Efficient Partially Blind Signatures from Isogenies

Dung Hoang Duong[1], Chunpeng Ge[2], Xuan Thanh Khuc[1], and Willy Susilo[1]

[1] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Australia.
{hduong,xtkhuc,wsusilo}@uow.edu.au

[2] Quan Cheng Laboratory, The Joint SDU-NTU Centre for Artificial Intelligence Research(C-FAIR), and School of Software, Shandong University, China.
{chunpeng.ge}@sdu.edu.cn

Abstract. (Partially) blind signatures are foundational cryptographic primitives that enable privacy-preserving authentication in digital systems. Blind signatures allow a user to obtain a signature on a message without revealing its content, while partially blind signatures permit the controlled inclusion of agreed-upon public information into the signed message. These primitives are central to applications such as electronic voting, anonymous credentials, and digital cash.
The first isogeny-based construction of (partially) blind signatures, CSI-Otter, were proposed by Katsumata et al. at CRYPTO'23. However, its concurrent security was later broken by Katsumata et al. (CRYPTO'24) and Do et al. (EUROCRYPT'24). These findings imply that CSI-Otter is secure only in sequential settings. Recently, Hanzlik et al. (ASIACRYPT'25) has proposed a novel framework for concurrently secure blind signatures which can be instantiated from isogenies with smaller signature size (but larger public key and secret key size). It is not known yet whether their construction can be extended to partially blind signatures.
In this paper, we present a new and efficient construction of partially blind signatures based on isogenies with substantially smaller signature and public key sizes than CSI-Otter. This makes our construction the most compact post-quantum partially blind signature scheme known to date. As similar to CSI-Otter, our scheme uses small challenge space resulting in only achieving sequential concurrent security. Our design follows the Abe-Okamoto paradigm in the group action setting, building upon the framework introduced by Tessaro and Zhu (EUROCRYPT'22), whose security is based on the Discrete Logarithm Problem. We rigorously prove the security of our scheme in the Algebraic Group Action Model and the Random Oracle Model, under the hardness assumption of the Group Action Discrete Logarithm Problem.

Keywords: Partially Blind Signature, Abe-Okamoto, Isogenies

1 Introduction

Blind signatures (BS), first introduced by David Chaum in 1983 [17], are a form of digital signature that enables a signer to sign a message without seeing its content. This property is achieved by allowing the message to be *blinded* before being presented for signing and *unblinded* afterward to yield a valid signature. The resulting signature can be publicly verified, just like a regular digital signature, yet the signer remains oblivious to the exact message they signed. This technique plays a central role in privacy-preserving cryptographic protocols, most notably electronic cash (e-cash) systems, where users can withdraw coins from a bank anonymously and later spend them without being traced [17], and electronic voting [18,27].

While blind signatures provide complete unlinkability between the message and the signer, this strong privacy guarantee can be misused, as the signer has no control over the message content being signed. To address this limitation, partially blind signatures (PBS) were introduced by Abe and Fujisaki [1] as a refinement. In a PBS scheme, the message contains a visible, agreed-upon portion of information—denoted info—which is embedded in the final signature and jointly known to both signer and user [1,2]. This visible part may include metadata such as timestamps, election or session identifiers, monetary denominations, or transaction contexts. By combining partial transparency with the privacy of blind signing, PBS achieves a more nuanced balance between user anonymity and signer accountability which makes PBS a powerful tool in many privacy-preserving applications such as electronic voting with ballot binding [2,35], anonymous credential systems [15], e-cash systems [1] and privacy-preserving access control [39].

Since the introduction nearly 30 years ago, there have been very few partially blind signatures proposed. The most efficient and well-known scheme is the one proposed by Abe and Okamoto [2]. The scheme is based on the classical OR-proof technique for obtaining witness indistinguishable protocols and its security proof involved rewinding argument. Unfortunately, there are issues with the proof of one-more unforgeability in [2]. In [32], Kastner et al. revisited the Abe-Okamoto partially blind signature scheme and provided a new comprehensive analysis of its one-more unforgeability which addressed all issues in the original security proof in [2]. In an independent work, Tessaro and Zhu [44] proposed an Abe-Okamoto-like partially blind signature, which is as efficient as Abe-Okamoto scheme, and proved its security in the algebraic group model (AGM) [26].

In terms of post-quantum (partially) blind signatures, there has been active research based on lattices. The most efficient one to date is the blind signature proposed by Beullens et al. [10] with signature size around 22KB. Isogeny-based cryptography is an alternative promising candidate for post-quantum cryptography with the benefit of providing short signature/public key sizes. In [33], Katsumata et al. proposed isogeny-based (partially) blind signatures, called $\mathsf{CSI\text{-}Otter}$, from class group actions. Their scheme is based on sigma protocols of an OR-relation in which the prover is trying to convince the verifier that he owns one of two secret keys corresponding to the public key. Hence they can utilize the proof techniques by Kastner et al. [32] to Abe-Okamoto scheme in their security proof. As a result, their blind signature scheme has public key size 128B and signature size 8KB, which are smallest compared to existing post-quantum blind signatures.

Unfortunately, Katsumata et al. [34] and Do et al. [21] independently presented attacks against the concurrent security of CSI-Otter and Schnorr-type blind signatures constructed from sigma protocols with small challenge space. It suggested that CSI-Otter and similar constructions [36,23] should be used in the sequential setting, i.e., where the adversary must close one session before opening another, we assume the last session is never closed.

Very recently, Hanzlik et al. [28] proposed a novel framework for concurrently secure blind signatures which can be instantiated from isogenies with smaller signature size (but larger public key and secret key size). However, it is not known yet whether their construction can be extended to partially blind signatures (see Section 2.6 for a discussion).We wonder whether we can design an efficient isogeny-based partially blind signatures, even with sequential concurrent security. Nevertheless, sequential concurrent security is still meaningful in many applications [31] (see Appendix B for some illustrated examples).

Our contribution: In this paper, we provide an affirmative answer to the above question. Our contribution can be summarized as follows.

- We first propose a blind signature from class group actions. Our scheme can be considered as a class group action version of the scheme BS_3 in [44] with some differences, since isogenies do not offer as rich algebraic structure. We prove that our scheme is sequentially concurrent secure in the Algebraic Group Action Model (AGAM) [22]. As a result, our scheme achives signature size around 8KB, as similar to CSI-Otter.
- We extend our construction to a partially blind signature with sequentially concurrent security from class group action. We also provide the security proof in in the Algebraic Group Action Model (AGAM) [22]. Notably, our partially blind signature scheme has the same signature size with our signature scheme (i.e., 8KB), and is just 1/3 of that of partially CSI-Otter.
- We also present the possibility to optimize our (partially) blind signatures, as in CSI-Otter, with the expense of increased public key size. As a result, our partially blind signatures have signature size around 4KB.

Due to page limit, we present only our construction of partially blind signature in Section 4, and leave the proofs as well as the construction of blind signature and optimizations to the Appendix. Table 1 provides a comparison between our schemes with existing post-quantum schemes.

2 Technical Overview

We now explain our constributions in detail. We first review the construction of BS_3 in [44], We then present the difficulty when translating the construction to isogeny setting and our solution for constructing the blind signature, and extend it to achieve partially blind signature. We also present the idea of our proof techniques in Algebraic Group Action Model (AGAM).

Scheme	|pk|	|sk|	|sig|
Blind signatures			
BlindOR [6]	34.3 KB	3.19 KB	893 KB
HKLN20 [29]	444 KB	4.28 MB	7.73 MB
LNP22 [37]	1.3 MB	75 KB	150 KB
dPK22 [20]	60 KB	135 KB	102.6 KB
AKSY22 [3]	839 KB	1.6 MB	45.19 KB
CSI-Otter [33]	128 B	16 B	8 KB
Tanuki 1 (CSIDH) [28]	64 B	32 B	8.47 KB
Tanuki 2 (CSIDH) [28]	960 B	480 B	2.54 KB
Tanuki 3 (CSIDH) [28]	3.31 KB	1.66 KB	5.18 KB
Ours BS	**128 B**	**16 B**	**8 KB**
CSI-Otter (ζ_4) [33]	512 B	16 B	4 KB
Ours BS_{ζ_4}	**320 B**	**16 B**	**4 KB**
Partially blind signatures			
dPK22 [20]	60 KB	135 KB	102.6 KB
CSI-Otter [33]	128 B	16 B	24 KB
Ours PBS	**64 B**	**16 B**	**8 KB**
Ours PBS_{ζ_4}	**320 B**	**16 B**	**4 KB**

Table 1: Comparison of public, secret, and signature sizes of post-quantum blind and partially blind signatures.

2.1 Reviewing the blind signature BS_3 in [44]

The blind signature BS_3 in [44] can be seen as an Abe-Okamoto-like blind signature [2]. Let $\mathcal{G}$ be a group of order prime p with generator g and $\mathsf{H} : \{0,1\} \to \mathbb{Z}_p^*$ is a hash function modeled as a random oracle model. The scheme can be described through the interactions between a signer with $(\mathsf{sk}, \mathsf{pk}) = (x, (X := g^x, Z)) \in \mathbb{Z}_p \times \mathcal{G}^2$ and a user with $\mathsf{pk} = (X, Z)$ depicted in Fig. 1.

2.2 Translating into class group action setting

Note that in the blind signature BS_3 in Fig. 1, we are able to compute $C \leftarrow g^t Z^y$ as well as randomize the commitment (A, C) to (A', C') because of the fact that $\mathcal{G}$ is a $\mathbb{Z}_p$-module. However, this does not extend to the class group action from isogenies. In this paper, we consider the CSIDH group action [16] defined as $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ where $\mathcal{G}$ is an ideal class group and $\mathcal{X}$ is the set of elliptic curves. Furthermore, $\mathcal{G}$ is cyclic $\mathcal{G} = \langle \mathbf{g} \rangle$ and can be expressed as $\mathcal{G} \cong \mathbb{Z}_N$ for some N;

$\mathsf{Signer}(\mathsf{sk} = x, \mathsf{pk} = (X, Z))$		$\mathsf{User}(\mathsf{pk} = (X, Z), m)$
$a, t \leftarrow\$\ \mathbb{Z}_p; y \leftarrow\$\ \mathbb{Z}_p^*$		
$A \leftarrow g^a; C \leftarrow g^t Z^y$	$\xrightarrow{(A, C)}$	
		$r_1, r_2 \leftarrow\$\ \mathbb{Z}_p; \gamma_1, \gamma_2 \leftarrow\$\ \mathbb{Z}_p^*$ $A' \leftarrow g^{r_1} A^{\gamma_1/\gamma_2}; C' \leftarrow C^{\gamma_1} g^{r_2}$
	$\xleftarrow{c}$	$c' \leftarrow \mathsf{H}(A', C', m); c \leftarrow c' \cdot \gamma_2$
$s \leftarrow a + c \cdot y \cdot x$	$\xrightarrow{(s, y, t)}$	
		$s' \leftarrow (\gamma_1/\gamma_2) \cdot s + r_1$ $y' \leftarrow \gamma_1 \cdot y; t' \leftarrow \gamma_1 \cdot t + r_2$ If $c' = \mathsf{H}(g^{s'} X^{-c' \cdot y'}, g^{t'} Z^{y'}, m)$ Return $\sigma = (s', t', y', c')$

Fig. 1: The blind signature BS_3 in [44]

we then write the action of $\mathbf{g}^a \in \mathcal{G}$, with $a \in \mathbb{Z}_N$, on an elliptic curve $E \in \mathcal{X}$ as $a \star E$; see Section C.2 for the detail.

Now let us see how we can translate BS_3 into this isogeny setting. First we fix an elliptic curve $E_0 \in \mathcal{X}$, the public key $\mathsf{pk} = (E_1, Z)$ can be computed as $E_1 = x \star E_0$ and $Z = z \star E_0$ for some $x, z \leftarrow\$\ \mathbb{Z}_N$, and the secret key is $\mathsf{sk} = x$; we here forget the z. It is easy to compute the committed value A by $A := a \star E_0$ for some $a \leftarrow\$\ \mathbb{Z}_N$. However, we are unable to compute C as in BS_3 from $t \leftarrow\$\ \mathbb{Z}_N$ and y sampled from the challenge space - normally $\{0, 1\}$[3]. We can compute $t \star Z$ but do not know how to integrate y into Z as Z^y in BS_3. The same also happens when we try to randomize A, C to A', C'.

The solution comes from a special property of isogenies that is more expressive than group action, that is using *quadratic twist*, as in $\mathsf{CSI\text{-}Otter}$ [33]: given any $A = a \star E_0$ for unknown $a \in \mathbb{Z}_N$, we can efficiently compute its quadratic twist $-a \star E_0$, denoted by A^{-1}, without knowing a (see [16,9] and references therein for further details). And hence we will switch to use the challenge space $\{-1, 1\}$ instead of $\{0, 1\}$, as having been used in several work, e.g., [9,33].

Now, in order to compute the committed value C, we sample $t \leftarrow\$\ \mathbb{Z}_N$ and $y \leftarrow\$\ \{-1, 1\}$ and compute C by $C \leftarrow t \star Z^y$, as similar to BS_3. Our BS scheme is hence described as in Fig. 2. Here we use the fact that $\{-1, 1\}$ is a multiplica-

[3] Here, for convenience, we reduce to the case of small challenge space of cardinality 2, e.g., $\{0, 1\}$. For the real scheme, our challenge space is $\{0, 1\}^\lambda$ for a security parameter λ.

tive group and $c^{-1} = c$ for all $c \in \{-1, 1\}$. Hence instead of having γ_1/γ_2 in randomization of A in A', we use $\gamma_1 \cdot \gamma_2$ instead, which is equal to γ_1/γ_2.

$$
\begin{array}{lcl}
\mathsf{Signer}(\mathsf{sk} = x, \mathsf{pk} = (E_1, Z)) & & \mathsf{User}(\mathsf{pk} = (E_1, Z), m) \\
\hline
a, t \leftarrow\!\!\$\; \mathbb{Z}_N; y \leftarrow\!\!\$\; \{-1, 1\} & & \\
A \leftarrow a \star E_0; C \leftarrow t \star Z^y & \xrightarrow{(A, C)} & \\
 & & r_1, r_2 \leftarrow\!\!\$\; \mathbb{Z}_N; \gamma_1, \gamma_2 \leftarrow\!\!\$\; \{-1, 1\} \\
 & & A' \leftarrow r_1 \star A^{\gamma_1 \cdot \gamma_2}; C' \leftarrow r_2 \star C^{\gamma_1} \\
 & \xleftarrow{c} & c' \leftarrow \mathsf{H}(A', C', m); c \leftarrow c' \cdot \gamma_2 \\
s \leftarrow a - c \cdot y \cdot x & \xrightarrow{(s, y, t)} & \\
 & & s' \leftarrow (\gamma_1 \cdot \gamma_2) \cdot s + r_1 \\
 & & y' \leftarrow \gamma_1 \cdot y; t' \leftarrow \gamma_1 \cdot t + r_2 \\
 & & \text{If } c' = \mathsf{H}(s' \star E_1^{c' \cdot y'}, t' \star Z^{y'}, m) \\
 & & \quad \text{Return } \sigma = (s', t', y', c')
\end{array}
$$

Fig. 2: Our isogeny-based blind signature BS

2.3 Proof techniques

Despite the similarity between BS_3 in Fig. 1 and our BS scheme in Fig. 2, we cannot just follow the security proof for BS_3 straight away. Tessaro and Zhu [44] proved the security of BS_3 in the algebraic group model (AGM) [26]. Let us discuss a bit about the differences.

In AGM, an adversary $\mathcal{A}$ is considered to be *algebraic*, i.e., if the adversary is given a list of group elements $(X_1, \ldots, X_n)$ and later returns a group element Y, then $\mathcal{A}$ needs to also give a representation $\mathbf{y} = (y_1, \ldots, y_n)$ of Y in terms of the group elements it has received so far $(X_1, \ldots, X_n)$, i.e., $Y = X_1^{y_1} \cdot X_2^{y_2} \cdots X_n^{y_n}$. In such case then the discrete log $\log_g(Y)$ can be represented by an equation $\log_g(Y) = \sum_{i=1}^{n} y_i \cdot \log_g(X_i)$. Tessaro and Zhu then defined an event E to capture such 'equations' in the event when the adversary $\mathcal{A}$ against the one-more unforgeability of BS_3 returns the outputs. Then we can bound the probability of the event WIN that $\mathcal{A}$ wins by $\Pr[\mathsf{WIN}] = \Pr[\mathsf{WIN} \wedge E] + \Pr[\mathsf{WIN} \wedge (\neg E)]$.

Tessaro and Zhu then bound $\Pr[\mathsf{WIN} \wedge E]$ by reducing to the advantage of solving the *weighted fractional ROS (WFROS)* problem which is upper-bounded by $\frac{(Q_H + Q_{S_1} + 1)(Q_H + 3Q_{S_1} + 1)}{p-1}$ where Q_H is the number of queries $\mathcal{A}$ makes to the random oracle H and Q_{S_1} is the number of queries that $\mathcal{A}$ makes to the signer in the first round. $\Pr[\mathsf{WIN} \wedge (\neg E)]$ is bounded by the advantage of solving the discrete logarithm problem; see [44, Section 5] for the detail.

In our paper, we prove the security of our blind signature scheme BS in the Algebraic Group Action Model (AGAM) [22] which can be seen as an adaptation of the AGM to class group actions (with twists). Due to the group action settings, given input $X_1, \ldots, X_n \in \mathcal{X}$, when an algebraic adversary $\mathcal{A}$ outputs $Y \in \mathcal{X}$, it has to give also a representation for only one of the X_i's, i.e., it gives (z, i, b) for $z \in \mathbb{Z}_N$, $i \in \{1, \ldots, n\}$ and $b \in \{-1, 1\}$, such that $Y = z \star X_i^b$. In the real case when the challenge space is $\{-1, 1\}^\lambda$, the input and output are vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n, \mathbf{Y} \in \mathcal{X}^\lambda$, it is more involved, since each component Y_i, for $i \in \{1, \ldots, \lambda\}$, can be expressed in terms of one of the components of the vectors $\mathbf{X}_i$'s, and hence $\mathbf{Y}$ can not be expressed in terms of one vector $\mathbf{X}_i$ but will involve many components of the vectors $\mathbf{X}_i$'s which require a careful analysis. So in constrast to [44], we do not define a WFROS problem, we will dig deeply in the relation and the possibilities appeared in the event E and provide the upper bound for $\Pr[\mathsf{WIN} \wedge E]$ accordingly, which we follow the techniques from [31,28]. Bounding $\Pr[\mathsf{WIN} \wedge (\neg E)]$ will be done in a similar manner to [44].

2.4 Extending to Partially Blind

The blind signature scheme BS_3 is easily extended to partially blind signature, called PBS_3, in [44] by setting the public component Z in BS_3 to be associated with the info. Concretely, the public key now just consists of $X = g^x$. And when computing the commitment (A, C) the signer needs to first compute the tag Z from the info through a function $\mathsf{F} : \{0,1\}^* \to \mathcal{G}$ modeled as a random oracle. The security proof is proceeded as similar to that of BS_3 but with some care because of the involvement of the public information info. In particular, in estimating the bound $\Pr[\mathsf{WIN} \wedge (\neg E)]$, Tessaro and Zhu needs to rely also on the hardness of the so-called rel-Dlog problem [30].

In our setting, we also do the same as in [44] to extend our blind signature scheme BS to a partially blind signature scheme PBS: our public key now consists of only $E_1 = x \star E_0$. In contrast to [44], we need to employ a function $\mathsf{G} : \{0,1\}^* \to \mathbb{Z}_N$ modeled as a random oracle model, and for a public information info, we compute Z by $Z := \mathsf{G}(y) \star E_0$ for some string y. The security proof is now following that of the blind signature scheme BS. Similar to [44], we need to put some care because of the involvement of public information, i.e., the difference between BS and PBS. In particular, in bounding $\Pr[\mathsf{WIN} \wedge (\neg E)]$, we need to rely on the hardness of the multi-target discrete logarithm group action problem, written as $\mathsf{MT\text{-}GA\text{-}DLOG}$. We note that $\mathsf{MT\text{-}GA\text{-}DLOG}$ has been used in many digital signatures from group actions, e.g., [9,43,12,8].

2.5 Optimization

Normally in group action setting, as in the case of our (partially) blind signature, the underlying protocol has small challenge space $\{-1, 1\}$ or $\{0, 1\}$, and the protocol needs to repeat, say λ, times to attain the required security level. One popular way to extend the challenge space to a bigger set, e.g., $[0, \dots, S]$, as done in many existing schemes [9,43,12,19,11], with the expense of increasing the public key. This cannot be applied to our scheme since here we use the fact that $\{-1, 1\}$ is a multicative subgroup of the ring $\mathbb{Z}_N$, while $[0, \dots, S]$ does not has a multiplicative group structure. Katsumata et al. [33] then proposed a method for optimizing CSI-Otter by using the challenge space $\mathcal{C} = \{\zeta_d^i\}_{j\in[d]}$ where ζ_d is the d-th primitive root of unity over $\mathbb{Z}_N$. We also adapt such techniques to our schemes and use the same parameter $d = 4$ as in [33]. As a result, our (partially) blind signature schemes with ζ_4 has reduced 50% of the signature size. We defer to Section F for the detailed description.

2.6 Discussions

Polynomially Concurrent Security. Our construction is designed for compactness and therefore operates in a small-challenge regime enabled by efficient twists. This regime is known to be fundamentally incompatible with polynomially concurrent one-more security for Schnorr/Σ-style blind-signature paradigms: concurrency enables 'simulation-and-patching' strategies in which an adversary can efficiently simulate one coordinate/instance with constant expected effort, embed it into an otherwise honestly formed transcript, and then patch the signer's response to obtain an extra valid signature. Intuitively, because a single coordinate has only a constant-size challenge (e.g. $\{-1, 1\}$), the adversary can re-hash a small number of times until the global challenge is consistent with its simulated coordinate; once a valid response is obtained for the modified transcript, the adversary replaces the simulated coordinate's response components accordingly, in the same manner with the attack against CSI-Otter in [21]. This explains our focus on sequential one-more unforgeability, which matches many deployment models where the signer enforces 'no interleaving' (one outstanding session per authenticated identity/channel); see Section B for concrete examples.

Tanuki [28] achieves concurrent security by moving away from this small-challenge paradigm. At a high level, its concurrently secure instantiations enlarge the effective challenge space (and typically pay for it in larger public parameters, e.g., additional/expanded public-key components), which prevents constant-cost simulation of one coordinate under concurrency. In contrast, our approach requires the challenge space to retain a 'group structure' compatible with the Abe–Okamoto/Tessaro–Zhu style transformations and the isogeny-with-twist operations used to obtain compact signatures. Adopting the same 'expand challenge space/public parameters' approach is therefore not a drop-in modification for our scheme and would require a separate redesign and analysis.

Partially Blind Signatures from Tanuki. Tanuki [28] provides frameworks for (log- or polynomially) concurrently secure blind signatures from group actions. A natural question is whether one can 'simply add tags' to obtain PBS. We believe this is not immediate across Tanuki's frameworks for two structural reasons. First, PBS requires the agreed public information info to be cryptographically bound into the signed relation (not merely appended as metadata), so that signatures cannot be transferred or malleated across different tags. In Tanuki, however, blindness and one-more security rely on a carefully engineered transcript structure where commitments are blinded/permuted and the Fiat–Shamir challenge is sampled from a balanced distribution (e.g., fixed Hamming weight or fixed index multiplicities). This balancing is essential to avoid linkability through challenge invariants. Binding info in a PBS-correct way typically makes the verification relation depend on info (or introduces an info-derived public element such as $Z := \mathsf{G}(\mathsf{info}) \star R_0$), which changes what is committed/hashed and therefore requires re-establishing the required balance and transcript indistinguishability arguments. Second, Tanuki's concurrent-security proofs—especially in the polynomially concurrent setting—are sensitive to random-oracle programming points and transcript distributions across many interleavings (e.g., via multi-key challenges and, in the strongest framework, doubly blinded session-specific public keys). Making the relation tag-dependent introduces additional, possibly repeating/adaptive inputs, so a correct PBS variant would require reworking the concurrent simulation/extraction argument rather than a syntactic change. We therefore view concurrently secure PBS from group actions in the Tanuki sense as an important and non-immediate extension.

On proving security in the quantum version of AGAM. Duman et al. [22] define a quantum analogue of AGAM (QAGAM). In that model, an adversary may output a group-action element together with a quantum 'representation' state, and the representation can even be entangled with the element. They also show how to incorporate twists by extending the representation with an extra bit. However, lifting our security proof from AGAM+ROM to QAGAM would still require nontrivial changes. First, our reduction currently uses classical representation extraction in a direct way. In QAGAM, checking/using representations is more subtle; Duman et al. note that, w.l.o.g., the oracle/reduction may need to compute the element from the representation (or use semi-classical tests at the cost of error), precisely because representations are quantum objects. Second, a full post-quantum claim would also need to handle the quantum random oracle model (QROM). Our proof uses standard random-oracle programming/consistency arguments that do not automatically lift to superposition queries; in contrast, Duman et al.'s quantum reductions rely on QROM-specific tools such as the (semi-classical) one-way-to-hiding lemma to justify reprogramming. For these reasons, we state provable guarantees for classical adversaries in AGAM+ROM, and leave a full QAGAM+QROM treatment as future work, requiring a refactoring of both the representation layer and the random-oracle programming steps along the lines above.

3 Preliminaries

In this Section, we first define notations used in the paper. We then provide the definition of (partially) blind signatures. Due to page limit, we refer to Appendix C for all the required preliminaries on group actions, isogenies and algebraic group action models.

3.1 Notations

We denote by $\mathbb{N}$ and $\mathbb{Z}$ the set of natural numbers and integers respectively. We define the ring of integers modulo N as $\mathbb{Z}_N$ with representatives in $\left[-\frac{N}{2}, \frac{N}{2}\right) \cap \mathbb{Z}$. For a positive integer n, we denote $[n]$ to be the set $\{1, 2, \ldots, n\}$. We use bold letter for a vector, e.g., $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_N^n$. For a finite set S, we write $s \leftarrow\!\$\ S$ for sampling s at uniformly random over S. We denote by $\odot$ for the component-wise multiplication of vectors in $\mathbb{R}$, e.g., for $c \in \mathbb{R}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, we write $c \odot \mathbf{a}$ for $(c \odot a_1, \ldots, c \odot a_n)$ and $\mathbf{a} \odot \mathbf{b}$ for $(a_1 \odot b_1, \ldots, a_n \odot b_n)$. We also abuse such notation for writing $E^{\mathbf{c}}$ for the vector $(E^{c_1}, \ldots, E^{c_n})$; here E stands for an elliptic curve - see Section C.2 for more details. For a security parameter λ, all probabilistic polynomial-time algorithms in this paper take 1^λ as input. We write $(\mathcal{G}, \mathcal{X}, \star)$ to denote the group action $\star$ of the group $\mathcal{G}$ on the set $\mathcal{X}$ (see Section C for further details).

3.2 (Partially) Blind Signatures

We follow [33,2,32,31,44] to define a partially blind signature. Note that the blind signature is a special case of a partially blind signature in which no binding public information info is involved.

Definition 1 (Partially Blind Signature). *A three-move partially blind signature* PBS *with efficient decidable public key space* $\mathcal{PK}$ *consists of the following* PPT *algorithms.*

$\mathsf{PBS.Setup}(1^\lambda) \to \mathit{par}$*: On input the security parameter* 1^λ*, outputs the public parameter* par*.*

$\mathsf{PBS.KGen}(\mathit{par}) \to (\mathsf{pk}, \mathsf{sk})$*: On input the public parameter* par*, the key generation algorithm outputs a pair of public and secret keys* $(\mathsf{pk}, \mathsf{sk})$*.*

$\mathsf{PBS.S} = (\mathsf{PBS.S}_1, \mathsf{PBS.S}_2)$*: The signer consists of two phases:*

- $\mathsf{PBS.S}_1(\mathsf{sk}, \mathsf{info}) \to (\mathsf{state}_S, \rho_{S,1})$*: On input the secret key* sk *and a tag* info*, it outputs an internal signer state* state_S *and the first-sender message* $\rho_{S,1}$*.*
- $\mathsf{PBS.S}_2(\mathsf{state}_S, \rho_U) \to \rho_{S,2}$*: On input the signer state* state_S *and a user message* ρ_U*, it outputs a second-sender message* $\rho_{S,2}$*.*

$\mathsf{PBS.U} = (\mathsf{PBS.U}_1, \mathsf{PBS.U}_2)$*: The user consists of two phases:*

- $\mathsf{PBS.U}_1(\mathsf{pk}, \mathsf{info}, M, \rho_{S,1}) \to (\mathsf{state}_U, \rho_U)$*: On input the public key* $\mathsf{pk} \in \mathcal{PK}$*, a tag* info*, a message* M *and the first-sender message* $\rho_{S,1}$*, it outputs an internal user state* state_U *and a user message* ρ_U*.*
- $\mathsf{PBS.U}_2(\mathsf{state}_U, \rho_{S,2})$*: On input a user state* state_U *and a second-signer message* $\rho_{S,2}$*, outputs a signature* σ

$\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}, M, \sigma)$*: On input the public key* pk*, a tag* info*, a message* M *and a signature* σ*, it outputs* 1 *to indicate the signature is valid, and* 0 *otherwise.*

If the partially blind signature scheme only accepts a unique tag info, it is referred to as a blind signature (BS), and info is omitted from the scheme's syntax.

A valid partial blind signature must satisfy three essential properties: *correctness*, *partial blindness under chosen keys*, and *one-more unforgeability*, as formally defined below.

Definition 2 (Correctness). *A three-move partial blind signature scheme* PBS *is correct if for all public and secret key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathit{PBS.KeyGen}(\mathit{par})$ *with* $\mathit{par} \leftarrow \mathit{PBS.Setup}(1^\lambda)$*, we have*

$$\Pr\left[\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}, M, \sigma) = 1 \;\middle|\; \begin{array}{c} (\mathsf{state}_S, \rho_{S,1}) \leftarrow \mathsf{PBS.S}_1(\mathsf{sk}, \mathit{info}) \\ (\mathit{state}_U, \rho_U) \leftarrow \mathit{PBS.U}_1(\mathit{pk}, \mathit{info}, M, \rho_{S,1}) \\ \rho_{S,2} \leftarrow \mathit{PBS.S}_2(\mathit{state}_S, \rho_U) \\ \sigma \leftarrow \mathit{PBS.U}_2(\mathit{state}_U, \rho_{S,2}) \end{array}\right] = 1.$$

Definition 3 (Partial Blindness under Chosen Keys). *For a partial blind signature* PBS*, define the following game* $\mathsf{Blind}_{\mathsf{PBS}}$ *with an adversary* $\mathcal{A}$ *(playing the signer) as follows.*

Setup. *The challenger samples a bit* $\mathit{coin} \leftarrow\!\!\$\; \{0,1\}$ *and runs* $\mathcal{A}$ *on input* 1^λ*.*

Online Phase. $\mathcal{A}$ *outputs a tag* info*, two message* M_0^* *and* M_1^**, a public key* $\mathit{pk} \in \mathcal{PK}$*, the game checks if* pk *is valid and if so, it assigns* $(M_0, M_1) = (M^*_{\mathit{coin}}, M^*_{1-\mathit{coin}})$*. If* pk *is not valid, the game aborts and outputs* 0*. The adversary* $\mathcal{A}$ *is given access to oracles* U_1, U_2 *which behave as follows*

- **Oracle** U_1*. On input* $b \in \{0,1\}$ *and a first-signer message* $\rho_{S,1,b}$*, if the session* b *is not yet open, the oracle marks session* b *as* opened *and generates a state and a challenge as* $(\mathit{state}_{U,b}, \rho_{U,b}) \leftarrow\!\!\$\; \mathit{PBS.U}_1(\mathit{pk}, \mathsf{info}, M_b, \rho_{S,1})$*. It returns* $\rho_{U,b}$ *to* $\mathcal{A}$*.*
- **Oracle** U_2*. On input* $b \in \{0,1\}$ *and a second-signer message* $\rho_{S,2,b}$*, if the session* b *is* opened*, the oracle creates a signature* $\sigma_b \leftarrow \mathit{PBS.U}_2(\mathit{state}_{U,b}, \rho_{S,2,b})$*. It marks session* b *as* closed*. Oracle* U_2 *does not output anything.*

Output Determination. *When both sessions are closed and for* $b \in \{0,1\}$ *we have that* $\mathit{PBS.V}(\mathit{pk}, \mathit{info}, M_b, \sigma_b) = 1$*, the oracle returns the two signatures* $(\sigma_{\mathit{coin}}, \sigma_{1-\mathit{coin}})$ *to* $\mathcal{A}$*, where note that* σ_{coin} *(resp.* $\sigma_{1-\mathit{coin}}$*) is a valid signature for* M_0^* *(resp.* M_1^**) regardless of the choice of* coin*.* $\mathcal{A}$ *outputs a guess* coin^* *for* coin*. We say that* $\mathcal{A}$ *wins if* $\mathit{coin}^* = \mathit{coin}$*.*

We say that PBS *is partial blind under chosen keys if the probability that* $\mathcal{A}$ *wins is negligible.*

Definition 4 (Concurrent ℓ-one-more unforgeability (cOMUF)). *Let* $\mathsf{PBS} = (\mathsf{PBS.KGen}, \mathsf{PBS.S}_1, \mathsf{PBS.S}_2, \mathsf{PBS.Verify})$ *be as in Definition 3.2. For a security parameter* 1^λ *and a polynomially bounded* $\ell = \ell(n)$*, define the following game between a challenger and a PPT adversary* $\mathcal{A}$*.*

Setup. *The challenger samples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PBS.KGen}(1^\lambda)$ *and gives* pk *to* $\mathcal{A}$*. It initializes a counter* $\ell_{\mathsf{closed}} \leftarrow 0$*, a table of signer states indexed by session identifiers* $\mathsf{state}_S[\mathsf{sid}]$*, and a flag map* $\mathsf{opened}[\mathsf{sid}] \in \{\mathsf{false}, \mathsf{true}\}$*, initially* false *for all* sid*.*

Signing Oracles (concurrent access). *$\mathcal{A}$ may adaptively and concurrently query:*

- S_1 ***(open session).*** *On input a tag* info *chosen by $\mathcal{A}$, the oracle samples a fresh unused* sid*, sets* $\mathsf{opened}[\mathsf{sid}] \leftarrow \mathsf{true}$*, computes* $(\mathsf{state}_S[\mathsf{sid}], \rho_{S,1}) \leftarrow \mathsf{PBS.S}_1(\mathsf{sk}, \mathsf{info})$*, and returns* $(\mathsf{sid}, \rho_{S,1})$*.*
- S_2 ***(close session).*** *On input* (sid, ρ_U)*, if* $\ell_{\mathsf{closed}} \geq \ell$ *or* $\mathsf{opened}[\mathsf{sid}] = \mathsf{false}$ *then return* $\perp$*. Otherwise set* $\ell_{\mathsf{closed}} \leftarrow \ell_{\mathsf{closed}} + 1$ *and* $\mathsf{opened}[\mathsf{sid}] \leftarrow \mathsf{false}$*, compute* $\rho_{S,2} \leftarrow \mathsf{PBS.S}_2(\mathsf{state}_S[\mathsf{sid}], \rho_U)$*, and return* $\rho_{S,2}$*.*

Output & Win. *Eventually $\mathcal{A}$ outputs* $\{(\mathsf{info}_i, M_i, \sigma_i)\}_{i=1}^k$ *with all pairs* (info_i, M_i) *pairwise distinct. $\mathcal{A}$ wins iff* $k \geq \ell_{\mathsf{closed}} + 1$ *and* $\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}_i, M_i, \sigma_i) = 1$ *for all* $i \in [k]$*.*

We say PBS *is* concurrently ℓ-one-more unforgeable *if* $\Pr[\mathcal{A}$ *wins*$]$ *is negligible in n for all PPT $\mathcal{A}$.*

Definition 5 (Sequential ℓ-one-more unforgeability (sOMUF)). *The* sequential *game is identical to cOMUF except that $\mathcal{A}$ must respect the* no-interleaving *rule:* S_1 *may be queried only when* $\mathsf{opened}[\mathsf{sid}] = \mathsf{false}$ *for all* sid*, i.e., every open session must be closed via* S_2 *before a new session can be started. The winning condition and security notion are as in cOMUF.*

We say PBS *is* sequentially ℓ-one-more unforgeable *if* $\Pr[\mathcal{A}$ *wins*$]$ *is negligible in n for all PPT $\mathcal{A}$ subject to the no-interleaving rule.*

4 Efficient Partially Blind Signature in AGAM

In this section, we present our partially blind signature scheme, called PBS detailed in Fig. 3, that relies on the hardness of GA-DLOG and MT-GA-DLOG problems. The scheme is built from the blind signature scheme BS in Fig. 4 with the difference that the public key now is just $E_1 = x \star E_0$, and the component Z in the public key of BS plays the role of the tag related to info in our PBS; see Section D for the details. Here, we also require a function $\mathsf{G} : \{0,1\}^* \to \mathbb{Z}_N$ that is modeled as a random oracle model. This function is used to map the info into an element in $\mathbb{Z}_N$. Our scheme, illustrated in Figure 3, is described as the following.

- $\mathsf{PBS.Setup}(1^\lambda)$: On input security paramter, it invokes the group generator GrGen [4] to generate a $\mathsf{CEGAT} = (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$ as a public parameter par.
- $\mathsf{PBS.KeyGen}(\mathsf{par})$: On input the parameter $\mathsf{par} = (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$, samples $x \leftarrow\$ \mathbb{Z}_N$ and computes $E_1 := x \star E_0$. It sets $\mathsf{sk} := x$ and $\mathsf{pk} := E_1$ and returns $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{PBS.Sign}_1(\mathsf{sk}, \mathsf{info})$: On input the secret key $\mathsf{sk} = x$, it samples $\mathbf{a}, \mathbf{t} \leftarrow\$ \mathbb{Z}_N^\lambda$, $\mathbf{y} \leftarrow\$ \{-1, 1\}^\lambda$, computes $z = \mathsf{G}(\mathsf{info}), Z := z \star E_0$, $\mathbf{A} := \mathbf{a} \star E_0$, $\mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$, and returns a commitment $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ as the first-sender message and the state $\mathsf{state}_S = (\mathbf{a}, \mathbf{t}, \mathbf{y})$.

[4] Here and what follows, we denote GrGen is a group generator that, given a security parameter λ, generates a $\mathsf{CEGAT} = (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$ as in Section C.2.

- $\mathsf{PBS.Sign}_2(\mathsf{state}_S, \mathbf{c})$: On input the secret key $\mathsf{sk} = x$, the state $\mathsf{state}_{S,1} = (\mathbf{a}, \mathbf{t}, \mathbf{y})$ and a challenge $\mathbf{c} \in \{-1, 1\}^\lambda$, computes $\mathbf{s} = \mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x$ and returns the response $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$ as the second-sender message.[5]
- $\mathsf{PBS.User}_1(\mathsf{pk}, m, \rho_{S,1}, \mathsf{info})$: On input the public key $\mathsf{pk} := (E_1, Z)$, a message m, the information info, and the commitment $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ received from $\mathsf{BS.Sign}_1$, it samples $\gamma_1, \gamma_2 \leftarrow\$ \{-1, 1\}^\lambda$, $\mathbf{r}_1, \mathbf{r}_2 \leftarrow\$ \mathbb{Z}_N^\lambda$, and computes $\mathbf{A}' := \mathbf{r}_1 \star \mathbf{A}^{\gamma_1 \odot \gamma_2}$ and $\mathbf{C}' := \mathbf{r}_2 \star \mathbf{C}^{\gamma_1}$. It then computes $\mathbf{c}' = \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$ and $\mathbf{c} = \mathbf{c}' \odot \gamma_2$. It returns the challenge $\mathbf{c}$ and the state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$.
- $\mathsf{PBS.User}_2(\mathsf{state}_U, \rho_{S,2}, \mathsf{info})$: On input the state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$, info, and a response $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$, it computes $z = \mathsf{G}(\mathsf{info}), Z := z \star E_0$, and checks if $\mathbf{A} = \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$ and $\mathbf{C} = \mathbf{t} \star Z^{\mathbf{y}}$. If yes then it returns a signature $\sigma := (\mathbf{s}', \mathbf{y}', \mathbf{t}', \mathbf{c}')$ where $\mathbf{s}' := \gamma_1 \odot \gamma_2 \odot \mathbf{s} + \mathbf{r}_1$, $\mathbf{y}' := \mathbf{y} \odot \gamma_1$ and $\mathbf{t}' := \gamma_1 \odot \mathbf{t} + \mathbf{r}_2$; otherwise it returns $\perp$.
- $\mathsf{PBS.Ver}(\mathsf{pk}, m, \mathsf{info}, \sigma)$: On input $\mathsf{pk} = (E_1, Z)$, a message m, info, and a signature $\sigma = (\mathbf{s}, \mathbf{y}, \mathbf{t}, \mathbf{c})$, let $z = \mathsf{G}(\mathsf{info}), Z := z \star E_0$, and $\mathbf{c}^* := \mathsf{H}(\mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}, \mathbf{t} \star Z^{\mathbf{y}}, m)$. If $\mathbf{c} = \mathbf{c}^*$, then return 1; otherwise return 0.

We summarize the properties of our PBS scheme in Theorem 1 below.

Theorem 1. *Our PBS in Fig. 3 achieves correctness, perfectly blindness, and sequential one-more unforgeability under the hardness of GA-DLOG and MT-GA-DLOG problems. More concretely, let $\mathcal{A}$ be an algebraic adversary for the game sOMUF such that for each public information info, makes at most Q_{S_1} queries to $\mathsf{PBS.Sign}_1$, and Q_H queries to the random oracle H that start with info. Also let the total number of distinct public information info's queries by $\mathcal{A}$ to $\mathsf{PBS.Sign}_1$ to be bounded by Q_{info}. Then there exist an adversary $\mathcal{B}_1$ for the discrete logarithm GA-DLOG and and adversary $\mathcal{B}_2$ for the MT-GA-DLOG running in similar running time as $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathsf{sOMUF}}_{\mathcal{A},\mathsf{PBS}}(\lambda) \leq \left(1 + \frac{1}{Q_{\mathsf{info}}}\right) \mathbf{Adv}^{\mathsf{GA\text{-}DLOG}}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\mathsf{MT\text{-}GA\text{-}DLOG}}_{\mathcal{B}_2}(\lambda) + \frac{Q_{\mathsf{info}}\left(Q_H + Q_{S_1} + 1 + \binom{Q_H}{2}\right)}{2^\lambda}.$$

Due to the page limit, the full proofs are presented in Section E. The correctness can be easily checked by inspections. For blindness, we follow the proof framework as in that of [33, Theorem 4.2] to prove that for any valid public key pk, tag info, any first and second-signer messages $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ and $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$, and valid signature $\sigma = (\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$ there exists a unique and pair-wise distinct user state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$ that could have generate σ. We here sketch the proof idea of the sequential one-more unforgeability.

Let $\mathcal{A}$ be the adversary described in the Theorem. Without loss of generality, we assume that if $\mathcal{A}$ outputs the public information info^* then $\mathcal{A}$ makes exactly

[5] In fact, we do not need to include $\mathbf{c}$ in $\rho_{S,2}$ but it will be needed in the proof of the blindness, so we include $\mathbf{c}$ for completeness.

$\mathsf{PBS.Setup}(1^\lambda)$

$(\mathbb{Z}_N, \mathcal{X}, \star, E_0) \leftarrow \mathsf{GrGen}(1^\lambda)$

Select $\mathsf{H} : \{0,1\}^* \rightarrow \{-1,1\}^\lambda$

Select $\mathsf{G} : \{0,1\}^* \rightarrow \mathbb{Z}_N$

return $\mathsf{par} := (\mathbb{Z}_N, \mathcal{X}, \star, E_0, \mathsf{H}, \mathsf{G})$

$\mathsf{PBS.KeyGen}(\mathsf{par})$

$(\mathbb{Z}_N, \mathcal{X}, \star, E_0, \mathsf{H}, \mathsf{G}) \leftarrow \mathsf{par}$

$x \leftarrow\!\$\ \mathbb{Z}_N, E_1 := x \star E_0$

return $(\mathsf{sk}, \mathsf{pk}) = (x, E_1)$

$\mathsf{PBS.Sign}_1(\mathsf{sk}, \mathsf{info})$

$\mathbf{a}, \mathbf{t} \leftarrow\!\$\ \mathbb{Z}_N^\lambda, \mathbf{y} \leftarrow\!\$\ \{-1,1\}^\lambda$

$z := \mathsf{G}(\mathsf{info}), Z := z \star E_0$

$\mathbf{A} := \mathbf{a} \star E_0, \mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$

$\mathsf{state}_S := (\mathbf{a}, \mathbf{t}, \mathbf{y})$

$\rho_{S,1} := (\mathbf{A}, \mathbf{C})$

return $(\mathsf{state}_S, \rho_{S,1})$

$\mathsf{PBS.User}_1(\mathsf{pk}, m, \mathsf{info}, \rho_{S,1})$

parse $(\mathbf{A}, \mathbf{C}) \leftarrow \rho_{S,1}$

$\gamma_1, \gamma_2 \leftarrow\!\$\ \{-1,1\}^\lambda$

$\mathbf{r}_1, \mathbf{r}_2 \leftarrow\!\$\ \mathbb{Z}_N^\lambda$

$\mathbf{A}' := \mathbf{r}_1 \star \mathbf{A}^{\gamma_1 \odot \gamma_2}$

$\mathbf{C}' := \mathbf{r}_2 \star \mathbf{C}^{\gamma_1}$

$\mathbf{c}' := \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$

$\mathbf{c} = \mathbf{c}' \odot \gamma_2$

$\mathsf{state}_U \leftarrow (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$

$\rho_U := \mathbf{c}$

return $(\mathsf{state}_U, \rho_U)$

$\mathsf{PBS.Sign}_2(\mathsf{state}_S, \rho_U)$

parse $(\mathbf{a}, \mathbf{t}, \mathbf{y}) \leftarrow \mathsf{state}_S$

parse $\mathbf{c} \leftarrow \rho_U$

$\mathbf{s} := \mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x$

return $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$

$\mathsf{PBS.User}_2(\mathsf{state}_U, \rho_{S,2})$

parse $(\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2) \leftarrow \mathsf{state}_U$

parse $(\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c}) \leftarrow \rho_{S,2}$

$z = \mathsf{G}(\mathsf{info}), Z := z \star E_0$

if $\mathbf{A} \neq \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$ or $\mathbf{C} \neq \mathbf{t} \star Z^{\mathbf{y}}$

  return $\perp$

$\mathbf{s}' := \gamma_1 \odot \gamma_2 \odot \mathbf{s} + \mathbf{r}_1$

$\mathbf{y}' := \gamma_1 \odot \mathbf{y}$

$\mathbf{t}' := \gamma_1 \odot \mathbf{t} + \mathbf{r}_2$

  return $\sigma = (\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$

return $\sigma = \perp$

$\mathsf{PBS.Ver}(\mathsf{pk}, m, \mathsf{info}, \sigma)$

parse $(\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c}) \leftarrow \sigma$

$z = \mathsf{G}(\mathsf{info}), Z := z \star E_0$

$\mathbf{A} := \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$

$\mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$

if $\mathbf{c} = \mathsf{H}(\mathbf{A}, \mathbf{C}, m)$

  return 1

return 0

Fig. 3: The partially blind signature scheme PBS

Q_{S_1} queries to $\mathsf{PBS.Sign}_1$ and Q_1 queries to $\mathsf{PBS.Sign}_2$ that do not return $\perp$ for info^*. Then when $\mathcal{A}$ returns outputs, we know that $l(\mathsf{info}^*) = Q_{S_1}$.

In addition to the original $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game, for each query $(\mathsf{info}, \mathbf{A}, \mathbf{C}, m)$ to H, being algebraic, $\mathcal{A}$ also outputs the representation of $\mathbf{A}$ and $\mathbf{C}$ recorded as α and β respectively. Here $\alpha = (\mathbf{z_A}, \mathbf{e_A}, \mathsf{Mat}_A)$ with $\mathbf{z}_A \in \mathbb{Z}_N^\lambda$, $\mathbf{e_A} \in \{-1,1\}^\lambda$ and $\mathsf{Mat}_\mathbf{A} \in \{\perp, 1\}^{\lambda \times (2l\lambda + Q_{\mathsf{info}})}$, and similar to β. To be more precise, we denote by $\mathbf{A}_1, \mathbf{C}_1, \dots, \mathbf{A}_l, \mathbf{C}_l$ are the answers to l queries of $\mathcal{A}$ to $\mathsf{BS.Sign}_1$ and $Z_1, \dots, Z_{Q_{\mathsf{info}}}$ are the answers to Q_{info} queries of $\mathcal{A}$ to $\mathsf{BS.Sign}_1$. We concatenate them into a column vector $(\mathbf{A}_1 \| \mathbf{C}_1 \| \dots \| \mathbf{A}_l \| \mathbf{C}_l \| Z_1 \| \dots \| Z_{Q_{\mathsf{info}}})^T$ of length $2l\lambda + Q_{\mathsf{info}}$ and denote it by $\begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix}$. We therefore denote by $\mathsf{Mat}_\mathbf{A}$ (resp. $\mathsf{Mat}_\mathbf{C}$) a $\lambda \times (2l\lambda + Q_{\mathsf{info}})$ matrix with exactly one non-zero (e.g., number 1) in each row and $\perp$ otherwise. Hence $\mathsf{Mat}_\mathbf{A} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix}$ will be a vector of length λ whose each component is either $\mathbf{A}_{i,j}$ or $\mathbf{C}_{i,j}$ for $i \in [l]$ and $j \in [\lambda]$ or Z_i for $i \in [Q_{\mathsf{info}}]$. Furthermore, we denote $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ (resp. $\mathsf{Mat}_{\mathbf{A},\mathbf{C}}$, $\mathsf{Mat}_{\mathbf{A},Z}$) to be the $\lambda \times l\lambda$ (resp. $\lambda \times l\lambda$, $\lambda \times Q_{\mathsf{info}}$ matrix obtained from the odd columns (resp. even columns) of $\mathsf{Mat}_\mathbf{A}$, i.e., $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ will correspond to the $\mathbf{A}_{i,j}$ entries in $\begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix}$.

We denote by WIN the event that $\mathcal{A}$ wins the $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game, i.e., $\mathcal{A}$ outputs message-signature pairs $(m_k^*, \sigma_k^*)_{k \in [l+1]}$ are distinct and valid for info^*. Furthermore, we also denote $\mathsf{str}_k^* := \mathsf{info}^* \| \mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} \| \mathbf{t}_k^* \star Z_{\mathsf{Fid}(\mathsf{info}^*)}^{\mathbf{y}_k^*} \| m_k^*$. We note that when the validity of the signature (m_k^*, σ_k^*) is checked, then str_k^* is made to H, and there should be $\alpha_k = (\mathbf{z}_\mathbf{A}^{(k)}, \mathbf{e}_\mathbf{A}^{(k)}, \mathsf{Mat}_\mathbf{A}^{(k)})$ and $\beta_k = (\mathbf{z}_\mathbf{C}^{(k)}, \mathbf{e}_\mathbf{C}^{(k)}, \mathsf{Mat}_\mathbf{C}^{(k)})$ representing $\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*}$ and $\mathbf{t}_k^* \star Z^{\mathbf{y}_k^*}$ respectively in terms of the known commitments $\mathbf{A}_1, \mathbf{C}_1, \dots, \mathbf{A}_l, \mathbf{C}_l, Z_1, \dots, Z_{Q_{\mathsf{info}}}$ i.e.,

$$\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} = \mathbf{z}_\mathbf{A}^{(k)} \star \left(\mathsf{Mat}_\mathbf{A}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix} \right)^{\mathbf{e}_\mathbf{A}^{(k)}}$$

and

$$\mathbf{t}_k^* \star Z_{i_*}^{\mathbf{y}_k^*} = \mathbf{z}_\mathbf{C}^{(k)} \star \left(\mathsf{Mat}_\mathbf{C}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix} \right)^{\mathbf{e}_\mathbf{C}^{(k)}}. \tag{1}$$

Let E be the event in the $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game for which, after the validity of the output is checked, for each $k \in [l+1]$, the following conditions hold.

(i) $\mathbf{c}_{k,v}^* \odot \mathbf{y}_{k,v}^* = \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot \mathbf{e}_{\mathbf{A},v}^{(k)}$ for all $(u,v) \in I_\mathbf{A}^{(k)}$; and

(ii) All the $\mathbf{C}_i$ appearing in Eq. (1) correspond to info^*, i.e., $\mathbf{C}_i \overset{(303)}{=} \mathbf{t}_i \star Z_{i_*}^{\mathbf{y}_i}$ for such $i \in [Q_{S_1} + 1]$ that the corresponding column in $\mathsf{Mat}_\mathbf{C}$ is not $\perp$.

Then

$$\mathbf{Adv}^{\mathsf{sOMUF}}_{\mathcal{A},\mathsf{PBS}}(\lambda) = \Pr[\mathsf{WIN} \wedge E] + \Pr[\mathsf{WIN} \wedge (\neg E)].$$

It is clear that $\Pr[\mathsf{WIN} \wedge E]$ is less than the probability that (i) holds. We then bound that probability by carefully analyze how str^*_k is queried to H in the sequential game manner.

To bound $\Pr[\mathsf{WIN} \wedge (\neg E)]$, we need to partition the event $\mathsf{WIN} \wedge (\neg E)$ into two cases F_1 as the event in the $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}$ game that there exists $k \in [Q_{S_1}+1]$ such that either (i) does not hold, and F_2 as the event that (ii) does not hold. If E does not occur, then either F_1 or F_2 occurs and hence $\mathsf{WIN} \wedge (\neg E) = (\mathsf{WIN} \wedge F_1) \vee (\mathsf{WIN} \wedge F_2)$. We then bound $\Pr[\mathsf{WIN} \wedge F_1]$ by the advantage of solving $\mathsf{GA\text{-}DLOG}$, and $\Pr[\mathsf{WIN} \wedge F_2]$ by the advantage of solving $\mathsf{GA\text{-}DLOG}$ and $\mathsf{MT\text{-}GA\text{-}DLOG}$; see Section E for the details.

Acknowledgement

We thank the reviewers for their careful reading and constructive feedback, which helped us improve the paper. This work is partially supported by the ARC Linkage Project LP220100332. Willy Susilo is supported by the ARC Australian Laureate Fellowship FL230100033. Chunpeng Ge is supported by the National Natural Science Foundation of China (62572285, 62032025, 62172258, U21A20467).

References

1. Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1996.
2. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2000.
3. Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 39–53. ACM, 2022.
4. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439. Springer, 2020.

5. Nabil Alkeilani Alkadri, Rachid El Bansarkhani, and Johannes Buchmann. BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers*, volume 12059 of *Lecture Notes in Computer Science*, pages 484–502. Springer, 2020.
6. Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. Blindor: an efficient lattice-based blind signature scheme from or-proofs. In Mauro Conti, Marc Stevens, and Stephan Krenn, editors, *Cryptology and Network Security - 20th International Conference, CANS 2021, Vienna, Austria, December 13-15, 2021, Proceedings*, volume 13099 of *Lecture Notes in Computer Science*, pages 95–115. Springer, 2021.
7. Ward Beullens. Multivariate blind signatures revisited. In Maria Eichlseder and Sébastien Gambs, editors, *Selected Areas in Cryptography - SAC 2024 - 31st International Conference, Montreal, QC, Canada, August 28-30, 2024, Revised Selected Papers, Part I*, volume 15516 of *Lecture Notes in Computer Science*, pages 230–236. Springer, 2024.
8. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 464–492. Springer, 2020.
9. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology - ASIACRYPT 2019*, volume 11921 of *Lecture Notes in Computer Science*, pages 227–247. Springer, 2019.
10. Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 16–29. ACM, 2023.
11. Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. LESS is more: Code-based signatures without syndromes. In Abderrahmane Nitaj and Amr M. Youssef, editors, *Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings*, volume 12174 of *Lecture Notes in Computer Science*, pages 45–65. Springer, 2020.
12. Markus Bläser, Zhili Chen, Dung Hoang Duong, Antoine Joux, Tuong Ngoc Nguyen, Thomas Plantard, Youming Qiao, Willy Susilo, and Gang Tang. On digital signatures based on group actions: QROM security and ring signatures. In Markku-Juhani O. Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part I*, volume 14771 of *Lecture Notes in Computer Science*, pages 227–261. Springer, 2024.
13. Olivier Blazy, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. A code-based blind signature. In *2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen, Germany, June 25-30, 2017*, pages 2718–2722. IEEE, 2017.

14. Gilles Brassard and Moti Yung. One-way group actions. In *Advances in Cryptology - CRYPTO 1990*, pages 94–107, 1990.
15. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.
16. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, pages 395–427. Springer, 2018.
17. David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, pages 199–203. Plenum Press, New York, 1982.
18. David Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In Christoph G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, volume 330 of *Lecture Notes in Computer Science*, pages 177–182. Springer, 1988.
19. Tung Chou, Ruben Niederhagen, Edoardo Persichetti, Tovohery Hajatiana Randrianarisoa, Krijn Reijnders, Simona Samardjiska, and Monika Trimoska. Take your meds: Digital signatures from matrix code equivalence. In *Progress in Cryptology - AFRICACRYPT 2023*, 2023.
20. Rafaël del Pino and Shuichi Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 306–336. Springer, 2022.
21. Khue Do, Lucjan Hanzlik, and Eugenio Paracucchi. M&M'S: Mix and Match Attacks on Schnorr-Type Blind Signatures with Repetition. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 363–387. Springer, 2024.
22. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 406–435. Springer, 2023.
23. Dung Hoang Duong, Thanh Xuan Khuc, Youming Qiao, Willy Susilo, and Chuanqi Zhang. Blind signatures from cryptographic group actions. *IACR Cryptol. ePrint Arch.*, page 397, 2025.
24. Luca De Feo and Steven D. Galbraith. Seasign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, volume 11478 of *Lecture Notes in Computer Science*, pages 759–789. Springer, 2019.

25. Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77. Springer, 2006.
26. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2018.
27. Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992.
28. Lucjan Hanzlik, Yi-Fu Lai, Marzio Mula, Eugenio Paracucchi, Daniel Slamanig, and Gang Tang. Tanuki: New Frameworks for (Concurrently Secure) Blind Signatures from Post-Quantum Group Actions. In: Hanaoka, G., Yang, BY. (eds) Advances in Cryptology – ASIACRYPT 2025. ASIACRYPT 2025. *Lecture Notes in Computer Science*, vol 16248. Springer, Singapore.
29. Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 500–529. Springer, 2020.
30. Joseph Jaeger and Stefano Tessaro. Expected-time cryptography: Generic techniques and applications to concrete soundness. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 414–443. Springer, 2020.
31. Julia Kastner, Julian Loss, and Jiayu Xu. On Pairing-Free Blind Signature Schemes in the Algebraic Group Model. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part II*, volume 13178 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2022.
32. Julia Kastner, Julian Loss, and Jiayu Xu. The Abe-Okamoto Partially Blind Signature Scheme Revisited. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 279–309. Springer, 2022.
33. Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-Otter: Isogeny-Based (Partially) Blind Signatures from the Class Group Action with a Twist. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part*

III, volume 14083 of *Lecture Notes in Computer Science*, pages 729–761. Springer, 2023.

34. Shuichi Katsumata, Yi-Fu Lai, and Michael Reichle. Breaking parallel ROS: implication for isogeny and lattice-based blind signatures. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part I*, volume 14601 of *Lecture Notes in Computer Science*, pages 319–351. Springer, 2024.
35. Aggelos Kiayias and Moti Yung. Self-tallying elections and perfect ballot secrecy. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*, volume 2274 of *Lecture Notes in Computer Science*, pages 141–158. Springer, 2002.
36. Veronika Kuchta, Jason T. LeGrow, and Edoardo Persichetti. Post-quantum blind signatures from matrix code equivalence. *IACR Cryptol. ePrint Arch.*, page 274, 2025.
37. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Efficient lattice-based blind signatures via gaussian one-time signatures. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part II*, volume 13178 of *Lecture Notes in Computer Science*, pages 498–527. Springer, 2022.
38. Albrecht Petzoldt, Alan Szepieniec, and Mohamed Saied Emam Mohamed. A practical multivariate blind signature scheme. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, volume 10322 of *Lecture Notes in Computer Science*, pages 437–454. Springer, 2017.
39. David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2016.
40. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptol.*, 13(3):361–396, 2000.
41. Markus Rückert. Lattice-based blind signatures. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2010.
42. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.
43. Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim,*

Norway, May 30 - June 3, 2022, Proceedings, Part III, volume 13277 of *Lecture Notes in Computer Science*, pages 582–612. Springer, 2022.
44. Stefano Tessaro and Chenzhi Zhu. Short pairing-free blind signatures with exponential security. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 782–811. Springer, 2022.

A Related Work

Lattice-based cryptography has emerged as a strong candidate to construct many post-quantum cryptographic protocols including blind signatures. The first lattice-based blind signature was proposed by Rücket [41] capturing the classical Schnorr [42] and Okamoto-Shnorr signature [40]. Following Rücket, there have been many lattice-based blind signatures proposed [5,6]. Later Hauck et al. [29] pointed out that there are flaws in the security proof of Rücket's blind signature and those following that frameworking including BLAZE [5]. They then proposed a three-round lattice-based blind signature scheme with signature size around 7.73MB. Notably, Del Pino-Katsumata [20] and Agrawal et al. [3] presented round-optimal lattice-based blind signatures using trapdoor sampling. The former has signature size around 100KB and the latter one has signature size around 50KB. Beullens et al. [10] later constructed a lattice-based blind signature scheme with signature size of around 22KB. Note also that Del Pino and Katsumata [20] extended their technique to obtain partially blind signatures.

There exists a code-based blind signature by Blazy et al. [13] following the Fischlin approach [25] which is based on proofs of knowledge, with signature size around 3.1MB for only 82-bit security if instantiated with Hamming distance metric. Petzoldt et al. proposed in [38] a blind signature based on multivariate polynomials which was recently attacked by Beullens [7].

The first (partially) blind signature from isogenies was proposed, called $\mathsf{CSI\text{-}Otter}$, by Katsumata et al. [33]. Their construction framework was later adapted to construct blind signatures from code equivalence [36] and generic group actions [23]. However, soon later, Katsumata et al. [34] and Do et al. [21] independly provided attacks against the concurrent security of $\mathsf{CSI\text{-}Otter}$, which is also applicable to [36] and [23]. It is hence recommended that existing blind signatures from group actions constructed from the Schnorr-like sigma protocol with small challenge space can be used in the sequential manner only. Recently, Hanzlik et al. [HLM+25] proposed a novel framework for con-currently secure blind signatures which can be instantiated from isogenies with small signature size

B Sequential security in practical deployments

Our scheme achieves sequential one-more unforgeability. This is sufficient in many realistic deployments where the signer can enforce no interleaving, i.e., at most one outstanding signing session per authenticated identity, device, or channel. This assumption is standard in server-side issuance APIs: the issuer keeps session state (or a short-lived issuance token) and simply rejects a new request until the current session completes. In such settings, attacks that rely on mixing transcripts across concurrent sessions are ruled out by the workflow itself, while PBS is still essential because the issuer must bind agreed public metadata into the issued object. We highlight three concrete examples.

1. **E-cash/voucher withdrawal:** A bank issues unlinkable coins/vouchers by signing a user-chosen hidden serial number m (or a commitment to m). The public tag info binds policy attributes that must remain publicly verifiable, e.g., denomination, expiry epoch, and program/policy identifier—so a coin cannot be 'relabelled' across denominations or validity periods. The issuer naturally enforces sequentiality by allowing only one in-flight withdrawal per account/session.
2. **Anonymous access tokens via an authenticated issuance API:** A client obtains a signed token handle m that will later be presented to a gateway/resource server. The tag info binds the claims that the verifier must enforce (service-id/audience, scope, validity window, rate tier, policy version), while the hidden m preserves unlinkability of token usage. Issuers typically enforce sequentiality via per-credential session locking or rate limiting (e.g., one active issuance session per device key/API key).
3. **Ballot/credential authorization workflows:** An authority issues an unlinkable authorization handle m (one-time credential serial or commitment) while the tag info binds public context such as election-id, district/ballot style, eligibility class, and validity window. Issuance is commonly carried out in a controlled session (kiosk/device/authenticated channel), where sessions are naturally serialized per voter/device and interleaving is prevented by design.

C Group actions

C.1 Definitions.

Let $(\mathcal{G}, \circ)$ be a group with identity id and $\mathcal{X}$ a set. An action of $\mathcal{G}$ on $\mathcal{X}$ is a map $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ satisfying the following properties: (i) $\mathrm{id} \star x = x$ for all $x \in \mathcal{X}$ and the identity element $\mathrm{id} \in \mathcal{G}$; and (ii) $g \star (h \star x) = (g \circ h) \star x$ for all $g, h \in \mathcal{G}$ and $x \in \mathcal{X}$. A group action is said to be [4]:

- *transitive* if for all $x, y \in \mathcal{X}$, there exists $g \in \mathcal{G}$ such that $g \star x = y$;
- *faithful* if there does not exist $g \in \mathcal{G} \setminus \{\mathrm{id}\}$ such that $g \star x = x$ for all $x \in \mathcal{X}$, i.e., if $g \star x = x$ for all $x \in \mathcal{X}$ then $g = \mathrm{id}$;
- *free* if whenever there exists $x \in \mathcal{X}$ such that $g * x = x$ then $g = id$; and
- *regular* if it is free and transitive.

In a shorthand, we write $(\mathcal{G}, \mathcal{X}, \star)$ to denote the group action.

Definition 6 (Effective Group Action - EGA). *Let $(\mathcal{G}, \mathcal{X}, \star)$ be a group action satisfying the following properties:*

1. *The group $\mathcal{G}$ is finite and there exist efficient algorithms for membership testing, equality testing, (random) sampling, group operation and inversion.*
2. *The set $\mathcal{X}$ is finite and there exist efficient algorithms for membership testing and to compute a unique representation.*
3. *There exists a distinguished element $x_0 \in \mathcal{X}$ with known representation.*
4. *There exist an efficient algorithm to evaluate the group action, i.e., to compute $g \star x$ given $g \in \mathcal{G}$ and $x \in \mathcal{X}$.*

Then we call $x_0 \in \mathcal{X}$ the origin and $(\mathcal{G}, \mathcal{X}, \star, x_0)$ an effective group action (EGA).

In this paper, we restrict to the case of cyclic group actions, i.e., $\mathcal{G}$ being a cyclic group, to capture the isogeny settings ([16]). In addition, we focus on known-order groups ([4]). We hence will focus on *Cyclic Effective Group Action* (CEGA) defined as the following ([4,22]).

Definition 7 (Cyclic Effective Group Action - CEGA). *Let $(\mathcal{G}, \mathcal{X}, \star, x_0)$ be an effective group action satisfying the following properties:*

1. *The group $\mathcal{G}$ is cyclic of order N for some known $N \in \mathbb{N}$.*
2. *There exists a generator $g \in \mathcal{G}$, i.e. $\mathcal{G} = \langle g \rangle$, with known representation.*
3. *For any element $h \in \mathcal{G}$, the element $a \in \mathbb{Z}_N$ satisfying $h = g^a$ is efficiently computable.*
4. *The group action $\star$ is regular.*

We then say that $(\mathcal{G}, \mathcal{X}, \star, x_0)$ is a cyclic (known-order) effective group action (CEGA). We can therefore denote any CEGA equivalent by $(\mathbb{Z}_N, \mathcal{X}, \star, x_0)$ where

$$\begin{aligned} \star : \mathbb{Z}_N \times \mathcal{X} &\to \mathcal{X} \\ (a, x) &\mapsto \underbrace{(g \circ \cdots \circ g)}_{a \ times} \star x. \end{aligned}$$

For our purpose, we introduce CEGA with twists to capture a property to isogeny-based group actions in CSIDH [16].

Definition 8 (Cyclic Effective Group Action with Twists - CEGAT). *Let $(\mathbb{Z}_N, \mathcal{X}, \star, x_0)$ be a CEGA. We call it a cyclic effective group action with twists (CEGAT) if there is an efficient algorithm that given $x = a \star x_0$, compute $x^t := -a \star x_0$ without knowing $a \in \mathbb{Z}_N$.*

C.2 Isogenies

In this paper, we focus on CSIDH-based group action in [16], briefly defined as the following (following the description in [9,33,22]). Let p be a large prime number. Let E denote the elliptic curve over a finite field $\mathbb{F}_p$ and let $\mathbf{0}_E$ is the point at infinity on E. The curve E is called supersingular if and only if the number of $\mathbb{F}_p$-rational points over $\mathbb{F}_p$ is $p + 1$, i.e., $\#E(\mathbb{F}_p) = p + 1$.

Given two elliptic curves E, E', an isogeny ϕ is a morphism $\phi : E \to E'$, i.e., it is a map given by rational functions and it is also a group homomorphism such that $\phi(\mathbf{0}_E) = \mathbf{0}_{E'}$. The endomorphism $\mathrm{End}(E)$ consists of all isogenies from E to itself and denote by $\mathrm{End}_{\mathbb{F}_p}(E)$ the ring of endomorphisms defined over $\mathbb{F}_p$. For a supersingular curve E over $\mathbb{F}_p$ we have a strict inclusion $\mathrm{End}_{\mathbb{F}_p}(E) \subsetneqq \mathrm{End}(E)$.

Fix an elliptic curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_q$. This curve is supersingular and its $\mathbb{F}_p$-rational endomorphism ring $\mathrm{End}_{\mathbb{F}_p}(E_0)$ is $\mathcal{O} = \mathbb{Z}[\pi]$ where π is the Frobenius endomorphism $(x, y) \mapsto (x^p, y^p)$. Let $\mathcal{E}ll_p(\mathcal{O})$ be the set of elliptic curves defined over $\mathbb{F}_p$ with endomorphism ring $\mathcal{O}$. The ideal class group $\mathcal{G} := cl(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}ll_p(\mathcal{O})$. The class group $cl(\mathcal{O})$ was successfully computed for the parameters of CSIDH-512 in [9]. In particular, the authors also proved that $cl(\mathcal{O})$ is a cyclic group generated by $\mathfrak{g} = (3, \pi - 1)$. In addition, one can identify $cl(\mathcal{O})$ as $\mathbb{Z}_N$ via an efficient computable isomorphism $\varphi : \mathbb{Z}_N \to cl(\mathcal{O})$ mapping a to $\mathfrak{g}^a$. Hence, in what follows, when writing an action of $\mathfrak{g}^a$ on an elliptic curve $E \in \mathcal{E}ll_p(\mathcal{O})$, we use only the exponent a and write $a \star E$ instead.

For any $a \in \mathbb{Z}_N$ we have that $(a \star E)^t = (-a) \star E$. Therefore, the CSIDH-512 group action can be viewed as a CEGAT. From now on in this paper, we will focus only on CSIDH-512 group action, where we identify the group $cl(\mathcal{O})$ as $\mathbb{Z}_N$, denote by $\mathcal{X}$ the set $\mathcal{E}ll_p(\mathcal{O})$. We also use E^{-1} for the quadratic twist of E.

C.3 Group Action Hardness Assumptions

A group action is *one-way*, if for a random x, the function $f_x : \mathcal{G} \to \mathcal{X}$ defined by $f_x(g) := g \star x$ is one-way. The one-way assumption is formulated as the Group Action Inverse Problem (GAIP) ([14]) which asks to compute, given $x, y \in \mathcal{X}$, $g \in \mathcal{G}$ such that $g \star x = y$, assuming that g exists which is always the case if the action $\star$ is transitive. In this paper, we focus to CEGAT and hence GAIP can be formalized as the discrete logarithm problem for group actions ([22]) which we call the *Group Action Discrete Logarithm Problem.*

Definition 9 (Group Action Discrete Logarithm Problem - GA-DLOG). *Let CEGAT $= (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$ be a cyclic effective group action with twitst. We say that an adversary $\mathcal{A}$ solves the group action discrete logarithm problem GA-DLOG if $\mathcal{A}(E_0, a \star E_0) = a$ for $a \leftarrow\!\!\$\, \mathbb{Z}_N$.*

Many digital signatures from group actions (e.g., [9,43,19]) rely on the hardness of random instances of a multi-target version of the GA-DLOG, which is shown to reduce tightly to the normal GA-DLOG by [24] in the case that the class group structure is known, i.e., in case of CEGAT in this paper.

Definition 10 (Multi-target Group Action Discrete Logarithm Problem (MT-GA-DLOG)). *Let CEGAT $= (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$ be a cyclic effective group action with twitst. Given k curves $E_1, \ldots, E_k$ with $End(E_i) = \cdots = End(E_k) = End(E_0)$, find an element $a \in \mathbb{Z}_N$ such that $E_i = a \star E_j$ for some $i, j \in \{0, \ldots, k\}$ with $i \neq j$.*

C.4 Algebraic Group Action Model (AGAM)

The Algebraic Group Action Model (AGAM) is first defined by Duman et al. in [22] which is similar to the Algebraic Group Model (AGM) introduced in [26] in which one replaces the underlying algebraic structure to be an effective group action instead of a prime-order group. Duman et al. [22] also defined a quantum version for AGAM. However, it is not our focus in this paper, we hence do not define such a version.

Definition 11 (Algebraic Group Action Algorithms). *Let* $\mathsf{CEGA} = (\mathbb{Z}_N, \mathcal{X}, \star, x_0)$ *be a fixed cyclic effective group action. An algorithm* $\mathcal{A}$ *is called algebraic if for each output element* $y \in \mathcal{X}$ *it additionally provides a representation relative to a previously received set element. Concretely, if* $(x_1, \ldots, x_l) \in \mathcal{X}^l$ *is the list of received set elements so far,* $\mathcal{A}$ *additonally provides a group element* $a \in \mathbb{Z}_N$ *and an index* $i \in [l]$ *such that* $y = a \star x_i$*. We denote*

$$y_{(i,a)} = a \star x_i.$$

If an oracle is queried on some set elements, then $\mathcal{A}$ *also has to provide a representation for each set element contained in that query.*

Additionally, if the group action is CEGAT*, we extend the representation by a bit* b *indicating whether the base element was twisted before applying the group action, i.e., we denote*

$$y_{(i,a,b)} = \begin{cases} a \star x_i \text{ if } b = 0, \\ a \star x_i^t \text{ if } b = 1. \end{cases} \tag{2}$$

We require that all auxiliary input provided to the adversary which is not in $\mathcal{X}$ *does not depend on elements from* $\mathcal{X}$*.*

For convention in this paper, instead of using the bit $b \in \{0, 1\}$, we use $b \in \{-1, 1\}$ and write $y_{(i,a,b)} = a \star x_{bi}$. Note that for $b = 1$, we have $y_{(i,a,1)} = a \star x_i$ which is the case $b = 0$ in (2). When $b = -1$ then $y_{(i,a,-1)} = a \star x_{-i} = a \star x_i^t$ which is the case $b = 1$ in (2).

D Efficient Blind Signature in AGAM

D.1 Description of the Blind Signature Scheme

In this section, we provide the description of our Blind Signature BS that relies on the hardness of the group action discrete logarithm problem ($\mathsf{GA\text{-}DLOG}$), which is summarized in Fig. 4. The additional elliptic curve Z will allow us to develop a partially blind version of BS, called PBS, which will be discussed in Section . The description of the scheme consists of the following algorithms.

- $\mathsf{BS.Setup}(1^\lambda)$: On input the security paramter, it invokes the group generator GrGen to generate $\mathsf{CEGAT} = (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$ as a public parameter par.
- $\mathsf{BS.KeyGen}(\mathsf{par})$: On input the parameter $\mathsf{par} = (\mathbb{Z}_N, \mathcal{X}, \star, E_0)$, samples $x, z \leftarrow\!\$\ \mathbb{Z}_N$ and computes $E_1 := x \star E_0$, $Z := z \star E_0$. It sets $\mathsf{sk} := x$ and $\mathsf{pk} := (E_1, Z)$ and returns $(\mathsf{sk}, \mathsf{pk})$.

– $\mathsf{BS.Sign}_1(\mathsf{sk})$: On input the secret key $\mathsf{sk} = x$, it samples $\mathbf{a}, \mathbf{t} \leftarrow\$ \mathbb{Z}_N^\lambda$, $\mathbf{y} \leftarrow\$ \{-1,1\}^\lambda$, computes $\mathbf{A} := \mathbf{a} \star E_0$, $\mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$, and returns a commitment $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ as the first-sender message and the state $\mathsf{state}_S = (\mathbf{a}, \mathbf{t}, \mathbf{y})$.
– $\mathsf{BS.Sign}_2(\mathsf{state}_S, \mathbf{c})$: On input the secret key $\mathsf{sk} = x$, the state $\mathsf{state}_{S,1} = (\mathbf{a}, \mathbf{t}, \mathbf{y})$ and a challenge $\mathbf{c} \in \{-1,1\}^\lambda$, computes $\mathbf{s} = \mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x$ and returns the response $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$ as the second-sender message.[6]
– $\mathsf{BS.User}_1(\mathsf{pk}, m, \rho_{S,1})$: On input the public key $\mathsf{pk} := (E_1, Z)$, a message m, and the commitment $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ received from $\mathsf{BS.Sign}_1$, it samples $\gamma_1, \gamma_2 \leftarrow\$ \{-1,1\}^\lambda$, $\mathbf{r}_1, \mathbf{r}_2 \leftarrow\$ \mathbb{Z}_N^\lambda$, and computes $\mathbf{A}' := \mathbf{r}_1 \star \mathbf{A}^{\gamma_1 \odot \gamma_2}$ and $\mathbf{C}' := \mathbf{r}_2 \star \mathbf{C}^{\gamma_1}$. It then computes $\mathbf{c}' = \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$ and $\mathbf{c} = \mathbf{c}' \odot \gamma_2$. It returns the challenge $\mathbf{c}$ and the state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$.
– $\mathsf{BS.User}_2(\mathsf{state}_U, \rho_{S,2})$: On input the state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$ and a response $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$, it checks if $\mathbf{A} = \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$ and $\mathbf{C} = \mathbf{t} \star Z^{\mathbf{y}}$. If yes then it returns a signature $\sigma := (\mathbf{s}', \mathbf{y}', \mathbf{t}', \mathbf{c}')$ where $\mathbf{s}' := \gamma_1 \odot \gamma_2 \odot \mathbf{s} + \mathbf{r}_1$, $\mathbf{y}' := \mathbf{y} \odot \gamma_1$ and $\mathbf{t}' := \gamma_1 \odot \mathbf{t} + \mathbf{r}_2$; otherwise it returns $\perp$.
– $\mathsf{BS.Ver}(\mathsf{pk}, m, \sigma)$: On input the public key $\mathsf{pk} = (E_1, Z)$, a message m and a signature $\sigma = (\mathbf{s}, \mathbf{y}, \mathbf{t}, \mathbf{c})$, let $\mathbf{c}^* := \mathsf{H}(\mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}, \mathbf{t} \star Z^{\mathbf{y}}, m)$. If $\mathbf{c} = \mathbf{c}^*$, then return 1; otherwise return 0.

D.2 Correctness and Blindness

Theorem 2 (Correctness). *If the signer and user follow honestly the protocol in Fig. 4 then the verifier will return* 1 *(in line 705) with probability* 1.

Proof. First we show that $\mathbf{A} = \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$ in line 603. In fact, substituting $\mathbf{s} = \mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x$ (from line 503) gives us

$$\begin{aligned} \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}} &= (\mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x) \star E_1^{\mathbf{c} \odot \mathbf{y}} = \mathbf{a} \odot (-x \odot E_1)^{\mathbf{c} \odot \mathbf{y}} \\ &= \mathbf{a} \odot E_0 = \mathbf{A} \end{aligned} \tag{3}$$

where the last equation follows from line 302.

Now, with signature $(\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$ obtained from line 608, we have

$$\begin{aligned} \mathbf{s}' \star E_1^{\mathbf{c}' \odot \mathbf{y}'} &\overset{(407,605,606)}{=} (\gamma_1 \odot \gamma_2 \odot \mathbf{s} + \mathbf{r}_1) \star E_1^{\mathbf{c} \odot \gamma_2 \gamma_1 \odot \mathbf{y}} \\ &= \mathbf{r}_1 \star (\mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}})^{\gamma_1 \odot \gamma_2} \overset{(3)}{=} \mathbf{r}_1 \star A^{\gamma_1 \odot \gamma_2} \overset{(404)}{=} A' \end{aligned}$$

and

$$\begin{aligned} \mathbf{t}' \star Z^{\mathbf{y}'} &\overset{(606,607)}{=} (\gamma_1 \odot \mathbf{t} + \mathbf{r}_2) \star Z^{\gamma_1 \odot \mathbf{y}} = \mathbf{r}_2 \star (\mathbf{t} \star Z^{\mathbf{y}})^{\gamma_1} \\ &\overset{(302)}{=} \mathbf{r}_2 \star C_1^{\gamma} \overset{(405)}{=} C'. \end{aligned}$$

This implies (from line 406) that $\mathbf{c}' = \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$ and the verifier will always output 1. □

[6] In fact, we do not need to include $\mathbf{c}$ in $\rho_{S,2}$ but it will be needed in the proof of the blindness, so we include $\mathbf{c}$ for completeness.

$\mathsf{BS.Setup}(1^\lambda)$

$(\mathbb{Z}_N, \mathcal{X}, \star, E_0) \leftarrow \mathsf{GrGen}(1^\lambda)$
Select $\mathsf{H} : \{0,1\}^* \rightarrow \{-1,1\}^\lambda$
$\textbf{return } \mathsf{par} := (\mathbb{Z}_N, \mathcal{X}, \star, E_0, \mathsf{H})$

$\mathsf{BS.KeyGen}(\mathsf{par})$

$(\mathbb{Z}_N, \mathcal{X}, \star, E_0, \mathsf{H}) \leftarrow \mathsf{par}$
$x \leftarrow\!\!\$\; \mathbb{Z}_N, E_1 := x \star E_0$
$z \leftarrow\!\!\$\; \mathbb{Z}_N, Z := z \star E_0$
$\textbf{return } (\mathsf{sk}, \mathsf{pk}) = (x, (E_1, Z))$

$\mathsf{BS.Sign}_1(\mathsf{sk})$

$\mathbf{a}, \mathbf{t} \leftarrow\!\!\$\; \mathbb{Z}_N^\lambda, \mathbf{y} \leftarrow\!\!\$\; \{-1,1\}^\lambda$
$\mathbf{A} := \mathbf{a} \star E_0, \mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$
$\mathsf{state}_S := (\mathbf{a}, \mathbf{t}, \mathbf{y})$
$\rho_{S,1} := (\mathbf{A}, \mathbf{C})$
$\textbf{return } (\mathsf{state}_S, \rho_{S,1})$

$\mathsf{BS.User}_1(\mathsf{pk}, m, \rho_{S,1})$

$\textbf{parse } (\mathbf{A}, \mathbf{C}) \leftarrow \rho_{S,1}$
$\gamma_1, \gamma_2 \leftarrow\!\!\$\; \{-1,1\}^\lambda$
$\mathbf{r}_1, \mathbf{r}_2 \leftarrow\!\!\$\; \mathbb{Z}_N^\lambda$
$\mathbf{A}' := \mathbf{r}_1 \star \mathbf{A}^{\gamma_1 \odot \gamma_2}$
$\mathbf{C}' := \mathbf{r}_2 \star \mathbf{C}^{\gamma_1}$
$\mathbf{c}' := \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$
$\mathbf{c} = \mathbf{c}' \odot \gamma_2$
$\mathsf{state}_U \leftarrow (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$
$\rho_U := \mathbf{c}$
$\textbf{return } (\mathsf{state}_U, \rho_U)$

$\mathsf{BS.Sign}_2(\mathsf{state}_S, \rho_U)$

$\textbf{parse } (\mathbf{a}, \mathbf{t}, \mathbf{y}) \leftarrow \mathsf{state}_S$
$\textbf{parse } \mathbf{c} \leftarrow \rho_U$
$\mathbf{s} := \mathbf{a} - \mathbf{c} \odot \mathbf{y} \odot x$
$\textbf{return } \rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$

$\mathsf{BS.User}_2(\mathsf{state}_U, \rho_{S,2})$

$\textbf{parse } (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2) \leftarrow \mathsf{state}_U$
$\textbf{parse } (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c}) \leftarrow \rho_{S,2}$
$\textbf{if } \mathbf{A} \neq \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}} \text{ or } \mathbf{C} \neq \mathbf{t} \star Z^{\mathbf{y}}$
$\quad \textbf{return } \perp$
$\mathbf{s}' := \gamma_1 \odot \gamma_2 \odot \mathbf{s} + \mathbf{r}_1$
$\mathbf{y}' := \gamma_1 \odot \mathbf{y}$
$\mathbf{t}' := \gamma_1 \odot \mathbf{t} + \mathbf{r}_2$
$\quad \textbf{return } \sigma = (\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$
$\textbf{return } \sigma = \perp$

$\mathsf{BS.Ver}(\mathsf{pk}, m, \sigma)$

$\textbf{parse } (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c}) \leftarrow \sigma$
$\mathbf{A} := \mathbf{s} \star E_1^{\mathbf{c} \odot \mathbf{y}}$
$\mathbf{C} := \mathbf{t} \star Z^{\mathbf{y}}$
$\textbf{if } \mathbf{c} = \mathsf{H}(\mathbf{A}, \mathbf{C}, m)$
$\quad \textbf{return } 1$
$\textbf{return } 0$

Fig. 4: The blind signature scheme BS

Theorem 3 (Blindness). *The blind signature in Fig. 4 achieves perfect blindness.*

Proof. We follow the proof framework as in that of [33, Theorem 4.2]. It suffices to prove that for any valid public key pk, any first and second-signer message $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$ and $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$, and valid signature $\sigma = (\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$, there exists a unique and pair-wise distinct user state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$ that could have generated σ. That means that we will prove that fixing an arbitrary $(\mathsf{pk}, \rho_{S,1}, \rho_{S,2})$, there exists a bijection between a valid signature σ and state_U. Note that any public key $\mathsf{pk} = (E_1, Z)$ output by the adversary $\mathcal{A}$ can be efficiently checked to be valid supersingular elliptic curves.

Let's fix $\mathsf{sk} = x, \mathsf{pk} = (E_1, Z)$, $\rho_{S,1} = (\mathbf{A}, \mathbf{C})$, $\rho_{S,2} = (\mathbf{s}, \mathbf{t}, \mathbf{y}, \mathbf{c})$, and a valid signature $\sigma = (\mathbf{s}', \mathbf{t}', \mathbf{y}', \mathbf{c}')$. We define the user state $\mathsf{state}_U = (\mathbf{r}_1, \mathbf{r}_2, \gamma_1, \gamma_2)$ by $\gamma_1 = \mathbf{y}' \odot \mathbf{y}$, $\mathbf{r}_2 = \mathbf{t}' - \gamma_1 \odot \mathbf{t}$, $\gamma_2 = \mathbf{c} \odot \mathbf{c}'$ and $\mathbf{r}_1 = \mathbf{s}' - \gamma_1 \odot \gamma_2 \odot \mathbf{s}$. Now we have

$$\begin{aligned}\mathbf{A}' &= \mathbf{r}_1 \star \mathbf{A}^{\gamma_1 \odot \gamma_2} = (\mathbf{s}' - \gamma_1 \odot \gamma_2 \odot \mathbf{s}) \star \mathbf{A}^{\gamma_1 \odot \gamma_2} \\ &\overset{(603)}{=} \mathbf{s}' \star (E_1^{\mathbf{c} \odot \mathbf{y}})^{\gamma_1 \odot \gamma_2} = \mathbf{s}' \star E_1^{\mathbf{c} \odot \mathbf{y} \odot \gamma_1 \odot \gamma_2} \\ &= \mathbf{s}' \star E_1^{\mathbf{c}' \odot \mathbf{y}'}\end{aligned}$$

and

$$\begin{aligned}\mathbf{C}' &= \mathbf{r}_2 \star \mathbf{C}^{\gamma_1} = (\mathbf{t}' - \gamma_1 \odot \mathbf{t}) \star C^{\gamma_1} \\ &\overset{(603)}{=} \mathbf{t}' \star Z^{\mathbf{y} \odot \gamma_1} = \mathbf{t}' \star Z^{\mathbf{y}'}.\end{aligned}$$

Hence $\mathbf{c}' = \mathsf{H}(\mathbf{s}' \star E_1^{\mathbf{c}' \odot \mathbf{y}'}, \mathbf{t}' \star Z^{\mathbf{y}'}, m) = \mathsf{H}(\mathbf{A}', \mathbf{C}', m)$, which implies that state_U is indeed a user state that results in the valid signature σ. Moreover, for any choice of $\rho_{S,2}$ and any $\sigma \neq \sigma'$, it can be easily checked that the corresponding user states state_U and state'_U defined as above are distinct. Hence there is a bijection between a valid signature and a user state. This completes the proof. □

D.3 One-more Unforgeability

Theorem 4 (One-more Unforgeability). *For any algebraic adversary $\mathcal{A}$ for the game sOMUF making at most Q_{S_1} queries to $\mathsf{BS.Sign}_1$ and Q_H queries to the random oracle H, there exists an algorithm $\mathcal{B}$ for the $\mathsf{GA\text{-}DLOG}$ problem running in a similar running time as $\mathcal{A}$ such that*

$$\mathbf{Adv}_{\mathcal{A},\mathsf{BS}}^{\mathsf{sOMUF}}(\lambda) \leq 2\,\mathbf{Adv}_{\mathcal{B}}^{\mathsf{GA\text{-}DLOG}}(\lambda) + \frac{(Q_{S_1}+1)(Q_{S_1}+Q_H+1)}{2^{\lambda}}.$$

Proof. The flow of the proof is similar to that of [44, Theorem 5], but we do not reduce to any ROS problem, instead we will evaluate directly the probability. As similar to [44, Theorem 5], we fix an adversary $\mathcal{A}$ that makes at most Q_{S_1} queries to $\mathsf{BS.Sign}_1$ and Q_H queries to the random oracle H. Without loss of generality, we assume that $\mathcal{A}$ makes exactly Q_{S_1} queries to $\mathsf{BS.Sign}_1$ and exactly one query $(i, \mathbf{c}_1)$ to $\mathsf{BS.Sign}_2$ for each $i \in [Q_{S_1}]$. Then after $\mathcal{A}$ returns we know that $l = Q_{S_1}$.

In addition to the original $\mathsf{sOMUF}^{\mathsf{BS}}_{\mathcal{A}}$ game, for each query $(\mathbf{A}, \mathbf{C}, m)$ to H, since $\mathcal{A}$ is algebraic, it also outputs the representation of $\mathbf{A}$ and $\mathbf{C}$ recorded as α and β respectively. Here $\alpha = (\mathbf{z_A}, \mathbf{e_A}, \mathsf{Mat}_A)$ with $\mathbf{z}_A \in \mathbb{Z}_N^\lambda$, $\mathbf{e_A} \in \{-1,1\}^\lambda$ and $\mathsf{Mat}_\mathbf{A} \in \{0,1\}^{\lambda \times 2l\lambda}$, and similar to β. To be more precise, we denote by $\mathbf{A}_1, \mathbf{C}_1, \ldots, \mathbf{A}_l, \mathbf{C}_l$ are the answers to l queries of $\mathcal{A}$ to $\mathsf{BS.Sign}_1$. We arrange them concatenately into a column vector $(\mathbf{A}_1 \| \mathbf{C}_1 \| \ldots \| \mathbf{A}_l \| \mathbf{C}_l)^T$ of length $2l\lambda$ and denote it by $\binom{\mathbf{A}}{\mathbf{C}}$. We also denote by $\mathsf{Mat}_\mathbf{A}$ (resp. $\mathsf{Mat}_\mathbf{C}$) a $\lambda \times 2l\lambda$ matrix with exactly one non-zero (e.g., number 1) in each row and $\perp$ otherwise. Hence $\mathsf{Mat}_\mathbf{A} \cdot \binom{\mathbf{A}}{\mathbf{C}}$ will be a vector of length λ whose each component is either $\mathbf{A}_{i,j}$ or $\mathbf{C}_{i,j}$ for $i \in [l]$ and $j \in [\lambda]$. Furthermore, we denote $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ (resp. $\mathsf{Mat}_{\mathbf{A},\mathbf{C}}$) to be the $\lambda \times l\lambda$ matrix obtained from the odd block-of-size-λ-columns (resp. even block-of-size-λ-columns) of $\mathsf{Mat}_\mathbf{A}$, i.e., $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ will correspond to the $\mathbf{A}_{i,j}$ entries in $\binom{\mathbf{A}}{\mathbf{C}}$. For convenience, we also include in α (resp. β) a set $I_\mathbf{A}$ (resp. $I_\mathbf{C}$) of pairs (i,j), with $i \in [l], j \in [\lambda]$, of size $|I_\mathbf{A}| = \lambda$ that corresponding to $\mathsf{Mat}_\mathbf{A}$ (resp. $\mathsf{Mat}_\mathbf{C}$) indexing which $\mathbf{A}_{i,j}$ or $\mathbf{C}_{i,j}$ appears in the representation.

We denote by WIN the event that $\mathcal{A}$ wins the $\mathsf{sOMUF}_\mathcal{A}$ game, i.e., $\mathcal{A}$ outputs message-signature pairs $(m_k^*, \sigma_k^*)_{k \in [l+1]}$ are distinct and valid. Furthermore, we also denote $\mathsf{str}_k^* := \mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} \| \mathbf{t}_k^* \star Z^{\mathbf{y}_k^*} \| m_k^*$. We note that when the validity of the signature (m_k^*, σ_k^*) is checked, then str_k^* is made to H, and there should be $\alpha_k = (\mathbf{z}_\mathbf{A}^{(k)}, \mathbf{e}_\mathbf{A}^{(k)}, \mathsf{Mat}_\mathbf{A}^{(k)}, I_\mathbf{A}^{(k)})$ and $\beta_k = (\mathbf{z}_\mathbf{C}^{(k)}, \mathbf{e}_\mathbf{C}^{(k)}, \mathsf{Mat}_\mathbf{C}^{(k)}, I_\mathbf{C}^{(k)})$ representing $\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*}$ and $\mathbf{t}_k^* \star Z^{\mathbf{y}_k^*}$ respectively in terms of the known commitments $\mathbf{A}_1, \mathbf{C}_1, \ldots, \mathbf{A}_l, \mathbf{C}_l$, i.e.,

$$\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} = \mathbf{z}_\mathbf{A}^{(k)} \star \left(\mathsf{Mat}_\mathbf{A}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \right)^{\mathbf{e}_\mathbf{A}^{(k)}}$$

and

$$\mathbf{t}_k^* \star Z^{\mathbf{y}_k^*} = \mathbf{z}_\mathbf{C}^{(k)} \star \left(\mathsf{Mat}_\mathbf{C}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix} \right)^{\mathbf{e}_\mathbf{C}^{(k)}}.$$

Let E be the event in the $\mathsf{sOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ game for which, after the validity of the output is checked, for each $k \in [l+1]$, the following conditions hold.

$$\mathbf{c}_{k,v}^* \odot \mathbf{y}_{k,v}^* = \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot \mathbf{e}_{\mathbf{A},v}^{(k)} \ \forall (u,v) \in I_\mathbf{A}^{(k)} \tag{4}$$

$$\mathsf{Mat}_{\mathbf{C},\mathbf{A}}^{(k)} = 0 \tag{5}$$

Then

$$\mathbf{Adv}^{\mathsf{sOMUF}}_{\mathcal{A}}(\lambda) = \Pr[\mathsf{WIN} \wedge E] + \Pr[\mathsf{WIN} \wedge (\neg E)].$$

We provide upper bounds for the above two probabilities in the following lemmas. The Theorem 4 hence follows.

Lemma 1.

$$\Pr[\mathit{WIN} \wedge E] \leq \frac{Q_H + Q_{S_1} + 1 + \binom{Q_H}{2}}{2^\lambda}.$$

Lemma 2. *There exists an algorithm $\mathcal{B}$ for the GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathit{WIN} \wedge (\neg E)] \leq 2\,\mathbf{Adv}_{\mathcal{B}}^{\mathit{GA\text{-}DLOG}}(\lambda).$$

□

Proof of Lemma 1

Proof. It is clear that $\Pr[\mathsf{WIN} \wedge E]$ is less than the probability that (4) holds which we will prove to be negligible. We follow the proof techniques as in [28,31] to consider the following three cases.

1. There exists $k \in [l+1]$ such that str_k^* has never been queried to H by $\mathcal{A}$. In this case $\mathbf{c}_{k,v}^*$ is uniformly in $\{-1,1\}$ for each $(u,v) \in I_{\mathbf{A}}$. Hence (3) holds for each k with probability $\frac{1}{2^\lambda}$. By the union bound, the probability is bounded above by $\frac{Q_{S_1}+1}{2^\lambda}$.
2. There exists $k \in [l+1]$ such that str_k^* is queried to H during the gap when no sessions are open, say before the start of the j-th session. In this case, the corresponding $\mathbf{c}_{k,v}^*$ values have been determined for all $k \in [j-1], v \in [\lambda]$. Therefore the probability that such a query resulting in a forgery such that (3) holds is less than $\frac{1}{2^\lambda}$. It is also the same if the query is made after the last session. By the union bound, the probability is bounded above by $\frac{Q_H}{2^\lambda}$.
3. There exists $k \in [l+1]$ such that str_k^* is made after the beginning of a session, say j-th, and before its closure, i.e., during a session. since there are l sessions and $l+1$ queries, there should be another query also being made during the j-th session. Note that the values $\mathbf{c}_{k,v}^*$ have been determined for $k \in [j-1], v \in [\lambda]$. It follows that such 2 queries result in 2^λ vectors over $\{-1,1\}^{2\lambda}$ came from the undetermined $\mathbf{c}_{j,i}^*$ for $i \in [\lambda]$. Hence (3) holds with probability less than $\frac{2^\lambda}{2^{2\lambda}}$. By the union bound, the probability is bounded above by $\frac{\binom{Q_H}{2}}{2^\lambda}$

As a summary, the probability that (4) holds, and hence $\mathsf{WIN} \wedge E$ holds, is bounded by

$$\frac{Q_H + Q_{S_1} + 1 + \binom{Q_H}{2}}{2^\lambda}$$

which completes the proof. □

Proof of Lemma 2

Proof. We first partition the event $\mathsf{WIN} \wedge (\neg E)$ into two cases F_1 as the event in the $\mathsf{sOMUF}_{\mathcal{A}}$ game that there exists $k \in [Q_{S_1}+1]$ such that (4) does not hold, and F_2 as the event that there exists $k \in [Q_{S_1}+1]$ such that (5) does not hold. If E does not occur, then either F_1 or F_2 occurs. Therefore $\mathsf{WIN} \wedge (\neg E) = (\mathsf{WIN} \wedge F_1) \vee (\mathsf{WIN} \wedge F_2)$. We prove the following two claims.

Claim 1. *There exists $\mathcal{B}_1$ for the GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathsf{WIN} \wedge F_1] \leq \mathbf{Adv}^{\mathsf{GA\text{-}DLOG}}_{\mathcal{B}_1}(\lambda)$$

Claim 2. *There exists $\mathcal{B}_2$ for the GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathsf{WIN} \wedge F_2] \leq \mathbf{Adv}^{\mathsf{GA\text{-}DLOG}}_{\mathcal{B}_2}(\lambda)$$

From the above claims, we can construct an algorithm $\mathcal{B}$ for the $\mathsf{GA\text{-}DLOG}$ problem that runs either $\mathcal{B}_1$ or $\mathcal{B}_2$ with 1/2 probability, the Lemma 2 follows. □

Proof of Claim 1. Initially, $\mathcal{B}_1$ initializes as described in the $\mathsf{sOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ game. After $\mathcal{B}_1$ receives $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, W)$ from the $\mathsf{GA\text{-}DLOG}$ game, $\mathcal{B}_1$ samples $z \leftarrow\$ \mathbb{Z}_N$ and set $Z := z \star E_0$ and $E_1 := W$. Then $\mathcal{B}$ runs $\mathcal{A}$ on input $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, E_1, Z)$ and with access to the oracles $\hat{\mathbf{S}}_1$, $\hat{\mathbf{S}}_2$ and $\hat{\mathbf{H}}$ which work follows.

Oracle $\hat{\mathbf{S}}_1$: $\mathcal{B}_1$ samples $\mathbf{s}_{\mathsf{sid}}, \mathbf{t}'_{\mathsf{sid}} \leftarrow\$ \mathbb{Z}_N^{\lambda}$, $\mathbf{y}'_{\mathsf{sid}} \leftarrow\$ \{-1,1\}^{\lambda}$ and sets $\mathbf{A}_{\mathsf{sid}} := \mathbf{s}_{\mathsf{sid}} \star E_1^{\mathbf{y}'_{\mathsf{sid}}}$ and $\mathbf{C}_{\mathsf{sid}} := \mathbf{t}'_{\mathsf{sid}} \star E_0$. Then $\mathcal{B}_1$ returns $(\mathsf{sid}, \mathbf{A}_{\mathsf{sid}}, \mathbf{C}_{\mathsf{sid}})$ to the adversary $\mathcal{A}$.

Oracle $\hat{\mathbf{S}}_2$: When $\mathcal{A}$ queries $(i, \mathbf{c}_i)$, $\mathcal{B}_1$ sets $\mathbf{y}_i := \mathbf{c}_i \odot \mathbf{y}'_i$, $\mathbf{t}_i := \mathbf{t}'_i - \mathbf{y}_i \odot \mathbf{z}_i$, and returns $(\mathbf{s}_i, \mathbf{y}_i, \mathbf{t}_i)$ to $\mathcal{A}$.

Oracle $\hat{\mathbf{H}}$: It works the same as in the $\mathsf{sOMUF}^{\mathsf{BS}}_{\mathcal{A}}(\lambda)$ game.

When the adversary returns $(m_k^*, \sigma_k^*)_{k\in[Q_{S_1}+1]}$, $\mathcal{B}_1$ aborts if $\mathsf{WIN} \wedge F_1$ does not occur. It is clear that $\mathcal{B}_1$ simulates the $\mathsf{sOMUF}_{\mathcal{A}}$ game perfectly.

We now show that if $\mathsf{WIN} \wedge F_1$ occurs within the simulation, then $\mathcal{B}_1$ can find a solution for the $\mathsf{GA\text{-}DLOG}$ problem, i.e., find x such that $W = E_1 = x \star E_0$.

Now, if $\mathsf{WIN} \wedge F_1$ occurs, there exist $k \in [Q_{S_1}+1]$ and $j = \mathsf{Hid}(\mathsf{str}^*)$ such that (4) does not hold. Note that

$$\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} = \mathbf{z}_{\mathbf{A}}^{(k)} \star \left(\mathsf{Mat}_{\mathbf{A}}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix}\right)^{\mathbf{e}_{\mathbf{A}}^{(k)}}.$$

If $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}^{(k)} = 0$, then for every $i \in [\lambda]$, there exists $k_i \in [Q_{S_1}+1]$ and $h_i \in [\lambda]$ such that

$$\begin{aligned}
\mathbf{s}_{k,i}^* \star E_1^{\mathbf{c}_{k,i}^* \odot \mathbf{y}_{k,i}^*} &= \mathbf{z}_{\mathbf{A},i}^{(k)} \star (\mathbf{t}_{k_i,h_i} \star Z^{\mathbf{y}_{k_i,h_i}})^{\mathbf{e}_{\mathbf{A},i}^{(k)}} \\
&= (\mathbf{z}_{\mathbf{A},i}^{(k)} + \mathbf{e}_{\mathbf{A}_j,i} \odot \mathbf{t}_{k_i,h_i} + \mathbf{e}_{\mathbf{A}_j,i} \odot z) \star E_0
\end{aligned}$$

which implies that we can obtain the discrete log of $E_1 = W$ which is equal to

$$\mathbf{c}_{k,i}^* \odot \mathbf{y}_{k,i}^* \odot (\mathbf{z}_{\mathbf{A},i}^{(k)} + \mathbf{e}_{\mathbf{A}_j,i} \odot \mathbf{t}_{k_i,h_i} + \mathbf{e}_{\mathbf{A}_j,i} \odot z - \mathbf{s}_{k,i}^*).$$

If $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}^{(k)} \neq 0$, then there exists $(u,v) \in I_{\mathbf{A}}^{(k)}$ such that

$$\begin{aligned}
\mathbf{s}_{k,v}^* \star E_1^{\mathbf{c}_{k,v}^* \odot \mathbf{y}_{k,v}^*} &= \mathbf{z}_{\mathbf{A},i}^{(k)} \star (\mathbf{s}_{u,v} \star E_1^{\mathbf{c}_{u,v} \odot \mathbf{y}_{u,v}})^{\mathbf{e}_{\mathbf{A},v}^{(k)}} \\
&= (\mathbf{z}_{\mathbf{A},i}^{(k)} + \mathbf{e}_{\mathbf{A},v}^{(k)} \odot \mathbf{s}_{u,v}) \star E_1^{\mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot \mathbf{e}_{\mathbf{A},v}^{(k)}}
\end{aligned}$$

Since (4) does not hold, we can obtain the discrete log x by

$$\frac{\mathbf{z}_{\mathbf{A},i}^{(k)} + \mathbf{e}_{\mathbf{A},v}^{(k)} \odot \mathbf{s}_{u,v} - \mathbf{s}_{k,v}^{*}}{\mathbf{c}_{k,v}^{*} \odot \mathbf{y}_{k,v}^{*} - \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot \mathbf{e}_{\mathbf{A},v}^{(k)}}.$$

□

Proof of Claim 2. Initially, $\mathcal{B}_2$ initializes as described in the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{BS}}(\lambda)$ game. After $\mathcal{B}_2$ receives $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, W)$ from the $\mathsf{GA\text{-}DLOG}$ game, $\mathcal{B}_2$ samples $x \leftarrow\$ \mathbb{Z}_N$ and sets $E_1 := x \star E_0$ and $Z = W$. Then $\mathcal{B}$ runs $\mathcal{A}$ with input $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, E_1, Z)$ and with access to the oracles $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2$ and $\hat{\mathsf{H}}$. Since $\mathcal{B}$ knows the secret x, it can simulate al the oracles $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2$ and $\hat{\mathsf{H}}$ in the same way as in the $\mathsf{sOMUF}_{\mathcal{A}}$ game. After receiving the output $\{(m_k^*, \sigma_k^*)\}_{k \in [l+1]}$, $\mathcal{B}_2$ aborts if the event $\mathsf{WIN} \wedge F_2$ does not hold.

It is clear that $\mathcal{B}_2$ simulates the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{BS}}(\lambda)$ game perfectly. We will show that if $\mathsf{WIN} \wedge F_2$ occurs then $\mathcal{B}_2$ can compute the discrete logarithm of $Z = W$.

Now asssume that $\mathsf{WIN} \wedge F_2$ holds, there exists $k \in [Q_{S_1} + 1]$ such that (5) does not hold. Note that

$$\mathbf{t}_k^* \star Z^{\mathbf{y}_k^*} = \mathbf{z}_{\mathbf{C}}^{(k)} \star \left(\mathsf{Mat}_{\mathbf{C}}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \end{pmatrix}\right)^{\mathbf{e}_{\mathbf{C}}^{(k)}}.$$

Since (5) does not hold, there exists $(u, v) \in I_{\mathbf{C}}^{(k)}$ such that

$$\begin{aligned}
\mathbf{t}_{k,v}^* \star Z^{\mathbf{y}_{k,v}^*} &= \mathbf{z}_{\mathbf{C},v}^{(k)} \star \mathbf{A}_{u,v}^{\mathbf{e}_{\mathbf{C},v}^{(k)}} \\
&= \mathbf{z}_{\mathbf{C},v}^{(k)} \star (\mathbf{s}_{u,v} \star E_1^{\mathbf{c}_{u,v} \odot \mathbf{y}_{u,v}})^{\mathbf{e}_{\mathbf{C},v}^{(k)}} \\
&= (\mathbf{z}_{\mathbf{C},v}^{(k)} + \mathbf{e}_{\mathbf{C},v}^{(k)} \odot (\mathbf{s}_{u,v} + \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot x)) \star E_0
\end{aligned}$$

which implies that we can compute the discrete logarithm of $Z = W$ equal to

$$\mathbf{y}_{k,v}^* \star (\mathbf{z}_{\mathbf{C},v}^{(k)} + \mathbf{e}_{\mathbf{C},v}^{(k)} \odot (\mathbf{s}_{u,v} + \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot x) - \mathbf{t}_{k,v}^*).$$

□

E Proof of Theorem 1

We separate the proof of Theorem 1 into several separate proofs below.

E.1 Correctness and Blindness

The correctness is easy to check, as similar to that of the blind signature BS in Section D.2.

Since the algorithm $\mathsf{PBS.User}_1$ and $\mathsf{PBS.User}_2$ are almost the same as $\mathsf{BS.User}_1$ and $\mathsf{BS.User}_2$, the proof for blindness of the PBS scheme can be done as similar to that of BS in Theorem 3. The only difference is that in BS, Z is given in the public key whereas in PBS, Z is given as $\mathsf{G}(\mathsf{info}) \star E_0$. We summary the result as the following.

Theorem 5 (Blindness). *The partially blind signature in Fig. 3 is perfectly blind.*

E.2 One-more Unforgeability

The one-more unforgeability of the partially blind signature PBS is proved in the same manner with that of BS in Theorem 4, but with some modifications because of the difference between two schemes. To be more precise, in the proof for PBS , we need to additionally perform argument over the queries to G, guessing which info will lead to a one-more forgery. In addition, we need to also base on the hardness of the MT-GA-DLOG problem.

Theorem 6 (One-more Unforgeability). *Let $\mathcal{A}$ be an algebraic adversary for the game sOMUF such that for each public information info, makes at most Q_{S_1} queries to PBS.Sign$_1$, and Q_H queries to the random oracle H that start with info. Also let the total number of distinct public information info's queries by $\mathcal{A}$ to BS.Sign$_1$ to be bounded by Q_{info}. Then there exist an adversary $\mathcal{B}_1$ for the discrete logarithm GA-DLOG and and adversary $\mathcal{B}_2$ for the MT-GA-DLOG running in similar running time as $\mathcal{A}$ such that*

$$\mathbf{Adv}^{\mathsf{sOMUF}}_{\mathcal{A},\mathsf{PBS}}(\lambda) \leq \left(1+\frac{1}{Q_{\mathsf{info}}}\right)\mathbf{Adv}^{\mathsf{GA\text{-}DLOG}}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\mathsf{MT\text{-}GA\text{-}DLOG}}_{\mathcal{B}_2}(\lambda)$$
$$+\frac{Q_{\mathsf{info}}\left(Q_H + Q_{S_1} + 1 + \binom{Q_H}{2}\right)}{2^{\lambda}}.$$

Proof. Let $\mathcal{A}$ be the adversary described in the Theorem. Without loss of generality, we assume that if $\mathcal{A}$ outputs the public information info* then $\mathcal{A}$ makes exactly Q_{S_1} queries to PBS.Sign$_1$ and Q_1 queries to PBS.Sign$_2$ that do not return $\perp$ for info*. Then when $\mathcal{A}$ returns outputs, we know that $l(\mathsf{info}^*) = Q_{S_1}$.

In addition to the original $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game, for each query $(\mathsf{info}, \mathbf{A}, \mathbf{C}, m)$ to H, since $\mathcal{A}$ is algebraic, it also outputs the representation of $\mathbf{A}$ and $\mathbf{C}$ recorded as α and β respectively. Here $\alpha = (\mathbf{z}_{\mathbf{A}}, \mathbf{e}_{\mathbf{A}}, \mathsf{Mat}_A)$ with $\mathbf{z}_A \in \mathbb{Z}_N^{\lambda}$, $\mathbf{e}_{\mathbf{A}} \in \{-1,1\}^{\lambda}$ and $\mathsf{Mat}_{\mathbf{A}} \in \{\perp, 1\}^{\lambda\times 2l\lambda}$, and similar to β, as in the proof of Theorem 4.

Note that, since in our PBS, in contrast to BS, Z is not the public key, so the representations of $\mathbf{A}$ and $\mathbf{C}$ also need to involve the requested Z's corresponding to the requested info's. To be more precise, we denote by $\mathbf{A}_1, \mathbf{C}_1, \ldots, \mathbf{A}_l, \mathbf{C}_l$ are the answers to l queries of $\mathcal{A}$ to BS.Sign$_1$ and $Z_1, \ldots, Z_{Q_{\mathsf{info}}}$ are the answers to Q_{info} queries of $\mathcal{A}$ to BS.Sign$_1$. We concatenate them into a column vector $(\mathbf{A}_1\|\mathbf{C}_1\|\ldots\|\mathbf{A}_l\|\mathbf{C}_l\|Z_1\|\ldots\|Z_{Q_{\mathsf{info}}})^T$ of length $2l\lambda + Q_{\mathsf{info}}$ and denote it by $\begin{pmatrix}\mathbf{A}\\ \mathbf{C}\\ Z\end{pmatrix}$. We also denote by $\mathsf{Mat}_{\mathbf{A}}$ (resp. $\mathsf{Mat}_{\mathbf{C}}$) a $\lambda \times (2l\lambda + Q_{\mathsf{info}})$ matrix with exactly one non-zero (e.g., number 1) in each row and $\perp$ otherwise. Hence $\mathsf{Mat}_{\mathbf{A}} \cdot \begin{pmatrix}\mathbf{A}\\ \mathbf{C}\\ Z\end{pmatrix}$ will be a vector of length λ whose each component is either $\mathbf{A}_{i,j}$

or $\mathbf{C}_{i,j}$ for $i \in [l]$ and $j \in [\lambda]$ or Z_i for $i \in [Q_{\mathsf{info}}]$. Furthermore, we denote $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ (resp. $\mathsf{Mat}_{\mathbf{A},\mathbf{C}}$, $\mathsf{Mat}_{\mathbf{A},Z}$) to be the $\lambda \times l\lambda$ (resp. $\lambda \times l\lambda$, $\lambda \times Q_{\mathsf{info}}$ matrix obtained from the odd columns (resp. even columns) of $\mathsf{Mat}_{\mathbf{A}}$, i.e., $\mathsf{Mat}_{\mathbf{A},\mathbf{A}}$ will correspond to the $\mathbf{A}_{i,j}$ entries in $\begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix}$.

We denote by WIN the event that $\mathcal{A}$ wins the $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game, i.e., $\mathcal{A}$ outputs message-signature pairs $(m_k^*, \sigma_k^*)_{k \in [l+1]}$ are distinct and valid for info^*. Furthermore, we also denote $\mathsf{str}_k^* := \mathsf{info}^* \| \mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} \| \mathbf{t}_k^* \star Z_{\mathsf{Fid}(\mathsf{info}^*)}^{\mathbf{y}_k^*} \| m_k^*$. We note that when the validity of the signature (m_k^*, σ_k^*) is checked, then str_k^* is made to H, and there should be $\alpha_k = (\mathbf{z}_{\mathbf{A}}^{(k)}, \mathbf{e}_{\mathbf{A}}^{(k)}, \mathsf{Mat}_{\mathbf{A}}^{(k)})$ and $\beta_k = (\mathbf{z}_{\mathbf{C}}^{(k)}, \mathbf{e}_{\mathbf{C}}^{(k)}, \mathsf{Mat}_{\mathbf{C}}^{(k)})$ representing $\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*}$ and $\mathbf{t}_k^* \star Z^{\mathbf{y}_k^*}$ respectively in terms of the known commitments $\mathbf{A}_1, \mathbf{C}_1, \ldots, \mathbf{A}_l, \mathbf{C}_l, Z_1, \ldots, Z_{Q_{\mathsf{info}}}$ i.e.,

$$\mathbf{s}_k^* \star E_1^{\mathbf{c}_k^* \odot \mathbf{y}_k^*} = \mathbf{z}_{\mathbf{A}}^{(k)} \star \left(\mathsf{Mat}_{\mathbf{A}}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix} \right)^{\mathbf{e}_{\mathbf{A}}^{(k)}}$$

and

$$\mathbf{t}_k^* \star Z_{i_*}^{\mathbf{y}_k^*} = \mathbf{z}_{\mathbf{C}}^{(k)} \star \left(\mathsf{Mat}_{\mathbf{C}}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix} \right)^{\mathbf{e}_{\mathbf{C}}^{(k)}}. \tag{6}$$

Let E be the event in the $\mathsf{sOMUF}^{\mathsf{PBS}}_{\mathcal{A}}(\lambda)$ game for which, after the validity of the output is checked, for each $k \in [l+1]$, the following conditions hold.

(i) $\mathbf{c}_{k,v}^* \odot \mathbf{y}_{k,v}^* = \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot \mathbf{e}_{\mathbf{A},v}^{(k)}$ for all $(u,v) \in I_{\mathbf{A}}^{(k)}$; and

(ii) All the $\mathbf{C}_i$ appearing in Eq. (6) correspond to info^*, i.e., $\mathbf{C}_i \overset{(303)}{=} \mathbf{t}_i \star Z_{i_*}^{\mathbf{y}_i}$ for such $i \in [Q_{S_1}+1]$ that the corresponding column in $\mathsf{Mat}_{\mathbf{C}}$ is not $\perp$.

Then

$$\mathbf{Adv}^{\mathsf{sOMUF}}_{\mathcal{A},\mathsf{PBS}}(\lambda) = \Pr[\mathsf{WIN} \wedge E] + \Pr[\mathsf{WIN} \wedge (\neg E)].$$

We provide upper bounds for the above two probabilities in the following lemmas. The Theorem 6 hence follows.

Lemma 3.

$$\Pr[\mathit{WIN} \wedge E] \leq \frac{Q_{\mathit{info}} \left(Q_H + Q_{S_1} + 1 + \binom{Q_H}{2} \right)}{2^\lambda}.$$

Lemma 4. *There exists an algorithm $\mathcal{B}_\infty$ for the GA-DLOG problem and $\mathcal{B}_2$ for MT-GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathit{WIN} \wedge (\neg E)] \leq \left(1 + \frac{1}{Q_{\mathit{info}}}\right) \mathbf{Adv}^{\mathit{GA\text{-}DLOG}}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\mathit{MT\text{-}GA\text{-}DLOG}}_{\mathcal{B}_2}(\lambda).$$

$\square$

Proof of Lemma 3

Proof. It is clear that $\Pr[\mathsf{WIN} \wedge E]$ is less than the probability that (i) holds. The proof is similar to that of Lemma 1 with a difference that $\mathcal{A}$ first needs to samples a label i_* uniformly random from $[Q_{\mathsf{info}}]$ and we then proceed as in the proof of Lemma 1, resulting in the factor Q_{info} in the nominator. □

Proof of Lemma 4

Proof. We first partition the event $\mathsf{WIN} \wedge (\neg E)$ into two cases F_1 as the event in the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{PBS}}$ game that there exists $k \in [Q_{S_1} + 1]$ such that either (i) does not hold, and F_2 as the event that (ii) does not hold. If E does not occur, then either F_1 or F_2 occurs. Therefore $\mathsf{WIN} \wedge (\neg E) = (\mathsf{WIN} \wedge F_1) \vee (\mathsf{WIN} \wedge F_2)$. We prove the following two claims.

Claim 3. *There exists $\mathcal{C}_1$ for the GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathit{WIN} \wedge F_1] \leq \mathbf{Adv}_{\mathcal{C}_1}^{\mathit{GA\text{-}DLOG}}(\lambda)$$

Claim 4. *There exist $\mathcal{B}_2$ for the MT-GA-DLOG problem and $\mathcal{C}_3$ for the GA-DLOG problem running in a similar running time as $\mathcal{A}$ such that*

$$\Pr[\mathit{WIN} \wedge F_2] \leq \mathbf{Adv}_{\mathcal{B}_2}^{\mathit{MT\text{-}GA\text{-}DLOG}}(\lambda) + \frac{1}{Q_{\mathit{info}}} \mathbf{Adv}_{\mathcal{C}_3}^{\mathit{GA\text{-}DLOG}}(\lambda).$$

From the above claims, we can construct an algorithm $\mathcal{B}_1$ for the $\mathsf{GA\text{-}DLOG}$ problem that runs either $\mathcal{C}_1$ or $\mathcal{C}_3$ with $1/2$ probability, the Lemma 4 follows. □

The proof of Claim 3 is similar to that of Claim 1, and hence we omit it. The proof of Claim 4 also follows the same manner as that of Claim 2 with some differences, and hence we will provide a sketch of the differences and reduction to $\mathsf{MT\text{-}GA\text{-}DLOG}$ instead.

Proof of Claim 4. Initially, $\mathcal{B}_2$ initializes as described in the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{PBS}}(\lambda)$ game. After $\mathcal{B}_2$ receives $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, W_1, \ldots, W_{Q_{\mathsf{info}}})$ from the $\mathsf{MT\text{-}GA\text{-}DLOG}$ game, $\mathcal{B}_2$ samples $x \leftarrow\!\!\$\ \mathbb{Z}_N$ and sets $E_1 := x \star E_0$ and $Z_i = W_i$ for $i \in [Q_{\mathsf{info}}]$. Then $\mathcal{B}$ runs $\mathcal{A}$ with input $(\mathbb{Z}_N, \mathcal{X}, \star, E_0, E_1)$ and with access to the oracles $\hat{\mathsf{G}}, \hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2$ and $\hat{\mathsf{H}}$. Since $\mathcal{B}$ knows the secret x, it can simulate all the oracles $\hat{\mathbf{S}}_1, \hat{\mathbf{S}}_2$ and $\hat{\mathsf{H}}$ in the same way as in the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{PBS}}$ game. For any query of $\mathcal{A}$ to $\hat{\mathsf{G}}$ regarding the information info_i, $\mathcal{B}_2$ will answer with the next available Z_i (for $i \in [Q_{\mathsf{info}}]$. After receiving the output $\{\mathsf{info}^*, (m_k^*, \sigma_k^*)_{k \in [l+1]}\}$, $\mathcal{B}_2$ aborts if the event $\mathsf{WIN} \wedge F_2$ does not hold.

It is clear that $\mathcal{B}_2$ simulates the $\mathsf{sOMUF}_{\mathcal{A}}^{\mathsf{PBS}}(\lambda)$ game perfectly. We will show that if $\mathsf{WIN} \wedge F_2$ occurs then $\mathcal{B}_2$ can solve the $\mathsf{MT\text{-}GA\text{-}DLOG}$ problem with instances $Z_1, \ldots, Z_{Q_{\mathsf{info}}}$.

Now asssume that $\mathsf{WIN} \wedge F_2$ holds, there exists $k \in [Q_{S_1}+1]$ such that (5) does not hold. Note that

$$\mathbf{t}_k^* \star Z_{i_*}^{\mathbf{y}_k^*} = \mathbf{z}_{\mathbf{C}}^{(k)} \star \left(\mathsf{Mat}_{\mathbf{C}}^{(k)} \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{C} \\ Z \end{pmatrix}\right)^{\mathbf{e}_{\mathbf{C}}^{(k)}}.$$

Since (ii) does not hold, we will consider two sub-events: $F_{2,1} : \mathsf{Mat}_{\mathbf{C},\mathbf{A}} = 0$ and $F_{2,2} : \mathsf{Mat}_{\mathbf{C},\mathbf{A}} \neq 0$.

If $F_{2,1}$ holds, i.e., $\mathsf{Mat}_{\mathbf{C},\mathbf{A}} = 0$, there exists $u \in [Q_{\mathsf{info}}]$ with $u \neq i_*$ and $v \in [\lambda]$ such that either (a) $\mathbf{t}_{k,v}^* \star Z_{i_*}^{\mathbf{y}_{k,v}^*} = \mathbf{z}_{\mathbf{C},v}^{(k)} \star Z_u^{\mathbf{e}_{\mathbf{C},v}^{(k)}}$ or (b) $\mathbf{t}_{k,v}^* \star Z_{i_*}^{\mathbf{y}_{k,v}^*} = \mathbf{z}_{\mathbf{C},v}^{(k)} \star \mathbf{C}_{u,v_u}^{\mathbf{e}_{\mathbf{C},v}^{(k)}}$ for some $u \in [Q_{S_1}+1], v_u \in [\lambda]$. Since (ii) does not hold, either (a) or (b) happens, and hence we can solve the $\mathsf{MT\text{-}GA\text{-}DLOG}$ problem. Therefore $\Pr[F_{2,1}] \leq \mathbf{Adv}_{\mathcal{B}_2}^{\mathsf{MT\text{-}GA\text{-}DLOG}}(\lambda)$.

If $F_{2,2}$ holds, then there exists $(u,v) \in I_{\mathbf{C}}^{(k)}$ such that

$$\begin{aligned}
\mathbf{t}_{k,v}^* \star Z_{i_*}^{\mathbf{y}_{k,v}^*} &= \mathbf{z}_{\mathbf{C},v}^{(k)} \star \mathbf{A}_{u,v}^{\mathbf{e}_{\mathbf{C},v}^{(k)}} \\
&= \mathbf{z}_{\mathbf{C},v}^{(k)} \star (\mathbf{s}_{u,v} \star E_1^{\mathbf{c}_{u,v} \odot \mathbf{y}_{u,v}})^{\mathbf{e}_{\mathbf{C},v}^{(k)}} \\
&= (\mathbf{z}_{\mathbf{C},v}^{(k)} + \mathbf{e}_{\mathbf{C},v}^{(k)} \odot (\mathbf{s}_{u,v} + \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot x)) \star E_0
\end{aligned}$$

which implies that we can compute the discrete logarithm of $Z_{i_*} = W_{i_*}$ equal to

$$\mathbf{y}_{k,v}^* \star (\mathbf{z}_{\mathbf{C},v}^{(k)} + \mathbf{e}_{\mathbf{C},v}^{(k)} \odot (\mathbf{s}_{u,v} + \mathbf{c}_{u,v} \odot \mathbf{y}_{u,v} \odot x) - \mathbf{t}_{k,v}^*).$$

In this case, then we can construct an algorithm $\mathcal{C}_3$ against $\mathsf{GA\text{-}DLOG}$ problem to embed the challenge W into one of the $Z_1, \ldots, Z_{Q_{\mathsf{info}}}$. Hence

$$\Pr[F_{2,2}] \leq \frac{1}{Q_{\mathsf{info}}} \mathbf{Adv}_{\mathcal{C}_3}^{\mathsf{GA\text{-}DLOG}}(\lambda).$$

□

F Optimization

In this section, we adapt the techniques from [33] used to optimizing CSI-Otter to optimizing our (paritally) blind signatures. We sketch our the interactions between signer and user and leave the full description and security proofs to the full version of the paper.

Here we use $\mathbb{Z}_d$ to denote the set $\{0,\ldots,d-1\}$. For any positive integer d and $a \in \mathbb{Z}$, we write $[a]_d$ for $(a \mod d) \in \mathbb{Z}_d$. We will write ζ (instead of ζ_d) for the d-root of unity and denote by $\langle\zeta\rangle$ to be the group $\{1,\zeta,\zeta^2,\ldots,\zeta^{d-1}\}$ the group of d-th primitive roots of unity over $\mathbb{Z}_N^\times$, i.e., ζ satisfies $\zeta^d = 1$ and $\zeta^j \neq 1$ for all $j \in [d-1]$. The action of $(r,c) \in \mathbb{Z}_N \times \mathbb{Z}_d$ on a curve $E_0 \in \mathcal{X}$ is defined to be $\mathbf{g}^{r\zeta^c} \star E_0$ or $(r\zeta^c) \star E_0$ for simplicity. Note that when $d = 2$ then $\zeta = -1$ and this is identical to previous sections. Note that unlike the case $d = 2$ where we can easily compute quadratic twist, for $d \geq 3$, the signer needs to perform extra calculations to support the user's computation. The descriptions of the interactions between signer and user in our optimized blind signature BS_{ζ_d} and partially blind signature PBS_{ζ_d} are given in Fig. 5 and Fig. 6 respectively. Here we will use E to replace for E_0 as in BS and PBS, i.e., $\mathsf{CEGAT} = (\mathbb{Z}_N, \mathcal{X}, \star, E)$. For simplicity, we just describe the schemes with challenge space $\mathbb{Z}_d$. For the real instantiations, one need to "repeat" κ times to attain the required security level. For our instantiation, as in [33], $d = 4$ and $\kappa = 64$. Note that the hash function H in these two basic schemes is now $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_d$.

$\mathsf{Signer}(\mathsf{sk} = x, \mathsf{pk} = ((E_j)_{j\in[d]}, Z))$ with $E_j = (x\zeta^j) \star E$ for $j \in [d]$		$\mathsf{User}(\mathsf{pk} = ((E_j)_{j\in[d]}, Z), m)$
$a, t \leftarrow\$ \mathbb{Z}_N; y \leftarrow\$ \mathbb{Z}_d$ $C \leftarrow t \star Z^y$ $A_j \leftarrow (a\zeta^j) \star E \ \forall\, j \in [d]$	$\xrightarrow{((A_j)_{j\in[d]}, C)}$	
		$r_1, r_2 \leftarrow\$ \mathbb{Z}_N$ $\gamma_1, \gamma_2 \leftarrow\$ \mathbb{Z}_d$ $C' \leftarrow r_2 \star C^{\zeta^{\gamma_1}}$ $A'_j \leftarrow (r_1\zeta^j) \star A^{\zeta^{[\gamma_1+\gamma_2]_d}} \ \forall\, j \in [d]$ $c' \leftarrow \mathsf{H}((A'_j)_{j\in[d]}, C', m)$
	$\xleftarrow{c}$	$c \leftarrow [c' - \gamma_2]_d$
$s \leftarrow a - \zeta^c \cdot y \cdot x$	$\xrightarrow{(s, y, t)}$	
		$s' \leftarrow \zeta^{[\gamma_1 \cdot \gamma_2]_d} \cdot s + r_1$ $y' \leftarrow \zeta^{\gamma_1} \cdot y$ $t' \leftarrow \zeta^{\gamma_1} \cdot t + r_2$ If $c' = \mathsf{H}(((s'\zeta^j) \star E_j^{\zeta^{c'} \cdot y'})_{j\in[d]}, t' \star Z^{y'}, m)$ Return $\sigma = (s', t', y', c')$

Fig. 5: Our optimized blind signature BS_{ζ_4}

The security of BS_{ζ_4} and PBS_{ζ_4} now also reply on the so-called ζ_d-Group Action Discrete Logarithm Problem, denoted by $\zeta_d - \mathsf{GA\text{-}DLOG}$ defined as the following [33].

Definition 12 (ζ_d-Ring Group Action Discrete Logarithm Problem (rGA-DLOG)). *Given $(E, S) \in \mathcal{X}^{d+1}$ where $S = (((s\zeta_d^j) \star E_0)_{j\in[d]}, s \leftarrow\!\!\$\ \mathbb{Z}_N$ and $d|\lambda(N))$ (here λ is the Carmichael function), the ζ_d-ring group action discrete logarithm problem (ζ_d-rGA-DLOG) is to recover s.*

Here the Carmichael function $\lambda(N)$ of a positive integer N is the smallest positive integer m such that $a^m \equiv 1 \mod N$ holds for every a coprime to N. Katsumata et al. [33] provided a structural analysis on ζ_d-rGA-DLOG for CSIDH-512 [16] and display a few weak and hard instances depending on ζ_d. They also showed that for some carefully-chosen d (depending on N), ζ_d-rGA-DLOG is essentially as hard as the original GA-DLOG.

Here we show the correctness of BS_{ζ_4}. The correctness for PBS_{ζ_4} can be done similarly. First of all, for each $j \in [d]$ we have

$$\begin{aligned}(s\zeta^j) \star E_j^{y\cdot\zeta^c} &= (a\zeta^j - \zeta^{[c+j]_d} \cdot y \cdot x) \star ((x\zeta^j) \star E)^{y\cdot\zeta^c} \\ &= (a\zeta^j) \star E = A_j. \end{aligned} \tag{7}$$

Hence

$$\begin{aligned}(s'\zeta^j) \star E_j^{y'\cdot\zeta^{c'}} &= (r_1\zeta^j + s\zeta^j \cdot \zeta^{[\gamma_1+\gamma_2]_d}) \star E_j^{y\cdot\zeta^{[\gamma_1+\gamma_2]_d}\cdot\zeta^c} \\ &= (r_1\zeta^j) \star A_j^{\zeta^{[\gamma_1+\gamma_2]_d}}\end{aligned}$$

where the last equation follows from Eq. (7); and

$$\begin{aligned}t' \star Z^{y'} &= (\zeta^{\gamma_1} \cdot t + r_2) \star Z^{y\cdot\zeta^{\gamma_1}} \\ &= r_2 \star C^{\zeta^{\{}\gamma_1} = C'.\end{aligned}$$

Therefore $c' = \mathsf{H}((((s'\zeta^j) \star E_j^{\zeta^{c'}\cdot y'})_{j\in[d]}, t' \star Z^{y'}, m)$ and the verifier will output 1.

$$
\begin{array}{lcl}
\mathsf{Signer}(\mathsf{sk} = x, \mathsf{pk} = ((E_j)_{j \in [d]}, \mathsf{info})) & & \mathsf{User}(\mathsf{pk} = ((E_j)_{j \in [d]}, \mathsf{info}), m) \\
\quad \text{with } E_j = (x\zeta^j) \star E \text{ for } j \in [d] & & \\
\hline
a, t, z \leftarrow\!\!\$\, \mathbb{Z}_N; y \leftarrow\!\!\$\, \mathbb{Z}_d & & \\
Z \leftarrow \mathcal{G}(\mathsf{info}) \star E & & \\
C \leftarrow t \star Z^y & & \\
A_j \leftarrow (a\zeta^j) \star E \ \forall\ j \in [d] & \xrightarrow{((A_j)_{j \in [d]}, C)} & \\
 & & r_1, r_2 \leftarrow\!\!\$\, \mathbb{Z}_N \\
 & & \gamma_1, \gamma_2 \leftarrow\!\!\$\, \mathbb{Z}_d \\
 & & C' \leftarrow r_2 \star C^{\zeta^{\gamma_1}} \\
 & & A'_j \leftarrow (r_1\zeta^j) \star A^{\zeta^{[\gamma_1+\gamma_2]_d}} \ \forall\ j \in [d] \\
 & & c' \leftarrow \mathsf{H}((A'_j)_{j \in [d]}, C', m) \\
 & \xleftarrow{c} & c \leftarrow [c' - \gamma_2]_d \\
s \leftarrow a - \zeta^c \cdot y \cdot x & \xrightarrow{(s, y, t)} & Z \leftarrow \mathcal{G}(\mathsf{info}) \star E \\
 & & s' \leftarrow \zeta^{[\gamma_1 \cdot \gamma_2]_d} \cdot s + r_1 \\
 & & y' \leftarrow \zeta^{\gamma_1} \cdot y \\
 & & t' \leftarrow \zeta^{\gamma_1} \cdot t + r_2 \\
 & & \text{If } c' = \mathsf{H}(((s'\zeta^j) \star E_j^{\zeta^{c'} \cdot y'})_{j \in [d]}, t' \star Z^{y'}, m) \\
 & & \quad \text{Return } \sigma = (s', t', y', c')
\end{array}
$$

Fig. 6: Our optimized partially blind signature PBS_{ζ_4}