Re-Randomize and Extract: A Novel Commitment Construction Framework Based on Group Actions

Kaijie Jiang[1,2], Anyu Wang[1,2,7(✉)], Hengyi Luo[3,4], Guoxiao Liu[2,5], Gang Tang[6], Yanbin Pan[3,4], and Xiaoyun Wang[1,2,7,8,9,10]

[1] Institute for Advanced Study, BNRist, Tsinghua University, Beijing, China
jkj21@mails.tsinghua.edu.cn,
{anyuwang,xiaoyunwang}@tsinghua.edu.cn,
[2] State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University, Beijing, China
[3] State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
{luohengyi, panyanbin}@amss.ac.cn
[4] University of Chinese Academy of Sciences, Beijing, China
[5] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
lgx22@mails.tsinghua.edu.cn
[6] University of Birmingham, Birmingham, UK
g.tang.1@bham.ac.uk
[7] Zhongguancun Laboratory, Beijing, China
[8] National Financial Cryptography Research Center, Beijing, China
[9] Shandong Institute of Blockchain, Jinan, China
[10] Key Laboratory of Cryptologic Technology and Information Security, School of Cyber Science and Technology, Shandong University, Qingdao, China

Abstract. Cryptographic group actions have attracted growing attention as a useful tool for constructing cryptographic schemes. Among their applications, commitment schemes are particularly interesting as fundamental primitives, playing a crucial role in protocols such as zero-knowledge proofs, multi-party computation, and more.

In this paper, we introduce a novel framework to construct commitment schemes based on cryptographic group actions. Specifically, we propose two key techniques for general group actions: re-randomization and randomness extraction. Roughly speaking, a re-randomization algorithm introduces randomness within an orbit for any input element, while a randomness extractor maps this randomness to uniformity over the message space. We demonstrate that these techniques can significantly facilitate the construction of commitment schemes, providing a flexible framework for constructing either perfectly hiding or perfectly binding commitments, depending on the type of extractor involved. Moreover, we extend our framework to support the construction of commitments with additional desirable properties beyond hiding and binding, such as dual-mode commitments and enhanced linkable commitments. These extensions are achieved by further adapting the extractor to satisfy trapdoor or homomorphic properties. Finally, we instantiate all our proposed

commitment schemes using lattices, specifically leveraging the lattice isomorphism problem (LIP) and the lattice automorphism problem (LAP) as underlying cryptographic assumptions. To the best of our knowledge, this is the first commitment scheme construction based on LIP/LAP. Additionally, we use LIP to provide a repair and improvement to the tensor isomorphism-based non-interactive commitment scheme proposed by D'Alconzo, Flamini, and Gangemi (*ASIACRYPT* 2023), which was recently shown to be insecure by an attack from Gilchrist, Marco, Petit, and Tang (*CRYPTO* 2024).

Keywords: Cryptographic group action · Dual-mode commitment · Homomorphic commitment · Lattice isomorphism problem

1 Introduction

Cryptographic group action is a powerful tool in the design of cryptographic schemes, with its study tracing back to Brassard and Yung in 1991 [14], where the concept of one-way group action was first introduced. Due to the generality of group actions in mathematics, various cryptographic assumptions with specific algebraic structures fall under the category of cryptographic group actions. These include isogenies [15], lattices [7], linear codes [11], polynomial isomorphisms [65,23], and trilinear forms [70]. Many of these assumptions are believed to resist quantum attacks, making them particularly well-suited for post-quantum cryptography. Recently, the framework of cryptographic group actions has demonstrated its effectiveness in cryptographic constructions, including commitment schemes [46,27], ring and group signatures [10,12,9], threshold signatures [20,30,6], threshold ring signatures [67], blind signatures [50], key exchanges [15], updatable encryption schemes [54,59], trapdoor claw-free functions [2], dual-mode trapdoor functions [39], verifiable random functions [52], and robustly reusable fuzzy extractors [73]. These wide-ranging applications highlight the versatility of cryptographic group actions in building secure and efficient cryptographic schemes.

Among these applications of cryptographic group actions, the commitment scheme is perhaps the simplest but also one of the most important cryptographic primitives. A commitment scheme is a protocol between two parties, a sender A and a receiver B, where A wants to commit to a message m for B. In essence, A can place the message into a "digital sealed envelope", referred to as a commitment. Later, when A wants to reveal the message to B, A opens the envelope. This process must satisfy two fundamental properties. First, the digital envelope must not reveal any information about the message prior to its opening. This property is known as *hiding*. Second, A must not be able to open the same commitment to a different message $m' \neq m$, a property referred to as *binding*. These properties make commitment schemes indispensable in many cryptographic applications, such as zero-knowledge protocols [35], multi-party computation [31], digital auctions [63], confidential transactions [68], and electronic commerce [22].

The construction of commitment schemes has been an active research area for decades, with numerous classical results showing that commitment schemes can be based on fundamental cryptographic assumptions [62,5,34,42,41]. Moreover, commitment schemes often serve as building blocks for more complex cryptographic protocols. In many cases, it is desirable for commitment schemes to possess additional properties beyond hiding and binding, such as homomorphic commitments [38,66], linkable commitments [27], dual-mode commitments [37], and others.

1.1 Our Results and Techniques

In this paper, we introduce a novel group action-based framework for constructing commitment schemes. Unlike previous construction frameworks, we observe that group actions can be naturally equipped with a *re-randomization* and an *extraction* process, which inherently facilitates the construction of commitments.

Re-Randomization and Randomness Extraction for Group Action. For a group action $(G, X, \star)$, where G is a group and X is a set, a re-randomization algorithm R takes as input an element $x \in X$ and outputs a pair $(x' = g \star x, g) \in X \times G$, such that the output distribution depends only on the orbit $\mathcal{O}(x)$ under the group action, and that g is uniformly distributed over the group elements that map x to x'. Intuitively, the re-randomization algorithm applies a random group action to x, effectively "forgetting" all specific information about x except its orbit $\mathcal{O}(x)$. We note that this notion of re-randomization can be seen as an abstraction of random self-reduction for lattice isomorphisms, where Gaussian sampling is used to obscure the details of the lattice basis while retaining only the geometric structure of the lattice [43,26,8]. Moreover, this self-reducing property emerges in various isomorphisms of algebraic objects and is often studied under the framework of random self-reducibility [61]. Thus, this abstraction of re-randomization represents a common characteristic in instantiations of cryptographic group actions.

An extractor roughly extracts the randomness over the group G to uniformity over the message space M, which we assume to have a group structure. We note that such functionality can be naturally achieved using a randomness extractor [4,42], which utilizes a public random seed to convert a non-uniform random variable into a near-uniform one, and is broadly adopted in the construction of cryptographic primitives. However, in our framework, we particularly focus on extractors that better align with the algebraic structure of group actions. The simplest type of extractor we introduce is a deterministic extractor E, which is required to be a deterministic function that takes as input elements $g \in G$, such that the output $\mathsf{E}(g)$ is uniformly distributed over M when g is drawn uniformly from the group elements that map x to x', where $x, x' \in X$ are in the same orbit. When combined with the re-randomization algorithm R, the extractor E can effectively extract the randomness introduced by R into uniformity over the message space M.

Basic Commitment Construction Framework. Equipped with the re-randomization algorithm R and a deterministic extractor E, a commitment scheme can be inherently constructed based on the group action $(G, X, \star)$. Initially, we set the commitment key to be $Ck = x$ for some $x \in X$. To commit to a message $m \in M$, we evoke the re-randomization algorithm to obtain a pair $(x', g) \leftarrow \mathsf{R}(x)$, and then set $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, x')$ as the commitment, with $d = g$ as the open value. To open the message, we output the message $\mathsf{E}(d)^{-1} \cdot c_1$ if the condition $d \star Ck = c_2$ holds. On the one hand, due to the uniformity of the output of E, this commitment scheme is inherently perfectly hiding. On the other hand, the binding property can be satisfied under certain computational assumptions on the group action.

Leveraging this basic model, we can adapt the extractor to obtain more commitment constructions. We show that a perfectly binding and computationally hiding commitment can be achieved using a *local constant extractor*, which extracts randomness for some orbits and returns constants for others. Additionally, we demonstrate that using the classical randomness extractor [4,42], we can construct a statistically hiding and computationally binding commitment, as well as a perfectly binding and computationally hiding commitment.

Construction Framework for Dual-Mode Commitment. As mentioned earlier, commitments often require additional properties beyond hiding and binding when used in cryptographic constructions such as zero-knowledge protocols. Dual-mode commitment is a desirable scheme that operates in two indistinguishable key generation modes: one for perfectly binding commitment and another for perfectly hiding commitment. The latter mode also generates a trapdoor, enabling the opening of commitments to arbitrary messages. We demonstrate that the local constant extractor E can be adapted into a *trapdoor extractor*, allowing the efficient computation of the "inverse" of E when the related stabilizer under the group action is available as a trapdoor. Building on this extractor, we show that a construction framework for dual-mode commitment can be naturally established by combining our constructions of perfectly binding commitment and perfectly hiding commitment. To the best of our knowledge, this is the first dual-mode commitment construction based on general group actions.

Enhanced Linkable Commitment Based on Group Actions. In [27], the authors introduced the *linkable bit commitment*, which allows a sender to generate two-bit commitments for the same message b to prove to the receiver that these commitments correspond to that message, without revealing the actual content. In this paper, we extend this concept to the *enhanced linkable commitment*, which can disclose the difference $m_1 \cdot m_2^{-1}$ between the committed messages m_1 and $m_2 \in M$ (assuming that M has a group structure). This enhanced linkable commitment inherently demonstrates that c_1 and c_2 commit to the same message by checking whether $m_1 \cdot m_2^{-1} = 1$, and it can also disclose the difference $m_1 \cdot m_2^{-1}$ for distinct messages. This can be viewed as an analogue to the homomorphic commitment [36,21,38,66], which requires that the commitments

c_1 and c_2 for two messages m_1 and m_2 satisfy that $c_1 \cdot c_2^{-1}$ is a commitment to $m_1 \cdot m_2^{-1}$. However, enhanced linkable commitment does not generally possess this homomorphic property because the commitment space C is not required to have a group structure, rendering $c_1 \cdot c_2^{-1}$ undefined.

To establish a construction framework for enhanced linkable commitment, we introduce the concept of a *homomorphic extractor*, which is a deterministic extractor and also a group homomorphism from G to M. We then present two constructions of enhanced linkable commitment schemes using the homomorphic extractor, one based on a perfectly hiding commitment and the other on a perfectly binding commitment. It is worth noting that constructing a homomorphic commitment scheme under post-quantum cryptographic assumptions remains challenging. For example, [69] states that "developing a homomorphic commitment scheme based on isogeny assumptions would signify a significant breakthrough in this domain." From this perspective, enhanced linkable commitments that follow our construction framework may be seen as viable alternatives to homomorphic commitments in many cases.

Instantiating Using Lattices. We instantiate all the proposed commitment schemes using lattices. Specifically, due to the well-established connection between lattices and quadratic forms [26], we consider the action of $\mathrm{GL}_n(\mathbb{Z})/\{\pm\mathbf{I}_n\}$ on positive definite quadratic forms. In this setting, all computational assumptions involved in the constructions correspond to the lattice isomorphism or lattice automorphism problem, whose hardness is extensively studied [43,24,48,49,57] and has been adopted in the construction of various cryptographic schemes [26,25,8,53]. As discussed earlier, the randomization algorithm can be realized based on the random self-reduction technique established in [43,26,8]. For the extractors, we provide a simple yet interesting deterministic extractor based on the determinant. Specifically, we set n to be any even number and define the extractor $\mathsf{E}(\mathbf{U}) = \det(\mathbf{U})$. Since the determinant is a group homomorphism, this extractor is inherently homomorphic as required by the enhanced linkable commitment. Furthermore, we show that the local constant extractor and trapdoor extractor can be adapted from the deterministic extractor by leveraging specific distributions on positive definite quadratic forms or the structure of stabilizer groups. As a result, the corresponding commitment schemes can be instantiated directly by utilizing these randomization algorithms and extractors.

Repair and Improvement of the Non-Interactive Commitment in [27]. In [27], the authors presented the first non-interactive bit commitment scheme based on group actions, instantiated with a special tensor isomorphism problem. However, this instantiation was shown to be insecure in [33]. We introduce a novel method to realize a non-interactive commitment scheme based on the hardness of the lattice isomorphism problem, which is not vulnerable to the attack in [33]. Moreover, our scheme leverages the 'direct sum' hardness of the lattice isomorphism problem, a property not known to be available in other isomorphism problems, enabling a substantial expansion of the message space.

1.2 Related Works

There have been many commitment schemes based on the framework of group actions, with most of these constructions focusing on bit commitments. In [14], Brassard and Yung introduced the first group action-based bit commitment under the assumption of one-way group actions. Ji et al. presented, among other constructions, two interactive bit commitment schemes relying on cryptographic assumptions related to non-abelian group actions [46]. D'Alconzo et al. [27] introduced the first non-interactive linkable bit commitment scheme based on group actions, which was instantiated using a special tensor isomorphism problem.

In [28], Kaafarani et al. presented a lossy identity scheme based on the class group action of CSIDH. Due to the connection between lossy identity schemes and dual commitments established in [64], their construction inherently provides a dual commitment scheme. However, their construction follows the DDH-based framework of commitment schemes, which essentially requires that the group is abelian. In contrast, our framework focuses on general group action-based constructions, which have the potential to be instantiated using a wide variety of cryptographic algebraic structures.

1.3 Outline

Section 2 presents the basic definitions and preliminaries. Section 3 introduces the general commitment framework based on group actions. Section 4 details the construction framework for dual-mode commitments and enhanced linkable commitments. Section 5 provides instantiations of the constructions using lattices. Section 6 discusses the repair and improvement of the non-interactive commitment scheme introduced in [27]. Finally, Section 7 summarizes the paper and discusses some open problems.

2 Preliminary

2.1 Notations

- Let $[n] = \{1, 2, \ldots, n\}$ for a positive integer n. The size of a finite set A is denoted by $|A|$.
- We use $x \leftarrow \mathcal{D}$ to denote that x is sampled from a distribution $\mathcal{D}$. In this paper, we focus solely on discrete distributions. For a finite set S, we write $s \leftarrow S$ to indicate that s is drawn uniformly from S.
- Given two random variables x and x' following distributions $\mathcal{D}$ and $\mathcal{D}'$ over X respectively, their statistical distance is defined as

$$\Delta(x, x') \coloneqq \frac{1}{2} \sum_{a \in X} \left| \Pr[x = a] - \Pr[x' = a] \right| .$$

- For a multivariate distribution $\mathcal{D}$, let $\mathcal{D}[j]$ represent the marginal distribution of the j-th variable.

2.2 Group Action

Definition 2.1 (Group Action) *Let G be a group with identity element e, and let X be a set. We say G acts on X if there is an operator $\star : G \times X \to X$ satisfying $e \star x = x$ and $g \star (h \star x) = (gh) \star x$ for all $g, h \in G$ and $x \in X$. The notation $(G, X, \star)$ will be used to denote such a group action.*

For a group action $(G, X, \star)$, the *orbit* of an element $x \in X$ is denoted by $\mathcal{O}(x) \coloneqq \{g \star x : g \in G\}$. The *stabilizer* of $x \in X$ is the subgroup of G defined as $\mathrm{Stab}(x) \coloneqq \{g \in G : g \star x = x\}$. Additionally, the set $\mathcal{I}(x, y) \coloneqq \{g \in G : g \star x = y\}$ is used to represent the elements of G mapping x to y. It is evident that $\mathcal{I}(x, y) = g \cdot \mathrm{Stab}(x)$ for any $g \in \mathcal{I}(x, y)$.

Cryptographic Assumptions for Group Actions. We briefly introduce the cryptographic assumptions related to group actions as outlined in [1,7,27,14,19]. We assume a group action $(G, X, \star)$ with an associated distribution $\mathcal{D}_{G,X}$ on $G \times X$, which we sometimes denote as $(G, X, \star, \mathcal{D}_{G,X})$ for brevity. We note that both G and X may be infinite, while we always assume that the distribution $\mathcal{D}_{G,X}$ is discrete.

Definition 2.2 (One-Way Group Action) *Let $\mathcal{F}$ be a family of group actions such that for a security parameter λ, $\mathcal{F}(1^\lambda)$ returns a group action $(G, X, \star)$ with distribution $\mathcal{D}_{G,X}$ over $G \times X$. Then $(G, X, \star)$ is said to be $\mathcal{D}_{G,X}$-one-way if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Pr[\mathcal{A}(x, g \star x) \star x = g \star x \mid (g, x) \leftarrow \mathcal{D}_{G,X}] \leq \mathsf{negl}(\lambda).$$

When working with cryptographic constructions, decisional assumptions related to group actions are commonly utilized, such as *Group Action Pseudorandomness* (GAPR) [46] and the *decisional Group Action Inversion Problem* (d-GAIP) [27]. The GAPR problem essentially requires distinguishing whether x' belongs to $\mathcal{O}(x)$ or is random, given $x, x' \in X$. This can be viewed as a generalization of the decisional Diffie-Hellman problem [46]. When the set X consists of only two orbits, this problem is referred to as the 2GAPR problem. The d-GAIP roughly entails determining whether x is in $\mathcal{O}(x_1)$ or in $\mathcal{O}(x_2)$, given that $x \in \mathcal{O}(x_1) \cup \mathcal{O}(x_2)$. It is demonstrated in [27] that there exists a reduction from the 2GAPR problem to d-GAIP when the two orbits are of similar size. In this paper, we further extend d-GAIP to the *decisional Group Action Orbit Problem* (d-GAOP), which requires distinguishing between two classes of orbits. Specifically, given two classes of disjoint orbits $\{\mathcal{O}(x)\}_{x \in X_0}$ and $\{\mathcal{O}(x)\}_{x \in X_1}$, where $X_0, X_1 \subseteq X$, d-GAOP means distinguishing whether x' is in $\cup_{x \in X_0} \mathcal{O}(x)$ or in $\cup_{x \in X_1} \mathcal{O}(x)$ for a given x'. The formal definition is as follows. We note that in this definition, we can utilize two distributions $\mathcal{D}^{(0)}_{G,X}$ and $\mathcal{D}^{(1)}_{G,X}$ to describe the sets of orbits $\{\mathcal{O}(x)\}_{x \in X_0}$ and $\{\mathcal{O}(x)\}_{x \in X_1}$, by letting $\Pr[x_j \in \cup_{x \in X_j} \mathcal{O}(x)] = 1$ for $(g_j, x_j) \leftarrow \mathcal{D}^{(j)}_{G,X}$, where $j \in \{0, 1\}$.

Definition 2.3 (d-GAOP) *Let $\mathcal{F}$ be a family of group actions such that for a security parameter λ, $\mathcal{F}(1^\lambda)$ returns a group action $(G, X, \star)$ with distributions $\mathcal{D}^{(0)}_{G,X}$ and $\mathcal{D}^{(1)}_{G,X}$ over $G \times X$. The d-GAOP assumption requires that for all PPT adversaries $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{c}(g_0, x_0) \leftarrow \mathcal{D}^{(0)}_{G,X}, (g_1, x_1) \leftarrow \mathcal{D}^{(1)}_{G,X} \\ y_0 = g_0 \star x_0, y_1 = g_1 \star x_1 \\ b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(y_b)\end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Additionally, we introduce the *search Group Action Stabilizer Problem* (s-GASP), which relates to finding a non-trivial element in a stabilizer. Specifically, given an $x \in X$ such that $\mathrm{Stab}(x) \neq \{e\}$, the goal is to find an $h \in G$ such that $x = h \star x$ and $h \neq e$. The formal definition is as follows.

Definition 2.4 (s-GASP) *Let $\mathcal{F}$ be a family of group actions such that for a security parameter λ, $\mathcal{F}(1^\lambda)$ returns a group action $(G, X, \star)$ with distribution $\mathcal{D}_{G,X}$ over $G \times X$. The s-GASP assumption requires that for all PPT adversaries $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Pr[\mathcal{A}(y) \star y = y, \mathcal{A}(y) \neq e \mid y = h \star x, (h, x) \leftarrow \mathcal{D}_{G,X}, \mathit{Stab}(x) \neq \{e\}] \leq \mathsf{negl}(\lambda).$$

We note that the problem of finding a non-trivial graph automorphism can be viewed as an instance of s-GASP by considering the action of the permutation group $\mathcal{S}_n$ on a graph with n vertices [58]. This problem currently has no known polynomial-time algorithm [56].

2.3 Commitment Scheme

Definition 2.5 *A commitment scheme Π_{com} is a triple of PPT algorithms* (**Gen**, **Com**, **Open**) *for a security parameter λ and message space M, commitment space C, open space D.*

1. **Gen**: $\mathbf{Gen}(1^\lambda) \to Ck$, *generates the public commitment key Ck.*
2. **Com**: *For any $m \in M$, $\mathbf{Com}_{Ck}(m) \to (c, d) \in C \times D$, $c = c(m)$ is the commitment value and $d = d(m)$ as the opening value.*
3. **Open**: $\mathbf{Open}_{Ck}(c, d) \to \tilde{m} \in M \cup \{\perp\}$, *where $\perp$ is returned if c is not a valid commitment to any message.*

The correctness of a commitment scheme requires that, for any $m \in M$, $\mathbf{Open}_{Ck}(\mathbf{Com}_{Ck}(m)) = m$. The security of a commitment scheme consists of *hiding* and *binding*, which is defined below.

1. **Hiding.** It is computationally hard for an adversary $\mathcal{A}$ to generate two messages $m_0, m_1 \in M$ such that $\mathcal{A}$ can distinguish between their corresponding commitment values c_0, c_1. Formally, for all PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there is a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{c}Ck \leftarrow \mathbf{Gen}\left(1^\lambda\right), (m_0, m_1, \alpha) \leftarrow \mathcal{A}_1(Ck) \\ b \xleftarrow{\$} \{0,1\}, (c, d) \leftarrow \mathbf{Com}_{Ck}\left(m_b\right), \tilde{b} \leftarrow \mathcal{A}_2(c; \alpha)\end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

2. **Binding.** It is computationally hard for an adversary $\mathcal{A}$ to generate a triple (c, d, d'), referred to as a collision, such that (c, d) and (c, d') are valid commitments for m and m' such that $m \neq m'$. Formally, for all PPT adversaies $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr\left[\begin{matrix} m \neq m', \\ m, m' \neq \perp \end{matrix} \middle| \begin{matrix} Ck \leftarrow \mathbf{Gen}\left(1^{\lambda}\right), (c, d, d') \leftarrow \mathcal{A}(Ck) \\ m \leftarrow \mathbf{Open}_{Ck}(c, d), m' \leftarrow \mathbf{Open}_{Ck}\left(c, d'\right) \end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

3 Basic Commitments from Group Actions

This section presents our basic construction framework. We begin by formally introducing the re-randomization algorithm for group actions.

3.1 Re-Randomization for Group Actions

Definition 3.1 (Re-Randomized Algorithm) *For a group action $(G, X, \star)$, a re-randomized algorithm R takes as input $x \in X$ and outputs a pair $(g \star x, g) \in X \times G$ according to a distribution, denoted as $\mathsf{R}(x)$, such that:*

- *For any $x \in X$, $x' \in \mathcal{O}(x)$, and $(x'', g) \leftarrow \mathsf{R}(x)$, the marginal distributions of the first variable are identical for $\mathsf{R}(x)$ and $\mathsf{R}(x')$; g is uniformly distributed on $\mathcal{I}(x, x'')$.*

To be more precise, let $f_x(t_1, t_2)$ be the probability mass function of $\mathsf{R}(x)$. For $x \in X, x', x'' \in \mathcal{O}(x)$, a re-randomized algorithm requires that $\sum_{t_2 \in G} f_x(t_1, t_2) = \sum_{t_2 \in G} f_{x'}(t_1, t_2)$ for any $t_1 \in X$, and that $f_x(t_1 = x'', g)$ is a constant for $g \in \mathcal{I}(x, x'')$. Intuitively, a re-randomized algorithm applies a random $g \in G$ to the input x, effectively 'forgetting' all information about x except for its orbit information.

To facilitate the construction of commitment schemes, we aim to extract the randomness of $g \in \mathcal{I}(x, x'')$ introduced by re-randomization to ensure the hiding property. A simple way to achieve this is by employing a *randomness extractor* [4,42], commonly used in cryptographic constructions. Such an extractor can be naturally integrated into our framework to yield commitment constructions. Before delving into these constructions, we show that some simpler variants of extractors can be defined and used effectively in constructing commitment schemes. Moreover, we demonstrate that these variant extractors are closely related to the structure of group actions and can be extended to construct commitments with advanced functionalities, which we will discuss in Section 4.

3.2 Commitment Scheme Using Deterministic Extractor

Definition 3.2 (Deterministic Extractor) *For a group action $(G, X, \star)$ with a distribution $\mathcal{D}_{G,X}$ on $G \times X$, and M is a finite group. A deterministic extractor is a deterministic and efficient algorithm $\mathsf{E} : G \to M$ such that:*

– *for* $(h, x) \leftarrow \mathcal{D}_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\mathsf{E}(g)$ *is uniformly distributed on* M *for* $g \leftarrow \mathcal{I}(y', y'')$.

We note that when combining with a re-randomized algorithm R for $(G, X, \star)$, it holds that, for $y = h \star x$, $(h, x) \leftarrow \mathcal{D}_{G,X}$, the pairs $(y', \mathsf{E}(g))$ and (y', u) follow the same distribution, where $(y', g) \leftarrow \mathsf{R}(y)$ and $u \leftarrow M$. Intuitively, a deterministic extractor converts the uniformity over the action on the orbit $\mathcal{O}(x)$ into the uniformity over the finite group M, where x is required to sample from $(h, x) \leftarrow \mathcal{D}_{G,X}$ (i.e., some orbits are excluded by the distribution $\mathcal{D}_{G,X}$).

In the following, we demonstrate how a commitment scheme can be constructed using a re-randomized algorithm R and a deterministic extractor E.

Commitment Scheme 3.1 : $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *for a security parameter* λ *and message space* M.

– $\mathbf{Gen}(1^\lambda)$: Sample $(h, x) \leftarrow \mathcal{D}_{G,X}$ and output $Ck = y = h \star x$.
– $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ such that $g \star y = y'$. Compute $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y')$ and set $d = g$. Output (c, d).
– $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d)^{-1} \cdot c_1$; otherwise, output $\perp$.

Theorem 3.1 *Suppose that the group action* $(G, X, \star, \mathcal{D}_{G,X})$ *satisfies the s-GASP assumption,* R *is a re-randomization algorithm, and* E *is a deterministic extractor. Then Commitment 3.1 is perfectly hiding and computationally binding.*

Proof. The correctness is obvious. We prove the hiding and binding below.

Perfectly Hiding: Assume that $\mathbf{Com}_y(m) \to (c, d)$, then the distribution of $c = (\mathsf{E}(g_1) \cdot m, y_1)$ and (u, y_1) are identical, where $u \leftarrow M$ and $y_1 \leftarrow \mathsf{R}(y)[1]$, due to the properties of R and E. Thus for any $m, m' \in M$, $\mathbf{Com}_y(m) \to (c, d)$ and $\mathbf{Com}_y(m') \to (c', d')$, the distribution of the c and c' are the same.

Computationally Binding: Given an s-GASP instance $(h, x) \leftarrow \mathcal{D}_{G,X}$. Send the $Ck = y = h \star x$ to the adversary $\mathcal{A}$ in the binding game, then if $\mathcal{A}$ outputs a commitment $c = (c_1, c_2) = (\mathsf{E}(g_1) \cdot m, y_1) = (\mathsf{E}(g_1') \cdot m', y_1)$ with $m' \neq m$. Then $\mathsf{E}(g_1') \neq \mathsf{E}(g_1)$, thus $g_1 \neq g_1'$ because E is a deterministic algorithm. Note that $g_1, g_1' \in \mathcal{I}(y, y_1)$, so $e \neq g_1^{-1} \cdot g_1' \in \mathrm{Stab}(y)$, this solves the s-GASP assumption. □

3.3 Commitment Scheme Using Local Constant Extractor

In this subsection, we demonstrate how to adapt the deterministic extractor to achieve a perfectly binding and computationally hiding commitment scheme.

Definition 3.3 (Local Constant Extractor) *For a group action* $(G, X, \star)$ *with distributions* $\mathcal{D}_{G,X}$ *and* $\mathcal{D}'_{G,X}$ *on* $G \times X$*, and* M *is a finite group. A local constant extractor is a deterministic and efficient algorithm* $\mathsf{E} : G \to M$ *such that:*

– *For* $(h, x) \leftarrow \mathcal{D}_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\mathsf{E}(g)$ *is uniformly distributed on* M *for* $g \leftarrow \mathcal{I}(y', y'')$.

– *For* $(h, x) \leftarrow \mathcal{D}'_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\mathsf{E}(g)$ *is a constant for* $g \in \mathcal{I}(y', y'')$*.*

Like the deterministic extractor, the local constant extractor can be combined with a re-randomized algorithm R for the group action $(G, X, \star)$. For $y = h \star x$, $(h, x) \leftarrow \mathcal{D}_{G,X}$, it follows that $\mathsf{E}(g)$ is uniform on M for $(y', g) \leftarrow \mathsf{R}(y)$. For $y = h \star x$, $(h, x) \leftarrow \mathcal{D}'_{G,X}$, we have $\mathsf{E}(g_1) = \mathsf{E}(g_2)$ when $(y'_1, g_1), (y'_2, g_2)$ are sampled from $\mathsf{R}(y)$ with $y'_1 = y'_2$. Utilizing this property, we can construct the following commitment scheme.

Commitment Scheme 3.2 : $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *for a security parameter* λ *and message space* M*.*

– $\mathbf{Gen}(1^\lambda)$: Sample $(h, x) \leftarrow \mathcal{D}'_{G,X}$ and output $Ck = h \star x = y$.
– $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ such that $g \star y = y'$. Compute $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y')$ and set $d = g$. Output (c, d).
– $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d)^{-1} \cdot c_1$; otherwise, $\tilde{m} = \perp$.

Theorem 3.2 *Suppose that the group action* $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ *satisfies the d-GAOP assumption,* R *is a re-randomized algorithm, and* E *is a local constant extractor. Then Commitment 3.2 is perfectly binding and computationally hiding.*

Proof. Correctness is trivial. For perfect binding, it is evident due to E being a local constant extractor.

For computational hiding, the proof follows from a standard hybrid argument. Notably, when Ck is sampled from $\mathcal{D}_{G,X}$, Commitment 3.2 reduces to Commitment 3.1, which provides perfect hiding. Therefore, it suffices to replace Ck with the corresponding d-GAOP instance to complete the proof. Details can be found in Appendix A. □

3.4 Commitment Schemes Using Randomness Extractor

As mentioned earlier, the randomness extractor, which can be easily instantiated as discussed in [4,42], can also be applied within our construction framework. In essence, a randomness extractor utilizes a publicly known random seed z to convert a non-uniform random variable on G into a near-uniform random variable on M. We adapt this definition to the context of group actions, as follows.

Definition 3.4 (Randomness Extractor) *For a group action* $(G, X, \star)$ *with a distribution* $\mathcal{D}_{G,X}$*, and* M *is a finite group, a* (G, M, ϵ)*-randomness extractor is a deterministic and efficient algorithm* $\mathsf{E} : G \times \{0,1\}^\zeta \mapsto M$ *such that:*

– *For* $(h, x) \leftarrow \mathcal{D}_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\Delta((z, \mathsf{E}(g, z)), (z, u)) \leqslant \epsilon$*, where the random seed* $z \leftarrow \{0,1\}^\zeta$*,* $u \leftarrow M$*, and* $g \leftarrow \mathcal{I}(y', y'')$*.*

To ensure the existence of a randomness extractor with negligible function $\epsilon(\lambda)$, it is necessary to require that $|M| \leq |\mathcal{I}(y', y'')| = |\mathrm{Stab}(x)|$ for $(h, x) \leftarrow \mathcal{D}_{G,X}$ and any $y', y'' \in \mathcal{O}(x)$. Moreover, similar to the case of deterministic extractors, a randomness extractor can be combined with a re-randomization algorithm R for $(G, X, \star)$. Specifically, for $y = h \star x$, $(h, x) \leftarrow \mathcal{D}_{G,X}$, the statistical distance between $(\mathsf{E}(g, z), y', z)$ and (u, y', z) is at most ϵ, where $(y', g) \leftarrow \mathsf{R}(y)$, $z \leftarrow \{0,1\}^{\zeta}$ and $u \leftarrow M$. This property can be leveraged to construct a statistically hiding and computationally binding commitment scheme, as described below.

Commitment Scheme 3.3 : $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *for a security parameter* λ *and message space* M*.*

- $\mathbf{Gen}(1^{\lambda})$: Sample $(h, x) \leftarrow \mathcal{D}_{G,X}$ and output $Ck = y = h \star x$.
- $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ s.t. $g \star y = y'$, $z \leftarrow \{0,1\}^{\zeta}$. Compute $c = (c_1, c_2, c_3) = (\mathsf{E}(g, z) \cdot m, y', z)$ and set $d = g$. Output (c, d).
- $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d, c_3)^{-1} \cdot c_1$; otherwise, output $\tilde{m} = \perp$.

We note that the randomness extractor was employed in [42] to construct a constant-round commitment, which leverages the compressibility of collision-resistant hash functions to guarantee the min-entropy of the input variable. In contrast, in our construction, the min-entropy of the input arises from its uniformity over $\mathcal{I}(y, y')$, which is ensured by the re-randomization algorithm R.

Theorem 3.3 *Suppose the group action* $(G, X, \star, \mathcal{D}_{G,X})$ *satisfies the s-GASP assumption,* R *is a re-randomized algorithm, and* E *is a randomness extractor. Then Commitment 3.3 is statistically hiding and computationally binding.*

Proof. The proof is similar to Theorem 3.1 and can be found in Appendix A. □

Next, we show that Commitment 3.3 can be adapted to yield a perfectly binding and computationally hiding commitment, similar to the approach in Section 3.3. However, in Section 3.3, the extractor is required to satisfy a 'local constant' property, meaning that $\mathsf{E}(g)$ remains constant for $g \in \mathcal{I}(y', y'')$, where $(h, x) \leftarrow \mathcal{D}'_{G,X}$ and any $y', y'' \in \mathcal{O}(x)$. This property is challenging to reconcile with the randomness extractor in Definition 3.4 due to the influence of the random seed. To address this, we impose stronger restrictions on the group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$, specifically requiring that $|\mathrm{Stab}(x)| = 1$ for $(h, x) \leftarrow \mathcal{D}'_{G,X}$. In addition, we require that for the group action $(G, X, \star, \mathcal{D}_{G,X})$ and a finite group M, E is a (G, M, ϵ)-randomness extractor. This means that for $(h, x) \leftarrow \mathcal{D}_{G,X}$ and any $y', y'' \in \mathcal{O}(x)$, it holds that $\Delta((z, \mathsf{E}(g, z)), (z, u)) \leqslant \epsilon$, where the random seed $z \leftarrow \{0,1\}^{\zeta}$, $u \leftarrow M$, and $g \leftarrow \mathcal{I}(y', y'')$. With these restrictions in place, we can construct the following perfectly binding and computationally hiding commitment scheme.

Commitment Scheme 3.4 : $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *for a security parameter* λ *and message space* M*.*

- $\mathbf{Gen}(1^\lambda)$: Sample $(h, x) \leftarrow \mathcal{D}'_{G,X}$ and output $Ck = y = h \star x$.
- $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ s.t. $g \star y = y'$, $z \leftarrow \{0,1\}^\zeta$. Compute $c = (c_1, c_2, c_3) = (\mathsf{E}(g, z) \cdot m, y', z)$ and set $d = g$. Output (c, d).
- $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d, c_3)^{-1} \cdot c_1$; otherwise, output $\tilde{m} = \perp$.

Theorem 3.4 *Suppose that the group action* $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ *satisfies the d-GAOP assumption, where the distribution* $\mathcal{D}'_{G,X}$ *satisfies* $|Stab(x)| = 1$ *for* $(h, x) \leftarrow \mathcal{D}'_{G,X}$*.* R *is a re-randomized algorithm, and* E *is a randomness extractor for distribution* $\mathcal{D}_{G,X}$*. Then Commitment 3.4 is perfectly binding and computationally hiding.*

Proof. The proof is similar to Theorem 3.2 and can be found in Appendix A. □

4 Dual-Mode Commitment and Enhanced-Linkable Commitments from Group Actions

In many cases, it is desirable for commitment schemes to possess additional properties beyond hiding and binding, such as dual-mode commitment, enhanced-linkable commitment. Specifically, the dual mode commitment was proposed in [16] and plays an important role in zero-knowledge proof and security proof [55,51,64]; the enhanced-linkable commitment introduced in this paper can be seen as an intermediate between the linkable commitment [27] and the homomorphic commitment [66,36].

In this section, we demonstrate how to use the basic commitments in Section 3 to construct more versatile commitments, i.e. the dual-mode commitment and the enhanced-linkable commitment.

4.1 The Dual-Mode Commitment

We present the definition of a dual-mode commitment scheme following [64,55].

Definition 4.1 (Dual-Mode Commitment) A dual-mode commitment scheme is a tuple of PPT algorithms $\Pi_{dmc} = (\mathbf{Gen}, \mathbf{TGen}, \mathbf{Com}, \mathbf{TCom}, \mathbf{Open}, \mathbf{TCol})$, for a security parameter λ, message space M, commitment space C, and opening space D, such that

1. $(\mathbf{TGen}, \mathbf{TCom}, \mathbf{TCol})$ satisfies:
   - $\mathbf{TGen}(1^\lambda)$ outputs the public commitment key Ck and corresponding trapdoor information Td.
   - $\mathbf{TCom}(Ck, Td)$ outputs a commitment value $c \in C$ and a state St.
   - $\mathbf{TCol}(Ck, Td, St, m)$ outputs an opening value $d \in D$.

2. **Perfectly Binding**: (**Gen**, **Com**, **Open**) is a perfectly binding commitment scheme.
3. **Completeness**: For $(Ck, Td) \leftarrow \mathbf{TGen}$ and $m \in M$, it holds that

$$\Pr[\mathbf{Open}_{Ck}(c, d) = m \mid (c, d) \leftarrow \mathbf{Com}_{Ck}(m)] = 1.$$

4. **Trapdoor Property**: For $(Ck, Td) \leftarrow \mathbf{TGen}$ and $m \in M$, the following distributions are identical:

$$\{(c, d, m) \mid (c, d) \leftarrow \mathbf{Com}_{Ck}(m)\} \quad \text{and}$$

$$\{(c, d, m) \mid (c, St) \leftarrow \mathbf{TCom}(Ck, Td),\ d \leftarrow \mathbf{TCol}(Ck, Td, St, m)\}.$$

5. **Key Indistinguishability**: For all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{matrix} Ck_0 \leftarrow \mathbf{Gen}(1^\lambda), Ck_1 \leftarrow \mathbf{TGen}(1^\lambda) \\ b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(Ck_b) \end{matrix}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

According to the definition, it is natural to consider combining a perfectly binding commitment and a perfectly hiding commitment, such as Commitment 3.2 and Commitment 3.1, to satisfy the requirements of a dual-mode commitment. However, to achieve the trapdoor property, we need to adapt the local constant extractor into a *trapdoor extractor*, which is defined as follows.

Definition 4.2 (Trapdoor Extractor) *For a group action $(G, X, \star)$ with distributions $\mathcal{D}_{G,X}$ and $\mathcal{D}'_{G,X}$ on $G \times X$, and a finite group M, a trapdoor extractor is a tuple of algorithms (E, F), where E is the local constant extractor defined in Definition 3.3 with respect to the distributions $\mathcal{D}_{G,X}$ and $\mathcal{D}'_{G,X}$, and F is a PPT algorithm that takes $(u, g, x, Stab(x)) \in M \times G \times X \times \mathcal{P}(G)$ as input and outputs $g' \in G$, such that:*

- *For $(h, x) \leftarrow \mathcal{D}_{G,X}$ and any $u \in M$, $y', y'' \in \mathcal{O}(x)$, $g \in \mathcal{I}(y', y'')$, the output g' of $\mathsf{F}(u, g, y', Stab(y'))$ satisfies $g' \in \mathcal{I}(y', y'')$ and $\mathsf{E}(g') = u$.*
- *The distributions of $(\mathsf{E}(g), g)$ and (u, g') are identical for any $y', y'' \in \mathcal{O}(x)$, where $g \leftarrow \mathcal{I}(y', y'')$, $u \leftarrow M$, $g' \leftarrow \mathsf{F}(u, g, y', Stab(y'))$, and $(h, x) \leftarrow \mathcal{D}_{G,X}$.*

We note that the input $\mathrm{Stab}(y')$ for F serves as a trapdoor for computing the 'inverse' of E. Additionally, we require that $\mathrm{Stab}(y')$ for any involved $y' \in \mathcal{O}(x)$ can be generated by a polynomial number of elements, ensuring that $\mathrm{Stab}(y')$ can be represented in polynomial size. When combined with a re-randomization algorithm R for the group action $(G, X, \star)$, it follows from the definition that the distributions of $(\mathsf{E}(g), y', g)$ and (u, y', g') are identical for $(y', g) \leftarrow \mathsf{R}(y)$, $y = h \star x$, $(h, x) \leftarrow \mathcal{D}_{G,X}$, and $u \leftarrow M$, $g' \leftarrow \mathsf{F}(u, g, y, \mathrm{Stab}(y))$. Using the trapdoor extractor, we can construct the following dual-mode commitment scheme.

Commitment Scheme 4.1 : $\Pi_{dmc} = (\mathbf{Gen}, \mathbf{TGen}, \mathbf{Com}, \mathbf{TCom}, \mathbf{Open}, \mathbf{TCol})$, *for a security parameter* λ.

- $\mathbf{Gen}(1^\lambda)$: Sample $(h, x) \leftarrow \mathcal{D}'_{G,X}$ and ouput $Ck = h \star x = y$.
- $\mathbf{TGen}(1^\lambda)$: Sample $(h, x) \leftarrow \mathcal{D}_{G,X}$ and compute $\mathrm{Stab}(x)$ from x. Then, obtain $\mathrm{Stab}(y) = h \cdot \mathrm{Stab}(x) \cdot h^{-1}$, where $y = h \star x$. Finally, output $(Ck, Td) = (y, \mathrm{Stab}(y))$.
- $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ such that $g \star y = y'$. Compute $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y')$ and set $d = g$. Output (c, d).
- $\mathbf{TCom}(y, \mathrm{Stab}(y))$: Sample $(y', g) \leftarrow \mathsf{R}(y)$, $u \leftarrow M$ and set $c = (c_1, c_2) = (u, y')$, $St = (u, g)$. Output (c, St).
- $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d)^{-1} \cdot c_1$; otherwise, $\tilde{m} = \perp$.
- $\mathbf{TCol}(y, \mathrm{Stab}(y), (u, g), m)$: Compute $r = u \cdot m^{-1}$, then sample $\mathsf{F}(r, g, y, \mathrm{Stab}(y)) \to g'$ and set $d = g'$. Output d.

Theorem 4.1 *Suppose that the group action* $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ *satisfies the d-GAOP assumption, where it is efficient to compute* $Stab(x)$ *from* x *given* $(h, x) \leftarrow \mathcal{D}_{G,X}$. R *is a re-randomized algorithm, and* (E, F) *is a trapdoor extractor. Then Commitment 4.1 is a dual mode commitment.*

Proof. The **Completeness** and **Perfectly Binding** properties are self-evident, as illustrated in Theorem 3.2. Regarding the **Trapdoor Property**, it is apparent due to (E, F) serving as a trapdoor extractor. Concerning **Key Indistinguishability**, it can be straightforwardly reduced to the d-GAOP assumption. The detailed proof is outlined in Appendix A. □

4.2 The Enhanced-Linkable Commitment

Definition 4.3 (Enhanced Linkable Commitment) *Let* λ *be the security parameter,* $(M, \cdot)$ *be a finite group representing the message space, and* (C, D) *be the commitment and opening spaces, respectively. An enhanced linkable commitment scheme is a tuple* $\Pi_{elc} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open}, (\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC}))$, *where* $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *is a commitment scheme, and* $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ *is the link component. The scheme satisfies the following:*

1. *The link component consists of PPT algorithms* $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ *such that for two messages* $m_0, m_1 \in M$, *and* $(c_0, d_0) \leftarrow \mathbf{Com}(m_0)$, $(c_1, d_1) \leftarrow \mathbf{Com}(m_1)$, *the following hold:*
   - $\mathsf{Link}(c_0, c_1, d_L)$ *outputs* 0 *or* 1. *If* $\mathsf{Link}(c_0, c_1, d_L) = 1$, *we say that* d_L *is a linking value for the pair* (c_0, c_1).
   - $\mathsf{LinkE}(d_0, d_1)$ *outputs a linking value* d_L *such that* $\mathsf{Link}(c_0, c_1, d_L) = 1$. *That is, given the open values* (d_0, d_1) *for* (c_0, c_1), LinkE *extracts a linking value* d_L *for the pair* (c_0, c_1).
   - $\mathsf{LinkC}(c_0, c_1, d_L)$ *outputs* $m_0 \cdot m_1^{-1}$ *if* $\mathsf{Link}(c_0, c_1, d_L) = 1$, *and outputs* $\perp$ *otherwise. Thus, with the linking value* d_L *for* (c_0, c_1), LinkC *computes* $m_0 \cdot m_1^{-1}$.

2. ***Computationally Enhanced-Linkable-Hiding (ELH)****: For all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\mathcal{A} \text{ wins } \boldsymbol{ELH}(\Pi_{elc})\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Here, $\boldsymbol{ELH}(\Pi_{elc})$ *represents the game described in Figure 1.*

3. ***Computationally Enhanced-Linkable-Binding (ELB)****: For all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\begin{array}{c|c} \mathbf{Open}(c_i, d_i) = m_i,\ i \in \{0,1\}, & Ck \leftarrow \mathbf{Gen}(1^\lambda), \\ \mathsf{Link}(c_0, c_1, d_L) = 1, & \begin{pmatrix} m_0, m_1, c_0, c_1 \\ d_0, d_1, d_L \end{pmatrix} \leftarrow \mathcal{A}(Ck) \\ m_0 \cdot m_1^{-1} \neq \mathsf{LinkC}(c_0, c_1, d_L) & \end{array}\right] \leq \mathsf{negl}(\lambda).$$

4. ***Computationally Enhanced-Linkable-Unforgeable (ELU)***[1]*: For all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\mathcal{A} \text{ wins } \boldsymbol{ELU}(\Pi_{elc})\right] \leq \mathsf{negl}(\lambda).$$

Here, $\boldsymbol{ELU}(\Pi_{elc})$ *represents the game described in Figure 2.*

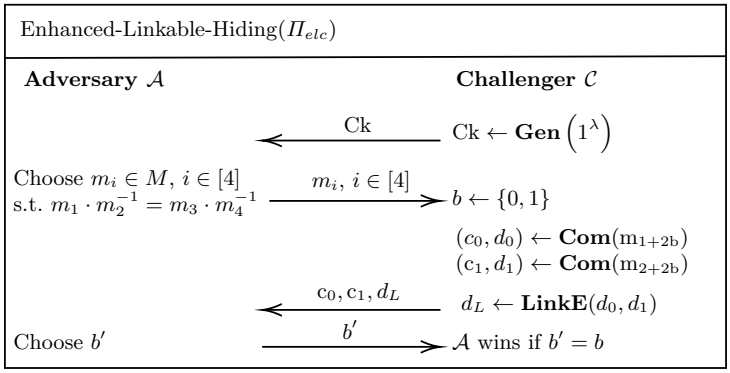

Fig. 1: The Enhanced Linkable Hiding Game

Intuitively, the **ELH** property requires that the linking value d_L for the pair (c_0, c_1) reveals only the information $m_0 \cdot m_1^{-1}$. This implies that an adversary with access to the commitments c_0 and c_1 of messages m_0 and m_1, along with the linking material d_L, gains no insight into m_0 and m_1 other than $m_0 \cdot m_1^{-1}$. The **ELB** property requires that it is infeasible for the sender to produce a fake linking value d_L such that $\mathsf{Link}(c_0, c_1, d_L) = 1$ and $m_0 \cdot m_1^{-1} \neq \mathsf{LinkC}(c_0, c_1, d_L)$. The **ELU** property requires that no PPT adversary can craft a linking value d_L for pair (c_0, c_1) without possessing any information about d_0 and d_1, where $(c_0, d_0) \leftarrow \mathbf{Com}(m_0)$ and $(c_1, d_1) \leftarrow \mathbf{Com}(m_1)$. Next, we demonstrate adjusting the deterministic extractor E to achieve an Enhanced Linkable Commitment.

[1] This property is referred to as computational link secrecy in [27].

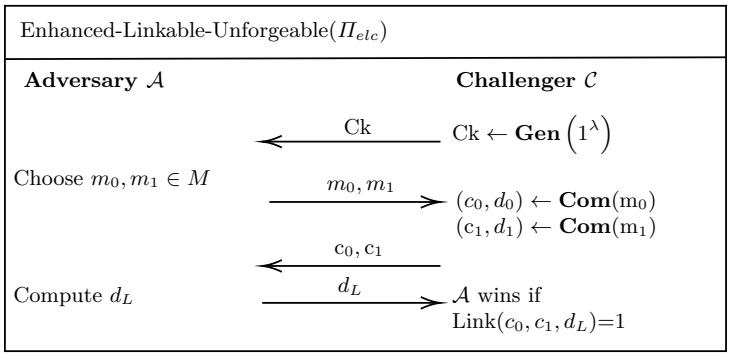

Fig. 2: The Enhanced Linkable Unforgeable Game

Definition 4.4 (Homomorphic Extractor) *For a group action $(G, X, \star)$ with a distribution $\mathcal{D}_{G,X}$ in $G \times X$, and M is an abelian group. A homomorphic extractor is a deterministic and efficient algorithm $\mathsf{E} : G \to M$ such that*

- *for $(h, x) \leftarrow \mathcal{D}_{G,X}$ and any $y', y'' \in \mathcal{O}(x)$, it holds that $\mathsf{E}(g)$ is uniformly distributed on M for $g \leftarrow \mathcal{I}(y', y'')$.*
- *$\mathsf{E} : G \to M$ is a surjective group homomorphism, i.e, for any $g_0, g_1 \in G$, $\mathsf{E}(g_0) \cdot \mathsf{E}(g_1)^{-1} = \mathsf{E}(g_0 \cdot g_1^{-1})$*

Then we define $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ in Definition 4.3 to correspond to Commitment 3.1.

Commitment Scheme 4.2 : $\Pi_{elc} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open}, (\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC}))$, *for a security parameter λ.*

- Π_{com} =(**Gen**, **Com**, **Open**) is Commitment 3.1.

If $Ck = y$, for two commitments of m_0 and m_1 in Commitment 3.1:

$$\mathbf{Com}_y(m_0) \to (c_0, d_0) = ((c_{01}, c_{02}), d_0) = ((\mathsf{E}(g_0) \cdot m_0, y_0), g_0)$$
$$\mathbf{Com}_y(m_1) \to (c_1, d_1) = ((c_{11}, c_{12}), d_1) = ((\mathsf{E}(g_1) \cdot m_1, y_1), g_1)$$

The link component $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ as follows:

- $\mathsf{Link}(c_0, c_1, d_L) = \begin{cases} 1 & \text{if } d_L \star c_{12} = c_{02}. \\ 0 & else. \end{cases}$
- $\mathsf{LinkE}(d_0, d_1) = d_0 \cdot d_1^{-1}$.
- $\mathsf{LinkC}(c_0, c_1, d_L)$: If $d_L \star c_{12} = c_{02}$, it outputs $\mathsf{E}(d_L)^{-1} \cdot c_{01} \cdot c_{11}^{-1}$; otherwise, it outputs $\perp$.

Note that if E is a homomorphic extractor and $d_L = g_0 \cdot g_1^{-1}$, then:

$$\mathsf{E}(d_L)^{-1} \cdot c_{01} \cdot c_{11}^{-1} = \mathsf{E}(g_0 \cdot g_1^{-1})^{-1} \cdot \mathsf{E}(g_0) \cdot m_0 \cdot (\mathsf{E}(g_1) \cdot m_1)^{-1} = m_0 \cdot m_1^{-1}.$$

where we have used the fact that M is an abelian group.

Theorem 4.2 *Suppose that group action* $(G, X, \star, \mathcal{D}_{G,X})$ *satisfies the s-GASP assumption,* R *is a re-randomization algorithm, and* E *is a homomorphic extractor. Then Commitment 4.2 is an Enhanced Linkable Commitment.*

Proof. We have proved that the Commitment 3.1 is secure under the s-GASP assumption. Thus, we only need to establish the security of the enhanced linkable commitment scheme. Enhanced linkable hiding can be straightforwardly derived from the properties of R and E, while enhanced linkable binding can be directly reduced to s-GASP. Moreover, enhanced linkable unforgeability can be reduced to s-GASP with the assistance of R. The detailed proof is outlined in Appendix A. □

Similarly, we can adjust the local constant extractor E to a homomorphic local constant extractor E so that if we have such E, we can transform Commitment 3.2 into an Enhanced Linkable Commitment.

Definition 4.5 (Homomorphic Local Constant Extractor) *For a group action* $(G, X, \star)$ *with distributions* $\mathcal{D}_{G,X}$ *and* $\mathcal{D}'_{G,X}$ *on* $G \times X$*, and* M *is an abelian group. A homomorphic local constant extractor is a deterministic and efficient algorithm* $\mathsf{E} : G \to M$ *such that:*

- *For* $(h, x) \leftarrow \mathcal{D}_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\mathsf{E}(g)$ *is uniformly distributed on* M *for* $g \leftarrow \mathcal{I}(y', y'')$*.*
- *For* $(h, x) \leftarrow \mathcal{D}'_{G,X}$ *and any* $y', y'' \in \mathcal{O}(x)$*, it holds that* $\mathsf{E}(g)$ *is a constant for* $g \in \mathcal{I}(y', y'')$*.*
- $\mathsf{E} : G \to M$ *is a surjective group homomorphism, i.e, for any* $g_0, g_1 \in G$*,* $\mathsf{E}(g_0) \cdot \mathsf{E}(g_1)^{-1} = \mathsf{E}(g_0 \cdot g_1^{-1})$

Commitment Scheme 4.3 : $\Pi_{elc} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open}, (\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC}))$*, for a security parameter* λ*.*

- $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ is Commitment 3.2.
- $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ is the same defined in Commitment 4.2.

Theorem 4.3 *Suppose that group action* $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ *satisfies the d-GAOP assumption and* $\mathcal{D}_{G,X}$*-one-way,* R *is a re-randomization algorithm, and* E *is a homomorphic local constant extractor. Then Commitment 4.3 is an Enhanced Linkable Commitment.*

Proof. This proof essentially uses the same techniques in Theorem 4.2 and Theorem 3.2, so we omit it here. The full proof can be found in Appendix A. □

5 Instantiation of the Commitments with Lattices

In this section, we demonstrate how to instantiate the commitment schemes introduced in Section 3 and Section 4 using lattices. We introduce some basic notations and definitions, which will be used throughout Section 5 and Section 6.

- Matrices and column vectors are denoted by bold letters, such as $\mathbf{A}$ and $\mathbf{a}$. The transpose of $\mathbf{A}$ is denoted by $\mathbf{A}^\top$, and $(\mathbf{A}^{-1})^\top$ is abbreviated as $\mathbf{A}^{-\top}$. The Euclidean norm of $\mathbf{a} \in \mathbb{R}^n$ is represented as $\|\mathbf{a}\|$.
- Let $\mathrm{GL}_n(\mathbb{Z})$ denote the general linear group of rank n over $\mathbb{Z}$, and let $\mathrm{GL}_n^{\pm}(\mathbb{Z})$ denote the quotient group $\mathrm{GL}_n(\mathbb{Z})/\{\pm\mathbf{I}_n\}$.
- Let $\mathrm{O}_n(\mathbb{R})$ denote the group of orthogonal matrices $\mathbf{O} \in \mathrm{GL}_n(\mathbb{R})$ such that $\mathbf{O}^\top\mathbf{O} = \mathbf{I}_n$, where $\mathbf{I}_n$ is the identity matrix.
- Let $\mathcal{S}_n^{>0}(\mathbb{Z})$ denote the set of $n \times n$ positive definite matrices over $\mathbb{Z}$.

5.1 Definitions Related to Lattices

A lattice $\mathcal{L}$ of rank n and dimension m is a set of points in $\mathbb{R}^m$ that can be expressed as integer combinations of n linearly independent basis vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Denote the basis of the lattice $\mathcal{L}$ as $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, then

$$\mathcal{L} \coloneqq \mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^n\}.$$

We focus on full-rank lattices, where $m = n$ and $\mathbf{B} \in \mathrm{GL}_n(\mathbb{R})$. The dual lattice of $\mathcal{L}$ is defined as $\mathcal{L}^* \coloneqq \{\mathbf{u} \in \mathbb{R}^n : \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z} \text{ for all } \mathbf{v} \in \mathcal{L}\}$. A lattice $\mathcal{L}$ is said to be *unimodular* if $\mathcal{L} = \mathcal{L}^*$. A unimodular lattice is called *even* if all its vectors have an even squared norm, and *odd* otherwise.

Lattice Isomorphism and Automorphism. Two n-dimensional lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are called isomorphic if there exists an orthogonal matrix $\mathbf{O} \in \mathrm{O}_n(\mathbb{R})$ such that $\mathcal{L}_2 = \{\mathbf{Ov} : \mathbf{v} \in \mathcal{L}_1\}$, which we denote as $\mathcal{L}_1 \cong \mathcal{L}_2$ or $\mathcal{L}_1 = \mathbf{O} \cdot \mathcal{L}_2$. For any lattice $\mathcal{L}$, we denote its isomorphism class by $[\mathcal{L}] = \{\mathbf{O} \cdot \mathcal{L} : \mathbf{O} \in \mathrm{O}_n(\mathbb{R})\}$. A related problem is the Lattice Isomorphism Problem (LIP) [43,26], which involves finding an isomorphism $\mathbf{O} \in \mathrm{O}_n(\mathbb{R})$ such that $\mathcal{L} = \mathbf{O} \cdot \mathcal{L}'$ for two given isomorphic lattices $\mathcal{L}$ and $\mathcal{L}'$.

The automorphism group $\mathrm{Aut}(\mathcal{L})$ of an n-dimensional lattice $\mathcal{L}$ consists of all orthogonal matrices that preserve $\mathcal{L}$, i.e., $\mathrm{Aut}(\mathcal{L}) = \{\mathbf{O} \in \mathrm{O}_n(\mathbb{R}) : \mathbf{Ov} \in \mathcal{L} \text{ for all } \mathbf{v} \in \mathcal{L}\}$. It is evident that $\mathrm{Aut}(\mathcal{L})$ includes the trivial automorphisms $\pm\mathbf{I}_n$. A relevant problem is the Lattice Automorphism Problem (LAP) [47,57], which involves finding a non-trivial automorphism $\mathbf{O} \in \mathrm{Aut}(\mathcal{L})$ for a given lattice $\mathcal{L}$.

Definition 5.1 (Reducible & Irreducible Lattices) *A lattice $\mathcal{L}$ is said to be reducible if it is isomorphic to a lattice of the form $\mathcal{L}_1 \oplus \mathcal{L}_2$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are lattices of dimension at least 1. If $\mathcal{L}$ is not reducible, it is called irreducible.*

Group Action Based on Lattices. Let $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ be a positive definite quadratic form over $\mathbb{Z}$, and let $[\mathbf{Q}]$ denote the set of quadratic forms equivalent to $\mathbf{Q}$, i.e. $[\mathbf{Q}] \coloneqq \{\mathbf{VQV}^\top : \mathbf{V} \in \mathrm{GL}_n(\mathbb{Z})\}$. A group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}], \star)$ can then be defined as [2]:

$$\mathbf{V} \star \mathbf{Q}' = \mathbf{VQ}'\mathbf{V}^\top \quad \text{for any } \mathbf{V} \in \mathrm{GL}_n^{\pm}(\mathbb{Z}), \mathbf{Q}' \in [\mathbf{Q}]. \tag{1}$$

[2] As observed in [7], the choice of representative in $\mathrm{GL}_n^{\pm}(\mathbb{Z})$ does not matter for this group action since $\mathbf{VQV}^\top = (-\mathbf{V})\mathbf{Q}(-\mathbf{V})^\top$.

This group action is closely related to lattice isomorphisms and automorphisms. Specifically, given two isomorphic n-dimensional lattices $\mathcal{L} \cong \mathcal{L}'$ with respective bases $\mathbf{B}$ and $\mathbf{B}'$, define $\mathbf{Q} = \mathbf{B}^\top\mathbf{B}$ and $\mathbf{Q}' = \mathbf{B}'^\top\mathbf{B}' \in \mathcal{S}_n^{>0}(\mathbb{Z})$. For any isomorphism $\mathbf{O}$ from $\mathcal{L}'$ to $\mathcal{L}$, there exists a unique matrix $\mathbf{V}^\top \in \mathrm{GL}_n(\mathbb{Z})$ such that $\mathbf{OB}' = \mathbf{BV}^\top$. This implies $\mathbf{VQV}^\top = (\mathbf{BV}^\top)^\top(\mathbf{BV}) = (\mathbf{OB}')^\top(\mathbf{OB}') = \mathbf{Q}'$, so we have $\mathbf{Q}' \in [\mathbf{Q}]$. The above correspondence defines a one-to-one map between $\mathcal{I}(\mathbf{Q}, \mathbf{Q}')$ and the set of isomorphisms from $\mathcal{L}'$ to $\mathcal{L}$ modulo the relation $(-\mathbf{O}) \sim \mathbf{O}$ [7]. Additionally, it has been shown in [47] that the map $\mathbf{V}^\top \mapsto \mathbf{BV}^\top\mathbf{B}^{-1}$ defines an isomorphism from $\mathrm{Stab}(\mathbf{Q})$ to $\mathrm{Aut}(\mathcal{L})/\{\pm\mathbf{I}_n\}$.

On the other hand, every quadratic form $\mathbf{Q}$ induces a unique upper-triangular lattice basis $\mathbf{B}_\mathbf{Q}$ such that $\mathbf{Q} = \mathbf{B}_\mathbf{Q}^\top\mathbf{B}_\mathbf{Q}$ through Cholesky decomposition, implying an efficient conversion between quadratic forms and lattices. Thus, lattice isomorphisms and automorphisms can be represented in two equivalent forms: lattice basis and quadratic form. In this paper, we primarily use quadratic forms to describe lattice problems.

Definition 5.2 (Lattice Isomorphism Problem, LIP) *Given two quadratic forms $\mathbf{Q}$ and $\mathbf{Q}'$ associated with isomorphic lattices $\mathcal{L}$ and $\mathcal{L}'$, the objective is to find an isomorphism $\mathbf{U} \in GL_n(\mathbb{Z})$ such that $\mathbf{UQU}^\top = \mathbf{Q}'$. When $\mathcal{L} \cong \mathbb{Z}^n$, this problem is referred to as the $\mathbb{Z}$LIP.*

Definition 5.3 (Decisional LIP$(\mathcal{L}_0, \mathcal{L}_1)$) [3] *Let $\mathcal{L}_0$ and $\mathcal{L}_1$ be non-isomorphic lattices, with the corresponding quadratic forms $\mathbf{Q}_0$ and $\mathbf{Q}_1$. Given a quadratic form $\mathbf{Q} \in [\mathbf{Q}_b]$, where $b \in \{0, 1\}$ is a uniformly random bit, the objective is to determine the value of b.*

Definition 5.4 (Lattice Automorphism Problem, LAP) *Given a quadratic form $\mathbf{Q}$ of a lattice $\mathcal{L}$ where $\mathrm{Aut}(\mathcal{L}) \neq \{\pm\mathbf{I}_n\}$, the objective is to find an automorphism $\mathbf{U} \in \mathrm{Stab}(\mathbf{Q})$ such that $\mathbf{U} \neq \pm\mathbf{I}_n$. When $\mathcal{L} \cong \mathbb{Z}^n$, this problem is referred to as the $\mathbb{Z}$LAP.*

Both LIP and LAP are considered computationally hard problems. LIP has been applied in the design of various cryptographic schemes [26,25,8,53]. Furthermore, it has been demonstrated in [47] that $\mathbb{Z}$LAP is equivalent to $\mathbb{Z}$LIP.

The Genus of a Lattice. The genus is a classification of lattices under $\mathbb{Z}_p$-equivalence, where $\mathbb{Z}_p$ represents the p-adic integers. Specifically, two lattices $\mathcal{L}_1$ and $\mathcal{L}_2 \subset \mathbb{R}^n$ are said to be $\mathbb{Z}_p$-equivalent if there exists a matrix $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z}_p)$ such that $\mathbf{UQ}_1\mathbf{U}^\top = \mathbf{Q}_2$, where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are the corresponding quadratic forms of $\mathcal{L}_1$ and $\mathcal{L}_2$.

Definition 5.5 (Genus [72]) *For an n-dimensional integral lattice $\mathcal{L}$, genus$(\mathcal{L})$ consists of all integral lattices that are $\mathbb{Z}_p$-equivalent to $\mathcal{L}$ for all primes p.*

[3] In earlier LIP-based works, 'Decisional LIP' has been referred to as 'Distinguish LIP'.

Given an integral lattice $\mathcal{L}$, a canonical label for genus($\mathcal{L}$) can be efficiently computed if the prime factorization of $\det(\mathcal{L})^2$ is known. Thus, for LIP, it is essential that the input lattices belong to the same genus.

Automorphism Group of $\mathbb{Z}^n$. The automorphism group of $\mathbb{Z}^n$ consists of signed permutation matrices of size $n \times n$, which have exactly one nonzero entry per row and column, and the nonzero entries are either ± 1. We denote this group as $\mathcal{S}_n^{\pm}$, and the group of permutation matrices of size $n \times n$ as $\mathcal{S}_n$.

It is known that $\mathcal{S}_n^{\pm}$ is the semi-direct product $D_n \rtimes \mathcal{S}_n$, where D_n represents the group of signed matrices, i.e., diagonal matrices with diagonal entries ± 1. Besides, we denote the standard basis of $\mathbb{Z}^n$ as $\{\mathbf{e}_i\}_{i\in[n]}$.

Lattices with Specific Automorphism Groups. We introduce the following types of lattices, which will be used in our construction. The existence of these lattices will be discussed later.

(1) Λ_0^n is the set of n-dimensional lattices with a trivial automorphism group, i.e., $\Lambda_0^n := \{\mathcal{L} : \mathrm{Aut}(\mathcal{L}) = \{\pm\mathbf{I}_n\}\}$.

(2) Λ_k^n, where $k > 0$, is the set of n-dimensional lattices with an automorphism group of size at least 2^{k+1}, i.e., $\Lambda_k^n := \{\mathcal{L} : |\mathrm{Aut}(\mathcal{L})| \geq 2^{k+1}\}$.

(3) Λ_+^n is the set of n-dimensional lattices where all automorphisms have determinant 1, i.e., $\Lambda_+^n := \{\mathcal{L} : \det(\mathbf{O}) = 1 \text{ for all } \mathbf{O} \in \mathrm{Aut}(\mathcal{L})\}$.

(4) Λ_-^n is the set of n-dimensional lattices where the number of automorphisms with determinant 1 equals the number of automorphisms with determinant -1, i.e., $\Lambda_-^n := \{\mathcal{L} : |A_1^{(\mathcal{L})}| = |A_{-1}^{(\mathcal{L})}|\}$, where $A_i^{(\mathcal{L})} = \{\mathbf{O} \in \mathrm{Aut}(\mathcal{L}) : \det(\mathbf{O}) = i\}$.

5.2 Instantiating the Re-Randomization Algorithm

As mentioned earlier, the re-randomization algorithm R fundamentally arises from the self-reduction of LIP. Intuitively, this self-reduction process leverages the well-known Gaussian sampling algorithm on lattices [32,13], which effectively conceals the details of the lattice basis while preserving the geometric structure of the lattice. There are two similar methods to realize this process, both of which can be adapted for the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}], \star)$ in the instantiation of the re-randomization algorithm R, described as follows.

The first method, explained in terms of the lattice basis [8,47], proceeds as follows. Given an input $\mathbf{Q}$, the algorithm R first computes the corresponding lattice basis $\mathbf{B}$. It then applies the LLL algorithm to remove the length information from the lattice basis $\mathbf{B}$. Next, it utilizes the Gaussian sampling algorithm on the lattice $\mathcal{L}(\mathbf{B})$ to generate $m = \mathsf{poly}(n)$ lattice vectors $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ such that V spans $\mathcal{L}(\mathbf{B})$ with overwhelming probability [43, Lemma 5.4]. Afterward, a uniformly random orthogonal matrix $\mathbf{O}$ is sampled, and the LLL algorithm is applied to the set $\mathbf{O} \cdot V$ to obtain a lattice basis $\mathbf{B}'$ and a transition matrix $\mathbf{U}^\top \in \mathrm{GL}_n(\mathbb{Z})$ such that $\mathbf{B} = \mathbf{O}\mathbf{B}'\mathbf{U}^\top$. The output is $(\mathbf{B}'^\top\mathbf{B}', \mathbf{U})$.

The second method, described in terms of quadratic forms [26], first applies the LLL algorithm to the Gram matrix $\mathbf{Q}$, eliminating the length information. Then, the Gaussian sampling algorithm is used to sample n linearly independent lattice vectors $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. By leveraging [60, Lemma 7.1] (which exploits the uniqueness of the Hermite Normal Form), these vectors are converted into the Gram matrix $\mathbf{Q}'$, yielding the transition matrix $\mathbf{U}^\top \in \mathrm{GL}_n(\mathbb{Z})$, such that $\mathbf{Q} = \mathbf{U}\mathbf{Q}'\mathbf{U}^\top$. The output is $(\mathbf{Q}', \mathbf{U})$.

Since the second method is simpler and more efficient, we recommend opting for this approach. For further details, please refer to [26].

Lemma 5.1 (Adapted from [26, Lemma 3.4]) *There exists an efficient randomized algorithm* R *that takes any* $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$ *as input and outputs* $(\mathbf{R}, \mathbf{U})$ *such that* $(\mathbf{R} = \mathbf{U}\mathbf{Q}\mathbf{U}^\top, \mathbf{U}) \in [\mathbf{Q}] \times GL_n^{\pm}(\mathbb{Z})$*, with the following properties:*

- *For any quadratic form* $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{Z})$*,* $\mathbf{Q}' \in [\mathbf{Q}]$*, and* $(\mathbf{Q}'', \mathbf{U}) \leftarrow \mathsf{R}(\mathbf{Q})$*, the marginal distributions of the first variable are identical for* $\mathsf{R}(\mathbf{Q})$ *and* $\mathsf{R}(\mathbf{Q}')$*;* $\mathbf{U}$ *is uniformly distributed on* $\mathcal{I}(\mathbf{Q}, \mathbf{Q}'')$*.*

5.3 The Deterministic Extractor and Commitment 3.1

It suffices to instantiate the deterministic extractor, as the commitment directly follows from the randomization algorithm R in Lemma 5.1 and the framework established in Commitment 3.1. Let n be an even positive integer, and let $\mathbf{Q}$ be the quadratic form of a lattice $\mathcal{L} \in \Lambda_-^n$. Define the distribution $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ which outputs a pair $(\mathbf{U}, \mathbf{Q})$, where $\mathbf{U}$ is sampled using the re-randomization algorithm $(\mathbf{Q}', \mathbf{U}) \leftarrow \mathsf{R}(\mathbf{Q})$.

Lemma 5.2 *For the group action* $(GL_n^{\pm}(\mathbb{Z}), [\mathbf{Q}], \star)$ *with distribution* $\mathcal{D}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ *on* $GL_n^{\pm}(\mathbb{Z}) \times [\mathbf{Q}]$ *and the group* $M = (\{\pm 1\}, \times)$*, define* $\mathsf{E} : GL_n^{\pm}(\mathbb{Z}) \to M$ *such that* $\mathsf{E}(\mathbf{U}) \mapsto \det(\mathbf{U})$*. Then,* E *is a deterministic extractor as in Definition 3.2.*

Proof. The function E is well-defined because $\det(\mathbf{U}) = \det(-\mathbf{U})$ for even n and any $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$. Since $\mathbf{Q} \in \Lambda_-^n$ and $\mathsf{E} : \mathrm{GL}_n^{\pm}(\mathbb{Z}) \to \{\pm 1\}$ is a surjective group homomorphism, $\det(\mathbf{U})$ is uniformly distributed over $\{\pm 1\}$ for any $\mathbf{Q}' \in [\mathbf{Q}]$ and $\mathbf{U} \leftarrow \mathrm{Stab}(\mathbf{Q}')$. Therefore, for $(\mathbf{U}', \mathbf{Q}) \leftarrow \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ and any $\mathbf{Q}', \mathbf{Q}'' \in [\mathbf{Q}]$, $\det(\mathbf{U}')$ is uniformly distributed over $\{\pm 1\}$ for $\mathbf{U}' \leftarrow \mathcal{I}(\mathbf{Q}', \mathbf{Q}'') = \mathbf{V} \cdot \mathrm{Stab}(\mathbf{Q}')$, where $\mathbf{V} \in \mathcal{I}(\mathbf{Q}', \mathbf{Q}'')$. □

Since E is a group homomorphism, the following corollary holds.

Corollary 5.1 *For* $\mathcal{D}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ *and* E *defined in Lemma 5.2,* E *is a homomorphic extractor as in Definition 4.4.*

How to Choose a Lattice from Λ_-^n. We now explain how to select a quadratic form $\mathbf{Q}$ of a lattice $\mathcal{L} \in \Lambda_-^n$, given an even n. In fact, for any $(n-1)$-dimensional lattice $\mathcal{L}_1$ and $(n-2)$-dimensional lattice $\mathcal{L}_2$, we have $\mathcal{L}_1 \oplus \mathbb{Z}$ and $\mathcal{L}_2 \oplus \mathbb{Z}^2 \in \Lambda_-^n$, particularly $\mathbb{Z}^n \in \Lambda_-^n$. Additionally, other methods for selecting a lattice from Λ_-^n exist, as discussed in [18, Section 3.4.2].

The s-GASP Assumption. For the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}], \star)$ with distribution $\mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$, given an s-GASP instance $\mathbf{Q}' \in [\mathbf{Q}]$, finding a non-trivial stabilizer $\pm\mathbf{I}_n \neq \mathbf{U} \in \mathrm{Stab}(\mathbf{Q}')$ is equivalent to solving LAP, as discussed in Section 5.1. Thus, the security of this instantiation reduces to the hardness of LAP.

A Discussion on E. An interesting question is whether we can identify a finite group M and a deterministic extractor $\mathsf{E} : G \to M$ such that $|M|$ is maximized. However, if we wish to preserve the homomorphic property, it seems challenging to extend E to extract additional bits. A more detailed discussion on this is provided in the Appendix B.

5.4 The Local Constant Extractor and Commitment 3.2

Let n be an even positive integer, and let $\mathbf{Q}_0, \mathbf{Q}_1$ be the quadratic forms corresponding to lattices $\mathcal{L}_0 \in \Lambda_-^n$ and $\mathcal{L}_1 \in \Lambda_+^n$, respectively. For the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}_0] \cup [\mathbf{Q}_1], \star)$, we define the distributions $\mathcal{D}^{(0)}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ and $\mathcal{D}^{(1)}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ such that $\mathcal{D}^{(b)}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ outputs a pair $(\mathbf{U}_b, \mathbf{Q}_b)$, where $\mathbf{U}_b$ is sampled using the re-randomization algorithm $(\mathbf{Q}'_b, \mathbf{U}_b) \leftarrow \mathsf{R}(\mathbf{Q}_b)$ for $b \in \{0, 1\}$. The following lemma can be derived similar to Lemma 5.2, and the proof is omitted here.

Lemma 5.3 *For the group action $(GL_n^{\pm}(\mathbb{Z}), [\mathbf{Q}_0] \cup [\mathbf{Q}_1], \star)$ with distributions $\mathcal{D}^{(0)}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ and $\mathcal{D}^{(1)}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ on $GL_n^{\pm}(\mathbb{Z}) \times [\mathbf{Q}_0] \cup [\mathbf{Q}_1]$ and the group $M = (\{\pm 1\}, \times)$, define $\mathsf{E} : GL_n^{\pm}(\mathbb{Z}) \to M$ such that $\mathsf{E}(\mathbf{U}) \mapsto \det(\mathbf{U})$. Then, E is a local constant extractor as in Definition 3.3.*

Corollary 5.2 *For the distributions $\mathcal{D}^{(0)}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$, $\mathcal{D}^{(1)}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$, and E defined in Lemma 5.3, E is a homomorphic local constant extractor as described in Definition 4.5.*

An instantiation of Commitment 3.2 can be achieved by applying the re-randomization algorithm R from Lemma 5.1 and the local constant extractor E from Lemma 5.3. In this context, the d-GAOP problem becomes equivalent to the decisional $\mathrm{LIP}(\mathcal{L}_0, \mathcal{L}_1)$, as discussed in Section 5.1. Therefore, an additional requirement for $\mathcal{L}_0$ and $\mathcal{L}_1$ is that they must belong to the same genus, with the details of this selection discussed below.

How to Choose Lattices from Λ_-^n and Λ_+^n . We present two methods for selecting $\mathcal{L}_0$ and $\mathcal{L}_1$ such that they belong to the same genus, noting that the dimension n must be an even number in both cases.

Method 1: This method leverages the properties of lattice direct sums. Specifically, to construct a pair of lattices $(\mathcal{L}_0, \mathcal{L}_1)$ that satisfy the desired conditions, we first identify two irreducible low-dimensional lattices $(\mathcal{N}_0, \mathcal{N}_1) \in \Lambda_-^k \times \Lambda_0^k$,

where k is an even number and $\text{genus}(\mathcal{N}_0) = \text{genus}(\mathcal{N}_1)$. Using the following proposition, we can deduce that setting $(\mathcal{L}_0, \mathcal{L}_1) = (\oplus_{i=1}^m \mathcal{N}_0, \oplus_{i=1}^m \mathcal{N}_1)$ for any m results in $(\mathcal{L}_0, \mathcal{L}_1) \in \Lambda_-^{mk} \times \Lambda_+^{mk}$ and $\text{genus}(\mathcal{L}_0) = \text{genus}(\mathcal{L}_1)$. The deduction of Proposition 5.1 is provided in the Appendix C.

Proposition 5.1 *For two k-dimensional irreducible lattices $(\mathcal{L}_1, \mathcal{L}_2)$ satisfying $\text{genus}(\mathcal{L}_1) = \text{genus}(\mathcal{L}_2)$ and any positive integer m, we have $\text{genus}(\oplus_{i=1}^m \mathcal{L}_1) = \text{genus}(\oplus_{i=1}^m \mathcal{L}_2)$ and $Aut(\oplus_{i=1}^m \mathcal{L}_j) = \{(\mathbf{S} \otimes \mathbf{I}_k) \cdot diag(\mathbf{A}_1, \cdots, \mathbf{A}_m) \mid \mathbf{S} \in \mathcal{S}_m^{\pm}, \mathbf{A}_i \in Aut(\mathcal{L}_j), i \in [m]\}$, $j \in \{1, 2\}$.*

The selection of lattices $(\mathcal{N}_0, \mathcal{N}_1)$ can proceed as follows. Set $\mathcal{N}_0 = \mathbb{Z}^k$ and $\mathcal{N}_1$ as a k-dimensional odd unimodular lattice with a trivial automorphism group. This selection requires that k be an even integer of at least 30. This is because such $\mathcal{N}_1$ only exists for dimensions $k > 28$, and an explicit construction of $\mathcal{N}_1$ for $k = 30$ can be found in [71].

It is generally believed that the 'direct sum' structure does not compromise the hardness of the LIP. On the one hand, $\mathbb{Z}^n = \oplus_{i=1}^n \mathbb{Z}$ has a 'direct sum' structure, yet $\mathbb{Z}$LIP is still considered hard, with the best-known algorithms having exponential complexity in the dimension n. Our construction can be viewed as an extension of $\mathbb{Z}$LIP by replacing the direct sum of $\mathbb{Z}$ with the direct sum of other small lattices. On the other hand, the 'direct sum' structure has been implicitly used in previous scheme constructions. For instance, in [26], the lattice structure $\mathcal{N} \oplus a(a+1)\mathcal{N}$ was employed, where a is an integer and $\mathcal{N}$ represents a lattice.

Method 2: This method employs a sampling algorithm from a given genus, a similar approach has also been mentioned in [72]. Specifically, for a given genus $\mathcal{G}$ (typically represented by its canonical label), lattices can be sampled from $\mathcal{G}$ according to a distribution related to the sizes of their automorphism groups, as stated below.

Proposition 5.2 ([44,17]) *For a given genus $\mathcal{G}$, there exists an efficient algorithm to sample from the distribution $\mathcal{D}(\mathcal{G})$, which selects $[\mathcal{L}] \in \mathcal{G}$ with relative mass $m(\mathcal{L}) := 1/|Aut(\mathcal{L})|$. In particular, for any $[\mathcal{L}] \in \mathcal{G}$, we have:*

$$\Pr_{[\mathcal{L}'] \leftarrow \mathcal{D}(\mathcal{G})} [\, [\mathcal{L}'] = [\mathcal{L}] \,] = \frac{m(\mathcal{L})}{\sum_{[\mathcal{L}'] \in \mathcal{G}} m(\mathcal{L}')}.$$

Note that $\mathbb{Z}^n$ is an odd unimodular lattice, and the Barnes-Wall lattices of dimension $n = 2^{2k+1}$ are even unimodular lattices up to scaling. Additionally, for a given dimension n, odd unimodular lattices form a single genus, as do even unimodular lattices. Considering $\mathbb{Z}^n$ or the Barnes-Wall lattices (up to scaling) as $\mathcal{L}_0 \in \Lambda_-^n$, then we can employ Proposition 5.2 to sample a lattice $\mathcal{L}_1$ from $\mathcal{D}(\mathcal{G})$, where $\mathcal{G} = \text{genus}(\mathcal{L}_0)$. According to Theorem 5.1, the automorphism group of $\mathcal{L}_1$ is overwhelmingly likely to be trivial for large n. Consequently, we obtain an instantiation $(\mathcal{L}_0, \mathcal{L}_1) \in \Lambda_-^n \times \Lambda_0^n \subset \Lambda_-^n \times \Lambda_+^n$.

Theorem 5.1 ([3]) *Let m be the mass of the given genus of positive definite unimodular lattices of rank n and m' be the mass of all the classes in the genus with nontrivial automorphisms. Then the ratio of the mass m'/m is upper bounded by $33(\sqrt{2\pi})^n/\Gamma(\frac{n}{2})$ for odd unimodular lattices of dimension $n \geq 43$ and by $2^{n+1}(\sqrt{2\pi})^n/\Gamma(\frac{n}{2})$ for even unimodular lattices of dimension $n \geq 144$. In particular, this ratio approaches 0 very rapidly as n increases.*

5.5 Commitments Based on the Randomness Extractor

In [4,42,26], a (k,ϵ)-random extractor $\mathcal{E} : \mathcal{X} \times \{0,1\}^n \to \{0,1\}^v$ with a distribution $X \leftarrow \mathcal{D}_{\mathcal{X}}$ is a computationally efficient algorithm such that if the min-entropy of X is greater than k, then $\Delta((\mathcal{E}(X,Z),Z),(V,Z)) \leq \epsilon$, where $X \leftarrow \mathcal{D}_{\mathcal{X}}, Z \leftarrow \{0,1\}^n, V \leftarrow \{0,1\}^v$. The *leftover hash lemma* [4] is a well-known technique for constructing randomness extractors, providing a (k,ϵ)-extractor with $k = \Theta(v)$ and $\epsilon = 2^{-\Theta(k)}$. Thus, the randomness extractor defined in Definition 3.4 essentially functions as an (k,ϵ)-extractor when we set $(\mathcal{X},\mathcal{D}_{\mathcal{X}},n)$ to $(G,\mathcal{D}_{G,X},\zeta)$ and $(M,\cdot) = (\{0,1\}^v,\oplus)$, requiring the distribution $\mathcal{D}_{G,X}$ to satisfy $|\mathrm{Stab}(x)| \geq 2^k$ for $(h,x) \leftarrow \mathcal{D}_{G,X}$. Thus, the following lemma can be derived:

Lemma 5.4 *For an even integer n and a quadratic form $\mathbf{Q}$ of a lattice $\mathcal{L} \in \Lambda_k^n$, consider the group action $(GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}],\star)$ with distribution $\mathcal{D}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ on $GL_n^{\pm}(\mathbb{Z}) \times [\mathbf{Q}]$ and $M = (\{0,1\}^v,\oplus)$, where $\mathcal{D}_{GL_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]}$ samples the pair $(\mathbf{U},\mathbf{Q})$ such that $\mathbf{U}$ is drawn according to $(\mathbf{Q}',\mathbf{U}) \leftarrow \mathsf{R}(\mathbf{Q})$.*

Then there is an efficient algorithm $\mathsf{E} : GL_n^{\pm}(\mathbb{Z}) \times \{0,1\}^{\zeta} \to M$, such that E is a randomness extractor as defined in Definition 3.4.

Instantiation of Commitment 3.3. First, we demonstrate how to find a quadratic form $\mathbf{Q}$ of a lattice $\mathcal{L} \in \Lambda_k^n$, given an even n. To maximize the number of bits committed, it is desirable for the automorphism group of $\mathcal{L}$ to be as large as possible. Feit [29] showed that for $n > 10$, $\mathbb{Z}^n$ has the largest automorphism group among all n-dimensional lattices, with $|\mathrm{Aut}(\mathbb{Z}^n)| = 2^n \cdot n!$. Furthermore, it is noteworthy that $\mathbb{Z}$LAP is equivalent to $\mathbb{Z}$LIP [47]. Therefore, by selecting $\mathbf{Q} \in [\mathbf{I}_n]$, k can approximately scale as $\Theta(n \log n)$. Using R from Lemma 5.1 and E from Lemma 5.4, we can effectively instantiate Commitment 3.3. In this context, the s-GASP problem reduces to $\mathbb{Z}$LIP.

Instantiation of Commitment 3.4. Given an even n, we demonstrate how to find a quadratic form $\mathbf{Q}_0$ for a lattice $\mathcal{L}_0 \in \Lambda_k^n$ and a quadratic form $\mathbf{Q}_1$ for a lattice $\mathcal{L}_1 \in \Lambda_0^n$ such that $\mathrm{genus}(\mathcal{L}_0) = \mathrm{genus}(\mathcal{L}_1)$. Similarly, to maximize the commitment of more bits, it is preferable for the automorphism group of $\mathcal{L}_0$ to be as large as possible. Thus, we set $\mathcal{L}_0 = \mathbb{Z}^n$, and we can apply Proposition 5.2 to sample a lattice $\mathcal{L}_1$ from $\mathcal{D}(\mathcal{G})$, where $\mathcal{G} = \mathrm{genus}(\mathcal{L}_0)$. According to Theorem 5.1, the automorphism group of $\mathcal{L}_1$ is overwhelmingly likely to be trivial for a sufficiently large n. As a result, we instantiate $(\mathcal{L}_0,\mathcal{L}_1) \in \Lambda_k^n \times \Lambda_0^n$, with $k = \Theta(n \log n)$. Using R from Lemma 5.1 and E from Lemma 5.4, we can

effectively instantiate Commitment 3.4. In this context, the d-GAOP problem is reduced to the decisional LIP$(\mathcal{L}_0, \mathcal{L}_1)$.

5.6 Dual-Mode Commitment and Enhanced-Linkable Commitments

To begin with, we demonstrate how to obtain a trapdoor extractor (E, F) that satisfies the conditions in Definition 4.2. Let n be an even integer, and let $\mathbf{Q}_0, \mathbf{Q}_1$ be quadratic forms of lattices $\mathcal{L}_0 \in \Lambda^n_-$ and $\mathcal{L}_1 \in \Lambda^n_+$, respectively. For the group action $(\mathrm{GL}^{\pm}_n(\mathbb{Z}), [\mathbf{Q}_0] \cup [\mathbf{Q}_1], \star)$, define distributions $\mathcal{D}^{(0)}_{\mathrm{GL}^{\pm}_n(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ and $\mathcal{D}^{(1)}_{\mathrm{GL}^{\pm}_n(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ on $\mathrm{GL}^{\pm}_n(\mathbb{Z}) \times [\mathbf{Q}_0] \cup [\mathbf{Q}_1]$, where $\mathcal{D}^{(b)}_{\mathrm{GL}^{\pm}_n(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ outputs a pair $((\mathbf{U}_b, \mathbf{Q}_b))$ such that $\mathbf{U}_b$ is sampled according to $(\mathbf{Q}'_b, \mathbf{U}_b) \leftarrow \mathsf{R}(\mathbf{Q}_b)$, for $b \in \{0, 1\}$.

Define the extractor $\mathsf{E} : \mathrm{GL}^{\pm}_n(\mathbb{Z}) \to \{\pm 1\}$ such that $\mathsf{E}(\mathbf{U}) \mapsto \det(\mathbf{U})$, which serves as a local constant extractor according to Lemma 5.3. We construct F as follows. For $(\mathbf{U}_0, \mathbf{Q}_0) \leftarrow \mathcal{D}^{(0)}_{\mathrm{GL}^{\pm}_n(\mathbb{Z}),[\mathbf{Q}_0]\cup[\mathbf{Q}_1]}$ and $u \in \{\pm 1\}$, with $\mathbf{Q}'_0, \mathbf{Q}''_0 \in [\mathbf{Q}_0]$ and $\mathbf{U} \in \mathcal{I}(\mathbf{Q}'_0, \mathbf{Q}''_0)$, F takes $(u, \mathbf{U}, \mathbf{Q}'_0, \mathrm{Stab}(\mathbf{Q}'_0))$ as input. If $\det(\mathbf{U}) = u$, F sets $\mathbf{U}' = \mathbf{U}$ and outputs $\mathbf{U}'$. Otherwise, F finds an element $\mathbf{V} \in \mathrm{Stab}(\mathbf{Q}'_0)$ such that $\det(\mathbf{V}) = -1$, sets $\mathbf{U}' = \mathbf{U}\mathbf{V}$, and then outputs $\mathbf{U}'$. Thus, we conclude the following lemma, with its proof provided in Appendix C.

Lemma 5.5 *The above pair of algorithms* (E, F) *forms a trapdoor extractor as described in Definition 4.2.*

Instantiation of Commitment 4.1. This instantiation closely resembles that of Section 5.4. For an even n, the instantiation involves not only finding a quadratic form $\mathbf{Q}_0$ of a lattice $\mathcal{L}_0 \in \Lambda^n_-$ and a quadratic form $\mathbf{Q}_1$ of a lattice $\mathcal{L}_1 \in \Lambda^n_+$ such that $\mathrm{genus}(\mathcal{L}_0) = \mathrm{genus}(\mathcal{L}_1)$, but also efficiently computing a polynomial-size generating set of $\mathrm{Stab}(\mathbf{Q}_0)$ from $\mathbf{Q}_0$. It is worth noting that in Lemma 5.5, we only required an element $\mathbf{V} \in \mathrm{Stab}(\mathbf{Q}''_0)$ with $\det(\mathbf{V}) = -1$. This implies the need to efficiently find an element $\mathbf{V} \in \mathrm{Stab}(\mathbf{Q}_0)$ such that $\det(\mathbf{V}) = -1$ from $\mathbf{Q}_0$ to finalize the instantiation. Furthermore, for all instantiation methods described in Section 5.4, $\mathrm{Aut}(\mathcal{L}_0)$ is consistently computationally feasible and succinctly represented. Specifically, in *Method 1*, the selected lattice $\mathcal{L}_0 = \oplus^m_{i=1}\mathcal{N}_0$, where $\mathrm{Aut}(\mathcal{N}_0)$ can be efficiently computed due to its constant dimension or because it has a known automorphism group, such as $\mathcal{N}_0 = \mathbb{Z}^k$. Thus, by Proposition 5.1, $\mathrm{Aut}(\mathcal{L}_0)$ can be fully determined. In *Method 2*, one might choose $\mathcal{L}_0 = \mathbb{Z}^n$ such that $\mathrm{Aut}(\mathcal{L}_0) = \mathcal{S}^{\pm}_n$. Consequently, following the discussions in Section 5.4, utilizing R in Lemma 5.1 and E in Lemma 5.5, we can effectively instantiate Commitment 4.1 based on the Decisional LIP$(\mathcal{L}_0, \mathcal{L}_1)$.

5.7 Enhanced-Linkable Commitments

Since Commitment 4.2 can be viewed as an extension of Commitment 3.1 with an additional linking component, and given that the extractor E defined in

Lemma 5.2 is already a homomorphic extractor, the corresponding linking component naturally arises from Commitment 3.1. Therefore, in line with the discussion in Section 5.3, by utilizing R from Lemma 5.1 and E from Lemma 5.2, we can efficiently instantiate Commitment 4.2 based on LAP.

Similarly, following the discussion in Section 5.4, by utilizing R from Lemma 5.1 and E from Lemma 5.3, we can efficiently instantiate Commitment 4.3 based on LIP and the Decisional LIP$(\mathcal{L}_0, \mathcal{L}_1)$.

6 Non-Interactive Commitment Based on Decisional LIP

In this section, we present how to derive a non-interactive commitment scheme based on the decisional LIP. Compared to the non-interactive commitment schemes proposed in [27], this scheme does not suffer from the attack in [33], while also offering an enhancement by expanding the message space without increasing the asymptotic size of the commitment.

Commitment Scheme 6.1 : *A non-interactive commitment scheme* $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *with security parameter* λ*, message space* $M = \{0,1\}^n$*, where* n *is an even integer.*

- $\mathbf{Gen}(1^\lambda)$: Generate a set of lattices $\{\mathcal{L}_i^0, \mathcal{L}_i^1\}_{i\in[n]}$ such that $\mathcal{L}_i^0$ and $\mathcal{L}_i^1$ belong to the same genus, and there exists a witness $\mathbf{w}$ to show that the 2^n lattices $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ are pairwise non-isomorphic. Compute a Gram matrix $\mathbf{Q}_i^j$ for the lattice $\mathcal{L}_i^j$, where $i \in [n], j \in \{0,1\}$. Output $Ck = \{\mathbf{Q}_i^0, \mathbf{Q}_i^1\}_{i=1}^n$.
- $\mathbf{Com}(\mathbf{m})$: For a message $\mathbf{m} = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$, define $\mathbf{Q_m} = \mathrm{diag}\{\mathbf{Q}_i^{m_i}\}_{i\in[n]}$. Use the re-randomization algorithm from Section 5.2 to generate $(\mathbf{Q}, \mathbf{U}) \leftarrow \mathsf{R}(\mathbf{Q_m})$. Output $(c, d) = (\mathbf{Q}, \mathbf{U})$.
- $\mathbf{Open}(c, d)$: Compute $\mathbf{Q}' = d^{-1} \cdot c \cdot d^{-\top}$. If there exists a message $\mathbf{m}' = (m_1', m_2', \ldots, m_n') \in \{0,1\}^n$ such that $\mathbf{Q}' = \mathrm{diag}\{\mathbf{Q}_i^{m_i'}\}_{i\in[n]}$, output $\mathbf{m}'$; otherwise, output $\perp$.

Theorem 6.1 *Assuming that it is difficult to distinguish among the* 2^n *lattices* $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$*, i.e., the decisional LIP*$(\oplus_{i=1}^n \mathcal{L}_i^{m_i}, \oplus_{i=1}^n \mathcal{L}_i^{m_i'})$ *is hard for any* $\mathbf{m}, \mathbf{m}' \in \{0,1\}^n$*, then Commitment 6.1 is perfectly binding and computationally hiding.*

Proof. The correctness is evident. The binding property arises from the 2^n lattices $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ being pairwise non-isomorphic. The hiding property can be reduced to the hardness of the decisional LIP$(\mathcal{L}, \mathcal{L}')$ for any $\mathcal{L}, \mathcal{L}' \in \{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$. The proof is provided in Appendix C □

Generation of the Lattices $\{\mathcal{L}_i^0, \mathcal{L}_i^1\}_{i\in[n]}$. Generating a witness that is short and efficiently verifiable for non-isomorphic lattices is a challenging task in general, as it remains unclear whether the lattice isomorphism problem is in coNP.

However, for Commitment 6.1, we can provide a straightforward instantiation of $\{\mathcal{L}_i^0, \mathcal{L}_i^1\}_{i\in[n]}$, ensuring that $\mathcal{L}_i^0$ and $\mathcal{L}_i^1$ belong to the same genus. Therefore, all lattices in $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ have the same genus. A short and efficiently verifiable witness $\mathbf{w}$ for their non-isomorphism can be easily constructed.

Observe that for any positive integer a and lattice $\mathcal{L}$, the lattices $a\cdot\mathcal{L}\oplus(a+1)\cdot\mathcal{L}$ and $\mathcal{L}\oplus a(a+1)\cdot\mathcal{L}$ are in the same genus [26, Section 8]. Thus, for Commitment 6.1, we can define $\mathcal{L}_i^0 = 2i\cdot\mathbb{Z}\oplus(2i+1)\cdot\mathbb{Z}$ and $\mathcal{L}_i^1 = \mathbb{Z}\oplus 2i(2i+1)\cdot\mathbb{Z}$. Moreover, the witness can be trivially set as the set of bases for $\mathcal{L}_i^j$, where $i\in[n]$ and $j\in\{0,1\}$. This witness can then be used to efficiently prove that the lattices in $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ are non-isomorphic, as stated in the following lemma.

Lemma 6.1 *Suppose* $\mathbf{B}_i^{(0)}$ *and* $\mathbf{B}_i^{(1)}$ *are bases of* $\mathcal{L}_i^0 = 2i\cdot\mathbb{Z}\oplus(2i+1)\cdot\mathbb{Z}$ *and* $\mathcal{L}_i^1 = \mathbb{Z}\oplus 2i(2i+1)\cdot\mathbb{Z}$ *respectively. Let the witness* $\mathbf{w} = \{\mathbf{B}_i^{(0)}, \mathbf{B}_i^{(1)}\}_{i\in[n]}$*, it can be efficiently verified that the* 2^n *lattices* $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ *are pairwise non-isomorphic.*

Proof. The receiver needs to verify that each $\mathbf{B}_i^j$ is a lattice basis of $\mathcal{L}_i^j$, for $j\in\{0,1\}$ and $i\in[n]$, which can be done efficiently. This verification suffices because for any distinct $\mathbf{s}\neq\mathbf{t}\in\{0,1\}^n$, the lattices $\oplus_{i=1}^n \mathcal{L}_i^{s_i}$ and $\oplus_{i=1}^n \mathcal{L}_i^{t_i}$ are not isomorphic. Consider two diagonal bases of $\mathcal{L}_\mathbf{s} = \oplus_{i=1}^n \mathcal{L}_i^{s_i}$ and $\mathcal{L}_\mathbf{t} = \oplus_{i=1}^n \mathcal{L}_i^{t_i}$, denoted as $\mathbf{B}_\mathbf{s} = \mathrm{diag}(a_1,\ldots,a_n)$ and $\mathbf{B}_\mathbf{t} = \mathrm{diag}(b_1,\ldots,b_n)$, respectively. It follows that $\mathcal{L}_\mathbf{s}$ and $\mathcal{L}_\mathbf{t}$ are isomorphic if and only if the diagonal elements of $\mathbf{B}_\mathbf{s}$ and $\mathbf{B}_\mathbf{t}$ are identical up to a signed permutation, especially the odd integers in both diagonals must match. This observation underscores the non-isomorphism of $\mathcal{L}_\mathbf{s}$ and $\mathcal{L}_\mathbf{t}$ by comparing the odd integers in their diagonals. □

Comparison with the Non-Interactive Commitment Scheme from [27]. The non-interactive commitment scheme in [27] is a bit-commitment based on d-GAIP. Specifically, their scheme is instantiated using a special case of the tensor isomorphism problem, where ranks serve as witnesses to distinguish non-isomorphic tensors. However, this instantiation is vulnerable due to the 'direct sum' structure in the tensor isomorphism problem, as demonstrated in [33]. In contrast, Commitment 6.1 is based on lattice isomorphism problems, which we believe exhibit 'direct sum'-hardness. Moreover, Commitment 6.1 has the advantage of utilizing 2^n different combinations $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i\in\{0,1\}}$ for committing, a feature that cannot be achieved with the approach from [27]. This effectively enlarges the message space of Commitment 6.1.

7 Conclusion

We propose two key techniques for general group actions: re-randomization and randomness extraction. We demonstrate that these techniques can significantly facilitate the construction of commitment schemes, providing a flexible framework for constructing commitment schemes with various properties, depending

on the type of extractor involved. Finally, we instantiate all our proposed commitment schemes using lattices, specifically leveraging the LIP and the LAP as underlying cryptographic assumptions. Additionally, we use LIP to provide a repair and improvement to the tensor isomorphism-based non-interactive commitment scheme proposed by [27].

As part of future research, it would be intriguing to explore the potential applicability of alternative assumptions for instantiation, such as code equivalence or isogenies. Furthermore, an investigation of the feasibility of devising additional cryptographic schemes using the methodologies and framework described in this paper could be an interesting avenue for exploration.

References

1. Alamati, N., Feo, L.D., Montgomery, H., Patranabis, S.: Cryptographic group actions and applications. 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. 411–439. Springer (2020). https://doi.org/10.1007/978-3-030-64834-3_14, https://doi.org/10.1007/978-3-030-64834-3_14
2. Alamati, N., Malavolta, G., Rahimi, A.: Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In: Theory of Cryptography Conference. pp. 266–293. Springer (2022)
3. Bannai, E.: Positive definite unimodular lattices with trivial automorphism groups, vol. 429. American Mathematical Soc. (1990)
4. Barak, B., Dodis, Y., Krawczyk, H., Pereira, O., Pietrzak, K., Standaert, F.X., Yu, Y.: Leftover hash lemma, revisited. In: Annual Cryptology Conference. pp. 1–20. Springer (2011)
5. Barak, B., Ong, S.J., Vadhan, S.: Derandomization in cryptography. In: Advances in Cryptology-CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings 23. pp. 299–315. Springer (2003)
6. 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, https://doi.org/10.1007/978-3-031-58868-6_18
7. Bencina, B., Budroni, A., Chi-Domínguez, J., Kulkarni, M.: Properties of lattice isomorphism as a cryptographic group action. 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. 170–201. Springer (2024). https://doi.org/10.1007/978-3-031-62743-9_6, https://doi.org/10.1007/978-3-031-62743-9_6
8. Bennett, H., Ganju, A., Peetathawatchai, P., Stephens-Davidowitz, N.: Just how hard are rotations of $\mathbb{Z}^n$? algorithms and cryptography with the simplest lattice. IACR Cryptol. ePrint Arch. p. 1548 (2021), https://eprint.iacr.org/2021/1548

9. Beullens, W., Dobson, S., Katsumata, S., Lai, Y.F., Pintore, F.: Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 95–126. Springer (2022)
10. 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, https://doi.org/10.1007/978-3-030-64834-3_16
11. Biasse, J.F., 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 12. pp. 45–65. Springer (2020)
12. 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, https://doi.org/10.1007/978-3-031-62743-9_8
13. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013. pp. 575–584. ACM (2013). https://doi.org/10.1145/2488608.2488680, https://doi.org/10.1145/2488608.2488680
14. Brassard, G., Yung, M.: One-way group actions. In: Advances in Cryptology-CRYPTO'90: Proceedings 10. pp. 94–107. Springer (1991)
15. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: an efficient post-quantum commutative group action. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 395–427. Springer (2018)
16. Catalano, D., Visconti, I.: Hybrid commitments and their applications to zero-knowledge proof systems. Theor. Comput. Sci. **374**(1-3), 229–260 (2007). https://doi.org/10.1016/J.TCS.2007.01.007, https://doi.org/10.1016/j.tcs.2007.01.007
17. Chenevier, G.: Statistics for kneser p-neighbors. arXiv preprint arXiv:2104.06846 (2021)
18. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups, Grundlehren der mathematischen Wissenschaften, vol. 290. Springer (1988). https://doi.org/10.1007/978-1-4757-2016-7, https://doi.org/10.1007/978-1-4757-2016-7
19. Couveignes, J.M.: Hard homogeneous spaces (2006)
20. Cozzo, D., Smart, N.P.: Sashimi: Cutting up csi-fish secret keys to produce an actively secure distributed signing protocol. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12100, pp. 169–186. Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_10, https://doi.org/10.1007/978-3-030-44223-1_10

21. Crépeau, C., Stuart, J.: Zero-knowledge mips using homomorphic commitment schemes. arXiv preprint arXiv:2304.09784 (2023)
22. Darwish, A., El-Gendy, M.M.: A new cryptographic voting verifiable scheme for e-voting system based on bit commitment and blind signature. Int J Swarm Intel Evol Comput **6**(158), 2 (2017)
23. Doröz, Y., Hoffstein, J., Pipher, J., Silverman, J.H., Sunar, B., Whyte, W., Zhang, Z.: Fully homomorphic encryption from the finite field isomorphism problem. In: IACR International Workshop on Public Key Cryptography. pp. 125–155. Springer (2018)
24. Ducas, L., Gibbons, S.: Hull attacks on the lattice isomorphism problem. IACR Cryptol. ePrint Arch. p. 194 (2023), https://eprint.iacr.org/2023/194
25. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.P.J.: Hawk: Module LIP makes lattice signatures fast, compact and simple. 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. 65–94. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_3, https://doi.org/10.1007/978-3-031-22972-5_3
26. Ducas, L., van Woerden, W.P.J.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. 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 III. Lecture Notes in Computer Science, vol. 13277, pp. 643–673. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_23, https://doi.org/10.1007/978-3-031-07082-2_23
27. D'Alconzo, G., Flamini, A., Gangemi, A.: Non-interactive commitment from non-transitive group actions. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 222–252. Springer (2023)
28. El Kaafarani, A., Katsumata, S., Pintore, F.: Lossy csi-fish: Efficient signature scheme with tight reduction to decisional CSIDH-512. 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. 157–186. Springer (2020). https://doi.org/10.1007/978-3-030-45388-6_6, https://doi.org/10.1007/978-3-030-45388-6_6
29. Feit, W.: Orders of finite linear groups. preprint (1995)
30. 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, https://doi.org/10.1007/978-3-030-45388-6_7
31. Frederiksen, T.K., Pinkas, B., Yanai, A.: Committed mpc: Maliciously secure multiparty computation from homomorphic commitments. In: Public-Key Cryptography–PKC 2018: 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I 21. pp. 587–619. Springer (2018)

32. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407, https://doi.org/10.1145/1374376.1374407
33. Gilchrist, V., Marco, L., Petit, C., Tang, G.: Solving the tensor isomorphism problem for special orbits with low rank points: Cryptanalysis and repair of an asiacrypt 2023 commitment scheme. In: Annual International Cryptology Conference. pp. 141–173. Springer (2024)
34. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptol. **9**(3), 167–190 (1996)
35. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. Journal of the ACM (JACM) **38**(3), 690–728 (1991)
36. Groth, J.: Homomorphic trapdoor commitments to group elements. Cryptology ePrint Archive, Paper 2009/007 (2009), https://eprint.iacr.org/2009/007
37. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4004, pp. 339–358. Springer (2006). https://doi.org/10.1007/11761679_21, https://doi.org/10.1007/11761679_21
38. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer (2008). https://doi.org/10.1007/978-3-540-78967-3_24, https://doi.org/10.1007/978-3-540-78967-3_24
39. Gupte, A., Vaikuntanathan, V.: How to construct quantum fhe, generically. In: Annual International Cryptology Conference. pp. 246–279. Springer (2024)
40. Hahn, A., O'Meara, O.: The Classical Groups and K-theory. Grundlehren der mathematischen Wissenschaften, Springer-Verlag (1989), https://books.google.com.sg/books?id=weruAAAAMAAJ
41. Haitner, I., Horvitz, O., Katz, J., Koo, C., Morselli, R., Shaltiel, R.: Reducing complexity assumptions for statistically-hiding commitment. J. Cryptol. **22**(3), 283–310 (2009). https://doi.org/10.1007/S00145-007-9012-8, https://doi.org/10.1007/s00145-007-9012-8
42. Halevi, S., Micali, S.: Practical and provably-secure commitment schemes from collision-free hashing. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 201–215. Springer (1996). https://doi.org/10.1007/3-540-68697-5_16, https://doi.org/10.1007/3-540-68697-5_16
43. Haviv, I., Regev, O.: On the lattice isomorphism problem. In: Chekuri, C. (ed.) Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 391–404. SIAM (2014). https://doi.org/10.1137/1.9781611973402.29, https://doi.org/10.1137/1.9781611973402.29

44. Hein, J.: Orthogonal modular forms: An application to a conjecture of birch, algorithms and computations. Dartmouth College (2016)
45. (https://math.stackexchange.com/users/2820/derek holt), D.H.: Normal subgroups of signed symmetric groups. Mathematics Stack Exchange, https://math.stackexchange.com/q/3441946, uRL:https://math.stackexchange.com/q/3441946 (version: 2019-11-19)
46. Ji, Z., Qiao, Y., Song, F., Yun, A.: General linear group action on tensors: A candidate for post-quantum cryptography. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11891, pp. 251–281. Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_11, https://doi.org/10.1007/978-3-030-36030-6_11
47. Jiang, K., Wang, A., Luo, H., Liu, G., Yu, Y., Wang, X.: Exploiting the symmetry of z n: Randomization and the automorphism problem. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 167–200. Springer (2023)
48. Jr., H.W.L., Silverberg, A.: Revisiting the gentry-szydlo algorithm. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8616, pp. 280–296. Springer (2014). https://doi.org/10.1007/978-3-662-44371-2_16, https://doi.org/10.1007/978-3-662-44371-2_16
49. Jr., H.W.L., Silverberg, A.: Lattices with symmetry. J. Cryptol. **30**(3), 760–804 (2017). https://doi.org/10.1007/s00145-016-9235-7, https://doi.org/10.1007/s00145-016-9235-7
50. Katsumata, S., Lai, Y.F., LeGrow, J.T., Qin, L.: Csi-otter: isogeny-based (partially) blind signatures from the class group action with a twist. In: Annual International Cryptology Conference. pp. 729–761. Springer (2023)
51. Kitagawa, F., Matsuda, T., Tanaka, K.: CCA security and trapdoor functions via key-dependent-message security. J. Cryptol. **35**(2), 9 (2022). https://doi.org/10.1007/S00145-022-09420-8, https://doi.org/10.1007/s00145-022-09420-8
52. Lai, Y.F.: Capybara and tsubaki: verifiable random functions from group actions and isogenies. Cryptology ePrint Archive (2023)
53. Leo Ackermann, Adeline Roux-Langlois, A.W.: Public-key encryption from the lattice isomorphism problem (extended abstract) (2024), https://wcc2024.sites.dmi.unipg.it/WCC_proceedings.pdf
54. Leroux, A., Roméas, M.: Updatable encryption from group actions. In: International Conference on Post-Quantum Cryptography. pp. 20–53. Springer (2024)
55. Lindell, Y.: An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In: Dodis, Y., Nielsen, J.B. (eds.) Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9014, pp. 93–109. Springer (2015). https://doi.org/10.1007/978-3-662-46494-6_5, https://doi.org/10.1007/978-3-662-46494-6_5
56. Lubiw, A.: Some np-complete problems similar to graph isomorphism. SIAM Journal on Computing **10**(1), 11–21 (1981)
57. Luo, H., Jiang, K., Pan, Y., Wang, A.: Cryptanalysis of rank-2 module-lip with symplectic automorphisms. Cryptology ePrint Archive (2024)

58. Mathon, R.: A note on the graph isomorphism counting problem. Inf. Process. Lett. **8**(3), 131–132 (1979). https://doi.org/10.1016/0020-0190(79)90004-8, https://doi.org/10.1016/0020-0190(79)90004-8
59. Meers, J., Riepel, D.: Cca secure updatable encryption from non-mappable group actions. In: International Conference on Post-Quantum Cryptography. pp. 137–169. Springer (2024)
60. Micciancio, D., Goldwasser, S.: Complexity of lattice problems - a cryptographic perspective, The Kluwer international series in engineering and computer science, vol. 671. Springer (2002)
61. Montgomery, H., Patranabis, S.: A computational category-theoretic approach to cryptography and average-case complexity. Mathematical Cryptology **3**(2), 24–52 (2023)
62. Naor, M.: Bit commitment using pseudorandomness. Journal of cryptology **4**, 151–158 (1991)
63. Ostrovsky, R., Persiano, G., Visconti, I.: Simulation-based concurrent non-malleable commitments and decommitments. In: Theory of Cryptography: 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings 6. pp. 91–108. Springer (2009)
64. Pan, J., Wagner, B.: Lattice-based signatures with tight adaptive corruptions and more. 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. 347–378. Springer (2022). https://doi.org/10.1007/978-3-030-97131-1_12, https://doi.org/10.1007/978-3-030-97131-1_12
65. Patarin, J.: Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In: International conference on the theory and applications of cryptographic techniques. pp. 33–48. Springer (1996)
66. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 129–140. Springer (1991). https://doi.org/10.1007/3-540-46766-1_9, https://doi.org/10.1007/3-540-46766-1_9
67. Pham, M.T.T., Duong, D.H., Li, Y., Susilo, W.: Threshold ring signature scheme from cryptographic group action. In: International Conference on Provable Security. pp. 207–227. Springer (2023)
68. Poelstra, A., Back, A., Friedenbach, M., Maxwell, G., Wuille, P.: Confidential assets. In: Zohar, A., Eyal, I., Teague, V., Clark, J., Bracciali, A., Pintore, F., Sala, M. (eds.) Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10958, pp. 43–63. Springer (2018). https://doi.org/10.1007/978-3-662-58820-8_4, https://doi.org/10.1007/978-3-662-58820-8_4
69. Sterner, B.: Commitment schemes from supersingular elliptic curve isogeny graphs. Mathematical Cryptology **1**(2), 40–51 (2021)
70. 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: Annual international conference on the theory and applications of cryptographic techniques. pp. 582–612. Springer (2022)

71. Wang, J.: A note on unimodular lattices with trivial automorphism groups. Discrete Mathematics **340**(4), 763–765 (2017)
72. van Woerden, W.: Dense and smooth lattices in any genus. Cryptology ePrint Archive, Paper 2024/1468 (2024), https://eprint.iacr.org/2024/1468
73. Zhou, Y., Liu, S., Han, S.: Robustly reusable fuzzy extractor from isogeny. Theoretical Computer Science p. 114677 (2024)

Supplementary Material

Appendix A Security Proofs

A.1 Security Proofs in Section 3

Theorem A.1 (Corresponding to Theorem 3.2) *Suppose that the group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ satisfies the d-GAOP assumption, R is a re-randomized algorithm, and E is a local constant extractor. Then Commitment 3.2 is perfectly binding and computationally hiding.*

Proof. Correctness is obvious. We prove the hiding and binding below.

Perfectly Binding: Assume an adversary $\mathcal{A}$ outputs a commitment $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y') = (\mathsf{E}(g') \cdot m', y')$ with $m' \neq m$ and $g, g' \in \mathcal{I}(y, y')$. Then $\mathsf{E}(g') \neq \mathsf{E}(g)$, this contradicts the property of the local constant extractor E.

Computationally Hiding: We use a hybrid argument to complete the proof.

Game 0. This is the standard hiding game in Commitment 3.2. Let $\mathcal{A}$ be an adversary against the hiding property. $\mathcal{CH}$ interacts with an $\mathcal{A}$ as below. Let S_i be the event that $\mathcal{A}$ wins in *Game i* and $\bar{S}_i$ be the event that $\mathcal{A}$ loses in *Game i*.

1. **Gen**: $\mathcal{CH}$ samples $(h, x) \leftarrow \mathcal{D}'_{G,X}$, sets $Ck = h \star x = y$, sends Ck to $\mathcal{A}$.
2. **Choose**: $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ and sends them to $\mathcal{CH}$.
3. **Challenge**: $\mathcal{CH}$ picks $b \leftarrow \{0, 1\}$ and generates $(c, d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge.
4. **Guess**: $\mathcal{A}$ outputs a bit b' and wins if its guess $b' = b$.

Game 1 is same as *Game 0* except that $\mathcal{CH}$ generates Ck from $\mathcal{D}_{G,X}$, this is the standard hiding game in Commitment 3.1.

1. **Gen**: $\mathcal{CH}$ samples $(h, x) \leftarrow \mathcal{D}_{G,X}$, sets $Ck = h \star x = y$, send Ck to $\mathcal{A}$.
2. **Choose**: $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ and sends them to $\mathcal{CH}$.
3. **Challenge**: $\mathcal{CH}$ picks $b \leftarrow \{0, 1\}$ and generates $(c, d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge.
4. **Guess**: $\mathcal{A}$ outputs a bit b' and wins if its guess $b' = b$.

Since Commitment 3.1 is perfectly hiding, thus $\Pr[S_1] = \frac{1}{2}$ for any adversary $\mathcal{A}$. If there is a PPT adversary $\mathcal{A}$ that wins the hiding game of Commitment 3.2 with probability larger than $\frac{1}{2} + \frac{1}{p(\lambda)}$. Then, we can construct a PPT $\mathcal{A}'$ that solves d-GAOP with probability $\frac{1}{2} + \frac{1}{2p(\lambda)}$.

For a d-GAOP instance y, $\mathcal{A}'$ sets the $Ck = y$ and sends it to $\mathcal{A}$, then $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ to $\mathcal{A}'$. $\mathcal{A}'$ picks a $b \leftarrow \{0, 1\}$ and generates $(c, d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge. $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{A}'$ returns 1, i.e. y is from distribution $\mathcal{D}'_{G,X}$; otherwise, $\mathcal{A}'$ returns 0, i.e. y is from distribution $\mathcal{D}_{G,X}$. Thus, the probability that $\mathcal{A}'$ wins is $\Pr[S_0 \mid y = g \star x, (g, x) \leftarrow \mathcal{D}'_{G,X}] + \Pr[\bar{S}_1 \mid y = g \star x, (g, x) \leftarrow \mathcal{D}_{G,X}]$, which is larger than $\frac{1}{2}(\frac{1}{2} + \frac{1}{p(\lambda)}) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \frac{1}{2p(\lambda)}$. □

Theorem A.2 (Corresponding to Theorem 3.3) *Suppose that the group action $(G, X, \star, \mathcal{D}_{G,X})$ satisfies the s-GASP assumption, R is a re-randomized algorithm, and E is a randomness extractor. Then Commitment 3.3 is statistically hiding and computationally binding.*

Proof. Correctness is easy to obtain. We prove the hiding and binding below.

Statistically Hiding: Assume that $\mathbf{Com}_y(m) = (c, d)$, the statistical distance between $c = (\mathsf{E}(g, z) \cdot m, y_1, z)$ and (u, y_1, z) is at most $\epsilon(\lambda)$, where $u \leftarrow M$, $z \leftarrow \{0,1\}^{\zeta}$, $(y_1, g) \leftarrow \mathsf{R}(y)$ and $\epsilon(\lambda)$ is a negligible function. Thus the commitment scheme is statistically hiding.

Computationally Binding: Given an s-GASP instance y. Send the $Ck = y$ to the adversary $\mathcal{A}$ in the binding game, if $\mathcal{A}$ outputs a commitment $c = (c_1, c_2, c_3) = (\mathsf{E}(g_1, z) \cdot m, y_1, z) = (\mathsf{E}(g_1', z) \cdot m', y_1, z)$ with $m' \neq m$ and $g, g' \in \mathcal{I}(y, y_1)$. Then $\mathsf{E}(g_1', z) \neq \mathsf{E}(g_1, z)$, thus $g_1 \neq g_1'$, because of $\mathsf{E}(\cdot, z)$ is a deterministic extraction algorithm. Thus $e \neq g_1^{-1} \cdot g_1' \in \mathrm{Stab}(y)$, this solves the s-GASP assumption. □

Theorem A.3 (Corresponding to Theorem 3.4) *Suppose that the group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ satisfies the d-GAOP assumption, where the distribution $\mathcal{D}'_{G,X}$ satisfies $|Stab(x)| = 1$ for $(h, x) \leftarrow \mathcal{D}'_{G,X}$. R is a re-randomized algorithm, and E is a randomness extractor for distribution $\mathcal{D}_{G,X}$. Then Commitment 3.4 is perfectly binding and computationally hiding.*

Proof. Correctness is obvious. We prove the hiding and binding below.

Perfectly Binding: Assume an adversary $\mathcal{A}$ outputs a commitment $c = (c_1, c_2, c_3) = (\mathsf{E}(g_1, z) \cdot m, y_1, z) = (\mathsf{E}(g_1', z) \cdot m', y_1, z)$ with $m' \neq m$ and $g, g' \in \mathcal{I}(y, y_1)$. Then $\mathsf{E}(g_1', z) \neq \mathsf{E}(g_1, z)$, thus $g_1 \neq g_1'$. Thus, $e \neq g_1^{-1} \cdot g_1' \in \mathrm{Stab}(y)$, contradicts $|\mathrm{Stab}(y)| = 1$.

Computationally Hiding: We use hybrid argument to complete the proof.

Game 0. This is the standard hiding game in Commitment 3.4. Let $\mathcal{A}$ be an adversary against the hiding property. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below. Let S_i be the event that $\mathcal{A}$ wins in *Game i*, and $\bar{S}_i$ be the event that $\mathcal{A}$ loses in *Game i*.

1. **Gen**: $\mathcal{CH}$ samples $(h, x) \leftarrow \mathcal{D}'_{G,X}$, sets the public key $Ck = h \star x = y$, sends Ck to $\mathcal{A}$.
2. **Choose**: $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ to $\mathcal{CH}$.
3. **Challenge**: $\mathcal{CH}$ picks $b \leftarrow \{0,1\}$ and generates $(c, d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge.
4. **Guess**: $\mathcal{A}$ outputs a bit b' and wins if its guess $b' = b$.

Game 1 is same as *Game 0* except that $\mathcal{CH}$ generates Ck from $\mathcal{D}_{G,X}$, this is the standard hiding game in Commitment 3.3.

1. **Gen**: $\mathcal{CH}$ samples $(h, x) \leftarrow \mathcal{D}_{G,X}$, sets the public key $Ck = h \star x = y$, sends Ck to $\mathcal{A}$.

2. **Choose**: $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ to $\mathcal{CH}$.
3. **Challenge**: $\mathcal{CH}$ picks $b \leftarrow \{0,1\}$ and generates $(c,d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge.
4. **Guess**: $\mathcal{A}$ outputs a bit b' and wins if its guess $b' = b$.

According to the Theorem 3.3, we know that it's statistically hiding, thus $\Pr[S_1] \leq \frac{1}{2} + \epsilon(\lambda)$ for any adversary $\mathcal{A}$. If there is a PPT adversary $\mathcal{A}$ that wins the hiding game of Commitment 3.4 with probability larger than $\frac{1}{2} + \frac{1}{p(\lambda)}$. Then, we get a distinguisher $\mathcal{A}'$ that solves d-GAOP with probability larger than $\frac{1}{2} + \frac{1}{2p(\lambda)} - \frac{1}{2}\epsilon(\lambda)$.

$\mathcal{A}'$ needs to simulate the $\mathcal{CH}$ to call $\mathcal{A}$. For the group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$, given a d-GAOP instance y, $\mathcal{A}'$ sets $Ck = y$ and sends it to $\mathcal{A}$, then $\mathcal{A}$ chooses two messages $m_0, m_1 \in M$ to $\mathcal{A}'$. $\mathcal{A}'$ picks a $b \leftarrow \{0,1\}$ and generates $(c,d) \leftarrow \mathbf{Com}_y(m_b)$, sends c to $\mathcal{A}$ as the challenge. $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{A}'$ returns 1, i.e., y is from distribution $\mathcal{D}'_{G,X}$; otherwise, $\mathcal{A}'$ returns 0, i.e. y is from distribution $\mathcal{D}_{G,X}$. Thus, the probability that $\mathcal{A}'$ wins is $\Pr[S_0 \mid y = g \star x, (g,x) \leftarrow \mathcal{D}'_{G,X}] + \Pr[\bar{S}_1 \mid y = g \star x, (g,x) \leftarrow \mathcal{D}_{G,X}]$, which is larger than $\frac{1}{2}(\frac{1}{2} + \frac{1}{p(\lambda)}) + \frac{1}{2}(\frac{1}{2} - \epsilon(\lambda)) = \frac{1}{2} + \frac{1}{2p(\lambda)} - \frac{1}{2}\epsilon(\lambda)$. □

A.2 Security Proofs in Section 4

Theorem A.4 (Corresponding to Theorem 4.1) *Suppose that the group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ satisfies the d-GAOP assumption, where it is efficient to compute $Stab(x)$ from x given $(h,x) \leftarrow \mathcal{D}_{G,X}$. R is a re-randomized algorithm, and (E, F) is a trapdoor extractor. Then Commitment 4.1 is a dual mode commitment.*

Proof. The **Completeness** and **Perfectly Binding** properties are evident, as demonstrated in Theorem 3.2. We establish the **Trapdoor Property** and **Key Indistinguishability** below.

Trapdoor Property: For $\mathbf{TGen}(1^\lambda) \to (y, \mathrm{Stab}(y))$, where $y = h \star x$, $(h,x) \leftarrow \mathcal{D}_{G,X}$, and for any $m \in M$, the distribution

$$\{(c,d,m) \mid (c,d) \leftarrow \mathbf{Com}_y(m)\}$$

is $\{(\mathsf{E}(g) \cdot m, y'), g, m\}$, where $(g,y') \leftarrow \mathsf{R}(y)$. The distribution of

$$\{(c,d,m) \mid (c,(u,g)) \leftarrow \mathbf{TCom}(y, \mathrm{Stab}(y)), g' \leftarrow \mathbf{TCol}(y, \mathrm{Stab}(y), (u,g), m)\}$$

is $\{(u,y'), g', m\}$, where $(g,y') \leftarrow \mathsf{R}(y)$, $u \leftarrow M$, $r = u \cdot m^{-1}$, and $g' \leftarrow \mathsf{F}(r, g, y, \mathrm{Stab}(y))$.

Therefore, we only need to demonstrate that the distributions of

$$\{(\mathsf{E}(g), y'), g \mid (g,y') \leftarrow \mathsf{R}(y)\}$$

and

$$\{(r,y'), g' \mid (g,y') \leftarrow \mathsf{R}(y), g' \leftarrow \mathsf{F}(r, g, y, \mathrm{Stab}(y)), r \leftarrow M\}$$

are identical, which holds true based on the property of F and E is a local constant extractor.

Key Indistinguishability: This is obvious. If there is a PPT adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\Pr\left[b=\tilde{b} \;\middle|\; \begin{array}{c} Ck_0 \leftarrow \mathbf{Gen}(1^\lambda), Ck_1 \leftarrow \mathbf{TGen}(1^\lambda) \\ b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(Ck_b) \end{array}\right] \geq \frac{1}{2}+\frac{1}{p(\lambda)}.$$

Then, we get a distinguish $\mathcal{A}'$ that solves d-GAOP with the same probability. For a d-GAOP instance y, $\mathcal{A}'$ sends y to $\mathcal{A}$, then $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs. It is evident that the probability of $\mathcal{A}'$ winning is the same as the probability of $\mathcal{A}$ winning.

Thus, we get a dual-mode commitment if the d-GAOP assumption holds. □

Theorem A.5 (Corresponding to Theorem 4.2) *Suppose that group action $(G, X, \star, \mathcal{D}_{G,X})$ satisfies the s-GASP assumption, R is a re-randomization algorithm, and E is a homomorphic extractor. Then Commitment 4.2 is an Enhanced Linkable Commitment.*

Proof. We have already proven that the Commitment 3.1 is secure under the s-GASP assumption. Now, we only need to prove the enhanced linkable commitment scheme is secure, in fact, it is perfectly enhanced linkable hiding, computationally enhanced linkable binding and computationally enhanced linkable unforgeable.

Perfectly enhanced linkable hiding: If $Ck = y$, assume that $\mathbf{Com}_y(m_i) \to (c_i, d_i)$ where $(c_i, d_i) = ((c_{i1}, c_{i2}), d_i) = ((\mathsf{E}(g_i) \cdot m_i, y_i), g_i)$, $(y_i, g_i) \leftarrow \mathsf{R}(y)$ for $i \in [4]$ and $m_1 \cdot m_2^{-1} = m_3 \cdot m_4^{-1}$.

We only need to demonstrate that the distributions of $(c_1, c_2, d_1 \cdot d_2^{-1})$ and $(c_3, c_4, d_3 \cdot d_4^{-1})$ are identical. For the distributions of $(c_1, c_2, d_1 \cdot d_2^{-1}) = ((\mathsf{E}(g_1) \cdot m_1, y_1), (\mathsf{E}(g_2) \cdot m_2, y_2), g_1 \cdot g_2^{-1})$, by the properties of the homomorphic extractor E, $c_{11} = \mathsf{E}(d_1 \cdot d_2^{-1}) \cdot m_1 \cdot m_2^{-1} \cdot c_{21}$. Thus, combined with the properties of R and $m_1 \cdot m_2^{-1} = m_3 \cdot m_4^{-1}$, we only need to show

$$\begin{aligned} &\Pr[c_{21} = \mathsf{E}(g_1) \cdot m_2, d_1 \cdot d_2^{-1} = g_1 \cdot g_2^{-1} \mid c_{12} = y_1, c_{22} = y_2] \\ =&\Pr[c_{31} = \mathsf{E}(g_3) \cdot m_3, d_3 \cdot d_4^{-1} = g_3 \cdot g_4^{-1} \mid c_{32} = y_3, c_{42} = y_4] \end{aligned}$$

where $g_i \leftarrow \mathcal{I}(y, y_i)$ for $i \in [4]$. According to the independence of the distribution of g_i and the first property of the homomorphic extractor E, we know that the above probabilities are the same. Therefore, the scheme is perfectly enhanced linkable hiding.

Computationally enhanced linkable binding: Assume there is a PPT $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\Pr\left[\begin{array}{c} \mathbf{Open}(c_i, d_i) = m_i,\ i \in \{0,1\}, \\ \mathsf{Link}(c_0, c_1, d_L) = 1, \\ m_0 \cdot m_1^{-1} \neq \mathsf{LinkC}(c_0, c_1, d_L) \end{array} \;\middle|\; \begin{array}{c} Ck \leftarrow \mathbf{Gen}(1^\lambda), \\ \begin{pmatrix} m_0, m_1, c_0, c_1 \\ d_0, d_1, d_L \end{pmatrix} \leftarrow \mathcal{A}(Ck) \end{array}\right] \geq \frac{1}{p(\lambda)}.$$

Then we can construct a PPT $\mathcal{A}'$ that solves s-GASP with a probability larger than $\frac{1}{p(\lambda)}$.

For an s-GASP instance y, $\mathcal{A}'$ sets the $Ck = y$ and sends Ck to $\mathcal{A}$ in the enhanced linkable binding game. Then, $\mathcal{A}$ returns $(m_1, m_2, c_1, c_2, d_L)$ satisfying $\mathsf{Link}(c_0, c_1, d_L) = 1$ and $m_1 \cdot m_2^{-1} \neq \mathsf{LinkC}(m_1, m_2, d_L)$ to $\mathcal{A}'$, where $(c_i, d_i) = ((\mathsf{E}(g_i) \cdot m_i, y_i), g_i)$ for $i \in [2]$, and $d_L \in \mathcal{I}(y_2, y_1)$. Note that $g_1 \cdot g_2^{-1}$ is also in $\mathcal{I}(y_2, y_1)$, thus $\mathsf{E}(g_1) \cdot m_1 \cdot (\mathsf{E}(g_2) \cdot m_2)^{-1} \cdot \mathsf{E}(d_L)^{-1} \neq m_1 \cdot m_2^{-1}$ implies $\mathsf{E}(g_1 \cdot g_2^{-1}) \neq \mathsf{E}(d_L)$. Therefore, $e \neq d_L \cdot g_2 \cdot g_1^{-1}$ is a non-trivial element in $\mathrm{Stab}(y_1)$, thus $e \neq g_1^{-1} \cdot d_L \cdot g_2$ is a non-trivial element of $\mathrm{Stab}(y)$, thus $\mathcal{A}'$ solves the s-GASP with a probability larger than $\frac{1}{p(\lambda)}$.

Computationally enhanced linkable unforgeable: Assume there is a PPT $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\Pr\left[\mathcal{A} \text{ wins } \mathbf{ELU}(\Pi_{elc})\right] \geq \frac{1}{p(\lambda)}.$$

Then we can construct a PPT $\mathcal{A}'$ that solves s-GASP with a probability larger than $\frac{1}{2p(\lambda)}$.

For an s-GASP instance y, $\mathcal{A}'$ sets the $Ck = y$ and sends Ck to the $\mathcal{A}$ in the enhanced linkable unforgeable game. Then $\mathcal{A}$ returns two messages m_1 and m_2 to $\mathcal{A}'$. $\mathcal{A}'$ uses Ck to generate two commitments of m_1 and m_2, $\mathbf{Com}_y(m_i) \to (c_i, d_i) = ((\mathsf{E}(g_i) \cdot m_i, y_i), g_i)$ for $i \in [2]$ and sends (c_1, c_2) to $\mathcal{A}$. Then $\mathcal{A}'$ obtains a $d_L \in \mathcal{I}(y_2, y_1)$ from $\mathcal{A}$, i.e., $d_L \star y_2 = y_1$. Note that $\mathcal{A}'$ has a $g_1 \cdot g_2^{-1} \in \mathcal{I}(y_2, y_1)$, and it's uniformly distributed in $\mathcal{I}(y_2, y_1)$ by the property of R. Thus, $d_L \cdot g_2 \cdot g_1^{-1}$ is uniformly distributed in $\mathrm{Stab}(y_1)$, so $g_1 \cdot d_L \cdot g_2$ is uniformly distributed in $\mathrm{Stab}(y)$. Therefore, $e \neq g_1 \cdot d_L \cdot g_2$ with a probability larger than $\frac{1}{2}$, and thus $\mathcal{A}'$ solves s-GASP with a probability larger than $\frac{1}{2p(\lambda)}$. □

Theorem A.6 (Corresponding to Theorem 4.3) *Suppose that group action $(G, X, \star, \mathcal{D}_{G,X}, \mathcal{D}'_{G,X})$ satisfies the d-GAOP assumption and $\mathcal{D}_{G,X}$-one-way, R is a re-randomization algorithm, and E is a homomorphic local constant extractor. Then Commitment 4.3 is an Enhanced Linkable Commitment.*

Proof. We have already proven that the Commitment 3.2 is secure under the d-GAOP assumption. Now, we only need to prove the enhanced linkable commitment scheme is secure, in fact, it is computationally enhanced linkable hiding, perfectly enhanced linkable binding and computationally enhanced linkable unforgeable.

Computationally enhanced linkable hiding: Assume there is a PPT $\mathcal{A}$ and a polynomial $p(\cdot)$ such that

$$\Pr\left[\mathcal{A} \text{ wins } \mathbf{ELH}\left(\Pi_{elc}\right)\right] \geq \frac{1}{2} + \frac{1}{p(\lambda)}.$$

Then we can construct a PPT $\mathcal{A}'$ that solves d-GAOP with a probability larger than $\frac{1}{2} + \frac{1}{2p(\lambda)}$.

For a d-GAOP instance y, $\mathcal{A}'$ sets the $Ck = y$ and sends Ck to $\mathcal{A}$. Then $\mathcal{A}$ chooses messages $m_i, i \in [4]$ such that $m_1 \cdot m_2^{-1} = m_3 \cdot m_4^{-1}$ and sends them to $\mathcal{A}'$. $\mathcal{A}'$ picks a $b \leftarrow \{0,1\}$ and outputs two commitments $\mathbf{Com}_y(m_{1+2b}) \rightarrow (c_1, d_1)$, $\mathbf{Com}_y(m_{2+2b}) \rightarrow (c_2, d_2)$, and sends (c_1, c_2, d_L) to $\mathcal{A}$ where $d_L = d_1 \cdot d_2^{-1}$. $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{A}'$ returns 1, i.e., y is from $\mathcal{D}'_{G,X}$; otherwise $\mathcal{A}'$ returns 0, i.e., y is from $\mathcal{D}_{G,X}$.

Note that if y is from $\mathcal{D}_{G,X}$, the distribution of (c_1, c_2, d_L) is perfectly hiding the bit b due to Theorem 4.2. In this case, the probability that $\mathcal{A}$ guesses b correctly is $\frac{1}{2}$. Thus, the probability that $\mathcal{A}'$ guesses the distribution of y correctly is larger than $\frac{1}{2} + \frac{1}{2p(\lambda)}$.

Perfectly enhanced linkable binding: If $Ck = y$, where $y = h \star x$ and $(h, x) \leftarrow \mathcal{D}'_{G,X}$. Assume $\mathcal{A}$ is an adversary in the enhanced linkable binding game. If $\mathcal{A}$ outputs $(m_1, m_2, c_1, c_2, d_L)$ satisfying $\mathsf{Link}(c_1, c_2, d_L) = 1$ and $m_1 \cdot m_2^{-1} \neq \mathsf{LinkC}(c_1, c_2, d_L)$, where $(c_i, d_i) = ((\mathsf{E}(g_i) \cdot m_i, x_i), g_i)$ for $i \in [2]$, and $d_L \in \mathcal{I}(x_2, x_1)$, $g_1 \cdot g_2^{-1} \in \mathcal{I}(x_2, x_1)$, and $\mathsf{E}(g_1) \cdot \mathsf{E}(g_2)^{-1} \cdot m_1 \cdot m_2^{-1} \cdot \mathsf{E}(g_2 \cdot g_1^{-1})^{-1} = m_1 \cdot m_2^{-1} \neq \mathsf{E}(g_1) \cdot \mathsf{E}(g_2)^{-1} \cdot m_1 \cdot m_2^{-1} \cdot \mathsf{E}(d_L)^{-1} = \mathsf{LinkC}(c_1, c_2, d_L)$, this means $\mathsf{E}(g_2 \cdot g_1^{-1}) \neq \mathsf{E}(d_L)$, which contradicts the property of E. Thus, it is perfectly enhanced linkable binding.

Computationally enhanced linkable unforgeable: For this property, it holds under $\mathcal{D}_{G,X}$-one-way and d-GAOP assumptions. We use hybrid argument to complete the proof.

Game 0 is the standard enhanced linkable unforgeable game in Commitment 4.3. Let $\mathcal{A}$ be an adversary against the enhanced linkable unforgeable game. Let S_i be the event that $\mathcal{A}$ wins in *Game i*. $\mathcal{CH}$ interacts with the adversary $\mathcal{A}$.

1. **Gen:** $\mathcal{CH}$ generates the $Ck \leftarrow \mathbf{Gen}(1^\lambda)$, where $Ck = y = h \star x, (h, x) \leftarrow \mathcal{D}'_{G,X}$. $\mathcal{CH}$ sends it to $\mathcal{A}$.
2. **Choose:** $\mathcal{A}$ chooses two messages m_1, m_2 and sends them to $\mathcal{CH}$.
3. **Challenge:** $\mathcal{CH}$ commits the commitments of m_1 and m_2, i.e. $\mathbf{Com}_y(m_i) \rightarrow (c_i, d_i) = ((\mathsf{E}(g_i) \cdot m_i, y_i), g_i)$ where $(y_i, g_i) \leftarrow \mathsf{R}(y)$, for $i \in [2]$. $\mathcal{CH}$ sends (c_1, c_2) to $\mathcal{A}$.
4. **Return:** $\mathcal{A}$ returns a $d_L \in G$ and $\mathcal{A}$ wins if $\mathsf{Link}(c_1, c_2, d_L) = 1$.

Game 1 is similar to *Game 0* except that $\mathcal{CH}$ generates the $Ck = y = h \star x, (h, x) \leftarrow \mathcal{D}_{G,X}$. This is the standard enhanced linkable unforgeable game in Commitment 4.2

1. **Gen:** $\mathcal{CH}$ generates the $Ck \leftarrow \mathbf{Gen}(1^\lambda)$, where $Ck = y = h \star x, (h, x) \leftarrow \mathcal{D}_{G,X}$. $\mathcal{CH}$ sends it to $\mathcal{A}$.
2. **Choose:** $\mathcal{A}$ chooses two messages m_1, m_2 and sends them to $\mathcal{CH}$.
3. **Challenge:** $\mathcal{CH}$ generates the commitments of m_1 and m_2, $\mathbf{Com}_y(m_i) \rightarrow (c_i, d_i) = ((\mathsf{E}(g_i) \cdot m_i, y_i), g_i)$ where $(y_i, g_i) \leftarrow \mathsf{R}(y)$, $i \in [2]$. $\mathcal{CH}$ sends (c_1, c_2) to $\mathcal{A}$.
4. **Return:** $\mathcal{A}$ returns a $d_L \in G$ and $\mathcal{A}$ wins if $\mathsf{Link}(c_1, c_2, d_L) = 1$.

Game 2 is similar to *Game 1* except that $\mathcal{CH}$ generates the commitment of m_2 in a different way.

1. **Gen:** $\mathcal{CH}$ generates the $Ck \leftarrow \mathbf{Gen}(1^\lambda)$, where $Ck = y = h \star x, (h, x) \leftarrow \mathcal{D}_{G,X}$. $\mathcal{CH}$ sends Ck to $\mathcal{A}$, and $\mathcal{CH}$ generates a $y' \in \mathcal{O}(y)$ where $(y', g') \leftarrow \mathsf{R}(y)$.
2. **Choose:** $\mathcal{A}$ chooses two messages m_1, m_2 sends them to $\mathcal{CH}$.
3. **Challenge:** $\mathcal{CH}$ generates the commitments of m_1 and m_2, $\mathbf{Com}_y(m_1) \to (c_1, d_1) = ((\mathsf{E}(g_1) \cdot m_1, y_1), g_1)$ where $(y_1, g_1) \leftarrow \mathsf{R}(y)$. $\mathbf{Com}_{y'}(m_2) \to (c_2, d_2) = ((\mathsf{E}(g_2) \cdot m_2, y_2), g_2)$ where $(y_2, g_2) \leftarrow \mathsf{R}(y')$. $\mathcal{CH}$ send (c_1, c_2) to $\mathcal{A}$.
4. **Return:** $\mathcal{A}$ returns a $d_L \in G$ and $\mathcal{A}$ wins if $\mathsf{Link}(c_1, c_2, d_L) = 1$.

Similar to the proof in Theorem 3.2, suppose the d-GAOP assumption, then $|\Pr[S_0] - \Pr[S_1]| \leq \epsilon(\lambda)$, where $\epsilon(\lambda)$ is a negligible function of λ. Next, we show $\Pr[S_1] = \Pr[S_2]$ and if $\Pr[S_2] \geq \frac{1}{p(\lambda)}$ for some polynomial $p(\cdot)$, then we construct an $\mathcal{A}'$ that solves the $\mathcal{D}_{G,X}$-one-way assumption with a probability larger than $\frac{1}{p(\lambda)}$.

Note that the distributions of (c_1, c_2) is independent of m_1 and m_2 in *Game 1* and *Game 2*, which are same as the distribution of $((u_1, y_1), (u_2, y_2))$ where $(h, x) \leftarrow \mathcal{D}_{G,X}$, $u_1 \leftarrow M$, $u_2 \leftarrow M$ and $y_1 \leftarrow \mathsf{R}(x)[1]$, $y_2 \leftarrow \mathsf{R}(x)[1]$ due to the properties of R and E, thus $\Pr[S_2] = \Pr[S_1]$. Then, if there is an $\mathcal{A}$ that wins *Game 2* with probability larger than $\frac{1}{p(\lambda)}$. For a $\mathcal{D}_{G,X}$-one-way instance $(x, y = h \star x)$, where $(h, x) \leftarrow \mathcal{D}_{G,X}$ and $\mathcal{A}'$ needs to find a $g \in G$ s.t. $g \star x = y$. $\mathcal{A}'$ sets the $Ck = y$ and sends Ck to the $\mathcal{A}$ that in the enhanced linkable unforgeable game. Then $\mathcal{A}$ returns two messages m_1 and m_2 to $\mathcal{A}'$. $\mathcal{A}'$ generates two commitments m_1 and m_2 respectively. Specifically, $\mathbf{Com}_y(m_1) \to (c_1, d_1) = ((\mathsf{E}(g_1) \cdot m_1, y_1), g_1)$ where $(y_1, g_1) \leftarrow \mathsf{R}(y)$, $\mathbf{Com}_x(m_2) \to (c_2, d_2) = ((\mathsf{E}(g_2) \cdot m_2, y_2), g_2)$ where $(y_2, g_2) \leftarrow \mathsf{R}(x)$ and sends (c_1, c_2) to $\mathcal{A}$. $\mathcal{A}$ outputs a d_L s.t. $d_L \star y_2 = y_1$ with a probability larger than $\frac{1}{p(\lambda)}$. Thus $\mathcal{A}'$ gets a $g = g_1^{-1} \cdot d_L \cdot g_2$ s.t. $g \star x = y$ with a probability larger than $\frac{1}{p(\lambda)}$. □

Appendix B Discussion regarding extractors E

For a group action $(G, X, \star)$ with a distribution $\mathcal{D}_{G,X}$ on $G \times X$. A natural question is whether we can find a finite group M and a deterministic extractor $\mathsf{E} : G \to M$ in Definition 3.2 such that $|M|$ is as large as possible. To extract more bits, a natural idea is to take a good distribution $\mathcal{D}_{G,X}$.

In lattice automorphisms, as we mentioned earlier, for $n > 10$, $\mathbb{Z}^n$ has the largest automorphism group for any n-dimensional lattice. Therefore, it is natural to think that setting $\mathbf{Q} \in [\mathbf{I}_n]$ is a good choice in Lemma 5.2.

The Optimal Homomorphic Extractor E under $\mathbb{Z}$LAP. For $\mathbf{Q} \in [\mathbf{I}_n]$ and the group action $(\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}], \star, \mathcal{D}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}),[\mathbf{Q}]})$ in Lemma 5.2, we show that the homomorphic extractor E can only be the determinant function, so extracting one bit is optimal in this case.

Note that for $\mathbf{Q} = \mathbf{U}\mathbf{U}^\top \in [\mathbf{I}_n]$, and for any $\mathbf{Q}' \in [\mathbf{I}_n]$, the set $\mathcal{I}(\mathbf{Q}, \mathbf{Q}') = \{\mathbf{V}\mathbf{U}\mathbf{S}\mathbf{U}^{-1} : \mathbf{S} \in \mathcal{S}_n^{\pm}\}$, where $\mathbf{V} \in \mathcal{I}(\mathbf{Q}, \mathbf{Q}')$. Therefore, for a homomorphic

extractor E, where $\mathsf{E} : \mathrm{GL}_n(\mathbb{Z}) \to M$ is a surjective group homomorphism, $\mathsf{E}|_{\mathcal{S}_n^{\pm}} : \mathcal{S}_n^{\pm} \to M$ is also a surjective group homomorphism. Hence, in this scenario, we can rephrase the problem of finding a homomorphic extractor E that maximizes $|M|$ as the following question:

Question 1. Let $\mathsf{E} : \mathrm{GL}_n(\mathbb{Z}) \to M$ be a surjective group homomorphism with $|M| > 2$, and let $\phi : \mathcal{S}_n^{\pm} \to M$ also be a surjective group homomorphism (which are the restrictions of E on $\mathcal{S}_n^{\pm}$). Is there a E that satisfies these conditions?

If such a E exists, consider $\ker(\phi)$, which is a normal subgroup of $\mathcal{S}_n^{\pm}$. We have $\mathsf{E} : \mathrm{GL}_n(\mathbb{Z}) \to M \simeq \mathcal{S}_n^{\pm}/\ker(\phi)$. This can be denoted as $f : \mathrm{GL}_n(\mathbb{Z}) \to \mathcal{S}_n^{\pm}/\ker(\phi)$, and then $f|_{\mathcal{S}_n^{\pm}} = (\mathcal{S}_n^{\pm} \xrightarrow{\phi} M \simeq \mathcal{S}_n^{\pm}/\ker(\phi))$ represents the canonical quotient map. The following theorem illustrates that such an f does not exist, leading to the conclusion that such an E does not exist.

Theorem B.1 *If the proper normal subgroup H of $\mathcal{S}_n^{\pm}$ and the group homomorphism $f : GL_n(\mathbb{Z}) \to \mathcal{S}_n^{\pm}/H$ is such that $f|_{\mathcal{S}_n^{\pm}}$ is the canonical quotient map $\pi : \mathcal{S}_n^{\pm} \to \mathcal{S}_n^{\pm}/H$, then $\mathcal{S}_n^{\pm}/H \simeq \{\pm 1\}$ and the induced $\widetilde{f} : GL_n(\mathbb{Z}) \to \{\pm 1\}$ is just $\det|_{GL_n(\mathbb{Z})}$, that is, $\ker(f) = SL_n(\mathbb{Z}) = \ker(\det|_{GL_n(\mathbb{Z})})$.*

To complete the proof of Theorem B.1, we need to introduce some structural properties of $\mathrm{GL}_n(\mathbb{Z})$ and $\mathcal{S}_n^{\pm}$.

Definition B.1 *For R a unital ring, we denote by $E_n(R)$ the subgroup of $GL_n(R)$ generated by all transvections $e_{ij}(r) = I_n + r\epsilon_{ij}$ (a.k.a. the elementary matrices over R) with $r \in R$, $1 \le i \ne j \le n$, where I_n is the identity matrix and ϵ_{ij} is the matrix whose (i, j) entry is 1 while all its other entries are zero.*

Lemma B.1 ([40, Thm 4.3.9]) *Let R be a commutative ring. If R is a Euclidean domain, then $SL_n(R) = E_n(R)$ for all n.*

Proposition B.1 $[GL_n(\mathbb{Z}), GL_n(\mathbb{Z})] = SL_n(\mathbb{Z})$.

Proof. Since $e_{ik}(s) = [e_{ij}(1), e_{jk}(s)] \in [\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})]$, $\forall s \in \mathbb{Z}$, $E_n(\mathbb{Z}) \subseteq [\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})]$. By Lemma B.1, $\mathrm{SL}_n(\mathbb{Z}) = E_n(\mathbb{Z}) \subseteq [\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})]$. And $\mathrm{GL}_n(\mathbb{Z})/\mathrm{SL}_n(\mathbb{Z}) \simeq C_2$ is abelian $\Rightarrow [\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})] \subseteq \mathrm{SL}_n(\mathbb{Z})$.
Thus $[\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})] = \mathrm{SL}_n(\mathbb{Z})$. □

And we can know that its normal subgroups have only the following possibilities [45]. We only consider the case of $n \ge 5$, and the following proposition provides a formal statement. We provide a proof in the following proposition.

Proposition B.2 *Assume $n \ge 5$. We regard $\mathcal{S}_n^{\pm}$ as the subgroup of $GL_n(\mathbb{Z})$ and $\det : \mathcal{S}_n^{\pm} \to \{\pm 1\}$ is taking determinant. Then all the normal subgroups of $\mathcal{S}_n^{\pm}$ are*

$$\{I_n\}, \{\pm I_n\}, \ker(\det) \cap D_n \triangleq D_{n-1}, D_n, D_{n-1} \rtimes A_n, D_n \rtimes A_n, D_{n-1} \rtimes \mathcal{S}_n, \ker(\det), \mathcal{S}_n^{\pm}.$$

Here A_n is the alternate permutation group and D_{n-1} is also $\ker(\det|_{D_n})$.

Lemma B.2 *If $H \trianglelefteq \mathcal{S}_n^{\pm}$ and $H \leq D_n$, then H is one of $\{I_n\}, \{\pm I_n\}, D_{n-1}, D_n$.*

Proof. If $H \neq \{I_n\}, \{\pm I_n\}$, then $\exists\, d = \mathrm{diag}(d_1, \ldots, d_n) \in H$ s.t. $d \neq \pm I_n$. Assume $d_i = 1, d_j = -1$. Since $H \trianglelefteq \mathcal{S}_n^{\pm}$, $(i\,j)d(i\,j)^{-1} \in H$. Then $\widetilde{d} := (i\,j)d(i\,j)^{-1}d \triangleq \mathrm{diag}(\widetilde{d}_1, \ldots, \widetilde{d}_n) \in H$ and $\widetilde{d}_i = \widetilde{d}_j = -1$, $\widetilde{d}_s = 1$, $\forall s \in \{1, \ldots, n\} \setminus \{i, j\}$. Denote such $\widetilde{d}$ by $d_{i,j}$. $\forall\, k \neq \ell \in \{1, \ldots, n\}$, take $\sigma = \begin{pmatrix} i & j \\ k & \ell \end{pmatrix} \in \mathcal{S}_n$ (means exchange the i-th row and k-th row, j-th row and ℓ-th row), then $d_{k,\ell} = \sigma d_{i,j}\sigma^{-1} \in H$. Thus $D_{n-1} = \langle d_{k,\ell} | k \neq \ell \in \{1, \ldots, n\} \rangle \subseteq H$. Note $[D_n : D_{n-1}] = 2$, so $H = D_n$ or D_{n-1}. □

Proof (of Proposition B.2). Note $\mathcal{S}_n^{\pm} = D_n \rtimes \mathcal{S}_n$ i.e. we can write the element in $\mathcal{S}_n^{\pm}$ as $d \cdot s$ where $d \in D_n$, $s \in \mathcal{S}_n$. We will use this notation in the proof. And we have the surjective group homomorphism $\pi :\ d \cdot s \in \mathcal{S}_n^{\pm} \mapsto s \in \mathcal{S}_n$. For $H \trianglelefteq \mathcal{S}_n^{\pm}$, denote $\pi|_H$ by π_H. Then $\mathrm{Im}(\pi_H) = \pi(H) \trianglelefteq \pi(G) = \mathcal{S}_n$. $n \geq 5$ implies $\mathrm{Im}(\pi_H) = \{I_n\}$, or A_n, or $\mathcal{S}_n$.
If $A_n \subseteq \mathrm{Im}(\pi_H)$, then $\exists d \in D_n$ s.t. $d \cdot (1\,2\,3) \in H$. Since $H \trianglelefteq G$, $d \cdot s \in H \Rightarrow s \cdot d = d^{-1}(ds)d \in H$. So $(1\,2\,3)d \in H$ and then $(1\,3\,2) = (1\,2\,3)d \cdot d(1\,2\,3) \in H$. This implies $A_n \subseteq H$. Consider $d' := \mathrm{diag}(-1, 1, 1, \ldots, 1)$, then $d'^{-1}(1\,2\,3)d' \cdot (1\,2\,3)^{-1} \in H$. Following the same discussion as in the Lemma B.2 and noting that $d'^{-1}(1\,2\,3)d' \cdot (1\,2\,3)^{-1} = d_{1,3}$, we have $D_{n-1} \subseteq H$.

(1) $\mathrm{Im}(\pi_H) = \mathcal{S}_n$, then $A_n \subseteq H$, $D_{n-1} \subseteq H$.
  i) $H \cap \mathcal{S}_n = \mathcal{S}_n$
    a) $H \cap D_n = D_n \Rightarrow \mathcal{S}_n^{\pm} = D_n \rtimes \mathcal{S}_n \subseteq H \subseteq \mathcal{S}_n^{\pm} \Rightarrow H = \mathcal{S}_n^{\pm}$.
    b) $H \cap D_n = D_{n-1} \Rightarrow D_{n-1} \rtimes \mathcal{S}_n \subseteq H$. $\forall d \cdot s \in H$, note $s \in \mathcal{S}_n \subseteq H \Rightarrow d \in H \Rightarrow d \in D_{n-1}$. So $H \subseteq D_{n-1} \rtimes \mathcal{S}_n$, and then $H = D_{n-1} \rtimes \mathcal{S}_n$.
  ii) $H \cap \mathcal{S}_n = A_n$. $\mathrm{Im}(\pi_H) = \mathcal{S}_n \Rightarrow \forall\, s \in \mathcal{S}_n \setminus A_n$, $\exists d_s \in D_n$ s.t. $d_s \cdot s \in H$. Note $s \notin H \Rightarrow d_s \notin H \Rightarrow d_s \notin D_{n-1}$. This means $D_n \nsubseteq H$ and $\forall d' \in D_n \setminus D_{n-1}, d'd_s^{-1} \in D_{n-1} \subseteq H \Rightarrow d's = d'd_s^{-1}d_s s \in H$.
  We already have $D_{n-1} \rtimes A_n \subseteq H$. $\forall s \in A_n$, if $d \in D_n \setminus D_{n-1}$ s.t. $ds \in H$, then $s \in A_n \subseteq H \Rightarrow d \in H \Rightarrow D_n \subseteq H$. It's a contradiction.
  So $\forall ds \in G$, $ds \in H \iff s \in \mathcal{S}_n \setminus A_n$, $d \in D_n \setminus D_{n-1}$ or $s \in A_n$, $d \in D_{n-1} \iff \det(ds) = 1$. Thus $H = \ker(\det)$.

(2) $\mathrm{Im}(\pi_H) = A_n$, then $A_n \subseteq H$, $D_{n-1} \subseteq H$. $\mathrm{Im}(\pi_H) = A_n \Rightarrow H \subseteq D_n \rtimes A_n$. $A_n \subseteq H, D_{n-1} \subseteq H \Rightarrow D_{n-1} \rtimes A_n \subseteq H$. So $D_{n-1} \rtimes A_n \leq H \leq D_n \rtimes A_n \Rightarrow H = D_{n-1} \rtimes A_n$ or $D_n \rtimes A_n$.

(3) $\mathrm{Im}(\pi_H) = \{I_n\}$, then $H \subseteq D_n$. By Lemma B.2, the normal subgroups of D_n are $\{I_n\}, \{\pm I_n\}, D_{n-1}, D_n$. □

Proof (of Theorem B.1). Firstly, we have $[\mathcal{S}_n^{\pm}, \mathcal{S}_n^{\pm}] = [D_{n-1} \rtimes A_n]$ since it's clearly the minimum normal subgroup that makes the quotient group abelian from Proposition B.2. So, if $D_{n-1} \rtimes A_n \leq H$, then $\mathrm{GL}_n(\mathbb{Z})/\ker(f)$ is abelian $\Rightarrow \mathrm{SL}_n(\mathbb{Z}) = [\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})] \leq \ker(f)$. $[\mathrm{GL}_n(\mathbb{Z}) : \mathrm{SL}_n(\mathbb{Z})] = 2$ and $\ker(f) \neq \mathrm{GL}_n(\mathbb{Z})$ implies $\ker(f) = \mathrm{SL}_n(\mathbb{Z}) = \ker(\det|_{\mathrm{GL}_n(\mathbb{Z})})$.
Next, we just need to show H cannot be any of $\{I_n\}, \{\pm I_n\}, D_{n-1}, D_n$.
$f(\mathrm{SL}_n(\mathbb{Z})) = f([\mathrm{GL}_n(\mathbb{Z}), \mathrm{GL}_n(\mathbb{Z})]) = [f(\mathrm{GL}_n(\mathbb{Z})), f(\mathrm{GL}_n(\mathbb{Z}))] = [\mathcal{S}_n^{\pm}/H, \mathcal{S}_n^{\pm}/H] =$

$[\mathcal{S}_n^{\pm}, \mathcal{S}_n^{\pm}]/H = (D_{n-1} \rtimes A_n)/H$.
$[f(\mathrm{GL}_n(\mathbb{Z})) : f(\mathrm{SL}_n(\mathbb{Z}))] \leq [\mathrm{GL}_n(\mathbb{Z}) : \mathrm{SL}_n(\mathbb{Z})] = 2$ and the equality holds if and only if $\ker(f) \leq \mathrm{SL}_n(\mathbb{Z})$.
So $[\mathcal{S}_n^{\pm}/H : (D_{n-1} \rtimes A_n)/H] \leq 2$ and the equality holds if and only if $\ker(f) \leq \mathrm{SL}_n(\mathbb{Z})$.$(*)$
For $(H = \{I_n\})$, $(H = D_{n-1})$, or $(H = \{\pm I_n\}$ and n is even), we have $H \leq D_{n-1} \rtimes A_n$ and then $[\mathcal{S}_n^{\pm}/H : (D_{n-1} \rtimes A_n)/H] = [\mathcal{S}_n^{\pm} : D_{n-1} \rtimes A_n] = 4$. It's a contradiction with $(*)$.
For $(H = D_n)$ or $(H = \{\pm I_n\}$ and n is odd), $H \leq \ker(f)$ and $-I_n, \mathrm{diag}(-1, 1, \ldots, 1) \notin \mathrm{SL}_n(\mathbb{Z})$, so $[\mathcal{S}_n^{\pm}/H : (D_{n-1} \rtimes A_n)/H]$ should be 1 by $(*)$ but it's obviously not 1. □

How to find a such E in other isomorphism problems From the preceding discussion, concerning a group action $(G, X, \star, \mathcal{D}_{G,X})$, it becomes evident that a natural E as outlined in Definition 3.2 is likely a surjective homomorphic mapping of G. For example, Similarly, let $\mathsf{E}(\cdot)$ be $\det(\cdot)$, in the group action of module-LIP or code equivalence problem, it is possible to extract more bits. The task of discovering an improved E and its corresponding group action is left to future research.

Appendix C Direct Sum of Lattices

The following lemma seems to be folklore, but we have not found any correct proof in the existing literature.

Lemma C.1 *For an n_1-dimensional irreducible lattice $\mathcal{L}_1$ and an n_2-dimensional lattice $\mathcal{L}_2$, if $\mathcal{L} = \mathcal{L}_1 \oplus \mathcal{L}_2$, then for any $\mathbf{O} \in Aut(\mathcal{L})$, either $\mathbf{O} \cdot \mathcal{L}_1 \subseteq \mathcal{L}_1$ or $\mathbf{O} \cdot \mathcal{L}_1 \subseteq \mathcal{L}_2$ always holds* [4].

Proof. Let $n = n_1 + n_2$, for any $\mathbf{O} \in \mathrm{Aut}(\mathcal{L}) \subset \mathrm{O}_n(\mathbb{R})$. Let $\mathbf{x} \in \mathcal{L}_1$ satisfies $\|\mathbf{x}\| = \lambda_1(\mathcal{L}_1)$, then $\exists\, j \in [2]$ s.t. $\mathbf{Ox} \in \mathcal{L}_j$ due to the orthogonal decomposition structure of $\mathcal{L}$, thus $\mathbb{U} = \mathbb{R}\mathbf{x} \subseteq \mathcal{L}_1 \otimes_{\mathbb{Z}} \mathbb{R}$ and $\mathbb{V} = \mathbf{O} \cdot \mathbb{U} \subseteq \mathcal{L}_j \otimes_{\mathbb{Z}} \mathbb{R}$.

In induction, assume that $\mathbb{U}$ is a rank k subspace of $\mathcal{L}_1 \otimes_{\mathbb{Z}} \mathbb{R}$, $k < n_1$ and $\mathbb{V} = \mathbf{O} \cdot \mathbb{U} \subset \mathcal{L}_j \otimes_{\mathbb{Z}} \mathbb{R}$. Because $\mathcal{L}_1$ is irreducible, therefore $\mathcal{L}_1 \neq (\mathbb{U} \cap \mathcal{L}_1) \oplus (\mathbb{U}^{\perp} \cap \mathcal{L}_1)$, let $\mathbf{v} \in \mathcal{L}_1 - ((\mathbb{U} \cap \mathcal{L}_1) \oplus (\mathbb{U}^{\perp} \cap \mathcal{L}_1))$ be a such vector, let $\pi_{\mathbb{U}}(\mathbf{v}) = \mathbf{s}$, note that $\mathbf{s} \notin \mathcal{L}_1$. Let $\mathbf{v}_1 \in \mathcal{L}_1$ be one of the shortest vectors satisfying $\pi_{\mathbb{U}}(\mathbf{v}_1) = \mathbf{s}$, note that $\mathbf{v}_1 \notin \mathbb{U}$. Let $A_1 = \{\mathbf{v} \in \mathcal{L}_1 : \pi_{\mathbb{U}}(\mathbf{v}) = \mathbf{s}\}$, $A = \{\mathbf{v} \in \mathcal{L} : \pi_{\mathbb{U}}(\mathbf{v}) = \mathbf{s}\}$, then $\{\mathbf{v} \in A_1 : \forall \mathbf{y} \in A_1, \|\mathbf{y}\| \geq \|\mathbf{v}\|\} = \{\mathbf{v} \in A : \forall \mathbf{y} \in A, \|\mathbf{y}\| \geq \|\mathbf{v}\|\}$, due to the orthogonal decomposition structure of $\mathcal{L}$.

We will show $\mathbf{Ov}_1 \in \mathcal{L}_j$, thus lifting the rank of $\mathbb{U}$, if we let $\mathbb{U} = (\mathbb{U}, \mathbf{v}_1)$. Note that $\mathbf{Ov}_1 \in \mathcal{L}$ is the one of the shortest vectors $\mathbf{y}$ satisfing $\pi_{\mathbb{V}}(\mathbf{y}) = \mathbf{Os}$, since $\pi_{\mathbb{V}}(\mathbf{O}(\cdot)) = \mathbf{O}(\pi_{\mathbb{U}}(\cdot))$. And let $B_j = \{\mathbf{v} \in \mathcal{L}_j : \pi_{\mathbb{V}}(\mathbf{v}) = \mathbf{Os}\}$, $B = \{\mathbf{v} \in \mathcal{L} :$

[4] Here, $\mathcal{L}_1$ denotes $\begin{pmatrix} \mathcal{L}_1 \\ 0 \end{pmatrix} \in \mathcal{L}$, and $\mathcal{L}_2$ denotes $\begin{pmatrix} 0 \\ \mathcal{L}_2 \end{pmatrix} \in \mathcal{L}$.

$\pi_{\mathbb{V}}(\mathbf{v}) = \mathbf{Os}\}$, then $\{\mathbf{v} \in B_j : \forall \mathbf{y} \in B_j, \|\mathbf{y}\| \geq \|\mathbf{v}\|\} = \{\mathbf{v} \in B : \forall \mathbf{y} \in B, \|\mathbf{y}\| \geq \|\mathbf{v}\|\}$, due to the orthogonal decomposition structure of $\mathcal{L}$. Thus, $\mathbf{Ov}_1 \in \mathcal{L}_j$.

By induction, for any $\mathbf{O} \in \mathrm{Aut}(\mathcal{L})$, $\exists\, j \in [2]$ s.t. $\mathbf{O} \cdot \mathcal{L}_1 \subseteq \mathcal{L}_j \otimes_{\mathbb{Z}} \mathbb{R}$, thus $\mathbf{O} \cdot \mathcal{L}_1 \subseteq \mathcal{L}_j$ due to $\mathcal{L} \cap (\mathcal{L}_j \otimes_{\mathbb{Z}} \mathbb{R}) = \mathcal{L}_j$. □

Proof (of Proposition 5.1). For the lattices $\mathcal{N}_i$, $i \in [4]$, it is easy to observe that if $\mathrm{genus}(\mathcal{N}_1) = \mathrm{genus}(\mathcal{N}_2)$ and $\mathrm{genus}(\mathcal{N}_3) = \mathrm{genus}(\mathcal{N}_4)$, then $\mathrm{genus}(\mathcal{N}_1 \oplus \mathcal{N}_3) = \mathrm{genus}(\mathcal{N}_2 \oplus \mathcal{N}_4)$, hence $\mathrm{genus}(\oplus_{i=1}^m \mathcal{L}_1) = \mathrm{genus}(\oplus_{i=1}^m \mathcal{L}_2)$. From Lemma C.1, we can observe that for any $\mathbf{O} \in \mathrm{Aut}(\oplus_{i=1}^m \mathcal{L}_1)$, $\mathbf{O}$ acts on irreducible lattice blocks $\mathcal{L}_1$ as a signed permutation among these blocks, thus $\mathrm{Aut}(\oplus_{i=1}^m \mathcal{L}_j) = \{(\mathbf{S} \otimes \mathbf{I}_k) \cdot \mathrm{diag}(\mathbf{A}_1, \cdots, \mathbf{A}_m) \mid \mathbf{S} \in \mathcal{S}_m^{\pm}, \mathbf{A}_i \in \mathrm{Aut}(\mathcal{L}_j), i \in [m]\}$, $j \in \{1, 2\}$. □

Lemma C.2 (Corresponding to Lemma 5.5) *The above pair of algorithms* (E, F) *forms a trapdoor extractor as described in Definition 4.2.*

Proof. We demonstrate that (E, F) forms a trapdoor extractor as described in Definition 4.2. The correctness is obvious, as it always holds that $\mathbf{U}' \in \mathcal{I}(\mathbf{Q}_0', \mathbf{Q}_0'')$ and $\mathsf{E}(\mathbf{U}') = u$.

It is notable that for $n > 10$, any n-dimensional lattice $\mathcal{L}$ satisfies $|\mathrm{Aut}(\mathcal{L})| \leq 2^n \cdot n!$ as mentioned in Section 5.5. Hence, it follows that $\mathrm{Stab}(\mathbf{Q}_0')$ always possesses a polynomial-sized generating set. Because $\mathcal{L}_0 \in \Lambda_{-}^n$ and $\mathsf{E} : \mathrm{GL}_n^{\pm}(\mathbb{Z}) \to \{\pm 1\}$ is a surjective group homomorphism, there always exists an $\mathbf{V} \in S$ such that $\det(\mathbf{V}) = -1$. Therefore, in Algorithm F, finding such an element $\mathbf{V} \in \mathrm{Stab}(\mathbf{Q}_0')$ is efficient given a polynomial-sized generating set of $\mathrm{Stab}(\mathbf{Q}_0')$. Thus, F is a probabilistic polynomial-time algorithm.

Moreover, due to $\mathcal{L}_0 \in \Lambda_{-}^n$, the distributions of $(\mathsf{E}(\mathbf{U}), \mathbf{U})$ and $(u, \mathbf{U}')$ are identical for any $\mathbf{Q}_0', \mathbf{Q}_0'' \in [\mathbf{Q}_0]$, where $\mathbf{U} \leftarrow \mathcal{I}(\mathbf{Q}_0', \mathbf{Q}_0'')$, $u \leftarrow M$, $g' \leftarrow \mathsf{F}(u, \mathbf{U}, \mathbf{Q}_0', \mathrm{Stab}(\mathbf{Q}_0'))$, and $(\mathbf{U}_0, \mathbf{Q}_0) \leftarrow \mathcal{D}^{(0)}_{\mathrm{GL}_n^{\pm}(\mathbb{Z}), [\mathbf{Q}_0] \cup [\mathbf{Q}_1]}$ □

Theorem C.1 (Corresponding to Theorem 6.1) *Assuming that it is difficult to distinguish among the* 2^n *lattices* $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i \in \{0,1\}}$*, i.e., the decisional* $\mathit{LIP}(\oplus_{i=1}^n \mathcal{L}_i^{m_i}, \oplus_{i=1}^n \mathcal{L}_i^{m_i'})$ *is hard for any* $\mathbf{m}, \mathbf{m}' \in \{0, 1\}^n$*, then Commitment 6.1 is perfectly binding and computationally hiding.*

Proof. The correctness is obvious. We prove the hiding and binding properties.

Perfect Binding: This property holds because the receiver can use the witness $\mathbf{w}$ to verify that the lattices $\{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i \in \{0,1\}}$ are pairwise non-isomorphic, ensuring that the commitment is perfectly binding.

Computational Hiding: For $\mathbf{Gen}(1^\lambda) \to Ck = \{\mathbf{Q}_i^0, \mathbf{Q}_i^1\}_{i=1}^n$. If there exists a polynomial $p(\cdot)$ and a PPT $\mathcal{A}$ wins the hiding game in Commitment 6.1 with a probability larger than $\frac{1}{2} + \frac{1}{p(\lambda)}$, then we can construct a PPT adversary $\mathcal{A}'$ that solves some decisional $\mathrm{LIP}(\mathcal{L}, \mathcal{L}')$ with a probability larger than $\frac{1}{2} + \frac{1}{p(\lambda)}$, where $\mathcal{L}, \mathcal{L}' \in \{\oplus_{i=1}^n \mathcal{L}_i^{j_i}\}_{j_i \in \{0,1\}}$. $\mathcal{A}'$ sends Ck to $\mathcal{A}$. Then $\mathcal{A}$ returns $\mathbf{m}_0, \mathbf{m}_1 \in M$. For a decisional $\mathrm{LIP}(\mathbf{Q}_{\mathbf{m}_0}, \mathbf{Q}_{\mathbf{m}_1})$ instance $\mathbf{Q}$, $\mathcal{A}$ uses the re-randomization algorithm from Section 5.2 to generate $(\mathbf{Q}', \mathbf{U}) \leftarrow \mathsf{R}(\mathbf{Q})$, and sends $\mathbf{Q}'$ to $\mathcal{A}$. Then $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs. It is evident that the probability of $\mathcal{A}'$ winning is the same as the probability of $\mathcal{A}$ winning. □