(Partially) Blind Signatures from Cryptographic Group Actions

Dung Hoang Duong[1], Xuan Thanh Khuc[1], Youming Qiao[2], Willy Susilo[1], and Chuanqi Zhang[3]

[1] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Australia.
{hduong,xtkhuc,wsusilo}@uow.edu.au
[2] School of Computer Science and Engineering, the University of New South Wales, Australia.
jimmyqiao86@gmail.com
[3] Department of Software Systems & Cybersecurity, Faculty of Information Technology, Monash University, Australia.
chuanqi.zhang@monash.edu

Abstract. Blind and partially blind signatures are important tools in building privacy-preserving systems such as electronic cash, anonymous authentication, and credential schemes. Blind signatures allow a user to obtain a valid signature without revealing the message to the signer, protecting user privacy. Partially blind signatures extend this idea by letting both parties agree on some public information - like validity periods or conditions - while still hiding the rest of the message, balancing privacy with accountability.
In this paper, we provide a generic construction of (partially) blind signatures from cryptographic group actions following the framework of the blind signature CSI-Otter introduced by Katsumata et al. (CRYPTO'23) in the context of isogeny (commutative group action). We adapt and modify that framework to make it work even for non-commutative group actions. As a result, we obtain a (partially) blind signature from abstract group actions which are proven to be secure in the random oracle model (ROM). We also propose an instantiation based on a variant of linear code equivalence, interpreted as a symmetric group action.

Keywords: (Partially) Blind Signatures, Group Actions, Canonical Forms

1 Introduction

Blind signatures, introduced by Chaum [28] in 1982, are interactive protocols between a signer, who holds a secret key, and a user, who holds a message, to jointly create a signature on a message in such a way that the message is oblivious to the signer at the signing time. Blind signatures have found many applications such as in e-cash [28, 30], in e-voting [29, 60], and in blockchains [26, 52, 75], and much more; see [44] and references therein for a rich list of applications and references.

One approach to construct a blind signature is to design a Schnorr-like sigma protocol [31] (or identification scheme) which has *module* structures [49] enabling the randomization of the interaction. The Schnorr blind signature was generalized by Pointcheval and Stern [70] and Abe and Okamoto [1]. The security proof of Abe and Okamoto contained a bug that has recently been fixed by Kastner, Loss and Xu [55] who provided a generic proof for Abe-Okamoto style blind signature. At CRYPTO'23, Katsumata et al. [57] proposed the first isogeny-based blind signature in the context of *cryptographic group actions*, called CSI-Otter, inspired by the Abe-Okamoto's construction in which Katsumata et al. utilized the *quadratic twist* of an elliptic curve in a clever way to endow isogenies with richer structure than abstract group actions, but still weaker than module structures, that enables the blindness. The security proofs are obtained by adapting the framework by Kastner et al. [55] into their setting.

Cryptographic group actions were first introduced by Brassard and Yung [22] in the context of *one-way* group actions. It was then considered independently by Couveignes [35] in the context of *hard homogeneous spaces* and by Rostovsev and Stolbunov [72] in the context of isogenous elliptic curves. This line of research was largely ignored until the proposal of CSIDH by Castryck et al. [27] in which the authors considered supersingular elliptic curves defined over a large prime field, rather than to ordinary elliptic curves as in the previous work of Couveignes [35] and Rostovsev-Stolbunov [72], on which many efficient isogeny-based constructions are based, such as CSI-FiSh signature [15], threshold signature [43], ring signatures [14], group signature [13] and blind signature CSI-Otter [57].

In the context of non-commutative group actions, there have been several proposals that submitted to NIST's recent call for additional post-quantum signatures[4], including LESS [17], MEDS [33], ALTEQ [74], and HAWK [39]. There have been several analogous cryptographic constructions to the case of isogenies in this context, such as (inefficient) threshold signature [8] and ring signatures [7, 18]. However, due to the non-commutativity of the underlying groups, the cryptographic constructions in this setting are still limited. For example, public key encryptions based on non-commutative group actions are only recently shown with *quantum ciphertexts* [53].

Even though non-commutative group action constructions are less efficient than the isogeny counterparts in terms of key/signature sizes, those schemes however enjoy efficiency in terms of implementation. Furthermore, actions by "highly non-commutative" groups, such as symmetric and general linear groups, enjoy the property that most known quantum algorithmic techniques do not work for hidden subgroup problems for such groups [47]. These make non-commutative group actions an appealing candidate for post-quantum cryptography, so it is desirable to develop more advanced cryptographic schemes to increase cryptographic functionalities based on them. In particular, it brings to us the following question:

> *Can we construct a (partially) blind signature from non-commutative group actions?*

[4] https://csrc.nist.gov/news/2023/additional-pqc-digital-signature-candidates

1.1 Our Contribution

In this paper, we provide an affirmative answer to the above question. Our contribution in this paper is two-fold and can be summarized as follows.

- We provide a framework to construct a Schnorr-type blind signature from abstract group actions. Our framework follows closely with the construction of CSI-Otter [57] with modifications to adjust for the case of generic groups. In particular, because we do not have *twists* as in the case of elliptic curves, we need to *double* the public key, compared to that of CSI-Otter, in such a way that the *additional* public key element plays a twist role in our context; see Section 1.2 for more detail. Another contribution in this fold is a zero-knowledge proof for well-formed public key. In contrast to the case of isogenies, one can easily verify that the public key is valid, i.e., the public key is indeed a valid supersingular elliptic curve. In our setting, we need a zero-knowledge protocol allowing one to validate the public key. In addition, we also extend our blind signature into a *partially* blind signature.
- We provide an efficient instantiation from non-commutative group actions.[5] In order to provide an instantiation for non-commutative group actions, we require several conditions for the underlying group. If the group is non-commutative, to ensure the soundness of our protocol, we need a group with an efficient square-root algorithm. This is because in our protocol, given two accepted transcripts with the same commitment, an extractor can obtain only g^2, where g is the secret key. Hence, we need an efficient algorithm to compute g given g^2 in the group. In order to enable an efficient instantiation inherited from existing efficient schemes, we proposed a variant of LESS that instead of having monomial matrix $\mathrm{Mon}(n, q)$ action as in LESS, we have a permutation group action which fulfils our purpose; see details in Section 1.2.

1.2 Technique Overview

In this section, we present in detail our contributions. We will first describe the core in the construction of CSI-Otter [57], on which our framework is based. We then present a variant of linear code equivalence problem in LESS from which we provide an instantiation for a blind signature following our framework.

Construction Framework. In CSI-Otter [57], the authors consider the CSIDH group action $* : G \times \mathcal{E} \to \mathcal{E}$ where G is an ideal class group and $\mathcal{E}$ is a set of elliptic curves. It can be assumed that the structure of G is known and we can express G as $G = \langle \mathfrak{g} \rangle \cong \mathbb{Z}_N$ for some positive integer $N \in \mathbb{N}$ and generator $\mathfrak{g} \in G$. In isogeny settings, an elliptic curve $E_0 \in \mathcal{E}$ is fixed and the public key is of the form $A = [\mathfrak{g}^a] * E_0$ for a random $a \leftarrow\!\!\$\, \mathbb{Z}_N$, and the first-sender message (i.e., commitment) is computed similarly as $Y = [\mathfrak{g}^y] * E_0$ for a random $y \leftarrow\!\!\$\, \mathbb{Z}_N$. In order to enable a Schnorr-type blind signature, the normal procedure for the

[5] We note that our framework can be instantiated for generic groups, both commutative and non-commutative.

user in blinding the message M would be: (i) randomize the commitment, which can be done by computing $[\mathfrak{g}^z] * Y$ for $z \leftarrow\!\$\, \mathbb{Z}_N$; (ii) randomize the public key A and public parameter E_0, which can be done by computing $[\mathfrak{g}^b] * A$ and $[\mathfrak{g}^d] * E_0$ for $d, b \leftarrow\!\$\, \mathbb{Z}_N$; (iii) associate $[\mathfrak{g}^z] * Y$, $[\mathfrak{g}^b] * A$ and $[\mathfrak{g}^d] * E_0$ into one element, say X; (iv) compute the hash value $c = H(X \| M)$; and lastly (v) use z, d, b to randomize c to obtain a randomized challenge c' and send to the signer. In the discrete logarithm setting [31], all steps (i)–(v) can be done easily, especially step (iii) since $[\mathfrak{g}^z] * Y$, $[\mathfrak{g}^b] * A$ and $[\mathfrak{g}^d] * E_0$ are all group elements on which we can do operation to create a group element X. However, it is not the case for isogeny setting since $[\mathfrak{g}^z] * Y$, $[\mathfrak{g}^b] * A$ and $[\mathfrak{g}^d] * E_0$ are all elliptic curves and we do not have operations on elliptic curves.

In order to overcome that problem, Katsumata et al. [57] have cleverly used *quadratic twist* in elliptic curves. Briefly speaking, given $A = [\mathfrak{g}^a] * E_0$ for an unknown $a \in \mathbb{Z}$, every one can easily compute its quadratic twist $[\mathfrak{g}^{-a}] * E_0$, which was denoted by A^{-1} in [57]. Now step (i)–(iii) above can be done together in [57] as follows: choose $(d, z) \leftarrow\!\$\, \{-1, 1\} \times \mathbb{Z}_N$ and set $X := [\mathfrak{g}^z] * Y^d$. For the proof to work, following the proof by Kastner-Loss-Xu [55] for Abe-Okamoto blind signature [1], Katsumata et al. [57] modified the above idea to use the OR composition of the underlying sigma protocol. Specifically, CSI-Otter uses two public key $A_0 = [\mathfrak{g}^a] * E_0, A_1 = [\mathfrak{g}^b] * E_0$ where $a, b \leftarrow\!\$\, \mathbb{Z}_N$, and the secret key is one of the a_0 or a_1; see [57] for the details.

Since quadratic twists exist only in the isogeny setting, to enable a construction analogous to CSI-Otter for abstract group actions, what we do is to *double* the public key. To be more precise, consider a group G acting on a set S by $* : G \times S \to S$ and fix an element $E_0 \in S$.[6] Our public key will consist of $A_b^{(c)} = g_b^c * E_0$ for $b \in \{0, 1\}, c \in \{-1, 1\}$ and the secret key is either $g_0 \in G$ or $g_1 \in G$. Here $A_b^{(-1)}$ will play the role for quadratic twists of $A_b^{(1)}$ as in the case of CSI-Otter. We also construct a protocol for an OR relation (cf. [40, Fig. 7]) as in CSI-Otter as the underlying sigma protocol for the blind signature. Our blind signature follows the same route as in CSI-Otter but with modifications; see Section 4 for the detail. We highlight below two noteworthy differences between our scheme with CSI-Otter:

- Firstly, in our scheme, one of the responses is of the form $r = hg_\delta^{-1} \in G$ where g_δ with $\delta \in \{0, 1\}$ is the secret key. For the verification, we need to compute the action on $A_\delta^{(-1)} = g_\delta^{-1} * E_0$ and expect the outcome to be one of the commitments $Y := h^{-1} * E_0$.
  In this case, if we use $r^{-1} = g_\delta h^{-1}$ then we need g and h^{-1} commute. However, requiring h^{-1} to commute with g would break the HVZK property of the underlying sigma protocol.
  Therefore, instead of having one response hg_δ^{-1}, we send two responses hg_δ^{-1} and $h^{-1}g_\delta$. This turns out to be useful in restoring HVZK.
- Secondly, in order to ensure the soundness of the underlying sigma protocol in our scheme, the extractor can obtain, from two given accepted transcripts

[6] We use the same notation as in CSI-Otter to make readers easily follow the flows of the construction.

with the same commitment, the square g_δ^2 of the secret key g_δ. Hence, we need an efficient square-root algorithm in the group G to compute g_δ from g_δ^2, which we provide in Section 6.

Another contribution is a zero-knowledge protocol to validate the public key. In contrast to the case of isogeny-based cryptography in which everyone can easily validate the public key – a valid supersingular elliptic curve, it is not the case for abstract group actions. Our protocol is presented in Fig. 1.

Extending to Partially Blind. We also extend our construction to obtain a partially blind signature from abstract (non-commutative) group actions, also following the framework by Katsumata et al. [57] in the isogeny setting. The idea is to keep the public key as in the blind signature consisting of $(A_b^{(c)})$ with $b \in \{0,1\}$ and $c \in \{-1,1\}$ as before, and add two more elements $A_2^{(1)} = g_2 * E_0$ and $A_2^{(-1)} := g_2^{-1} * E_0$ where $g_2 = \mathsf{G}(\mathsf{info})$ with G a function $\{0,1\}^* \to G$. We then modify the signer to prove that it knows at least two of the three secrets (g_0, g_1, g_2) generating $(A_b^{(c)})$ with $b \in \{0,1,2\}$ and $c \in \{-1,1\}$. Our first step is to provide a sigma protocol for a 2-out-of-3 relation in which the prover convinces the verifier that he knows two of the secrets, say g_0, g_2, among three g_0, g_1, g_2. Then we turn this protocol into a partially blind signature. Because of the 'extra' secret g_2 and hence the extra public key, which is associated with the tag info, the resulting transcript and hence signature size of the partially blind signature is increased by 50% compared to that of the blind signature, which happens similarly in the case of CSI-Otter [57].

Instantiation. As mentioned above, to instantiate our blind signatures with non-commutative group actions, we require the corresponding group G to satisfy the following:

- Given g^2, there exists an efficient algorithm to compute g.
- Reusing $g \in G$ twice still gives a secure protocol (see Definition 4 for a more precise requirement).

To identify a non-commutative group action satisfying the above seems like a tricky business. We will make further discussions in Section 6.1. To address the issues, we interpret the monomial code equivalence problem as a group action of the symmetric group. This is made possible by a canonical form algorithm ([40, Appendix B]). In contrast to LESS, where the group action is by the monomial group, our group actions avoid the attacks on reusing group elements as in [23,25] (Section 6.2). We also show that by selecting a family of permutations that satisfy the square-root requirement ([40, Appendix B]).

1.3 Related Work

Most of the existing post-quantum blind signatures are constructed from lattices. The first post-quantum blind signature was proposed by Rückert [73] following the design paradigm by Pointcheval and Stern [70]. However, Hauck et al. [50]

discovered a flaw in Rückert's security argument which results in many blind signatures [5, 21, 63, 65] following the design and security arguments of Rückert [73] being insecure. Hauck et al. [50] also introduced a new blind signature from linear hash functions [49] but it is impractical. Lyubashevsky et al. [64], del Pino-Katsumata [69], and Agrawal et al. [2] respectively proposed efficient two-round lattice-based blind signatures, which was further improved by Beullens et al. [16] with a two-round blind signature from standard lattice problems with signature size around 22 KB.

In another context, Petzoldt et al. [67] constructed a blind signature from multivariate quadratic equations. However, it has been recently broken by Beullens [12]. Blazy et al. [20] proposed a blind signature from codes but it had a flaw in the security proof, which was later fixed [19].

At CRYPTO 2023, Katsumata et al. [57] proposed the first construction of a (partially) blind signature from isogenies. Kuchta, LeGrow and Persichetti proposed a construction of blind signatures from matrix code equivalence [61] following the framework in [57], with a focus on matrix code equivalence. To resolve issues caused by non-commutativity, the authors of [61] make use of the actions of both A and its inverse transpose A^{-T} for an invertible matrix A, and require A to be (anti)symmetric. The security of the scheme relies on the hardness assumption of the Modified Inverse Matrix Code Equivalence Problem (MIMCE), which is a variant of the Inverse Matrix Code Equivalence Problem (IMCE). However, there have been some recent attacks [32, 45] on MIMCE, breaking the claimed bit security of the blind signature scheme proposed in [61]. Separately, Katsumata et al. [58] and Do et al. [38] independently propose a polynomial-time attack against the l-concurrent unforgeability of Schnorr-type blind signatures. Hence such attack is applicable to both [61] and our scheme in this paper, which means the security of our construction is limited to the sequential setting.

More recently, Hanzlik et al. [48] proposed a novel framework for efficient blind signatures from group actions which also achieves polynomially concurrent security. Their frameworks can be instantiated in both commutative and non-commutative group action settings. In a following work [62], Lai and Persichetti provided a compact and efficient instantiation of Tanuki from code equivalence. However, we note that neither [61] nor [48] discusses whether or how their blind signature can be made partially blind, so it is still open to construct a partially blind signature from non-commutative group actions, which we will resolve in this paper.

In this paper, we generalize those results in [57] to abstract group actions. Our framework is applicable for general (non-commutative) group actions, with security based on assumptions such as in Definition 9. One requirement for instantiating our scheme is the reuse of secret keys, which leads us to propose the use of symmetric group action to equivalence classes of linear codes under the action of the general linear group and the diagonal group. This viewpoint helps to thwart the attack of reusing keys as in [23, 25]; see Section 6 for the details.

A natural question is whether one can simply add tags to obtain PBS. We do not think this is immediate. Binding the agreed public information info into the

Scheme	Size			Computational Cost		
	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{pk}\rvert$	$\lvert\sigma\rvert$	Signer	User	Verifier
CSI-Otter [57]	$\mathsf{size}(G)+1$	$2\cdot\mathsf{size}(S)$	$2n\cdot\mathsf{size}(S)+4n$	$2n$	$4n$	$2n$
CSI-Otter (Partially Blind) [57]	$\mathsf{size}(G)+1$	$2\cdot\mathsf{size}(S)$	$6n\cdot\mathsf{size}(S)+6n$	$6n$	$13n$	$6n$
Tanuki [48]	$\mathsf{size}(G)$	$\mathsf{size}(S)$	$n\cdot\mathsf{size}(S)+n\cdot\mathsf{size}(G)$	n	$2n$	n
Our Blind Signature	$\mathsf{size}(G)+1$	$4\cdot\mathsf{size}(S)$	$4n\cdot\mathsf{size}(S)+2n$	$4n$	$8n$	$4n$
Our Partially Blind Signature	$\mathsf{size}(G)+1$	$6\cdot\mathsf{size}(S)$	$6n\cdot\mathsf{size}(S)+2n$	$6n$	$12n$	$6n$

Table 1: Efficiency comparison of blind and partially blind signature schemes. Here, $\mathsf{size}(G)$ and $\mathsf{size}(S)$ denote the size of a single element in the acting group G and the acted set S, respectively, and n is the challenge dimension (security parameter). All computational costs are measured in terms of the number of group actions. The (partially) blind signature schemes of CSI-Otter [57] correspond to the basic versions without optimization, while Tanuki [48] corresponds to Framework 1 with log-concurrent and sequential one-more unforgeability.

verification relation changes what is committed and hashed, so one would need to re-establish the transcript-balance and indistinguishability arguments used in Tanuki. Moreover, Tanuki's concurrent-security proofs are sensitive to random-oracle programming and transcript distributions across many interleavings, so making the relation tag-dependent appears to require revisiting the concurrent simulation and extraction arguments. Thus, concurrently secure PBS from group actions in the sense of Tanuki seems to be a non-trivial extension.

A related question is whether our CSI-Otter-like framework can be strengthened to achieve concurrent security in the sense of Tanuki [48]. Our setting has an additional constraint: the challenge space must carry a compatible group structure, since the proof relies on algebraic manipulations over challenges. Since Tanuki does not require this, adapting its concurrency techniques to our setting does not seem straightforward and would likely require substantial redesign. We therefore view Tanuki as the appropriate reference for stronger concurrent-security guarantees, while our work focuses on a simpler CSI-Otter-like construction tailored to our setting.

In Table 1, we provide a comparison between our construction with CSI-Otter [57] and Tanuki [48]. Since we do not have yet the concrete parameters for the modified group actions, we hence provide the comparison in terms of the size of the group and the set as well as the number of group actions.

2 Preliminaries

2.1 Notations

For a prime power q, let $\mathbb{F}_q$ be the field consisting of q elements. Denote by $\mathbb{F}_q^n$ the linear space of length-n row vectors over $\mathbb{F}_q$. Denote by $\mathrm{Mat}(m \times n, q)$ the linear space of $m \times n$ matrices over $\mathbb{F}_q$, and $\mathrm{Mat}(n, q) := \mathrm{Mat}(n \times n, q)$. We use $\mathrm{GL}(n, q)$ to denote the group of $n \times n$ invertible matrices over $\mathbb{F}_q$, and $\mathrm{D}(n, q)$ to denote the group of $n \times n$ invertible diagonal matrices over $\mathbb{F}_q$. The symmetric group on $\{1, \ldots, n\}$ is denoted by S_n. By encoding each $\sigma \in \mathrm{S}_n$ as a $n \times n$ permutation matrix over $\mathbb{F}_q$, we can embed S_n as a subgroup of $\mathrm{GL}(n, q)$. A matrix in $\mathrm{Mat}(n, q)$ is said to be monomial, if it is the product between a diagonal and a permutation matrix. The group of monomial matrices is denoted by $\mathrm{Mon}(n, q)$.

For a positive integer k, we denote $[k]$ to be the set $\{1, \ldots, k\}$. For integers a, b we denote by $[a : b]$ the set $\{a, a + 1, \ldots, b\}$. For a vector $\overrightarrow{h}$, denote by h_i the i-th entry of $\overrightarrow{h}$. We will also denote a vector by bold characters, e.g., $\mathbf{h}$. For a finite set S, we write $x \leftarrow\!\$\ S$ to denote x is sampled randomly from S. We use $\odot$ to denote the component-wise multiplication of vectors in $\mathbb{R}$. In particular, for $c \in \mathbb{R}$ and vectors $\mathbf{a} = (a_1, \cdots, a_k), \mathbf{b} = (b_1, \ldots, b_k)$, we write $c \odot \mathbf{a}$ for $(ca_1, \ldots, ca_k)$ and $\mathbf{a} \odot \mathbf{b} = (a_1 b_1, \ldots, a_k b_k)$. If $\mathbf{g}, \mathbf{h}$ are group element vectors, we also write $\mathbf{gh}$ for $(g_1 h_1, \ldots, g_k h_k)$. We also extend this component-wise notation for exponentiation, e.g., we write $\mathbf{a}^c$ for $(a_1^c, \ldots, a_k^c)$, $\mathbf{a}^{\mathbf{b}}$ for $(a_1^{b_1}, \ldots, a_k^{b_k})$, and for group action, e.g., we write the action of vector $\mathbf{a}$ on $\mathbf{s} \in S^k$ as $\mathbf{a} * \mathbf{s}$ for $(a_1 * s_1, \ldots, a_k * s_k)$ (here $*$ indicates the action operation - see Section 2.3 for group action definition).

2.2 Sigma Protocols

Definition 1 (Sigma Protocol). *A sigma protocol for an NP relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is a public-coin three-move interactive protocol between a prover $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ and a verifier $\mathcal{V}$ as follows.*

- *The prover on input a statement X and a witness W such that $(\mathsf{X}, \mathsf{W}) \in R$, runs $(\mathsf{com}, \mathsf{state}) \leftarrow \mathcal{P}_1(\mathsf{X}, \mathsf{W})$ and sents a commitment com to the verifier.*
- *The verifier samples a random challenge $\mathsf{ch} \leftarrow\!\$\ \mathcal{C}$ from the challenge space $\mathcal{C}$ and sends ch to the prover.*
- *Upon receiving the challenge ch, the prover $\mathcal{P}_2$ generates a response rsp and sents rsp to the verifier.*
- *The verifier runs $\mathcal{V}(\mathsf{X}, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ and outputs 1 to indicate acceptance, and 0 otherwise.*

A sigma protocol must satisfy correctness, honest-verifier zero-knowledge (HVZK), and special soundness defined as the following.

Correctness. It is required that if the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ follow the sigma protocol honestly, then the verifier would output 1 with probability 1.

Honest Verifier Zero-Knowledge (HVZK). There exists a PPT simulator Sim that given a statement X, a challenge $\mathsf{ch} \in \mathcal{C}$, outputs a valid transcipt $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ that is indistinguishable from a real transcript.

Special Soundness. There exists a determinsitic polynomial time extractor Ext that given two accepted transcripts $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ and $(\mathsf{com}, \mathsf{ch}', \mathsf{rsp}')$ with the same commitment com and different challenges $\mathsf{ch} \neq \mathsf{ch}'$, outputs W such that $(\mathsf{X}, \mathsf{W}) \in R$.

We also provide a definition for a hard instance generator for the NP relation R as follows.

Definition 2 (Hard Instance Generator). *An NP relation R is associated with an instance generator (IG) if IG, given as input the security parameter 1^n, outputs a statement-witness pair $(\mathsf{X}, \mathsf{W}) \in R$. Moreover, we say that the instance generator is hard if the following holds for any PPT adversary $\mathcal{A}$:*

$$\Pr[(\mathsf{X}, \mathsf{W}) \leftarrow \mathsf{IG}(1^n), \mathsf{W}' \leftarrow \mathcal{A}(\mathsf{X}) : (\mathsf{X}, \mathsf{W}') \in R] = \mathrm{negl}(n).$$

2.3 Cryptographic Group Actions

Let G be a group and S a set. An action of G on S is a map $* : G \times S \to S$ satisfying the following properties: (i) $\mathrm{id} * s = s$ for all $s \in S$ and the identity element $\mathrm{id} \in G$; and (ii) $g * (h * s) = gh * s$ for all $g, h \in G$ and $s \in S$. A group action is said to be [3]:

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

Given a group action $*$ of G on S, the orbit of an element $s \in S$ is defined as $\mathsf{Orb}(s) := \{g * s : \forall g \in G\}$. Note that if the group action is transitive, then $\mathsf{Orb}(s) = S$. The stabilizer of s is defined by $\mathsf{Stab}(s) := \{g \in G : g * s = s)\}$ which is a subgroup of G. The Orbit-Stabilizer theorem says that, if G is finite then $|G| = |\mathsf{Stab}(s)| \cdot |\mathsf{Orb}(s)|$.

In this paper, we shall mostly consider finite groups acting on finite sets. To use group actions in algorithms, we assume that group and set elements have natural encodings, as well as group operations, group actions, and random samplings of group and set elements can be efficiently computed; see [3, 22, 54] for more details and certain variations. In particular, we assume that uniform random samplings from the group G and the set S are efficient.

A group action is *one-way*, if for a random s, the function $f_s : G \to S$ defined by $f_s(g) := g * s$ is one-way. The one-way assumption is formulated as the *Group Action Inverse Problem* (GAIP) defined in the following.

Definition 3 (Group Action Inverse Problem (GAIP)). *Given a group action $* : G \times S \to S$, uniformly random $s \in S$, and uniformly random $t \in \mathsf{Orb}(s)$, find $g \in G$ such that $g * s = t$.*

Here we restrict to the case of transitive group actions, as in the isogeny-based setting [27, 35, 42], or we can restrict the element g to be in the orbit $\mathsf{Orb(s)}$ of s as in the case of non-commutative group actions [17, 33, 74].

For the purpose of our paper, we define what we call the *Inverse Group Action Problem* (IGAP), as follows.

Definition 4 (Inverse Group Action Problem (IGAP)). *Given a group action $* : G \times S \to S$, uniformly random $s \in S$, and a pair $(g * s, g^{-1} * s)$, find g.*

IGAP was called the *Inverse Linear Code Equivalence* (ILCE) problem [7] in the context of linear code equivalence underlying LESS, and called the *Inverse Matrix Code Equivalence* (IMCE) problem [33] in the context of matrix code equivalence underlying MEDS. Recently, [23, 25] introduced heuristic efficient algorithms for the ILCE problem, while it remains unclear whether similar efficient algorithms exist for the IMCE problem.

Note that the lattice isomorphism problems used in HAWK [39] and the alternating trilinear form equivalence problems used in ALTEQ [74] also admit formulations as group actions and corresponding IGAP. However, [24] gave a heuristic efficient algorithm to solve the corresponding IGAP for lattice isomorphism problems. The hardness of the corresponding IGAP for the alternating trilinear form equivalence problems is still unclear. We will discuss this again to justify our choice of instantiation in Section 6.1.

In this paper, we provide a generic framework of a blind signature for abstract group actions. For the framework to work, an instantiated group G needs to satisfy the following assumptions.

Assumption 1 (Square-root assumption)**.** Given an element $h \in G$ promised to be a square (i.e., $h = g^2$ for $g \in G$), there exists an efficient algorithm that outputs all $g \in G$ such that $g^2 = h$.

Assumption 1 is needed in the soundness proof of the underlying protocol. In the extraction argument, given two accepting transcripts with the same commitment, our extractor recovers an element of the form g^2, where g is the signer's secret. To complete extraction, we therefore require an efficient procedure that, given g^2, recovers the secret g, as stated in Assumption 1.

We note that square roots are not necessarily unique, i.e., it could be the case that $g^2 = (g')^2$ for some $g \neq g'$. Two standard ways to deal with this are: either requiring the square-root procedure to return all roots, or restricting it to a domain in which square roots are unique. Our instantiation follows the second approach; see [40, Remark 6]. In Section 6.1, we also discuss the applicability of Assumption 1 for MEDS, ALTEQ, LESS, and HAWK.

2.4 (Partially) Blind Signatures

We follow [1, 55–57] to define a three-move blind signature.

Definition 5 (Partially Blind Signature). *A three-move partially blind signature* PBS *with efficient decidable public key space* $\mathcal{PK}$ *consists of the following* PPT *algorithms.*

- $\mathsf{PBS.KGen}(1^n) \to (\mathsf{pk}, \mathsf{sk})$*: On input the security parameter* 1^n*, 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 partially blind signature must satisfy three essential properties: *correctness*, *partial blindness under chosen keys*, and *one-more unforgeability*, as formally defined below.

Definition 6 (Correctness). *A three-move partial blind signature scheme* PBS *is correct if for all public and secret key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PBS.KGen}(1^n)$*, we have*

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

Definition 7 (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* $\mathsf{coin} \leftarrow\!\!\$\, \{0,1\}$ *and runs* $\mathcal{A}$ *on input* 1^n*.*

Online Phase. $\mathcal{A}$ *outputs a tag* info*, two message* M_0^* *and* M_1^**, a public key* $\mathsf{pk} \in \mathcal{PK}$*, the game checks if* pk *is valid and if so, it assigns* $(M_0, M_1) = (M^*_{\mathsf{coin}}, M^*_{1-\mathsf{coin}})$*. If* pk *is not valid, the game aborts and outputs* 0*. The adversary* $\mathcal{A}$ *is given access to oracles* $\mathsf{U}_1, \mathsf{U}_2$ *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* $(\mathsf{state}_{U,b}, \rho_{U,b}) \leftarrow\!\!\$\ \mathsf{PBS.U}_1(\mathsf{pk}, \mathsf{info}, M_b, \rho_{S,1})$*. 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* $\sigma_b \leftarrow \mathsf{PBS.U}_2(\mathsf{state}_{U,b}, \rho_{S,2,b})$*. 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* $\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}, M_b, \sigma_b) = 1$*, the oracle returns the two signatures* $(\sigma_{\mathsf{coin}}, \sigma_{1-\mathsf{coin}})$ *to* $\mathcal{A}$*, where note that* σ_{coin} *(resp.* $\sigma_{1-\mathsf{coin}}$*) 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* $\mathsf{coin}^* = \mathsf{coin}$*.*

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

Definition 8 (One-More Unforgeability). *For a partial blind signature* PBS *and* $l \in \mathbb{N}$*, we define* l*-one-more unforgeability via the following game between a challenger and an adversary* $\mathcal{A}$*:*

Setup. *The challenger samples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PBS.KGen}(1^n)$ *and runs* $\mathcal{A}$ *on input* pk*. It initializes* $l_{\mathsf{closed}} = 0$ *and* $\mathsf{opened}_{\mathsf{sid}} = \mathsf{false}$ *for all* $\mathsf{sid} \in \mathbb{N}$*.*

Online Phase. $\mathcal{A}$ *is given access to two oracle* S_1 *and* S_2 *as follows.*
– **Oracle** S_1*: The oracle samples a fresh session identifier* sid*. It sets* $\mathsf{opened}_{\mathsf{sid}} \leftarrow \mathsf{true}$ *and generates* $(\mathsf{state}_{S,\mathsf{sid}}, \rho_{S,1}) \leftarrow \mathsf{PBS.S}_1(\mathsf{sk}, \mathsf{info})$*. It then returns* sid *and the first-sender message* $\rho_{S,1}$ *to* $\mathcal{A}$*.*
– **Oracle** S_2*: On input a user message* ρ_U *and a session identifier* sid*, if* $l_{\mathsf{closed}} \geq l$ *or* $\mathsf{opened}_{\mathsf{sid}} = \mathsf{false}$*, then it returns* $\perp$*. Otherwise, it increments* l_{closed} *and* $\mathsf{opened}_{\mathsf{sid}} = \mathsf{false}$*. It then computes the second-signer message* $\rho_{S,2} \leftarrow \mathsf{PBS.S}_2(\mathsf{state}_{S,\mathsf{sid}}, \rho_U)$ *and returns* $\rho_{S,2}$ *to* $\mathcal{A}$*.*

Output Determination. *When* $\mathcal{A}$ *outputs distinct tuples of message-signature pairs* $(M_1, \sigma_1, \mathsf{info}), \ldots, (M_k, \sigma_k, \mathsf{info})$*, we say that* $\mathcal{A}$ *wins if* $k \geq l_{\mathsf{closed}} + 1$ *and for all* $i \in [k]$*,* $\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}, M_i, \sigma_i) = 1$*.*

We say that the blind signature PBS *is* l*-one-more unforgeable if the probability that* $\mathcal{A}$ *wins is negligible.*

3 Sigma Protocol for Validating Public Key

Unlike the isogeny setting of CSI-Otter [57], public-key validity is not immediate for generic group actions. In this section, we give a sigma protocol for proving that a public key is well formed, namely for the relation

$$R = \{(\mathsf{X} = (A^{(1)}, A^{(-1)}), \mathsf{W} = g) \mid A^{(b)} = g^b * E,\ \forall b \in \{-1, 1\}\}. \tag{1}$$

Here, G acts on S and $E \in S$ is fixed. The protocol is shown in Fig. 1. It is a variant of the GMW-type protocol for generic group actions; see, e.g., [18]. Proofs are given in [40, Appendix C].

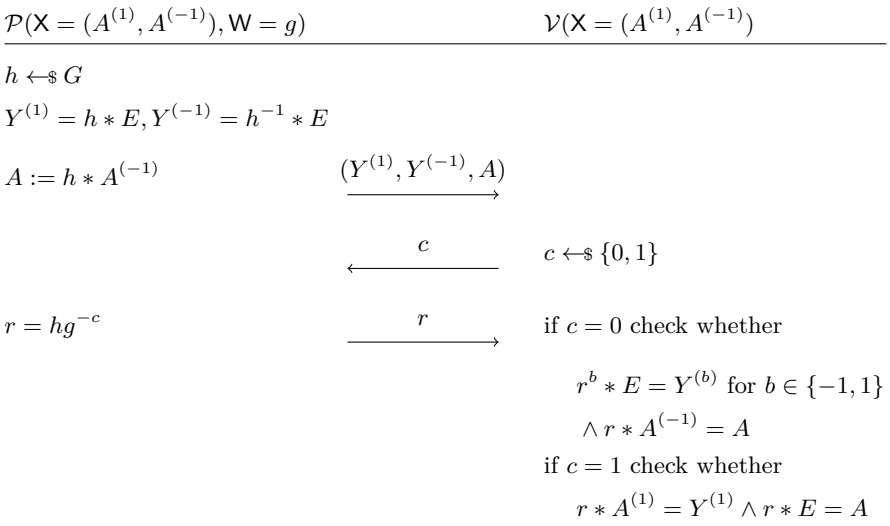

Fig. 1: Sigma Protocol for Validating Public Key

For this protocol, we assume the hardness of the following *Group Action Strong Decisional Diffie–Hellman* (GASDDH) problem, analogous to the strong decisional Diffie–Hellman assumption in [68, Assumption 2].

Definition 9 (Group Action Strong Decisional Diffie–Hellman (GASDDH)). *Given a group action $* : G \times S \to S$ and $s \in S$, distinguish the distributions between $(s,\ g * s,\ g^{-1} * s,\ h * s,\ (hg^{-1}) * s,\ h^{-1} * s)$ and $(s,\ \alpha(g, s),\ \alpha(g^{-1}, s),\ \alpha(h, s),\ \alpha(hg^{-1}, s),\ \alpha(f, s))$ for uniformly random $g, h, f \in G$.*

If no PPT adversary can distinguish the two distributions above, then the simulated and real transcripts for $c = 1$ are clearly indistinguishable as well. In particular, this problem is at least as hard as the following problem of *Group Action Inverse Decisional Diffie-Hellman* (GAIDDH), in analogy to the inverse decisional Diffie-Hellman assumption first studied in [68].

Definition 10 (Group Action Inverse Decisional Diffie–Hellman (GAIDDH)). *Given a group action $* : G \times S \to S$ and $s \in S$, distinguish the distributions between $(s,\ h * s,\ h^{-1} * s)$ and $(s,\ h * s,\ f * s)$ for uniformly random $h, f \in G$.*

We observe that there is a worst-case reduction from the problem of GAIDDH to the problem of GASDDH. Indeed, suppose we have a PPT algorithm to solve GASDDH, then given a group action $\alpha : G \times S \to S$, $s \in S$, and two distributions $(s,\ \alpha(h, s),\ \alpha(h^{-1}, s))$ and $(s,\ \alpha(h, s),\ \alpha(f, s))$ for some uniformly random $h, f \in G$, we can treat it as an instance applicable to the algorithm for GASDDH by taking $g = \mathrm{id} = g^{-1}$, where id is the identity element in G. It follows that GAIDDH can also be solved efficiently in this case.

4 Our Blind Signature Scheme

In this section, we present our blind-signature construction for generic group actions, following the CSI-Otter-style framework of Katsumata et al. [57]. Our construction is obtained by applying the blinding transformation to the underlying OR sigma protocol (see [40, Fig. 7]); the full proofs are given in [40, Appendix E]. Let G be a group acting on a set S, and fix an element $E \in S$. We assume that square roots in G can be computed efficiently. Let $H : \{0,1\}^* \to \{-1,1\}^n$ be a hash function modeled as a random oracle. The blind signature BS consists of the following algorithms, which are summarized in Fig. 2.

$\mathsf{BS.KGen}(1^n)$: On input the security parameter 1^n, it samples a bit $\delta \in \{0,1\}$, $(g_0, g_1) \in G^2$ and computes $A_b^{(c)} = g_b^c * E$ for $b \in \{0,1\}$ and $c \in \{-1,1\}$. It outputs a public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$ and a secret key $\mathsf{sk} = (\delta, g_\delta)$.

Before any signing interaction starts, the public key is verified once at setup/registration time using the sigma protocol in Fig. 1; accordingly, the online algorithms are defined only for valid public keys.

$\mathsf{BS.S}_1(\mathsf{sk})$: On input the secret key $\mathsf{sk} = (\delta, g_\delta)$, the signer first samples $\mathbf{h}_\delta \leftarrow\!\$\ G^n$, and sets $\mathbf{Y}_\delta = \{\mathbf{h}_{\delta,k}^c * E\}$ for $c \in \{-1,1\}, k \in [n]$. Here given a vector $\mathbf{c} = (c_1, \ldots, c_n) \in \{-1,1\}^n$, we write $\mathbf{Y}_\delta^{(\mathbf{c})}$ for the vector $(\mathbf{h}_{\delta,k}^{c_k} * E)_{k\in[n]}$. We will extend this convenience accordingly in the scheme.

Then it samples $(\mathbf{c}^*_{1-\delta}, \mathbf{r}^{*(1)}_{1-\delta}, \mathbf{r}^{*(-1)}_{1-\delta}) \leftarrow\!\$\ \{-1,1\}^n \times G^n \times G^n$ and set $\mathbf{r}^*_{1-\delta} := \{\mathbf{r}^{*(c)}_{1-\delta,k}\}_{c\in\{-1,1\},k\in[n]}$. Next it computes $\mathbf{Y}_{1-\delta} = \{\mathbf{r}^{*(c)}_{1-\delta,k} * A_{1-\delta}^{(c\odot\mathbf{c}^*_{1-\delta,k})}\}_{c\in\{-1,1\},k\in[n]}$. It outputs the signer state $\mathsf{state}_S = (\mathbf{h}_\delta, \mathbf{c}^*_{1-\delta}, \mathbf{r}^*_{1-\delta})$ and the first-sender message $\rho_{S,1} = (\mathbf{Y}_0, \mathbf{Y}_1)$.

$\mathsf{BS.U}_1(\mathsf{pk}, M, \rho_{S,1})$: On input the public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$, a message M, and the first-sender message $\rho_{S,1} = (\mathbf{Y}_0, \mathbf{Y}_1)$, it samples, for $b \in \{0,1\}$, $(\mathbf{d}_b, \mathbf{z}_b) \leftarrow\!\$\ \{-1,1\}^n \times G^n$ and sets $\mathbf{Z}_b = \mathbf{z}_b * \mathbf{Y}_b^{(\mathbf{d}_b)}$. Then it computes $\mathbf{c} = H(\mathbf{Z}_0 \| \mathbf{Z}_1 \| M)$ and sets $\mathbf{c}^* = \mathbf{c} \odot \mathbf{d}_0 \odot \mathbf{d}_1 \in \{-1,1\}^n$. It outputs the internal user state $\mathsf{state}_U = (\mathbf{d}_b, \mathbf{z}_b)_{b\in\{0,1\}}$ and a user message $\rho_U = \mathbf{c}^*$.

$\mathsf{BS.S}_2(\mathsf{state}_S, \rho_U)$: On input the internal state $\rho_S = (\mathbf{h}_\delta, \mathbf{c}^*_{1-\delta}, \mathbf{r}^*_{1-\delta})$ and a user message $\rho_U = \mathbf{c}^*$, it computes $\mathbf{c}^*_\delta = \mathbf{c}^* \odot \mathbf{c}^*_{1-\delta} \in \{-1,1\}^n$ and $\mathbf{r}^*_\delta = \{\mathbf{h}_{\delta,k}^c g_\delta^{-c\odot\mathbf{c}^*_{\delta,k}}\}_{c\in\{-1,1\},k\in[n]}$. It outputs the second-sender message $\rho_{S,2} = (\mathbf{c}^*_b, \mathbf{r}^*_b)_{b\in\{0,1\}}$.

$\mathsf{BS.U}_2(\mathsf{state}_U, \rho_{S,2})$: On input the internal user state $\mathsf{state}_U = (\mathbf{d}_b, \mathbf{z}_b)_{b\in\{0,1\}})$ and $\rho_{S,2} = (\mathbf{c}^*_b, \mathbf{r}^*_b)_{b\in\{0,1\}}$, it sets $\mathbf{c}_b = \mathbf{c}^*_b \odot \mathbf{d}_b, \mathbf{r}_b^{(\mathbf{d}_b)} = \mathbf{z}_b(\mathbf{r}_b^{*(\mathbf{d}_b)})$ for $b \in \{0,1\}$. Then it checks if

$$\mathbf{c}_0 \odot \mathbf{c}_1 = H(\mathbf{r}_0^{(\mathbf{d}_0)} * \mathbf{A}_0^{(\mathbf{c}_0)} \| \mathbf{r}_1^{(\mathbf{d}_1)} * A_1^{(\mathbf{c}_1)} \| M). \tag{2}$$

If it holds then it outputs a signature $\sigma = (\mathbf{c}_b, \mathbf{r}_b^{(1)}, \mathbf{r}_b^{(-1)})_{b\in\{0,1\}}$.

$\mathsf{BS.KGen}(1^n)$

101 : $\delta \leftarrow\!\!\$\ \{0,1\}$
102 : $(g_0, g_1) \leftarrow\!\!\$\ G^2$
103 : $(A_0^{(1)}, A_0^{(-1)}) \leftarrow (g_0 * E, g_0^{-1} * E)$
104 : $(A_1^{(1)}, A_1^{(-1)}) \leftarrow (g_1 * E, g_1^{-1} * E)$
105 : **return** $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$
106 : $\mathsf{sk} = (\delta, g_\delta)$

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

201 : **parse** $(\delta, g_\delta) \leftarrow \mathsf{sk}$
202 : $\mathbf{h}_\delta \leftarrow\!\!\$\ G^n$
203 : $\mathbf{Y}_\delta = \{\mathbf{h}_{\delta,k}^c * E\}_{c\in\{-1,1\},k\in[n]}$
204 : $(\mathbf{c}^*_{1-\delta}, \mathbf{r}^{*(1)}_{1-\delta}, \mathbf{r}^{*(-1)}_{1-\delta}) \leftarrow\!\!\$\ \{-1,1\}^n \times G^n \times G^n$
205 : set $\mathbf{r}^*_{1-\delta} := \{\mathbf{r}^{*(c)}_{1-\delta,k}\}_{c\in\{-1,1\},k\in[n]}$
206 : $\mathbf{Y}_{1-\delta} = \{\mathbf{r}^{*(c)}_{1-\delta,k} * A_{1-\delta}^{(c\odot\mathbf{c}^*_{1-\delta,k})}\}_{c\in\{-1,1\},k\in[n]}$
207 : $\mathsf{state}_S \leftarrow (\mathbf{h}_\delta, \mathbf{c}^*_{1-\delta}, \mathbf{r}^*_{1-\delta})$
208 : $\rho_{S,1} = (\mathbf{Y}_0, \mathbf{Y}_1)$
209 : **return** $(\mathsf{state}_S, \rho_{S,1})$

$\mathsf{BS.U}_1(\mathsf{pk}, M, \rho_{S,1})$

301 : **parse** $(\mathbf{Y}_0, \mathbf{Y}_1) \leftarrow \rho_{S,1}$
302 : **for** $b \in \{0,1\}$
303 : $\quad(\mathbf{d}_b, \mathbf{z}_b) \leftarrow\!\!\$\ \{-1,1\}^n \times G^n$
304 : $\quad\mathbf{Z}_b^{(1)} \leftarrow \mathbf{z}_b * \mathbf{Y}_b^{(\mathbf{d}_b)}$
305 : $\quad\mathbf{Z}_b^{(-1)} \leftarrow \mathbf{z}_b^{-1} * \mathbf{Y}_b^{(-\mathbf{d}_b)}$
306 : $\mathbf{c} \leftarrow H(\mathbf{Z}_0^{(1)} \| \mathbf{Z}_0^{(-1)} \| \mathbf{Z}_1^{(1)} \| \mathbf{Z}_1^{(-1)} \| M)$
307 : $\mathbf{c}^* \leftarrow \mathbf{c} \odot \mathbf{d}_0 \odot \mathbf{d}_1 \in \{-1,1\}^n$
308 : $\mathsf{state}_U \leftarrow (\mathbf{d}_b, \mathbf{z}_b)_{b\in\{0,1\}}$
309 : **return** $(\mathsf{state}_U, \rho_U = \mathbf{c}^*)$

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

401 : **parse** $(\mathbf{h}_\delta, \mathbf{c}^*_{1-\delta}, \mathbf{r}^*_{1-\delta}) \leftarrow \mathsf{state}_S$
402 : **parse** $\mathbf{c}^* \leftarrow \rho_U$
403 : $\mathbf{c}^*_\delta \leftarrow \mathbf{c}^* \odot \mathbf{c}^*_{1-\delta} \in \{-1,1\}^n$
404 : $\mathbf{r}^*_\delta = \{\mathbf{h}_{\delta,k}^c g_\delta^{-c\cdot\mathbf{c}^*_{\delta,k}}\}_{c\in\{-1,1\},k\in[n]}$
405 : **return** $\rho_{S,2} = (\mathbf{c}^*_b, \mathbf{r}^*_b)_{b\in\{0,1\}}$

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

501 : **parse** $(\mathbf{d}_b, \mathbf{z}_b)_{b\in\{0,1\}} \leftarrow \mathsf{state}_U$
502 : **parse** $(\mathbf{c}^*_b, \mathbf{r}^*_b)_{b\in\{0,1\}} \leftarrow \rho_{S,2}$
503 : **for** $b \in \{0,1\}$
504 : $\quad\mathbf{c}_b \leftarrow \mathbf{c}^*_b \odot \mathbf{d}_b$
505 : $\quad\mathbf{r}_b^{(1)} \leftarrow \mathbf{z}_b(\mathbf{r}_b^{*(\mathbf{d}_b)})$
506 : $\quad\mathbf{r}_b^{(-1)} \leftarrow \mathbf{z}_b^{-1}(\mathbf{r}_b^{*(-\mathbf{d}_b)})$
507 : $\mathbf{c}' \leftarrow H(\mathbf{r}_0^{(1)} * A_0^{(\mathbf{c}_0)} \| \mathbf{r}_0^{(-1)} * A_0^{(-\mathbf{c}_0)} \|$
508 : $\qquad\mathbf{r}_1^{(1)} * A_1^{(\mathbf{c}_1)} \| \mathbf{r}_1^{(-1)} * A_1^{(-\mathbf{c}_1)} \| M)$
509 : **if** $\mathbf{c}_0 \odot \mathbf{c}_1 = \mathbf{c}'$
510 : $\quad$**return** $\sigma = (\mathbf{c}_b, \mathbf{r}_b^{(1)}, \mathbf{r}_b^{(-1)})_{b\in\{0,1\}}$
511 : **return** $\sigma = \perp$

$\mathsf{BS.Verify}(\mathsf{pk}, M, \sigma)$

601 : **parse** $(\mathbf{c}_b, \mathbf{r}_b^{(1)}, \mathbf{r}_b^{(-1)})_{b\in\{0,1\}} \leftarrow \sigma$
602 : $\mathbf{c}' \leftarrow H(\mathbf{r}_0^{(1)} * A_0^{(\mathbf{c}_0)} \| \mathbf{r}_0^{(-1)} * A_0^{(-\mathbf{c}_0)} \|$
603 : $\qquad\mathbf{r}_1^{(1)} * A_1^{(\mathbf{c}_1)} \| \mathbf{r}_1^{(-1)} * A_1^{(-\mathbf{c}_1)} \| M)$
604 : **if** $\mathbf{c}_0 \odot \mathbf{c}_1 = \mathbf{c}'$
605 : $\quad$**return** 1
606 : **return** 0

Fig. 2: Blind Signature from Cryptographic Group Action. Here for example, given a vector $\mathbf{c} = (c_1, \ldots, c_n) \in \{-1,1\}^n$, we write $\mathbf{Y}_\delta^{(\mathbf{c})}$ for the vector $(\mathbf{h}_{\delta,k}^{c_k} * E)_{k\in[n]}$, and extend to others similarly to $\mathbf{Y}_{1-\delta}, \mathbf{r}^*_\delta, \mathbf{r}^*_{1-\delta}$ accordingly.

$\mathsf{BS.Verify}(\mathsf{pk}, M, \sigma)$: On input the public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$, a message M and a signature $\sigma = (\mathbf{c}_b, \mathbf{r}_b^{(1)}, \mathbf{r}_b^{(-1)})_{b\in\{0,1\}}$, it outputs 1 if the equation (2) holds, and 0 otherwise.

Due to page limit, we present the proofs in [40, Appendix E]. In order to prove the one-more unforgeability, we use the techniques developed in [56] and [57] that we summarize in [40, Appendix D]. We summarize the main result in the following Theorem.

Theorem 1. *Our blind signature is correct and blind under chosen keys. In addition, our blind signature satisfies the one-more unforgeability property under Assumption 1 and the hardness of IGAP problem. To be more precise, for $l \in \mathbb{N}$, if there exist an adversary $\mathcal{A}$ that makes Q hash queries to the random oracle and breaks the l-one-more unforgeability of BS with advantage $\epsilon_{\mathcal{A}} \geq \frac{C_1}{|\mathcal{C}|}\binom{Q}{l+1}$, then there exists an efficient algorithm $\mathcal{B}$ that breaks the IGAP problem with advantage $\epsilon_{\mathcal{B}} \geq C_2 \cdot \frac{\epsilon_{\mathcal{A}}^2}{\binom{Q}{l+1}^2 \cdot (l+1)^3}$ for some universal positive constants C_1 and C_2.*

5 Our Partially Blind Signature Scheme

In this section, we present our partially blind-signature construction for generic group actions, also following the framework of Katsumata et al. [57]. As in the blind-signature case, this construction is obtained from the corresponding base sigma protocol for the 2-out-of-3 relation, which extends the OR-proof idea used above; the full protocol and proofs are given in [40, Appendix F]. Let G be a group acting on a set S, and fix an element $E \in S$. We assume that square roots in G can be computed efficiently. Let $H : \{0,1\}^* \to \{-1,1\}^n$ be a hash function modeled as a random oracle. The partially blind signature PBS consists of the following algorithms:

$\mathsf{PBS.KGen}(1^n)$: On input the security parameter 1^n, it samples a bit $\delta \in \{0,1\}$, $(g_0, g_1) \in G^2$ and computes $A_b^{(c)} = g_b^c * E$ for $b \in \{0,1\}$ and $c \in \{-1,1\}$. It outputs a public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)}, A_2^{(1)}, A_2^{(-1)})$ and a secret key $\mathsf{sk} = (\delta, g_\delta)$.
Before any signing interaction starts, the public key is verified once at setup/registration time using the sigma protocol in Fig. 1; accordingly, the online algorithms are defined only for valid public keys.

$\mathsf{PBS.S}_1(\mathsf{sk}, \mathsf{info})$: On input the secret key $\mathsf{sk} = (\delta, g_\delta)$ and a tag info, for each $j \in \{0,1\}$, the signer first samples $(\mathbf{h}_{\delta,j}, \mathbf{h}_{2,j}) \leftarrow\!\$\ G^{2n}$. Then, it sets $\mathbf{Y}_{\delta,j} = \{\mathbf{h}_{\delta,j,k}^c * E\}$ for $c \in \{-1,1\}, k \in [n]$ and $\mathbf{Y}_{2,j} = \{\mathbf{h}_{2,j,k}^c * E\}$ for $c \in \{-1,1\}, k \in [n]$. Given a vector $\mathbf{c} = (c_1, \ldots, c_n) \in \{-1,1\}^n$, we write $\mathbf{Y}_{\delta,j}^{(\mathbf{c})}, \mathbf{Y}_{2,j}^{(\mathbf{c})}$ for the vectors $(\mathbf{h}_{\delta,j,k}^{c_k} * E)_{k\in[n]}$ and $(\mathbf{h}_{2,j,k}^{c_k} * E)_{k\in[n]}$. We will extend this convenience accordingly in the scheme. Then it samples $(\mathbf{c}^*_{[1-\delta+j]_3}, \mathbf{r}^{*(1)}_{1-\delta,j}, \mathbf{r}^{*(-1)}_{1-\delta,j}) \leftarrow\!\$\ \{-1,1\}^n \times G^n \times G^n$ and set $\mathbf{r}^*_{1-\delta,j} := \{\mathbf{r}^{*(c)}_{1-\delta,j,k}\}_{c\in\{-1,1\},k\in[n]}$. Next it computes $\mathbf{Y}_{1-\delta,j} = \{\mathbf{r}^{*(c)}_{1-\delta,j,k} * A_{1-\delta}^{(c\odot\mathbf{c}^*_{[1-\delta+j]_3,k})}\}_{c\in\{-1,1\},k\in[n]}$. It outputs the signer

state $\mathsf{state}_S = (\mathbf{h}_{\delta,j}, \mathbf{c}^*_{[1-\delta+j]_3}, \mathbf{r}^*_{1-\delta,j})$ for each $j \in \{0,1\}$ and the first-sender message $\rho_{S,1} = (\mathbf{Y}_{k,j})$ for all $(k,j) \in [0:2] \times \{0,1\}$.

$\mathsf{PBS.U}_1(\mathsf{pk}, \mathsf{info}, M, \rho_{S,1})$: On input the public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$, a tag info, a message M, and the first-sender message $\rho_{S,1} = (\mathbf{Y}_{k,j})$ for all $(k,j) \in [0:2] \times \{0,1\}$, it samples $\mathbf{d}_k \leftarrow\!\$\ \{-1,1\}^n$ for $k \in [0:2]$. Then, for each $j \in \{0,1\}$, it samples $\mathbf{z}_{k,j} \leftarrow\!\$\ G^n$ and sets $\mathbf{Z}^{(1)}_{k,j} = \mathbf{z}_{k,j} * \mathbf{Y}^{(\mathbf{d}_{[k+j]_3})}_{k,j}$ and $\mathbf{Z}^{(-1)}_{k,j} = \mathbf{z}^{-1}_{k,j} * \mathbf{Y}^{(-\mathbf{d}_{[k+j]_3})}_{k,j}$. Next, it computes $\mathbf{c} = H((\mathbf{Z}^{(1)}_{k,j} \| \mathbf{Z}^{(-1)}_{k,j})_{(k,j)\in[0:2]\times\{0,1\}} \| \mathsf{info} \| M)$ and sets $\mathbf{c}^* = \mathbf{c} \odot \mathbf{d}_0 \odot \mathbf{d}_1 \odot \mathbf{d}_2 \in \{-1,1\}^n$. It outputs the internal user state $\mathsf{state}_U = (\mathbf{d}_k, (\mathbf{z}_{k,j})_{j\in\{0,1\}})_{k\in[0:2]}$ and a user message $\rho_U = \mathbf{c}^*$.

$\mathsf{PBS.S}_2(\mathsf{state}_S, \rho_U)$: On input the internal state $\mathsf{state}_S = (\mathbf{h}_{\delta,j}, \mathbf{c}^*_{[1-\delta+j]_3}, \mathbf{r}^*_{1-\delta,j})$ for each $j \in \{0,1\}$ and a user message $\rho_U = \mathbf{c}^*$, it computes $\mathbf{c}^*_{[3-\delta]_3} = \mathbf{c}^* \odot \mathbf{c}^*_{[1-\delta]_3} \odot \mathbf{c}^*_{[2-\delta]_3} \in \{-1,1\}^n$. Next, for each $j \in \{0,1\}$, it computes $\mathbf{r}^*_{\delta,j} = \{\mathbf{h}^c_{\delta,j,k} g_\delta^{-c\odot\mathbf{c}^*_{[\delta+j]_3,k}}\}_{c\in\{-1,1\},k\in[n]}$ and $\mathbf{r}^*_{2,j} = \{\mathbf{h}^c_{2,j,k} g_2^{-c\odot\mathbf{c}^*_{[2+j]_3,k}}\}_{c\in\{-1,1\},k\in[n]}$. It outputs the second-sender message $\rho_{S,2} = (\mathbf{c}^*_k, (\mathbf{r}^*_{k,j})_{j\in\{0,1\}})_{kk\in[0:2]}$.

$\mathsf{PBS.U}_2(\mathsf{state}_U, \rho_{S,2})$: On input the internal user state $\mathsf{state}_U = (\mathbf{d}_k, (\mathbf{z}_{k,j})_{j\in\{0,1\}})_{k\in[0:2]}$ and $\rho_{S,2} = (\mathbf{c}^*_k, (\mathbf{r}^*_{k,j})_{j\in\{0,1\}})_{k\in[0:2]}$, it computes $g_2 = \mathsf{G}(\mathsf{info})$. Next, for each $k \in [0:2]$, it sets $\mathbf{c}_k = \mathbf{c}^*_k \odot \mathbf{d}_k$. Then, for each $j \in \{0,1\}$, it computes $\mathbf{r}^{(1)}_{k,j} = \mathbf{z}_{k,j}(\mathbf{r}^{*(\mathbf{d}_{[k+j]_3})}_{k,j})$ and $\mathbf{r}^{(-1)}_{k,j} = \mathbf{z}^{-1}_{k,j}(\mathbf{r}^{*(-\mathbf{d}_{[k+j]_3})}_{k,j})$. Then it checks if

$$\mathbf{c}_0 \odot \mathbf{c}_1 \odot \mathbf{c}_2 = H((\mathbf{r}^{(1)}_{k,j} * A_k^{(\mathbf{c}_{[k+j]_3})}, \mathbf{r}^{(-1)}_{k,j} * A_k^{(-\mathbf{c}_{[k+j]_3})})_{(k,j)\in[0:2]\times\{0,1\}} \| \mathsf{info} \| M). \tag{3}$$

If it holds then it outputs a signature $\sigma = (\mathbf{c}_k, (\mathbf{r}^{(1)}_{k,j}, \mathbf{r}^{(-1)}_{k,j})_{j\in\{0,1\}})_{k\in[0:2]}$.

$\mathsf{PBS.Verify}(\mathsf{pk}, \mathsf{info}, M, \sigma)$: On input the public key $\mathsf{pk} = (A_0^{(1)}, A_0^{(-1)}, A_1^{(1)}, A_1^{(-1)})$, a tag info, a message M and a signature $\sigma = (\mathbf{c}_k, (\mathbf{r}^{(1)}_{k,j}, \mathbf{r}^{(-1)}_{k,j})_{j\in\{0,1\}})_{k\in[0:2]}$, it outputs 1 if the equation (3) holds, and 0 otherwise.

Due to page limit, we present the proofs in [40, Appendix F]. We summarize the main result in the following Theorem.

Theorem 2. *Our partially blind signature is perfectly correct and perfectly blind under chosen keys. In addition, our partially blind signature satisfies the one-more unforgeability property under Assumption 1 and the hardness of IGAP problem. Concretely, for $l \in \mathbb{N}$, if there exist an adversary $\mathcal{A}$ that makes Q hash queries to the random oracle and breaks the l-one-more unforgeability of PBS with advantage $\epsilon_{\mathcal{A}} \geq \frac{C_1}{|\mathcal{C}|}\binom{Q}{l+1}$, then there exists an efficient algorithm $\mathcal{B}$ that breaks the IGAP problem with advantage $\epsilon_{\mathcal{B}} \geq C_2 \cdot \frac{\epsilon^2_{\mathcal{A}}}{\binom{Q}{l+1}^2 \cdot (l+1)^3}$ for some universal positive constants C_1 and C_2.*

6 Instantiation based on Monomial Code Equivalence

In this section, we present a concrete blind signature protocol based on monomial code equivalence for our (partially) blind signatures. Before delving into the details of our instantiation, we outline the requirements for the underlying group G and explain how they guide our choice of instantiation in this paper.

6.1 Instantiation Requirements

To instantiate the generic framework proposed in Section 5, we need to specify the underlying group G and set S. The basic requirements are that the group operations and element sampling in G are efficient and that the set S is sufficiently large and also allows efficient sampling. Besides, we have three more specific requirements corresponding to the framework, which are:

1. There is an efficient algorithm that can compute all the square roots of any given group element in G.
2. The underlying group action $* : G \times S \to S$ satisfies that IGAP is hard, i.e., given a random $s \in S$, and a pair $(g * s, g^{-1} * s)$, computing g is hard.
3. For public key validation using the sigma protocol in Fig. 1, it is also required that the hardness of GASDDH holds for the underlying group action $* : G \times S \to S$. That is, given a random $s \in S$, it is hard to distinguish the distributions between $(s, g*s, g^{-1}*s, h*s, (hg^{-1})*s, h^{-1}*s)$ and $(s, \alpha(g,s), \alpha(g^{-1},s), \alpha(h,s), \alpha(hg^{-1},s), \alpha(f,s))$ for uniformly random $g, h, f \in G$.

Remark 1. The necessity of an efficient square-root algorithm arises directly from the special soundness proof of our protocol. In the witness extraction phase, the extractor combines responses from two distinct challenges to recover the secret key. However, due to the dual nature of our public key structure, which means both forward and inverse group actions are simultaneously used, this algebraic combination could yield the square of the secret key rather than the key itself. Thus, an efficient algorithm that can compute every square root is required to allow the extractor to recover the valid witness and thereby complete the security reduction, establishing that the honest prover does possess the secret key.

In the context of non-commutative cryptographic group actions, there are mainly three lines of problems that have been intensively studied: linear code equivalence [11, 17] as used in LESS, tensor isomorphism and its variations [33, 46, 54, 74] as used in MEDS and ALTEQ, and lattice isomorphism [9, 39, 51] as used in HAWK.

For MEDS and ALTEQ, the underlying group is $\mathrm{GL}(n,q)$. For $\mathrm{GL}(n,q)$, the matrix square-root problem was addressed in [10], but the algorithm there requires going to an extension field.[7] So, it is an interesting problem to devise a matrix square-root algorithm without going to the extension field. This seems to be the only bottleneck for using the group actions underlying MEDS and ALTEQ, because the IGAP for these actions seem still hard. For LESS, the underlying group is the monomial group $\mathrm{Mon}(n,q)$, consisting of matrices in $\mathrm{GL}(n,q)$ with each row and each column having exactly one non-zero entry. Although square-root search can be performed efficiently in this group, reusing group elements

[7] Although the square-root algorithm for $\mathrm{GL}(n,q)$ in [10] runs in poly-time, the entries of the output square-root matrix could go to an extension field of the original field $\mathbb{F}_q$. Thus, the proof for special soundness that extracts the secret key would fail if the algorithm cannot produce square-root solutions over $\mathbb{F}_q$.

there is risky, and for certain parameters, using the same group element twice can result in an insecure protocol according to attacks in [23, 25]. For HAWK, the underlying group is the group $\mathrm{GL}(n, \mathbb{Z})/\{\pm I_n\}$, which also fails to meet the IGAP hardness requirement by a similar attack in [24]. We will continue to discuss these attacks in Section 6.2.

The first requirement motivates us to consider the symmetric group S_n, where square-root search can be done even more efficiently than group operations (see Proposition 2). In particular, $1/n$ of permutations are full cycles, whose square root is unique. This imposes an additional constraint on the permutations sampled as the secret key, to avoid complications in analyzing the number of square roots.[8]

Accordingly, a main task for us is to find a problem satisfying the hardness assumption of IGAP with symmetric group actions. We achieve this by proposing a new interpretation of monomial code equivalence problems in Section 6.2. We also provide the hardness evidence in Section 6.2 (see Corollary 1) and [40, Appendix A].

Remark 2. For public key validation using the sigma protocol in Fig. 1, we need to assume the hardness of GASDDH (defined in Definition 9) under the choices of G and S in our instantiation. Recall that there is a worst-case reduction from GAIDDH (as defined in Definition 10) to GASDDH, and DmILCE is the computational search version of GAIDDH in the instantiated case. So, we also leave open for further investigation whether a search-to-distinguish reduction exists; if it does, it would strengthen the justification for our current instantiation choice.

Note that our main goal is to show that the framework can be instantiated from a non-commutative group action satisfying the required properties. Since concrete parameter selections for the proposed instantiation would require a more detailed cryptanalytic study of the underlying hardness assumptions, we limit ourselves to structural efficiency comparisons and operation counts rather than concrete size estimates.

6.2 A New Interpretation of Monomial Code Equivalence

A linear code over $\mathbb{F}_q$ is a subspace of $\mathbb{F}_q^n$. An m-dimensional code in $\mathbb{F}_q^n$ is represented by $C \in \mathrm{Mat}(m \times n, q)$, whose rows form a basis of the code. The monomial code equivalence problem is the following.

Problem 1 (Monomial code equivalence (LCE))**.** For $n \in \mathbb{N}$, let $m \in [n]$. Let $C, C' \in \mathrm{Mat}(m \times n, q)$ be of rank m. Decide if there exist $A \in \mathrm{GL}(m, q)$, $D \in \mathrm{D}(n, q)$, and $P \in \mathrm{S}_n$, such that $ACDP = C'$. If yes, compute such A, D, and P.

[8] In general, we only need an efficient algorithm that can compute all square roots. This implies that the number of square roots is polynomially bounded, since otherwise merely outputting them would already be inefficient. In other words, uniqueness of square roots is not a necessary condition, but it is a sufficient one.

Equivalently, we can formulate monomial code equivalence as asking if there exist $A \in \mathrm{GL}(m,q)$ and $M \in \mathrm{Mon}(n,q)$, such that $ACM = C'$. By writing $M \in \mathrm{Mon}(n,q)$ as DP where $D \in \mathrm{D}(n,q)$ and $P \in \mathrm{S}_n$, we can define a symmetric group action as below.

Monomial code equivalence as a symmetric group action. Let $C, C' \in \mathrm{Mat}(m \times n, q)$. In Definition 1, three matrices, $A \in \mathrm{GL}(m,q)$, $D \in \mathrm{D}(n,q)$, and $P \in \mathrm{S}_n$, are used to define equivalence between C and C'. As a result, there is more than one way to interpret the group action behind monomial code equivalence.

The first, most straightforward, way is to consider the action of the group $\mathrm{GL}(m,q) \times (\mathrm{D}(n,q) \rtimes \mathrm{S}_n) = \mathrm{GL}(m,q) \times \mathrm{Mon}(n,q)$[9] on the set $\mathrm{Mat}(m \times n, q)$ that consists of all generator matrices of m-dimensional codes in $\mathbb{F}_q^n$.

The second approach is to consider the monomial group $\mathrm{Mon}(n,q)$ acting on the set of m-dimensional codes in $\mathbb{F}_q^n$. This is the natural action from the viewpoint of coding theory, as seen in [17,23,25,36,66]. This group-theoretic perspective is further developed in [34] through a general framework that partitions the set of m-dimensional codes in $\mathbb{F}_q^n$ into orbits under subgroups of $\mathrm{Mon}(n,q)$, and thus the equivalence testing can be reframed as comparing canonical representatives within those subgroup-induced orbits. In this setting, the LCE problem corresponds to the finest partition induced by the trivial subgroup.

We take the third approach by formulating it as the symmetric group S_n acting on a set S. In particular, this set S is the set of equivalence classes of m-dimensional codes in $\mathbb{F}_q^n$ under scalar multiplications on the coordinates. Note that this is actually in line with the second approach in the sense that the monomial group $\mathrm{Mon}(n,q)$ can be seen to act on the equivalence classes of $\mathrm{Mat}(m \times n, q)$ under left multiplication by $\mathrm{GL}(m,q)$. Nevertheless, what sets this perspective apart is that the objects of the group action are the equivalence classes of codes, in contrast to the generator matrices or codes themselves as in other approaches.

Let us examine this set S in more detail. Recall that $m \in [n]$. For $C_1, C_2 \in \mathrm{Mat}(m \times n, q)$, we define an equivalence relation $\sim$ as $C_1 \sim C_2$ if and only if there exists some $A \in \mathrm{GL}(m,q)$ and $D \in \mathrm{D}(n,q)$ such that $C_1 = AC_2D$. Note that this equivalence relation partitions $\mathrm{Mat}(m \times n, q)$ into orbits of $\mathrm{Mat}(m \times n, q)$ under the action of $\mathrm{GL}(m,q) \times \mathrm{D}(n,q)$. Denote by $[C]_\sim := \{ACD : A \in \mathrm{GL}(m,q), D \in \mathrm{D}(n,q)\}$ the equivalence class determined by $\sim$ and corresponding to $C \in \mathrm{Mat}(m \times n, q)$. Let $\mathrm{Mat}(m \times n, q)/\sim = \{[C]_\sim : C \in \mathrm{Mat}(m \times n, q)\}$ be the set of equivalence classes under $\sim$. This is the set S to be acted on.

We wish to define an action of S_n on $\mathrm{Mat}(m \times n, q)/\sim$. For $P \in \mathrm{S}_n$, since $[C]_\sim := \{ACD : A \in \mathrm{GL}(m,q), D \in \mathrm{D}(n,q)\}$ is a set of matrices, a natural map is to send $[C]_\sim$ to $[C]_\sim P := \{ACDP : A \in \mathrm{GL}(m,q), D \in \mathrm{D}(n,q)\}$. For S_n to act on $\mathrm{Mat}(m \times n, q)/\sim$, we need to show that $[C]_\sim P$ is an element in $\mathrm{Mat}(m \times n, q)/\sim$ by the following proposition.

Proposition 1. *Let $[C]_\sim \in \mathrm{Mat}(m \times n, q)/\sim$, $P \in \mathrm{S}_n$, and $[C]_\sim P$ be as above. Then $[C]_\sim P = [CP]_\sim$.*

[9] Here $\rtimes$ denotes semidirect product of groups.

Proof. Recall that $C \in \mathrm{Mat}(m \times n, q)$. Let $A \in \mathrm{GL}(m, q)$, and $D \in \mathrm{D}(n, q)$. Note that for any $P \in \mathrm{S}_n$, we have $\mathrm{D}(n, q) = P^{-1}\,\mathrm{D}(n, q)P = \{P^{-1}DP : D \in \mathrm{D}(n, q)\}$. This is because one can verify that $P^{-1}DP$ is a diagonal matrix with ith diagonal entry being the $P(i)$th entry of D. Therefore, $[C]_\sim P := \{ACDP : A \in \mathrm{GL}(m, q), D \in \mathrm{D}(n, q)\} = \{ACPP^{-1}DP : A \in \mathrm{GL}(m, q), D \in \mathrm{D}(n, q)\} = \{ACPD' : A \in \mathrm{GL}(m, q), D' \in \mathrm{D}(n, q)\} = [CP]_\sim$. □

This ensures that the map $\alpha : \mathrm{S}_n \times \mathrm{Mat}(m \times n, q)/\sim \to \mathrm{Mat}(m \times n, q)/\sim$ by $P \in \mathrm{S}_n$ sending $[C]_\sim$ to $[C]_\sim P = [CP]_\sim$ is a well-defined group action. Now we can state our security problem in this instantiated case as follows:

Problem 2 (Diagonal-masked inverse linear code equivalence (DmILCE))**.** For $n \in \mathbb{N}$, let $m \in [n]$. Let $\{C_0, C_1, C_2\} \subseteq \mathrm{Mat}(m \times n, q)$. Decide if there exist $A_1, A_2 \in \mathrm{GL}(m, q)$, $D_1, D_2 \in \mathrm{D}(n, q)$, and $P \in \mathrm{S}_n$, such that $C_1 = A_1 C_0 D_1 P$ and $C_2 = A_2 C_0 D_2 P^{-1}$. If yes, find the secret permutation P.

As preliminary evidence for the hardness of Problem 2, we refer to the Gröbner-basis experiments reported in [40, Appendix A].

Comparison with reusing the monomial matrix. Our instantiation has two main ingredients. First, we choose the group G to be the symmetric group, since square-root computation is efficient there (for the class of elements we use). Second, we hide elements of the acted set S inside suitable equivalence classes, since a direct symmetric-group action is not secure enough. In a related setting, recent works [23, 25] have considered the following relaxed monomial code equivalence problem with multiple samples.

Problem 3 (Inverse linear code equivalence (ILCE))**.** For $n \in \mathbb{N}$, let $m \in [n]$. Let $\{C_0, C_1, C_2\} \subseteq \mathrm{Mat}(m \times n, q)$. Decide if there exist $A \in \mathrm{GL}(m, q)$ and $M \in \mathrm{Mon}(n, q)$, such that $C_1 = AC_0M$ and $C_2 = A^{-1}C_0M^{-1}$. If yes, find the secret monomial matrix M.

From the group action viewpoint, this problem corresponds to reusing monomial group actions (compared to our symmetric group actions) and there is only a single general linear group $\mathrm{GL}(m, q)$ acting on the left of the generator matrices (compared to our equivalence classes under a composite group $\mathrm{GL}(m, q) \times \mathrm{D}(n, q)$) used to hide secret information during communication. Therefore, by applying the parity-check matrices of the linear codes, one can get rid of the left general group actions A and A^{-1} and leave the monomial matrix M as the only variable matrix. In this way, Problem 3 reduces to solving a homogeneous linear system, and therefore [23] managed to give a heuristic algorithm to tackle it for $m = n/2$ and [25] further refined it to solve the cases for any code rate. We also note that the subsequent attacks [24, 32, 45] on other secret-reuse problems all rely on this crucial linearization technique.

However, their techniques does not help in attacking our Problem 2, whether experimentally or theoretically. The main reason is that in an equivalence class, the way to scale the columns of the generator matrices can vary, so treating

these scalars as additional variables and combining them with the common permutation as another variable matrix would make a quadratic equation system instead of a linear system, which makes the heuristic analysis dedicated to linear systems in [23–25, 32, 45] completely inapplicable to our case. Furthermore, the linearization techniques for solving quadratic polynomial systems in [59] also cannot apply due to the insufficient number of equations. Besides, a key step of the algorithms in [23, 25] is to guess the entries of the common monomial matrix and check whether setting one of them to 1 can lead to the system having a solution. This is because in a homogeneous system, if M is a solution, and cM is also a solution for all $c \in \mathbb{F}_q$. Again, since the scalars for the commitment and keys keep updating in our protocol, guessing the diagonal values is more costly, and checking the existence of solutions to a system of quadratic equations is harder. These essential gaps differentiate our use of monomial code equivalence from those broken in [23, 25], which is also supported by our experimental results in [40, Appendix A]; see [40, Remark 4].

We now relate our problem to a recently introduced problem that underpins the hardness assumption used in the key exchange protocol of [41].

Problem 4 (General diagonal-masked linear code equivalence (DmLCE) [41])**.** For $n \in \mathbb{N}$, let $m \in [n]$. Let $\{C_i : i \in [n]\} \subseteq \mathrm{Mat}(m \times n, q)$. Decide if there exist $\{A_i : i \in [n-1]\} \subseteq \mathrm{GL}(m, q)$, $\{D_i : i \in [n-1]\} \subseteq \mathrm{D}(n, q)$, and an $n \times n$ permutation matrix P, such that $A_i C_i D_i P = C_{i+1}$ for all $i \in [n-1]$. If yes, find the secret permutation matrix P.

By [41, Fact 2], DmLCE reduces to the following 2-DmLCE problem.

Problem 5 (2-samples diagonal-masked linear code equivalence (2-DmLCE) [41])**.** For odd prime $n \in \mathbb{N}$, let $m \in [n]$. Let $C_1, C_2, C_3 \in \mathrm{Mat}(m \times n, q)$. Decide if there exist $A_1, A_2 \in \mathrm{GL}(m, q)$, $D_1, D_2 \in \mathrm{D}(n, q)$, and an $n \times n$ permutation matrix P, such that $A_1 C_1 D_1 P = C_2$ and $A_2 C_2 D_2 P = C_3$. If yes, compute all instances of such a common permutation matrix P.

We note that Problem 5 (2-DmLCE) is exactly equivalent to our Problem 2 (DmILCE), by moving P^{-1} in $C_2 = A_2 C_0 D_2 P^{-1}$ to the left-hand side.

Corollary 1. *Problem 4 (DmLCE)* $\leq_p$ *Problem 2 (DmILCE).*

This ensures that our underlying computational problem, DmILCE, of the IGAP assumption for this instantiation, is as hard as a known conjectured hard problem, DmLCE [41]. We leave further cryptanalysis of DmILCE as an intriguing open question, and we believe that research on this problem is inherently meaningful and could be beneficial to the entire community, as it is a natural variant of the standard code-based assumptions.

Remark 3. A very recent work by Alecci and D'Alconzo [4] has investigated an equivalent algebraic formulation of the DmILCE problem. Using tools from algebraic geometry, including Plücker coordinates and fields of invariant rational functions, they construct polynomial equations whose roots are the entries of the

common permutation matrix. However, the resulting equations are not practical for cryptographic parameters. Specifically, the polynomials have a degree of $2k$ and contain $2(k!)^2$ monomials, which grow exponentially. Thus, their work provides a theoretical algebraic model for stripping away the diagonal mask, but also shows that this direct invariant-theoretic approach is infeasible at cryptographic sizes. This offers indirect support for the hardness of the DmILCE problem, while not a formal proof of hardness.

6.3 Square-root and Canonical Form Algorithms

To achieve the soundness of the underlying sigma protocol for our blind signature, we need an algorithm to efficiently compute a certain square root of $g_\delta^2 \in \mathrm{S}_n$. Note that finding g_δ costs less than a group operation which corresponds to matrix multiplication in our instantiation. The proof can be found in [40, Appendix B].

Proposition 2. *There is an $O(n)$-time square-root algorithm that inputs $\sigma \in \mathrm{S}_n$ and outputs a square root of σ if it exists.*

When blinding the message concatenated with the set elements (e.g., $\mathbf{Z}_0^{(1)}$, $\mathbf{Z}_0^{(-1)}$, $\mathbf{Z}_1^{(1)}$, $\mathbf{Z}_1^{(-1)} \in S^n$) under a hash function H, it is less practical to feed the entire equivalence class into H. By contrast, we can efficiently compute the *canonical forms* that can uniquely represent the underlying equivalence classes. Specifically, for an authentic signature, to ensure that the resulting hash values c and c' are the same, the user (and other verifiers) need to perform a canonical form algorithm for the equivalence classes. Canonical forms are a well-studied topic for graphs and matrix tuples [6, 71]. Given a group G acting on a set S, a canonical form algorithm for this group action takes $s \in S$ and outputs $s^* \in \mathsf{Orb}(s) := \{g * s : \forall g \in G\}$. Furthermore, for another $s' \in \mathsf{Orb}(s)$, the algorithm should output the same s^*. In our case, G and S can be interpreted as $\mathrm{GL}(m, q) \times \mathrm{D}(n, q)$ and $\mathrm{Mat}(m \times n, q)$, respectively. Accordingly, we establish the following proposition, which strengthens the result in [37, Proposition 4] by eliminating the failure probability. Again, the proof can be found in [40, Appendix B].

Proposition 3. *There is an $O(m^2 n)$-time canonical form algorithm for the action of $\mathrm{GL}(m, q) \times \mathrm{D}(n, q)$ on $\mathrm{Mat}(m \times n, q)$.*

Acknowledgement

We would like to thank the anonymous reviewers for their careful reading and constructive feedback, which helped us improve the paper. We also thank Lucjan Hanzlik, Jesús-Javier Chi-Domínguez, Edoardo Persichetti, Veronika Kuchta, and Jason LeGrow for their insightful comments and helpful discussions. This work is partially supported by the ARC Linkage Project LP220100332, the ARC Discovery Project DP200100950, and the Sydney Quantum Academy, Sydney,

NSW, Australia. Part of this work was done when Chuanqi was a PhD student and Youming was an associate professor at the Centre for Quantum Software and Information, University of Technology Sydney, Australia.

References

1. Abe, M., Okamoto, T.: Provably secure partially blind signatures. In: Advances in Cryptology - CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 271–286. Springer (2000). https://doi.org/10.1007/3-540-44598-6_17
2. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Practical, round-optimal lattice-based blind signatures. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 39–53. ACM (2022). https://doi.org/10.1145/3548606.3560650
3. Alamati, N., Feo, L.D., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. In: 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. Lecture Notes in Computer Science, vol. 12492, pp. 411–439. Springer (2020). https://doi.org/10.1007/978-3-030-64834-3_14
4. Alecci, G., D'Alconzo, G.: Linear code equivalence via plücker coordinates. arXiv preprint arXiv:2603.09869 (2026)
5. Alkadri, N.A., Bansarkhani, R.E., Buchmann, J.: BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In: Bonneau, J., Heninger, N. (eds.) Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers. Lecture Notes in Computer Science, vol. 12059, pp. 484–502. Springer (2020). https://doi.org/10.1007/978-3-030-51280-4_26
6. Babai, L.: Canonical form for graphs in quasipolynomial time: preliminary report. In: Charikar, M., Cohen, E. (eds.) Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019. pp. 1237–1246. ACM (2019). https://doi.org/10.1145/3313276.3316356
7. Barenghi, A., Biasse, J., Ngo, T., Persichetti, E., Santini, P.: Advanced signature functionalities from the code equivalence problem. Int. J. Comput. Math. Comput. Syst. Theory **7**(2), 112–128 (2022). https://doi.org/10.1080/23799927.2022.2048206
8. Battagliola, M., Borin, G., Meneghetti, A., Persichetti, E.: Cutting the GRASS: threshold group action signature schemes. In: Oswald, E. (ed.) Topics in Cryptology - CT-RSA 2024 - Cryptographers' Track at the RSA Conference 2024, San Francisco, CA, USA, May 6-9, 2024, Proceedings. Lecture Notes in Computer Science, vol. 14643, pp. 460–489. Springer (2024). https://doi.org/10.1007/978-3-031-58868-6_18
9. Benčina, B., Budroni, A., Chi-Domínguez, J.J., Kulkarni, M.: Properties of lattice isomorphism as a cryptographic group action. In: International Conference on Post-Quantum Cryptography. pp. 170–201. Springer (2024)
10. Berthomieu, J., Faugere, J.C., Perret, L.: Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. Journal of Complexity **31**(4), 590–616 (2015). https://doi.org/10.1016/j.jco.2015.04.001

11. Beullens, W.: Not enough LESS: an improved algorithm for solving code equivalence problems over $\mathbb{F}_q$. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 387–403. Springer (2020). https://doi.org/10.1007/978-3-030-81652-0_15
12. Beullens, W.: Multivariate blind signatures revisited. IACR Cryptol. ePrint Arch. p. 720 (2024), https://eprint.iacr.org/2024/720
13. Beullens, W., Dobson, S., Katsumata, S., Lai, Y., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In: Dunkelman, O., Dziembowski, S. (eds.) 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. Lecture Notes in Computer Science, vol. 13276, pp. 95–126. Springer (2022). https://doi.org/10.1007/978-3-031-07085-3_4
14. Beullens, W., Katsumata, S., Pintore, F.: Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In: Moriai, S., Wang, H. (eds.) 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. Lecture Notes in Computer Science, vol. 12492, pp. 464–492. Springer (2020). https://doi.org/10.1007/978-3-030-64834-3_16
15. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient isogeny based signatures through class group computations. In: Advances in Cryptology - ASIACRYPT 2019. Lecture Notes in Computer Science, vol. 11921, pp. 227–247. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_9
16. Beullens, W., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Lattice-based blind signatures: Short, efficient, and round-optimal. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023. pp. 16–29. ACM (2023). https://doi.org/10.1145/3576915.3616613
17. Biasse, J., Micheli, G., Persichetti, E., Santini, P.: LESS is more: Code-based signatures without syndromes. In: Progress in Cryptology - AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Cairo, Egypt, July 20-22, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12174, pp. 45–65. Springer (2020). https://doi.org/10.1007/978-3-030-51938-4_3
18. Bläser, M., Chen, Z., Duong, D.H., Joux, A., Nguyen, T.N., Plantard, T., Qiao, Y., Susilo, W., Tang, G.: On digital signatures based on group actions: QROM security and ring signatures. In: Saarinen, M.O., Smith-Tone, D. (eds.) Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14771, pp. 227–261. Springer (2024). https://doi.org/10.1007/978-3-031-62743-9_8
19. Blazy, O., Gaborit, P., Mac, D.T.: A correction to a code-based blind signature scheme. In: Code-Based Cryptography - 9th International Workshop, CBCrypto 2021, Munich, Germany, June 21-22, 2021 Revised Selected Papers. Lecture Notes in Computer Science, vol. 13150, pp. 84–94. Springer (2021). https://doi.org/10.1007/978-3-030-98365-9_5
20. Blazy, O., Gaborit, P., Schrek, J., Sendrier, N.: A code-based blind signature. In: 2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen,

Germany, June 25-30, 2017. pp. 2718–2722. IEEE (2017). https://doi.org/10.1109/ISIT.2017.8007023
21. Bouaziz-Ermann, S., Canard, S., Eberhart, G., Kaim, G., Roux-Langlois, A., Traoré, J.: Lattice-based (partially) blind signature without restart. IACR Cryptol. ePrint Arch. p. 260 (2020), https://eprint.iacr.org/2020/260
22. Brassard, G., Yung, M.: One-way group actions. In: Advances in Cryptology - CRYPTO 1990. pp. 94–107 (1990). https://doi.org/10.1007/3-540-38424-3_7
23. Budroni, A., Chi-Domínguez, J., D'Alconzo, G., Di Scala, A.J., Kulkarni, M.: Don't use it twice! solving relaxed linear equivalence problems **15491**, 35–65 (2024). https://doi.org/10.1007/978-981-96-0944-4_2
24. Budroni, A., Chi-Domínguez, J.J., Franch, E.: Don't use it twice: Reloaded! on the lattice isomorphism group action. Cryptology ePrint Archive (2025)
25. Budroni, A., Natale, A.: On the sample complexity of linear code equivalence for all code rates. Cryptography and Communications pp. 1–23 (2025)
26. Buser, M., Dowsley, R., Esgin, M.F., Gritti, C., Kermanshahi, S.K., Kuchta, V., LeGrow, J.T., Liu, J.K., Phan, R.C., Sakzad, A., Steinfeld, R., Yu, J.: A survey on exotic signatures for post-quantum blockchain: Challenges and research directions. ACM Comput. Surv. **55**(12), 251:1–251:32 (2023). https://doi.org/10.1145/3572771
27. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT). pp. 395–427. Springer (2018). https://doi.org/10.1007/978-3-030-03332-3_15
28. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982. pp. 199–203. Plenum Press, New York (1982). https://doi.org/10.1007/978-1-4757-0602-4_18
29. Chaum, D.: Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Günther, C.G. (ed.) Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings. Lecture Notes in Computer Science, vol. 330, pp. 177–182. Springer (1988). https://doi.org/10.1007/3-540-45961-8_15
30. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: Goldwasser, S. (ed.) Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings. Lecture Notes in Computer Science, vol. 403, pp. 319–327. Springer (1988). https://doi.org/10.1007/0-387-34799-2_25
31. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. Lecture Notes in Computer Science, vol. 740, pp. 89–105. Springer (1992). https://doi.org/10.1007/3-540-48071-4_7
32. Chi-Domínguez, J.J.: Weak instances of the inverse matrix code equivalence problem. Cryptology ePrint Archive (2025), https://eprint.iacr.org/2025/1909
33. Chou, T., Niederhagen, R., Persichetti, E., Randrianarisoa, T.H., Reijnders, K., Samardjiska, S., Trimoska, M.: Take your meds: Digital signatures from matrix code equivalence. In: Progress in Cryptology - AFRICACRYPT 2023 (2023). https://doi.org/10.1007/978-3-031-37679-5_2

34. Chou, T., Persichetti, E., Santini, P.: On linear equivalence, canonical forms, and digital signatures. Designs, Codes and Cryptography pp. 1–43 (2025). https://doi.org/10.1007/s10623-025-01576-1
35. Couveignes, J.M.: Hard homogeneous spaces. IACR Cryptology ePrint Archive (2006), http://eprint.iacr.org/2006/291
36. D'Alconzo, G., Di Scala, A.J.: Representations of group actions and their applications in cryptography. Finite Fields and Their Applications **99**, 102476 (2024). https://doi.org/10.1016/j.ffa.2024.102476
37. D'Alconzo, G., Meneghetti, A., Signorini, E.: Group factorisation for smaller signatures from cryptographic group actions. Designs, Codes and Cryptography **94**(2), 42 (2026). https://doi.org/10.1007/s10623-025-01787-6
38. Do, K., Hanzlik, L., Paracucchi, E.: M&m's: Mix and match attacks on schnorr-type blind signatures with repetition. In: 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. Lecture Notes in Computer Science, vol. 14656, pp. 363–387. Springer (2024). https://doi.org/10.1007/978-3-031-58751-1_13
39. Ducas, L., Postlethwaite, E.W., Pulles, L.N., Woerden, W.v.: Hawk: Module lip makes lattice signatures fast, compact and simple. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 65–94. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_3
40. Duong, D.H., Khuc, X.T., Qiao, Y., Susilo, W., Zhang, C.: Blind signatures from cryptographic group actions. Cryptology ePrint Archive, Paper 2025/397 (2025), https://eprint.iacr.org/2025/397
41. Duong, D.H., Qiao, Y., Zhang, C.: Diffie-Hellman Key Exchange from Commutativity to Group Laws. In: 17th Innovations in Theoretical Computer Science Conference (ITCS 2026). Leibniz International Proceedings in Informatics (LIPIcs), vol. 362, pp. 52:1–52:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2026). https://doi.org/10.4230/LIPIcs.ITCS.2026.52, https://eprint.iacr.org/2025/1677
42. Feo, L.D., Galbraith, S.D.: Seasign: Compact isogeny signatures from class group actions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019. Lecture Notes in Computer Science, vol. 11478, pp. 759–789. Springer (2019). https://doi.org/10.1007/978-3-030-17659-4_26
43. Feo, L.D., Meyer, M.: Threshold schemes from isogeny assumptions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 187–212. Springer (2020). https://doi.org/10.1007/978-3-030-45388-6_7
44. Fuchsbauer, G., Wolf, M.: Concurrently secure blind schnorr signatures. In: Joye, M., Leander, G. (eds.) 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 II. Lecture Notes in Computer Science, vol. 14652, pp. 124–160. Springer (2024). https://doi.org/10.1007/978-3-031-58723-8_5
45. Gilchrist, V., Marco, L., Petit, C., Tang, G.: On the security of two blind signatures from code equivalence problems. IACR Communications in Cryptology **2**(4) (2026). https://doi.org/10.62056/aesg893y6
46. Grochow, J.A., Qiao, Y.: Isomorphism problems for tensors, groups, and cubic forms: completeness and reductions (2019), arXiv:1907.00309 [cs.CC]

47. Hallgren, S., Moore, C., Rötteler, M., Russell, A., Sen, P.: Limitations of quantum coset states for graph isomorphism. J. ACM **57**(6), 34:1–34:33 (Nov 2010). https://doi.org/10.1145/1857914.1857918
48. Hanzlik, L., Lai, Y., Mula, M., Paracucchi, E., Slamanig, D., Tang, G.: Tanuki: New frameworks for (concurrently secure) blind signatures from post-quantum groups actions. In: Advances in Cryptology – ASIACRYPT 2025. pp. 35–69. Springer Nature Singapore, Singapore (2026). https://doi.org/10.1007/978-981-95-5113-2_2
49. Hauck, E., Kiltz, E., Loss, J.: A modular treatment of blind signatures from identification schemes. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11478, pp. 345–375. Springer (2019). https://doi.org/10.1007/978-3-030-17659-4_12
50. Hauck, E., Kiltz, E., Loss, J., Nguyen, N.K.: Lattice-based blind signatures, revisited. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12171, pp. 500–529. Springer (2020). https://doi.org/10.1007/978-3-030-56880-1_18
51. Haviv, I., Regev, O.: On the lattice isomorphism problem. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 391–404 (2014)
52. Heilman, E., Baldimtsi, F., Goldberg, S.: Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D.S., Brenner, M., Rohloff, K. (eds.) Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9604, pp. 43–60. Springer (2016). https://doi.org/10.1007/978-3-662-53357-4_4
53. Hhan, M., Morimae, T., Yamakawa, T.: From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 639–667. Springer (2023). https://doi.org/10.1007/978-3-031-30545-0_22
54. Ji, Z., Qiao, Y., Song, F., Yun, A.: General linear group action on tensors: A candidate for post-quantum cryptography. In: Theory of cryptography conference. pp. 251–281. Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_11
55. Kastner, J., Loss, J., Xu, J.: On Pairing-Free Blind Signature Schemes in the Algebraic Group Model. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) 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. Lecture Notes in Computer Science, vol. 13178, pp. 468–497. Springer (2022). https://doi.org/10.1007/978-3-030-97131-1_16
56. Kastner, J., Loss, J., Xu, J.: The Abe-Okamoto Partially Blind Signature Scheme Revisited. In: Agrawal, S., Lin, D. (eds.) 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. Lecture Notes in Computer Science, vol. 13794, pp. 279–309. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_10

57. Katsumata, S., Lai, Y., LeGrow, J.T., Qin, L.: CSI-Otter: Isogeny-Based (Partially) Blind Signatures from the Class Group Action with a Twist. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III. Lecture Notes in Computer Science, vol. 14083, pp. 729–761. Springer (2023). https://doi.org/10.1007/978-3-031-38548-3_24
58. Katsumata, S., Lai, Y., Reichle, M.: Breaking parallel ROS: implication for isogeny and lattice-based blind signatures. In: 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. Lecture Notes in Computer Science, vol. 14601, pp. 319–351. Springer (2024). https://doi.org/10.1007/978-3-031-57718-5_11
59. Kipnis, A., Shamir, A.: Cryptanalysis of the hfe public key cryptosystem by relinearization. In: Annual International Cryptology Conference. pp. 19–30. Springer (1999). https://doi.org/10.1007/3-540-48405-1_2
60. Kucharczyk, M.: Blind signatures in electronic voting systems. In: Kwiecien, A., Gaj, P., Stera, P. (eds.) Computer Networks - 17th Conference, CN 2010, Ustroń, Poland, June 15-19, 2010. Proceedings. Communications in Computer and Information Science, vol. 79, pp. 349–358. Springer (2010). https://doi.org/10.1007/978-3-642-13861-4_37
61. Kuchta, V., LeGrow, J.T., Persichetti, E.: Post-quantum blind signatures from matrix code equivalence. IACR Cryptol. ePrint Arch. p. 274 (2025), https://eprint.iacr.org/2025/274
62. Lai, Y.F., Persichetti, E.: LEAF: Compact and efficient blind signature from code-based assumptions. Cryptology ePrint Archive, Paper 2025/1585 (2025), https://eprint.iacr.org/2025/1585
63. Le, H.Q., Susilo, W., Khuc, T.X., Bui, M.K., Duong, D.H.: A blind signature from module latices. In: 2019 IEEE Conference on Dependable and Secure Computing, DSC 2019, Hangzhou, China, November 18-20, 2019. pp. 1–8. IEEE (2019). https://doi.org/10.1109/DSC47296.2019.8937613
64. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Efficient lattice-based blind signatures via gaussian one-time signatures. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) 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. Lecture Notes in Computer Science, vol. 13178, pp. 498–527. Springer (2022). https://doi.org/10.1007/978-3-030-97131-1_17
65. Papachristoudis, D., Hristu-Varsakelis, D., Baldimtsi, F., Stephanides, G.: Leakage-resilient lattice-based partially blind signatures. IET Inf. Secur. **13**(6), 670–684 (2019). https://doi.org/10.1049/IET-IFS.2019.0156
66. Persichetti, E., Santini, P.: A new formulation of the linear equivalence problem and shorter less signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 351–378. Springer (2023). https://doi.org/10.1007/978-981-99-8739-9_12
67. Petzoldt, A., Szepieniec, A., Mohamed, M.S.E.: A practical multivariate blind signature scheme. In: Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10322, pp. 437–454. Springer (2017). https://doi.org/10.1007/978-3-319-70972-7_25

68. Pfitzmann, B.P., Sadeghi, A.R.: Anonymous fingerprinting with direct non-repudiation. In: Advances in Cryptology—ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings 6. pp. 401–414. Springer (2000). https://doi.org/https://doi.org/10.1007/3-540-44448-3_31
69. del Pino, R., Katsumata, S.: A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In: Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II. Lecture Notes in Computer Science, vol. 13508, pp. 306–336. Springer (2022). https://doi.org/10.1007/978-3-031-15979-4_11
70. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of cryptology **13**(3), 361–396 (2000). https://doi.org/10.1007/s001450010003
71. Qiao, Y., Sun, X.: Canonical forms for matrix tuples in polynomial time. In: 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). pp. 780–789. IEEE (2024). https://doi.org/10.1109/FOCS61266.2024.00054
72. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. IACR Cryptol. ePrint Arch. p. 145 (2006), http://eprint.iacr.org/2006/145
73. Rückert, M.: Lattice-based blind signatures. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6477, pp. 413–430. Springer (2010). https://doi.org/10.1007/978-3-642-17373-8_24
74. Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In: 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. Lecture Notes in Computer Science, vol. 13277, pp. 582–612. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_21
75. Yi, X., Lam, K.: A new blind ECDSA scheme for bitcoin transaction anonymity. In: Proceedings of AsiaCCS 2019. pp. 613–620. ACM (2019). https://doi.org/10.1145/3321705.3329816