\overline{w}_b - d_b u_b) G + d_b u_b G - \overline{c}_b \overline{d} X \Rightarrow \\ &= \underbrace{(\overline{w}_b - d_b u_b)}_{=: w_b} G + d_b U_b - \overline{c}_b \overline{d} X \end{aligned}$$

Hence, obtaining the equation:

$$V_b = w_b G + d_b U_b - \overline{c}_b \overline{d} X \tag{8}$$

ATTACK RATIONALE. The core of the attack consists in ensuring that Equation 7 holds for signature $\sigma_b := (w_b, c_b, d_b, U_b, V_b)$. For that purpose, $\mathcal{A}$ has to choose $\overline{c}_b$ and $\overline{d}$ such that $\overline{c}_b \overline{d} = c_b d_b$. By a simple substitution of $\overline{c}_b \overline{d} = c_b d_b$ in Equation 8, the term $\overline{c}_b \overline{d} X$ becomes $c_b d_b X$, so σ_b is verified as valid. Due to the independence of $\overline{d}$ with respect to b, $\mathcal{A}$ can set $\overline{d}$ to be a non-zero constant. This means that $\mathcal{A}$ can calculate $\overline{c}_b$ as:

$$\overline{c}_b := \frac{c_b d_b}{\overline{d}} \tag{9}$$

Finally, $\mathcal{A}$ computes $w_b := \overline{w}_b - d_b u_b$, and the valid signature on message m: $\sigma = (w_b, c_b, d_b, U_b, V_b)$.

Building upon this core procedure, the complete attack is as follows: the adversary opens ℓ concurrent sessions with the signer, receiving commitments $\overline{U}_i$ and $\overline{V}_i$ for $i \in [\ell]$. Then, for each session, $\mathcal{A}$ calculates the commits and challenges (blinded or not) as explained above: (*i*) sets the constant value $\overline{d}_i \leftarrow\$ \mathbb{Z}_p^*$; (*ii*) uses Equation 7 to obtain $V_i := \overline{V}_i - \overline{d}_i\overline{U}_i$; (*iii*) makes $U_{i,b} := u_{i,b}G$, $b \in \{0,1\}$, $u_{i,b} \leftarrow\$ \mathbb{Z}_p^*$; (*iv*) computes $(c_{i,b}, d_{i,b}) := \mathcal{H}(U_{i,b}, V_i, \mathsf{m}_i)$; and finally (*v*) uses Equation 9 to calculate $\overline{c}_{i,b}$.

After that, $\mathcal{A}$ computes the coefficients p_i, $i \in [\ell]$, and the constant term p_0 of the linear polynomial $\mathcal{P}$ exactly as in the attack described in Section 4, using Equation 5. $\mathcal{A}$ can then employ the same trick of setting the commit $U^\star$ to a value with a known discrete logarithm: it generates a random $u^\star \leftarrow\$ \mathbb{Z}_p^*$ and calculates the commitments and challenges of the forged signature following Equation 3:

$$\begin{cases} V^\star := \langle \mathbf{p}, \overline{\mathbf{V}} \rangle - \langle \mathbf{p}, \overline{d}_i\overline{U}_i \rangle \\ U^\star := u^\star G \\ (c^\star, d^\star) := \mathcal{H}(U^\star, V^\star, \mathsf{m}^\star) \end{cases} \tag{10}$$

The attacker chooses once again $\mathbf{b} \in \{0,1\}^\ell$ as the binary representation of $c^\star d^\star + p_0$ and sends the challenges accordingly (i.e., sets $u_i = u_{i,b_i}$, $U_i = U_{i,b_i}$, $c_i = c_{i,b_i}$, $d_i = d_{i,b_i}$ $\overline{c}_i = \overline{c}_{i,b_i}$ and sends $\overline{c}_i$, $\overline{d}_i$). After receiving answers $\overline{\mathbf{w}}$, the adversary can output the legitimate signatures:

$$\sigma_i = (\overline{w}_i - d_i u_i, c_i, d_i, U_i, V_i), \quad \text{for } i \in [\ell]$$

All those signatures are valid for the reasons explained at the beginning of this section. Finally, the attacker can produce the forged signature by calculating $w^\star := \langle \mathbf{p}, \overline{\mathbf{w}} \rangle - d^\star u^\star$ and outputting $\sigma^\star = (w^\star, c^\star, d^\star, U^\star, V^\star)$. Its validity can be verified by following the definitions given in (10):

$$\begin{aligned} V^\star &= \langle \mathbf{p}, \overline{\mathbf{V}} \rangle - \langle \mathbf{p}, \overline{d}_i\overline{U}_i \rangle \Rightarrow \\ &= \langle \mathbf{p}, \overline{\mathbf{w}} \rangle G - \langle \mathbf{p}, \overline{c}_{i,b_i}\overline{d}_i \rangle X \Rightarrow \\ &= (w^\star + d^\star u^\star)G - c^\star d^\star X \Rightarrow \\ &= w^\star G + d^\star U^\star - c^\star d^\star X \ \square \end{aligned}$$

EXPERIMENTAL RESULTS. Table 2 shows the average time, in seconds, to break one-more unforgeability of m–BZ blind signature scheme. Our experiments use the same setup and safe elliptic curves as in the practical results of Section 4. We can notice that, while the usage of one hash function in m–BZ hinders our first ROS-like attack, it also allows forgeries to be obtained in this second attack with 25% less time on average.

6. Conclusion

In this paper, we show how to successfully break one-more unforgeability of BZ blind signature scheme. This is accomplished by means of a new ROS-like attack capable of

Elliptic Curve	Expected security level	Time (s)
P-224	112	4.0
P-256	128	4.7
P-384	192	12.4
P-521	256	31.2

Table 2. Time measurements of attack on *m*–BZ instantiated over different secure elliptic curves.

circumventing BZ's the zero-knowledge step, designed to be an obstacle to such types of attacks. Further, we propose modifications in the original BZ, leading to the so-called *m*–BZ scheme: by using one secure hash function instead of two, we can effectively prevent the described attack. Unfortunately, however, *m*–BZ is susceptible to a second attack, as we can built a new probabilistic polynomial-time algorithm to forge a valid signature from ℓ concurrent sessions. Our experimental results show that the one-more unforgeability property can be broken in seconds using common off-the-shelf equipment, even when BZ and *m*–BZ are instantiated with different secure elliptic curves. Both attacks are written in SageMath and uploaded in the following git repository. These results indicate that, like other Schnorr-based strategies, it is hard to build a secure blind signature scheme using BZ's underlying structure.

Acknowledgements

This study was financed by the Brazilian FAPESP (grant 2020/09850-0), CNPq (grant 307732/2023-1), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – (Finance Code 001). We thank the anonymous reviewers of SBSEG'25 for their suggestions and comments.

References

Barreto, P., Jr, M. S., and Zanon, G. (2022). Succinct non-interactive arguments of knowledge from supersingular isogenies. In *Anais do XXII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 181–194, Porto Alegre, RS, Brasil. SBC.

Barreto, P., Reich, D., Jr., M. S., and Zanon, G. (2023). Blind signatures from zero knowledge in the kummer variety. In *Anais do XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 139–152, Porto Alegre, RS, Brasil. SBC.

Barreto, P., Simplicio, M., Ricardini, J., and Patil, H. (2020). Schnorr-based implicit certification: Improving the security and efficiency of vehicular communications. *IEEE Transactions on Computers*, 70(3):393–399.

Barreto, P. and Zanon, G. (2023). Blind signatures from zero-knowledge arguments. Cryptology ePrint Archive, Paper 2023/067.

Bellare, M., Crites, E., Komlo, C., Maller, M., Tessaro, S., and Zhu, C. (2022). Better than advertised security for non-interactive threshold signatures. In *Annual International Cryptology Conference (Crypto'22)*, pages 517–550. Springer.

Bellare, M., Namprempre, C., Pointcheval, D., and Semanko, M. (2003). The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215.

Benhamouda, F., Lepoint, T., Loss, J., Orrù, M., and Raykova, M. (2021). On the (in)security of ros. In *Advances in Cryptology – EUROCRYPT 2021*, pages 33–53, Cham. Springer International Publishing.

Boldyreva, A. (2002). Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public Key Cryptography — PKC 2003*, pages 31–46, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chairattana-Apirom, R., Tessaro, S., and Zhu, C. (2024). Pairing-free blind signatures from cdh assumptions. In *Advances in Cryptology – CRYPTO 2024*, pages 174–209, Cham. Springer Nature Switzerland.

Chaum, D. (1983). Blind signatures for untraceable payments. In *Advances in Cryptology*, pages 199–203, Boston, MA. Springer US.

Coron, J. (2000). On the exact security of full domain hash. In *Advances in Cryptology — CRYPTO 2000*, pages 229–235, Berlin, Heidelberg. Springer Berlin Heidelberg.

Denis, F., Jacobs, F., and Wood, C. (2023). RSA Blind Signatures. RFC 9474.

Fuchsbauer, G. and Wolf, M. (2024). Concurrently secure blind schnorr signatures. In *Advances in Cryptology – EUROCRYPT 2024*, pages 124–160, Cham. Springer Nature Switzerland.

Fujioka, A., Okamoto, T., and Ohta, K. (1993). A practical secret voting scheme for large scale elections. In *Advances in Cryptology — AUSCRYPT '92*, pages 244–251, Berlin, Heidelberg. Springer Berlin Heidelberg.

Galindo, D. and Garcia, F. (2009). A schnorr-like lightweight identity-based signature scheme. In *Progress in Cryptology – AFRICACRYPT 2009*, pages 135–148, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hanzlik, L., Loss, J., and Wagner, B. (2023). Rai-choo! evolving blind signatures to the next level. In *Advances in Cryptology – EUROCRYPT 2023*, pages 753–783, Cham. Springer Nature Switzerland.

Hauck, E., Kiltz, E., and Loss, J. (2019). A modular treatment of blind signatures from identification schemes. In *Advances in Cryptology – EUROCRYPT 2019*, pages 345–375, Cham. Springer International Publishing.

Hauck, E., Kiltz, E., Loss, J., and Nguyen, N. (2020). Lattice-based blind signatures, revisited. In *Advances in Cryptology – CRYPTO 2020*, pages 500–529, Cham. Springer International Publishing.

Katsumata, S., Lai, Y., LeGrow, J., and Qin, L. (2024). Csi-otter: isogeny-based (partially) blind signatures from the class group action with a twist. *Designs, Codes and Cryptography*, 92(11):3587–3643.

Katz, J., Loss, J., and Rosenberg, M. (2021). Boosting the security of blind signature schemes. In *Advances in Cryptology – ASIACRYPT 2021*, pages 468–492, Cham. Springer International Publishing.

Lysyanskaya, A. (2023). Security analysis of rsa-bssa. In *Public-Key Cryptography – PKC 2023*, pages 251–280, Cham. Springer Nature Switzerland.

Qin, X., Cai, C., and Yuen, T. (2021). One-more unforgeability of blind ecdsa. In *Computer Security – ESORICS 2021*, pages 313–331, Cham. Springer International Publishing.

Randall, K. (2023). Nist Special Publication (SP) 800-186, Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters — csrc.nist.gov. `https://csrc.nist.gov/pubs/sp/800/186/final`. [Accessed 05-02-2025].

Schnorr, C. (1990). Efficient identification and signatures for smart cards. In *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 239–252, New York, NY. Springer New York.

Schnorr, C. (2001). Security of blind discrete log signatures against interactive attacks. In *Information and Communications Security*, pages 1–12, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wagner, D. (2002). A generalized birthday problem. In *Advances in Cryptology — CRYPTO 2002*, pages 288–304, Berlin, Heidelberg. Springer Berlin Heidelberg.

Wuille, P., Nick, J., and Ruffing, T. (2020a). Schnorr signatures for secp256k1. Bitcoin improvement proposals.

Wuille, P., Nick, J., and Towns, A. (2020b). Taproot: Segwit version 1 spending rules. Bitcoin improvement proposals.