Updatable, Aggregatable, Succinct Mercurial Vector Commitment from Lattice

Hongxiao Wang[1], Siu-Ming Yiu[1(✉)], Yanmin Zhao[1], and Zoe L. Jiang[2]

[1] The University of Hong Kong, China
{hxwang, smyiu, ymzhao}@cs.hku.hk
[2] Harbin Institute of Technology, Shenzhen, China
zoeljiang@hit.edu.cn

Abstract. Vector commitments (VC) and their variants attract a lot of attention due to their wide range of usage in applications such as blockchain and accumulator. Mercurial vector commitment (MVC), as one of the important variants of VC, is the core technique for building more complicated cryptographic applications, such as the zero-knowledge set (ZKS) and zero-knowledge elementary database (ZK-EDB). However, to the best of our knowledge, the only post-quantum MVC construction is trivially implied by a generic framework proposed by Catalano and Fiore (PKC '13) with lattice-based components which causes *large* auxiliary information and *cannot satisfy* any additional advanced properties, that is, updatable and aggregatable.
A major difficulty in constructing a *non-black-box* lattice-based MVC is that it is not trivial to construct a lattice-based VC that satisfies a critical property called "mercurial hiding". In this paper, we identify some specific features of a new falsifiable family of basis-augmented SIS assumption (BASIS) proposed by Wee and Wu (EUROCRYPT '23) that can be utilized to construct the mercurial vector commitment from lattice *satisfying* updatability and aggregatability with *smaller* auxiliary information. We *first* extend stateless update and differential update to the mercurial vector commitment and define a *new* property, named updatable mercurial hiding. Then, we show how to modify our constructions to obtain the updatable mercurial vector commitment that satisfies these properties. To aggregate the openings, our constructions perfectly inherit the ability to aggregate in the BASIS assumption, which can break the limitation of *weak* binding in the current aggregatable MVCs. In the end, we show that our constructions can be used to build the various kinds of lattice-based ZKS and ZK-EDB directly within the existing framework.

Keywords: Vector commitment · Mercurial commitment · Lattice · Zero-knowledge elementary database.

1 Introduction

Vector commitment (VC) [21,8] allows the committer to commit a vector of messages and later opens the commitment at one or multiple specific indices. In general, a VC should have these properties: *succinct*, *binding*, and *hiding*. The *succinct* property means that the sizes of the commitment and the opening are *polylogarithmic* with the dimension of the vector. The *binding* property requires that one cannot open the commitment at the same index to different values. The *hiding* property means that no one can learn the committed vector from the commitment until it is revealed. There are many variants of VC proposed, for example, *updatable* VC [8,26,28,27] supports the committer to update the message inside the commitment and provide the update information for the verifier to update the corresponding commitment and opening. The functional VC [20,3,28] allows opening the commitment to a function of the committed data. Subvector commitment (SVC) [17,15], also named *aggregatable* VC [28] supports the committers to aggregate the openings to different indices as one opening.

Furthermore, one of the most important variants of VC is the mercurial vector commitment (MVC) [21,8] which introduces the *mercurial* property. The MVC allows the committer to generate a *hard* commitment of the input vector messages or a *soft* commitment of nothing. The *hard* commitment can be both *hard* and *soft* opened only to the unique value at each index, while the *soft* commitment can only be *soft* opened to *any* value. Furthermore, *mercurial hiding* requires that others cannot distinguish between the *soft* commitment and *hard* commitment with their associated openings. There are also many variants of MVC, such as the *updatable* MVC[8] and the *aggregatable* MVC [18]. The *updatable* MVC supports updating for both *hard* and *soft* commitment. The main difference between *updatable* MVC and *updatable* VC is that the old openings (even to the soft commitment) can be updated to the new openings to the new hard commitment via the update information; The *aggregatable* MVC allows the committer to aggregate *hard* and *soft* openings. The existing *aggregatable* MVC [18] is constructed in the Algebraic Group Model (AGM) model conceptually similar to the *weak* binding [15] which requires that the adversary is unable to generate the commitment without input the message and is only suitable for applications with external protocol constraints or consensus mechanisms, e.g. blockchain. This means that the existing *aggregatable* MVC *does not suffice* to build a secure zero-knowledge elementary database (ZK-EDB) straightforwardly.

Applications of MVC: MVC leads to many cryptography applications such as (l-ary) zero-knowledge set (ZKS) and zero-knowledge elementary database (ZK-EDB) [10,21,8] in which both utilize the soft commitment to denote non-existent elements and the soft openings to prove non-membership. The *updatable* MVCs enable to build the *updatable* ZKS and ZK-EDB [22,8] and the *aggregatable* MVCs can be used to construct ZKS and ZK-EDB with *batch verification* [18]. Unfortunately, to our best known, there is still a huge gap in (l-ary) ZKS or ZK-EDB between supporting updatability and batch verification and resisting the quantum computer attack.

Overall, the existing mercurial vector commitments satisfying advanced properties, i.e. *updatable* and *aggregatable* [18,8,21] are constructed from Diffie-Hellman (DH) assumptions and RSA assumptions which cannot resist the attack of *quan-*

tum computers. Although there exists a generic construction [8] of MVC which trivially implies the lattice-based MVC with the existing lattice-based components [19,28,26], it leads to *large* auxiliary information and *cannot support* such advanced properties, due to its black-box framework.

To solve these problems, informally, we consider that the main challenge of constructing non-black-box lattice-based vector commitments satisfying "mercurial hiding", i.e., MVC, lies in two aspects: (1) how to construct lattice-based vector commitments that satisfy *hiding*; (2) how to add indistinguishable redundant items into the commitments that support generating valid and indistinguishable (with the hard openings) openings, i.e., soft openings *without* trapdoors and messages. To address this, we find that the VC based on the BASIS assumption proposed by Wee and Wu [28] supports hiding the commitment. Thus, we focus on solving the former challenge based on their constructions.

We refer to Table 1 for a summary of the current state of the art.

Scheme	**AS**	**UD**	**AG**	$\lvert\mathsf{pp}\rvert$	$\lvert C\rvert$	$\lvert\mathsf{aux}\rvert$	$\lvert\pi\rvert$
[8]	RSA	✓	✗	$O(\lambda l)$	$O(\lambda)$	$O(\lambda l)$	$O(\lambda)$
[18]	l-DHE	✗*	✓	$O(\lambda l)$	$O(\lambda)$	$O(\lambda l)$	$O(\lambda)$
[19] + [28]†	SIS	✗	✗	$l^2\mathsf{poly}(\lambda, \log l)$	$O(\lambda^2 \cdot \mathcal{H})$‡	$O(\lambda^2 l \cdot \mathcal{H})$	$O(\lambda^2 \cdot \mathcal{H})$
Cons. A.1§	SIS	✓	✗	$l^2\mathsf{poly}(\lambda, \log l)$	$O(\lambda^2 \cdot \mathcal{H})$	$O(\lambda^2 l \cdot \mathcal{H})$	$O(\lambda^2 \cdot \mathcal{H})$
Cons. 3.1	BASIS	✓	✓	$l^2\mathsf{poly}(\lambda, \log l)$	$O(\lambda^2 \cdot \mathcal{H})$	$O((\lambda^2 + \lambda l) \cdot \mathcal{H})$	$O(\lambda^2 \cdot \mathcal{H})$

* Although it allows the committer to update the hard commitment, the soft commitment cannot update to a hard commitment.

† A lattice-based MVC can be trivially built by lattice-based components (e.g. [19] and [28]) in the generic framework [8].

‡ To simplify, we denote $\mathcal{H} = \log^2 \lambda + \log^2 l$.

§ The succinct version of Construction A.1 described in Section A.7 is used to compare.

Table 1: Comparison to current works on MVC. For each scheme, we report the size of the public parameters pp, the size of commitment C, the size of the auxiliary information aux, and the size of opening π as a function of the security parameter λ and the length l of the input vector. Constants and non-dominant terms are omitted and $\mathsf{poly}(\cdot)$ represents some arbitrary polynomial. We also indicate the *assumption* (**AS**) of each scheme based on and whether the scheme can support *update* (**UD**) and *aggregate* (**AG**).

1.1 Our Contributions

In this paper, we construct a lattice-based mercurial vector commitment satisfying updatability and aggregatability based on the BASIS assumption. Although the structured version of the BASIS assumption (denoted $\mathsf{BASIS}_{\mathsf{struct}}$) is not a standard lattice-based assumption, it is a *falsifiable* assumption [25,28]. Following the existing framework, our constructions can be used to directly build the lattice-based ZKS and ZK-EDB which support updating and batch verification. We summarize the main contributions of our work in the following.

– **Succinct mercurial vector commitment:** We provide two constructions of the non-black-box lattice-based mercurial vector commitment. One is based on the standard Short Integer Solution (SIS) and satisfies updatability. The other is based on $\mathsf{BASIS}_{\mathsf{struct}}$ assumption and supports updating and aggregating which its auxiliary information has been *greatly reduced* by a level compared to the other standard SIS-based constructions. As an additional contribution, we also revisit the lattice-based mercurial commitment and transform it into transparent setup in Appendix B.
– **Updatable mercurial vector commitment:** We generalize the definition of updatable MVC [8] and *first* introduce stateless update and differentially update from the VC [26,28] to MVC. Then, we *first* extend the stronger properties for updatable MVC, named updatable mercurial hiding and updatable hiding. Last, we provide two constructions of differentially updatable MVC respectively based on SIS and $\mathsf{BASIS}_{\mathsf{struct}}$ that satisfy updatable mercurial hiding and can be extended to updatable hiding.
– **Aggregatable mercurial vector commitment:** We propose the *first* construction of aggregatable mercurial vector commitment which can break the limitation of the AGM model and *weak* binding. It is also the *first* construction from lattice. We divide the mercurial binding into the same-set binding and different-set binding. Like [28], our construction supports aggregating the openings to the *bounded* message and achieves the same set binding and different set weak binding.
– **Application for ZKS (ZK-EDB):** We show the applications of our constructions at a high level. Our construction of succinct MVC is the standard one that can be used to build the lattice-based l-ary ZKS (ZK-EDB) straightly in the generic framework [21] and even the partially succinct MVC can also be directly used to build the ZKS (ZK-EDB). Following the framework [22,8,18], our updatable MVC and aggregatable MVC can be utilized to build the updatable ZKS (ZK-EDB) with batch verification.

1.2 Technique Overview

In this section, we provide a general overview of our technique for extending the vector commitment based on the BASIS assumption to mercurial vector commitment from lattices as well as the family of BASIS assumption. In the following description, we denote $D_{\mathbb{Z}^m}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$ and $\mathbf{x} = \mathbf{A}^{-1}(\mathbf{t}) \in \mathbb{Z}_q^m$ as a random vector distributed over the discrete Gaussian conditioned on $\mathbf{A}\mathbf{x} = \mathbf{t}$ for the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the target vector $\mathbf{t} \in \mathbb{Z}_q^n$. Let $\mathbf{e}_1 = [1, 0, ..., 0]^\intercal \in \mathbb{Z}_q^n$ be the first standard basis vector. By Theorem 2.5, if there exists a short matrix $\mathbf{R}$ satisfying $\mathbf{AR} = \mathbf{G}$ where

$\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^\mathsf{T}$ is the gadget matrix and $\mathbf{g}^\mathsf{T} = [1, 2, ..., 2^{\lfloor \log q \rfloor}]$, the matrix $\mathbf{R}$ is the gadget trapdoor for $\mathbf{A}$ and can be used to efficiently sample $\mathbf{x} \leftarrow \mathbf{A}^{-1}(\mathbf{t})$ by the algorithm $\mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{t}, s)$ with some Gaussian width s.

A general framework. We begin by describing a general framework of vector commitments based on the BASIS assumption [28].

- Setup: The public parameters pp including a collection of l matrices $\mathbf{A}_1, ..., \mathbf{A}_l \in \mathbb{Z}_q^{n\times m}$ and a trapdoor $\mathbf{T} = \mathbf{B}_l^{-1}(\mathbf{G}_l)$ for $\mathbf{B}_l$ as follows.

$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_l & -\mathbf{G} \end{array}\right], \qquad \mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_l \\ \mathbf{T}_\mathbf{G} \end{bmatrix}$$

- Commit: The commitment to a vector $\mathbf{x} = (x_1, ..., x_l) \in \mathbb{Z}_q^l$ is the vector $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ where

$$[\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}]^\mathsf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_l, \mathbf{T}, -\mathbf{x} \otimes \mathbf{e}_1, s_1)$$

  which $\mathbf{e}_1 = [1, 0..., 0]^\mathsf{T}$ is the first standard basis vector and the auxiliary information is $\mathsf{aux} = (\mathbf{v}_1, ..., \mathbf{v}_l)$.
- Open: An opening to index $i \in [\ell]$ is $\mathbf{v}_i$ from $\mathsf{aux} = (\mathbf{v}_1, ..., \mathbf{v}_l)$.
- Verify: A valid opening to index $i \in [\ell]$ and message x_i need satisfy the following condition

$$\|\mathbf{v}_i\| \le \beta, \qquad \mathbf{c} = \mathbf{A}_i \mathbf{v}_i + x_i \mathbf{e}_1$$

For correctness, by the $\mathsf{SampPre}$ in Theorem 2.5, we have

$$\begin{bmatrix} -x_1\mathbf{e}_1 \\ \vdots \\ -x_l\mathbf{e}_1 \end{bmatrix} = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_l & -\mathbf{G} \end{array}\right] \cdot \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_l \\ \hat{\mathbf{c}} \end{bmatrix}$$

For binding, Denote $\underline{\mathbf{A}_i}$ as $\mathbf{A}_i$ with the first row removed. The BASIS assumption is that it is hard to find a short vector $\mathbf{z}$ where $\underline{\mathbf{A}_i}\mathbf{z} = \mathbf{0}$ for any $i \in [\ell]$ even give the related matrix $\mathbf{B}_l$ and its trapdoor $\mathbf{T} = \mathbf{B}_l^{-1}(\mathbf{G}_l)$. Therefore, if the BASIS assumption holds, for all $i \in [\ell]$, there is no adversary can generate a commitment $\mathbf{c}$ with two openings $\mathbf{v}_i$, $\mathbf{v}_i'$ to *different* message x_i, x_i' ($x_i \neq x_i'$).

For *private openings*, by the Lemma 2.4, the commitment $\mathbf{c}$ is statistically close to uniform over $\mathbb{Z}_q^n$ and for each $i \in [\ell]$, the opening $\mathbf{v}_i$ is statistically close to $\mathbf{A}_i^{-1}(\mathbf{c} - x_i\mathbf{e}_1)$.

We observe the following *features* for the above constructions:

- The property of private openings implies that there exists a *simulating* algorithm that can generate the *fake* commitment $\mathbf{c}'$ without any message and *fake* openings $\mathbf{v}'_i$ only with x_i and the trapdoor of $\mathbf{A}_i$. The *fake* commitment and openings are valid and the distribution of them is statistically close to the *real* ones.
- If we extend $\mathbf{B}_l$ to $\mathbf{B}'_l$, the trapdoor $\mathbf{T}'$ of $\mathbf{B}'_l$ can also be extended from the trapdoor $\mathbf{T}$ of $\mathbf{B}_l$ as follows,

$$\mathbf{B}'_l = \left[\begin{array}{ccc|c} [\mathbf{A}_1|\mathbf{D}_1] & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & [\mathbf{A}_l|\mathbf{D}_l] & -\mathbf{G} \end{array}\right], \qquad \mathbf{T}' = \left[\begin{array}{c} \mathbf{T}_1 \\ \hline \mathbf{0} \\ \vdots \\ \mathbf{T}_l \\ \hline \mathbf{0} \\ \mathbf{T}_{\mathbf{G}} \end{array}\right]$$

The validity of the trapdoor $\mathbf{T}'$ is guaranteed by $|\mathbf{T}'| = |\mathbf{T}|$ and $\mathbf{B}'_l\mathbf{T}' = \mathbf{G}$ (by Theorem 2.5). Therefore, if we use $[\mathbf{A}_i|\mathbf{D}_i]$, $\mathbf{B}'_l$, $\mathbf{T}'$ to *replace* $\mathbf{A}_i$, $\mathbf{B}_l$, $\mathbf{T}$ in the above construction, the properties of correctness, binding, private openings still hold under the BASIS assumption.

Our approach. We adopt the strategy of *replacing* as mentioned before to construct the main part of mercurial vector commitment and keep the condition of $\mathbf{c} = [\mathbf{A}_i|\mathbf{D}_i]\mathbf{v}_i + x_i\mathbf{e}_1$ in the verification phase.

We provide two algorithms to generate statistically indistinguishable $\mathbf{D}_i$ in the commitment $(\mathbf{c}, \mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l))$ for each $i \in [\ell]$: one is $\mathbf{D}_i = \mathbf{A}_i\mathbf{R}_i$, and the other is $\mathbf{D}_i = \mathbf{G} - \mathbf{A}_i\mathbf{R}'_i$ which $\mathbf{R}_i$ and $\mathbf{R}'_i$ are randomly sampled over $\{0,1\}^{m\times m'}$ (indistinguishability is guaranteed by Lemma 2.3). When $\mathbf{D}_i = \mathbf{G} - \mathbf{A}_i\mathbf{R}'_i$, $\mathbf{R}'_i$ is the trapdoor for $[\mathbf{A}_i|\mathbf{D}_i]$ and a valid $\mathbf{v}_i$ can be sampled from $\mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}_i], \mathbf{R}'_i, \mathbf{c}-x_i\mathbf{e}_1, s)$ which is also statistically close to $[\mathbf{A}_i|\mathbf{D}_i]^{-1}(\mathbf{c}-x_i\mathbf{e}_1)$ (by Theorem 2.5). Therefore, we need an additional check for $\mathbf{D}_i = \mathbf{A}_i\mathbf{R}_i$ to differ between soft commitments and hard commitments in the hard verification and take $\mathbf{R}_i$ as the additional part in the hard opening.

The correctness and (mercurial) binding still hold after the above operations and we extend the private openings to the mercurial hiding by the following statistically close distributions for each $i \in [\ell]$:

$$\{(\mathbf{G}\hat{\mathbf{c}}, \mathbf{v}_i) : [\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}]^{\mathsf{T}} \leftarrow \mathsf{SampPre}(\mathbf{B}'_l, \mathbf{T}', -\mathbf{x} \otimes \mathbf{e}_1, s)\}$$

$$\{(\mathbf{G}\hat{\mathbf{c}}, \mathbf{v}_i) : \hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}}, \mathbf{v}_i \leftarrow [\mathbf{A}_i|\mathbf{D}_i]^{-1}(\mathbf{G}\hat{\mathbf{c}} - x_i\mathbf{e}_1)\}$$

$$\{(\mathbf{G}\hat{\mathbf{c}}, \mathbf{v}_i) : \hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}}, \mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}_i], \mathbf{R}'_i, \mathbf{G}\hat{\mathbf{c}} - x_i\mathbf{e}_1, s)\}$$

Following the two instantiations of BASIS assumption, we provide two constructions of our lattice-based mercurial vector commitment.

- If $\mathbf{A}_1, ..., \mathbf{A}_l$ are independently sampled, the above construction is based on the $\mathsf{BASIS}_{\mathsf{rand}}$ which can be reduced to standard SIS assumption. Therefore,

$\mathbf{D}_1, ..., \mathbf{D}_l$ are independent with each other and the size of $\mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l)$ is linear with the dimension of $\mathbf{x}$. It leads that the construction of mercurial vector commitment is partially succinct. But it can be transformed into succinct by a standard vector commitment. The formal description and analysis are shown in Appendix A.

- If $\mathbf{A}_1, ..., \mathbf{A}_l$ are structured by $\mathbf{A}_i = \mathbf{W}_i\mathbf{A}$ where $\mathbf{W}_i \in \mathbb{Z}_q^{n\times n}$ is a random invertible matrix for each $i \in [\ell]$ and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is sampled randomly. This construction is based on the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. And we set $\mathbf{D}_i = \mathbf{W}_i\hat{\mathbf{D}}$ where $\hat{\mathbf{D}} = \mathbf{AR}$ or $\hat{\mathbf{D}} = \mathbf{G}-\mathbf{AR}$ and $\mathbf{R}$ is randomly sampled over $\{0,1\}^{m\times m'}$. Thus, with the public matrix $\mathbf{W}_i$ for each $i \in [\ell]$, $\mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l)$ can be represented by $\hat{\mathbf{D}}$ whose size does not depend on the dimension of $\mathbf{x}$. It leads to this construction of mercurial vector commitment being fully succinct. We provide the full details in Section 3.

Updatable MVC. We extend stateless update and differential update in vector commitment [26,28] to mercurial vector commitment. In the vector commitment based on BASIS assumption, to update the message $\mathbf{x}$ in the commitment $\mathbf{c}$ and the associated openings $\mathbf{v}_i$ to $\mathbf{x}'$, we can first construct the target vector $\mathbf{u} = -\bar{\mathbf{x}} \otimes \mathbf{e}_1$ where $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x} = (x'_1 - x_1, ..., x'_l - x_l)$ is the difference between the updated messages and old message, then compute the commitment $\bar{\mathbf{c}}$ and the openings $\bar{\mathbf{v}}_i$ of $\bar{\mathbf{x}}$. and send the update information $U_i = \{\bar{\mathbf{c}}, \bar{\mathbf{v}}_i\}$ for users holding old commitment $\mathbf{c}$ and old opening $\mathbf{v}_i$ to update. Both $\mathbf{v}_i$ and $\bar{\mathbf{v}}_i$ are valid that satisfying

$$\mathbf{c} = \mathbf{A}_i\mathbf{v}_i + x_i\mathbf{e}_1, \qquad \bar{\mathbf{c}} = \mathbf{A}_i\bar{\mathbf{v}}_i + \bar{x}_i\mathbf{e}_1$$

By the linear homomorphism of BASIS assumption, $\mathbf{c}' = \mathbf{c}+\bar{\mathbf{c}}$ is the commitment to $\mathbf{x}' = \bar{\mathbf{x}} + \mathbf{x}$ with short opening $\mathbf{v}'_i = \bar{\mathbf{v}}_i + \mathbf{v}_i$.

However, in the mercurial vector commitment, to update the soft commitment i.e. add the message to a hard commitment, we have to sample a new $\mathbf{D}'$ in the updated commitment which leads to a different target vector $\bar{\mathbf{u}} = (\bar{\mathbf{u}}_1, ..., \bar{\mathbf{u}}_l)^\mathsf{T}$ as follows:

$$\bar{\mathbf{u}}_i = -\bar{x}_i\mathbf{e}_1 + (\mathbf{D}_i - \mathbf{D}'_i)\mathbf{v}_{i,2}$$

where $\mathbf{v}_{i,2}$ is phased from the old opening $\mathbf{v}_i = [\mathbf{v}_{i,1}|\mathbf{v}_{i,2}]^\mathsf{T}$.

Thanks to the indistinguishability between $\mathbf{D}'_i$ and $\mathbf{D}_i$ for each $i \in [\ell]$, our contributions of updatable mercurial vector commitment achieve a stronger property, named updatable mercurial hiding which was proposed by Catalano et al. [8] in mercurial commitment, and we extend this property to mercurial vector commitment. Informally speaking, the property requires that even given the old commitment $(\mathbf{c}, \mathbf{D})$ with its opening $\mathbf{v}_i$, the updated commitment $(\mathbf{c}', \mathbf{D}')$ with it opening $\mathbf{v}'_i$, and the update information $U_i = \{\bar{\mathbf{c}}, \mathbf{D}', \bar{\mathbf{v}}_i\}$, the adversary still cannot learn the type of old commitment. To prove this property, we define and provide the additional *simulating* update algorithms for the fake commitment and openings. The technique of update can be applied in both SIS-based MVC and $\mathsf{BASIS_{struct}}$-based MVC. We provide the full details of them in Appendix A.1, Section 3.1 respectively, and an extension to support updatable hiding in Appendix C.

Aggregatable MVC. To beak the limitation of the existing constructions only supports mercurial *weak* binding which the adversary has to use the $\mathsf{Hard\_com}$ algorithm (input some messages, possibly adversarially chosen) to generate the commitment rather than chosen arbitrarily during the attack. For the (mercurial) vector commitment based on $\mathsf{BASIS_{struct}}$ assumption, there exists an aggregate algorithm for the *bounded* message $\mathbf{x} \in \mathbb{Z}_p^l$, in which each entity of the target vector $\mathbf{u}$ is replaced from $-\mathbf{W}_i x_i \mathbf{e}_1$ to $-\mathbf{W}_i x_i \mathbf{u}_i$ where $\mathbf{u}_i$ is randomly sampled over $\mathbb{Z}_q^n$. For any set $S \subseteq [\ell]$, we have

$$\sum_{i \in S} \mathbf{W}_i^{-1}\mathbf{c} = \mathbf{A}\sum_{i \in S}\mathbf{v}_i + \sum_{i \in S} x_i \mathbf{u}_i$$

Therefore, $\hat{\mathbf{v}} = \sum_{i \in S} \mathbf{v}_i$ is the aggregated opening to all the indices in S. The security and the correctness are guaranteed by the leftover hash lemma and min-entropy. We show a detailed construction in Section 3.2 and a full analysis in Appendix D.

1.3 Related Work

The first mercurial commitment based on the DH assumption was proposed by Chase et al. [10]. Then, Catalano et al. [7] presented trapdoor mercurial commitments (TMC) based on a one-way function with higher efficiency but weaker assumption. Later Libert et al. [19] proposed the first lattice-based mercurial commitment that supports the commitment to a single message $x \in \{0,1\}^l$. Libert and Yung [21] proposed the concept of MVC and gave two constructions on it based on l-DHE (Diffie-Hellman Exponent) assumption and RSA assumption, respectively, which support commit on a l-length vector with compact proofs for both hard opening and soft opening.

Subsequently, Catalano et al. [8] provided a generic construction for MVC with a standard MC and a standard VC. Briefly speaking, to make a mercurial vector commitment to a vector $\mathbf{x} = (x_1, .., x_l)$, it first uses the standard MC to make the mercurial commitment $(\mathbf{c}_i, \mathbf{D}_i)$ of x_i for each $i \in [\ell]$ and then uses the standard VC to make the vector commitment C of $((\mathbf{c}_1, \mathbf{D}_1), ..., (\mathbf{c}_l, \mathbf{D}_l))$ and put all the mercurial commitments into the auxiliary information. During the phase of opening and verification, the vector commitment must be opened to the mercurial commitment $(\mathbf{c}_i, \mathbf{D}_i)$ on the index i then the mercurial commitment to x_i and finally verify both openings. The drawbacks of the generic construction are that (1) the size of the auxiliary information is large; (2) it is hard to extend other advanced properties into their framework.

The concept of VC was first proposed by Catalano and Fiore in [8]. They provided two different constructions of VC based on computational DH (CDH) assumptions and RSA assumptions. They also introduced many applications of VC and MVC, such as verifiable databases, zero-knowledge elementary databases, and universal dynamic accumulators. Subsequently, Lai and Malavolta [17] first proposed the primitive of SVC and presented two constructions under variants of the root assumption and the CDH assumption. Following their work [15,21], Li et al. [18] proposed the first definitions and constructions of MSVC based on the assumption l -DHE in the AGM model and Random Oracle (ROM). They introduced a hash function to aggregate the openings to the subvector. We can find that the above non-black-box constructions of MVCs are almost based on the l -DHE assumption and the RSA assumption.

Recently, a lot of work [6,3,28,26,1,13,5,4] has been done on lattice-based VC, which is regarded as the most possible candidate for the post-quantum cryptography primitive. Therefore, with the lattice-based MC [19] and VC (e.g. [28]), the black-box lattice-based MVC can be built trivially. Among them, Wee and Wu [28] proposed a variant of the SIS assumption, named BASIS assumption to build the lattice-based VC. Compared to standard SIS-based VC, their constructions support more advanced properties, e.g., updatable, aggregatable, and functional opening. Our work is mainly based on their assumptions.

2 Preliminaries

2.1 Notation

Let $\lambda \in \mathbb{N}$ denote the security parameter. For a positive integer l, denote the set $(1, ..., l)$ by $[\ell]$. For a positive integer q, we denote $\mathbb{Z}_q$ as the integers modulo q. We use bold uppercase letters to denote matrices like $\mathbf{A}$ and bold lowercase letters to denote vectors like $\mathbf{x}$. We use non-boldface letters to refer to the components: $\mathbf{x} = (x_1, ..., x_l)$ and $\mathbf{x}[S] := (x_i, i \in S)$ to be the subvector of $\mathbf{x}$ indexed by S. $\|\mathbf{x}\|$ is denoted as the infinity norm of the vector $\mathbf{x}$. When $\mathbf{X}$ is a matrix, $\|\mathbf{X}\| := \max_{i,j} |X_{i,j}|$. For matrices $\mathbf{A}_1, ..., \mathbf{A}_l \in \mathbb{Z}_q^{n\times m}$, let $\mathsf{diag}(\mathbf{A}_1, ..., \mathbf{A}_l) \in \mathbb{Z}_q^{nl\times ml}$ be the block diagonal matrix with blocks $\mathbf{A}_1, ..., \mathbf{A}_l$ along the main diagonal (and $\mathbf{0}$ elsewhere). We denote $\mathsf{poly}(\lambda)$ as a fixed function that is $O(\lambda^c)$ for some $c \in \mathbb{N}$ and $\mathsf{negl}(\lambda)$ as a function that is $o(\lambda^{-c})$ for all $c \in \mathbb{N}$.

2.2 Lattice Preliminaries

Lattice. Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be a full-rank matrix over $\mathbb{R}$. Then the n-dimensional lattice $\mathcal{L}$ generated by $\mathbf{B}$ is $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^n\}$. If $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for integers n, m, q, we define $\mathcal{L}^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{Ax} = \mathbf{0} \mod q\}$.

Definition 2.1 (SIS Assumption [2]). Let λ be a security parameter, and n, m, q, β be lattice parameters. The short integer solution assumption $\mathsf{SIS}_{n,m,q,\beta}$ holds if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{Ax} = \mathbf{0} \wedge 0 < \|\mathbf{x}\| \leq \beta \;\middle|\; \begin{array}{c}\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m};\\ \mathbf{x} \leftarrow \mathcal{A}(1^{\lambda}, \mathbf{A})\end{array}\right] = \mathsf{negl}(\lambda)$$

Discrete Gaussian over Lattice. For integer $m \in \mathbb{N}$, let $D_{\mathbb{Z}^m,s}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$ with width parameter $s \in \mathbb{R}^+$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times l}$ and a vector $\mathbf{v} \in \mathbb{Z}_q^n$, we donate $\mathbf{A}_s^{-1}(\mathbf{v})$ as the random variable $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$ conditioned on $\mathbf{Ax} = \mathbf{v} \mod q$. We extend $\mathbf{A}_s^{-1}$ to matrices by applying $\mathbf{A}_s^{-1}$ to each column of the input.

Lemma 2.2 (Gaussian Tail Bound [14]). *A sample from a discrete Gaussian with parameter s is at most $s\sqrt{m}$ away from its center with overwhelming probability,*

$$\Pr[\|\mathbf{r}\| > s\sqrt{m} | \mathbf{r} \leftarrow D_{\mathbb{Z}^m,s}] \leq 2^{-m}$$

Lemma 2.3 (Leftover Hash Lemma [16]). *Let n, m, q be lattice parameters and suppose $m \geq 2n\log q$. Then, the statistical distance between the following distributions is at most 2^{-n}:*

$$\{(\mathbf{A}, \mathbf{Ar}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{r} \xleftarrow{\$} \{0,1\}^m\} \approx \{(\mathbf{A}, \mathbf{u}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n\}$$

When sampling a matrix $\mathbf{R} = [\mathbf{r}_1|...|\mathbf{r}_{m'}] \in \mathbb{Z}^{m\times m'}$ where $\mathbf{r}_i \xleftarrow{\$} \{0,1\}^m$ for all $i \in [m']$, we will use the notation $\mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times m'}$.

Lemma 2.4 (Discrete Gaussian Preimages [28]). *Let n, q be lattice parameters and take $m \geq 2n\log q$. Take matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times l}$ where $l = \mathsf{poly}(n\log q)$. Let $\mathbf{C} = [\mathbf{A}|\mathbf{B}]$. Then for all target vectors $\mathbf{t} \in \mathbb{Z}_q^n$ and all width parameters for $s \geq \log m$, the distribution of $\{\mathbf{v} : \mathbf{v} \leftarrow \mathbf{C}_s^{-1}(\mathbf{t})\}$ is statistically close to the distribution $\{[\mathbf{v}_1|\mathbf{v}_2]^\mathsf{T} : \mathbf{v}_2 \leftarrow D_{\mathbb{Z}^l,s}, \mathbf{v}_1 \leftarrow \mathbf{A}_s^{-1}(\mathbf{t} - \mathbf{Bv}_2)\}$.*

Trapdoor. Our constructions will use the gadget trapdoors introduced in [24] and adapted in [28]. For any positive integer k, let $\mathbf{I}_k$ denote the identity matrix of order k. Let n be a positive integer, $q \in \mathsf{poly}(n)$ be a modulus, and $m' = n(\lceil \log q \rceil + 1)$. Define the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, ..., 2^{\lceil \log q \rceil}) \in \mathbb{Z}_q^{n\times m'}$.

Theorem 2.5 (Gadget Trapdoor [28,24]). *Let n, m, q, m' be lattice parameters. Then there exist efficient algorithms (*$\mathsf{TrapGen}$, $\mathsf{SampPre}$*) with the following syntax:*

- $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m, q)$*: On input the lattice dimension n, the modulus q, and the number of samples m, the trapdoor-generation algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$.*
- $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, s)$*: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$, a target vector $\mathbf{v} \in \mathbb{Z}_q^n$, and a Gaussian width parameter s, the preimage sampling algorithm outputs a vector $\mathbf{u} \in \mathbb{Z}_q^m$ satisfying $\mathbf{Au} = \mathbf{v}$.*

Moreover, for all $m \geq O(n\log q)$, the above algorithms satisfy the following properties:

- Trapdoor distribution*: The matrix $\mathbf{A}$ output by* $\mathsf{TrapGen}(n, q, m)$ *is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$. Moreover, $\mathbf{AR} = \mathbf{G}$ and $\|\mathbf{R}\| = 1$.*

– Preimage distribution*: Suppose* $\mathbf{R}$ *is a gadget trapdoor for* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *(i.e.,* $\mathbf{AR} = \mathbf{G}$*). Then, for all* $s \geq \sqrt{mm'}\|\mathbf{R}\|\omega(\sqrt{\log n})$*), and all target vectors* $\mathbf{v} \in \mathbb{Z}_q^n$*, the distribution of* $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, s)$ *is statistically close to* $\mathbf{A}_s^{-1}(\mathbf{v})$*.*

Remark 2.6. *More generally, the above properties hold if* $\mathbf{AR} = \mathbf{HG}$ *for some invertible matrix* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$*. In this case, we refer to* $\mathbf{H}$ *as the tag.*

Remark 2.7. *In the other situation, for* $m = \bar{m}+m'$ *and some* $\bar{m} > m'$*. A trapdoor for matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *can be a matrix* $\mathbf{R} \in \mathbb{Z}^{\bar{m}\times m'}$ *such that* $\mathbf{A}[\mathbf{R}|\mathbf{I}_{m'}]^\mathsf{T} = \mathbf{G}$ *and* $\|\mathbf{R}\| = 1$*. In particular, if* $\mathbf{A} = [\bar{\mathbf{A}}|\mathbf{G} - \bar{\mathbf{A}} \cdot \mathbf{R}]$*, where* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times \bar{m}}$*, then* $\mathbf{R}$ *is a trapdoor for* $\mathbf{A}$*.*

2.3 BASIS Assumption

Definition 2.8 (BASIS Assumption [28]). Let λ be a security parameter and n, m, q, β be lattice parameters. Let s be a Gaussian width parameter. Let Samp be an efficient sampling algorithm that takes a security parameter λ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ as input and outputs a matrix $\mathbf{B} \in \mathbb{Z}_q^{n'\times m'}$ along with auxiliary information aux. We say that the basis-augmented SIS (BASIS) assumption holds with respect to Samp if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left[\mathbf{Ax} = \mathbf{0} \wedge 0 < \|\mathbf{x}\| \leq \beta \;\middle|\; \begin{array}{c} \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}; \\ (\mathbf{B}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda, \mathbf{A}), \mathbf{T} \leftarrow \mathbf{B}_s^{-1}(\mathbf{G}'_n); \\ \mathbf{x} \leftarrow \mathcal{A}(1^\lambda, \mathbf{A}, \mathbf{B}, \mathbf{T}, \mathsf{aux}) \end{array}\right] = \mathsf{negl}(\lambda)$$

In other words, it requires that SIS assumption is hard with respect to $\mathbf{A}$ even given a trapdoor $\mathbf{T}$ for the related matrix $\mathbf{B}$.

Instantiation 2.9 ($\mathsf{BASIS}_\mathsf{rand}$ Assumption [28]). Let λ be a security parameter and n, m, q, β be lattice parameters. Let s be a Gaussian width parameter and l be a dimension. The BASIS assumption with random matrices ($\mathsf{BASIS}_\mathsf{rand}$) is that: the sampling algorithm $\mathsf{Samp}(\lambda, \mathbf{A})$ samples $i^* \xleftarrow{\$} [\ell]$, $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{(n+1)\times m}$ for all $i \in [\ell]/i^*$, $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^m$, sets $\mathbf{A}_{i^*} \leftarrow \begin{bmatrix} \mathbf{a}^\mathsf{T} \\ \mathbf{A} \end{bmatrix}$, and outputs

$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & \mathbf{A}_l & -\mathbf{G}_{n+1} \end{array}\right], \qquad \mathsf{aux} = i^*$$

Instantiation 2.10 ($\mathsf{BASIS}_\mathsf{struct}$ Assumption [28]). The parameters are the same as $\mathsf{BASIS}_\mathsf{rand}$. The BASIS assumption with structured matrices ($\mathsf{BASIS}_\mathsf{struct}$) is that: the sampling algorithm $\mathsf{Samp}(\lambda, \mathbf{A})$ samples $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ for all $i \in [\ell]$ and outputs

$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A} & & & -\mathbf{G}_n \\ & \ddots & & \vdots \\ & & \mathbf{W}_l\mathbf{A} & -\mathbf{G}_n \end{array}\right], \quad \mathsf{aux} = (\mathbf{W}_1, ..., \mathbf{W}_l)$$

Remark 2.11 (Hardness and Parameter Choices of BASIS [28]). The $\mathsf{BASIS}_{\mathsf{rand}}$ assumption can be reduced to the standard SIS assumption and the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption is conceptually similar to k-R-ISIS assumption [3] in which some instances are as hard as standard SIS. While $\mathsf{BASIS}_{\mathsf{struct}}$ assumption offers more structure and potentially more power to the adversary, it is believed to provide a similar level of security as the standard SIS assumption because there are no known concrete attacks specifically targeting the structured nature of $\mathsf{BASIS}_{\mathsf{struct}}$, and no faster combinatorial attacks on $\mathsf{BASIS}_{\mathsf{struct}}$ compared to standard SIS have been discovered. However, for now, there is not an analogous reduction for the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption or k-R-ISIS assumption to standard lattice assumption.

Following [28], to further support the security claims of $\mathsf{BASIS}_{\mathsf{struct}}$, its parameter choices can be the *same* as $\mathsf{BASIS}_{\mathsf{rand}}$ which means the quality of the basis *decreases* with the dimension. It is conjectured that its security is comparable with the hardness of SIS with a noise-bound polynomially scaling with the dimension of the vector that is similar to the q-type assumptions over groups [12].

2.4 Mercurial Vector Commitment

We provide the definition of (trapdoor) mercurial vector commitment.

Definition 2.12 (Mercurial Vector Commitment [21]). A succinct (trapdoor) mercurial vector commitment over message space $\mathcal{M}$ comprises the following algorithms:

- $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$: Input a security parameter λ and the dimension of vector l, and it outputs the public parameter pp and a trapdoor key tk optionally.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$: Input the public parameter pp and a vector message $\mathbf{x} \in \mathcal{M}^l$, and it outputs a hard commitment $(\mathbf{c}, \mathbf{D})$ and auxiliary information aux.
- $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$: Input the public parameter pp, the message x_i, the index i, and the auxiliary information aux, and it outputs a hard opening π_i to prove that x_i is committed at the index i in the hard commitment.
- $0/1 \leftarrow \mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i)$: Input the public parameter pp, the message x_i, the index i, commitment $(\mathbf{c}, \mathbf{D})$, and the hard opening π_i, and it outputs 0 or 1 to indicate whether π_i is a valid hard opening.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$: Input the public parameter pp, and it outputs a soft commitment $(\mathbf{c}, \mathbf{D})$ that is not bound to any vector message, and the corresponding auxiliary information aux.
- $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$: Input the public parameter pp, the $\mathsf{flag} \in \{\mathsf{hard}, \mathsf{soft}\}$ which indicates that the soft opening τ_i is for hard commitment or soft commitment, the message x, the index i and the auxiliary information aux, it outputs the soft opening τ_i. If $\mathsf{flag} = \mathsf{hard}$ and $x \neq x_i$ at the index i, the algorithm aborts and outputs $\perp$.

- $0/1 \leftarrow \mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i)$: Input the public parameter pp, the commitment pair $(\mathbf{c}, \mathbf{D})$, the message x, the index i, and soft opening τ_i, it outputs 0 or 1 to indicate whether τ_i is a valid soft opening.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$: Input the public parameter pp and trapdoor key tk, it outputs the *fake commitment* pair $(\mathbf{c}, \mathbf{D})$ and its corresponding auxiliary information aux.
- $\pi \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it outputs the *hard equivocation* π.
- $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it outputs the *soft equivocation* τ.

Remark 2.13 (Proper MVC [19]). Including all currently known constructions, the soft opening of a hard commitment is a proper part of the hard opening to the same message. Therefore, $\mathsf{Soft\_verify}$ performs a proper subset of the tests done by $\mathsf{Hard\_verify}$. Such mercurial (vector) commitments are called *proper* mercurial (vector) commitments.

Correctness. The correctness of a trapdoor mercurial vector commitment is as follows. Specifically, for all security parameters λ, all vector message $\mathbf{x} \in \mathcal{M}^l$, and the public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$, the following conditions must hold with an overwhelming probability.

- For a hard commitment $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$, a hard opening $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$ and a soft opening $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{hard}, x_i, i, \mathsf{aux})$ for the hard commitment, there must have $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i) = 1$ and $\mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i) = 1$.
- For a soft commitment $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$, a soft opening $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{soft}, x, i, \mathsf{aux})$ for the soft commitment, there must have $\mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i) = 1$.
- For a fake commitment $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}\ (\mathsf{pp}, tk)$, where tk is the trapdoor key for the scheme, a hard equivocation $\pi \leftarrow \mathsf{Equiv\_Hopen}\ (\mathsf{pp}, tk, x_i, i, \mathsf{aux})$ and a soft equivocation $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$ for the fake commitment, there must have $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi) = 1$ and $\mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau) = 1$.

Mercurial binding. For a *proper* mercurial vector commitment, given the public parameter pp, for any adversary $\mathcal{A}$ outputs a commitment $(\mathbf{c}, \mathbf{D})$, an index $i \in [\ell]$ and the openings to some values (x, π), (x', π') (or (x, τ), (x', π')), the following probability should be $\mathsf{negl}(\lambda)$.

$$\Pr\left[\begin{array}{c} \mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i) = 1 \\ \wedge\ x_i \neq x'_i\ \wedge \\ \mathsf{Soft\_verify}(\mathsf{pp}, x'_i, i, (\mathbf{c}, \mathbf{D}), \pi'_i) = 1 \end{array} \middle| \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^l); \\ \{(\mathbf{c}, \mathbf{D}), i, (x_i, \pi_i), (x'_i, \pi'_i)\} \leftarrow \mathcal{A}(1^\lambda, 1^l, \mathsf{pp}) \end{array}\right]$$

Mercurial hiding. Given the public parameter pp, for any $\mathbf{x}$, and an index i, no efficient adversary can distinguish between hard commitment with its soft opening $\{\mathbf{x}, \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x}), \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{Hard}, x, i, \mathsf{aux})\}$ and soft commitment with its soft opening $\{\mathbf{x}, \mathsf{Soft\_com}(\mathsf{pp}), \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{Soft}, x, i, \mathsf{aux})\}$. Generally, use an equivocation game to prove.

Equivocation game. There are three related conditions for equivocation games that have to be satisfied by mercurial commitments. Each is defined by a pair of games, one *real* and one *ideal*. Given the public parameter pp and the trapdoor tk, no adversary $\mathcal{A}$ can distinguish between them.

- $\mathsf{Hcom\_Hopen\ Equivocation}$: $\mathcal{A}$ picks a vector $\mathbf{x} = (x_1, ..., x_l)$ and an index $i \in [\ell]$. In the real game, $\mathcal{A}$ will receive $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Hard\_open}\ (\mathsf{pp}, x_i, i, \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ will obtain $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, $\pi_i \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$.
- $\mathsf{Hcom\_Sopen\ Equivocation}$: $\mathcal{A}$ picks a vector $\mathbf{x} = (x_1, ..., x_l)$ and an index $i \in [\ell]$. In the real game, $\mathcal{A}$ will receive $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\tau_i \leftarrow \mathsf{Soft\_open}\ (\mathsf{pp}, \mathsf{hard}, x_i, i, \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ will obtain $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Fake\_com}(\mathsf{pp},\ tk)$, $\tau_i \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$.
- $\mathsf{Scom\_Sopen\ Equivocation}$: In the real game, $\mathcal{A}$ will get $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$ and choose x_i for some index $i \in [\ell]$, finally receive $\tau_i \leftarrow \mathsf{Soft\_open}\ (\mathsf{pp}, \mathsf{soft}, x_i, i, \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ first obtains $(\mathbf{c}, \mathbf{D}) \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, then chooses x_i for some index $i \in [\ell]$, finally receives $\tau_i \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$.

Succinctness. A mercurial vector commitment is succinct if there exists a universal polynomial $\mathsf{poly}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, $|(\mathbf{c}, \mathbf{D})| = \mathsf{poly}(\lambda, \log l)$, and $|\pi_i| = \mathsf{poly}(\lambda, \log l)$ for all $i \in [\ell]$.

3 Succinct Mercurial Vector Commitments Based on BASIS

In this section, we show how to construct a non-black-box succinct mercurial vector commitment based on $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. Then we describe the variants of our constructions that satisfy updatability and aggregatability.

Construction 3.1 (MVC Based on $\mathsf{BASIS}_{\mathsf{struct}}$). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let l be the vector dimension. The detailed construction is shown as follows.

- $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$: Input a security parameter λ and a vector dimension l , it first obtains $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$. Then for each $i \in [\ell]$, it samples an invertible matrix $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$. Next, it completes $\mathbf{R}_i = \mathbf{R}\mathbf{G}^{-1}(\mathbf{W}_i^{-1}\mathbf{G}) \in \mathbb{Z}_q^{m\times m'}$ for each $i \in [\ell]$ and constructs $\mathbf{B}_l \in \mathbb{Z}_q^{nl\times(lm+m')}$

and $\tilde{\mathbf{R}} \in \mathbb{Z}_q^{(lm+m')\times lm'}$ as follows:

$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}_l\mathbf{A} & -\mathbf{G} \end{array}\right], \qquad \tilde{\mathbf{R}} = \left[\frac{\mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l)}{\mathbf{0}^{m'\times lm'}}\right] \tag{3.1}$$

After that, it samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_l, \tilde{\mathbf{R}}, \mathbf{G}_{nl}, s_0)$. It outputs the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ and the trapdoor key $tk = \tilde{\mathbf{R}}$ optionally.

– $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$: Input the public parameter pp and a message $\mathbf{x} \in \mathbb{Z}_q^l$, it first phases $\mathbf{T}$ as $(\mathbf{T}_1, ..., \mathbf{T}_l, \mathbf{T}_\mathbf{G})^\mathsf{T}$ where $\mathbf{T}_i \in \mathbb{Z}_q^{m\times m'l}$ for each $i \in [\ell]$ and $\mathbf{T}_\mathbf{G} \in \mathbb{Z}_q^{m'\times m'l}$, then samples $\hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m\times m'}$ and constructs $\mathbf{B}'_l \in \mathbb{Z}_q^{nl\times\left(l\left(m+m'\right)+m'\right)}$, $\mathbf{T}' \in \mathbb{Z}_q^{\left(l\left(m+m'\right)+m'\right)\times m'l}$ as follows,

$$\mathbf{B}'_l = \left[\begin{array}{ccc|c} [\mathbf{W}_1\mathbf{A}|\mathbf{W}_1\mathbf{A}\hat{\mathbf{R}}] & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & [\mathbf{W}_l\mathbf{A}|\mathbf{W}_l\mathbf{A}\hat{\mathbf{R}}] & -\mathbf{G} \end{array}\right], \quad \mathbf{T}' = \begin{bmatrix} \frac{\mathbf{T}_1}{\mathbf{0}^{m'\times m'l}} \\ \vdots \\ \frac{\mathbf{T}_l}{\mathbf{0}^{m'\times m'l}} \\ \mathbf{T}_\mathbf{G} \end{bmatrix}$$

Next, it constructs the target vector $\mathbf{u}$ and uses $\mathbf{T}'$ to sample the preimage as follows,

$$\mathbf{u} = \begin{bmatrix} -x_1\mathbf{W}_1\mathbf{e}_1 \\ \vdots \\ -x_l\mathbf{W}_l\mathbf{e}_1 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_l \\ \hat{\mathbf{c}} \end{bmatrix} \leftarrow \mathsf{SampPre}\left(\mathbf{B}'_l, \mathbf{T}', \mathbf{u}, s_1\right) \tag{3.2}$$

where $\mathbf{e}_1 = [1, 0, ..., 0]^\mathsf{T} \in \mathbb{Z}_q^n$ is the first standard basis vector. Last, it computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}} \in \mathbb{Z}_q^n$, $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \in \mathbb{Z}_q^{n\times m'}$. It outputs the hard commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{x}, \mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{R}}\}$.

– $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$: Input the public parameter pp, the message x_i, the index i, and the auxiliary information $\mathsf{aux} = \{\mathbf{x}, \mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{R}}\}$. It outputs the hard opening $\pi_i = \{\mathbf{v}_i, \hat{\mathbf{R}}\}$.

– $0/1 \leftarrow \mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i)$: Input the public parameter pp, the message x_i, the index i, the hard commitment $(\mathbf{c}, \mathbf{D})$, and the hard opening π_i, check if the following conditions hold to verify the opening.

$$\|\mathbf{v}_i\| \le \beta, \qquad \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\mathbf{v}_i + x_i\mathbf{e}_1 \tag{3.3}$$

$$\|\hat{\mathbf{R}}\| \le 1, \qquad \mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \tag{3.4}$$

If they all hold, it outputs 1; Otherwise, it outputs 0.

- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$: Input the public parameter pp, it first samples $\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}$ and $\hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m \times m'}$, then computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = \mathbf{G} - \mathbf{A}\hat{\mathbf{R}}$. It outputs the soft commitment $(\mathbf{c}, \mathbf{D})$ and $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}\}$.
- $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$: Input the public parameter pp, the $\mathsf{flag} \in \{\mathsf{hard}, \mathsf{soft}\}$ which indicates that the soft opening τ_i is for hard commitment or soft commitment, the message x, the index i and the auxiliary information aux.
  If $\mathsf{flag} = \mathsf{hard}$ and x equals x_i in aux, then it outputs $\mathbf{v}_i$ in aux; Otherwise, it outputs $\perp$.
  If $\mathsf{flag} = \mathsf{soft}$, it uses trapdoor $\hat{\mathbf{R}}$ with tag $\mathbf{W}_i$ to sample the preimage as follows,
  $$\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A} | \mathbf{W}_i\mathbf{G} - \mathbf{W}_i\mathbf{A}\hat{\mathbf{R}}], \hat{\mathbf{R}}, \mathbf{c} - x_i\mathbf{W}_i\mathbf{e}_1, s_1)$$
  and outputs the soft opening $\tau_i = \mathbf{v}_i$.
- $0/1 \leftarrow \mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i)$: Input the public parameter pp, the commitment pair $(\mathbf{c}, \mathbf{D})$, the message x, the index i, and soft opening τ_i, check if Eq. 3.3 holds. If it holds, it outputs 1; Otherwise, it outputs 0.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$: Input the public parameter pp and trapdoor key tk. It first samples $\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}$, $\hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m \times m'}$ and then computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$, $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$. It generates the fake commitment pair $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}\}$.
- $\pi \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it uses $\mathbf{R}_i$ from tk to sample the preimage as follows,
  $$\mathbf{v} \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A} | \mathbf{W}_i\mathbf{A}\hat{\mathbf{R}}], \mathbf{R}_i, \mathbf{c} - x_i\mathbf{W}_i\mathbf{e}_1, s_1) \tag{3.5}$$
  It generates the equivocation hard opening $\pi = \{\mathbf{v}, \hat{\mathbf{R}}\}$.
- $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it computes the Eq. 3.5 to obtain $\mathbf{v}$. It generates the equivocation soft opening $\tau = \mathbf{v}$.

Theorem 3.2 (Correctness). *For $n = \lambda$, $m = O(n \log q)$, $s_0 = O(lm^2 \log(ln))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$, and $\beta = \sqrt{l(m + m') + m'} \cdot s_1$, then the Construction 3.1 is correct.*

Proof. Suppose polynomial $l = l(\lambda)$, $m \geq m' = O(n \log q)$, for all $\mathbf{x} \in \mathbb{Z}_q^l$ and index $i \in [\ell]$. Let $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ where $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$ and $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, $\pi_i \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$, and $\tau_i \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$. Consider $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i)$ and $\mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i)$:

Following the same parameters and constructions of $\mathbf{B}_l$ and $\tilde{\mathbf{R}}$ in $\mathsf{BASIS}_{\mathsf{struct}}$, we have $\|\mathbf{T}\| \leq \sqrt{lm + m'} \cdot s_0$.

By the construction and Lemma 2.2, $\|\mathbf{T}'\| = \|\mathbf{T}\| \leq \sqrt{lm+m'} \cdot s_0$, $\|\hat{\mathbf{R}}\| = 1$ and $\|\mathbf{R}_i\| = 1$. Suppose $s_1 \geq \sqrt{(l(m+m')+m')lm'}\|\mathbf{T}'\| \cdot \omega(\sqrt{\log(nl)}) = O(l^{3/2}m^{3/2}\log(nl)\cdot s_0)$ (opening to hard commitment), $s_1 \geq \sqrt{(m+m')m'}\|\hat{\mathbf{R}}\| \cdot \omega(\sqrt{\log(n)}) = O(m\log(n))$ (opening to soft commitment), and $s_1 \geq \sqrt{(m+m')m'}$ $\|\mathbf{R}_i\| \cdot \omega(\sqrt{\log(n)}) = O(m\log(n))$ (opening to fake commitment). Then, by Theorem 2.5 and Remark 2.6, if the opening $\mathbf{v}_i$ is generated by $\mathsf{Hard\_open}$, $\mathsf{Soft\_open}$ or $\mathsf{Equiv\_Hopen}$, it should satisfy $\mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\mathbf{v}_i + x_i\mathbf{e}_1$ and $\|\mathbf{v}_i\| \leq \sqrt{l(m+m')+m'} \cdot s_1 \leq \beta$ so the verification algorithm accepts with overwhelming probability. □

Theorem 3.3 (Mercurial Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n\log q)$, and $s_0 = O(lm^2\log(nl))$. Under the $\mathsf{BASIS_{struct}}$ assumption with parameters $(n-1, m, q, 2(m+m')\beta, s_0, l)$, Construction 3.1 satisfies mercurial binding.*

Proof. Since our construction is a *proper* mercurial vector commitment in which the hard opening contains its corresponding soft opening as a proper subset. Thus, we only need to consider the hard-soft case. We now define a sequence of hybrid experiments:

– Hyb_0: This is the real mercurial binding experiment:
  - The challenger starts by sampling $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ for each $i \in [\ell]$. Then it constructs $\tilde{\mathbf{R}}$ and $\mathbf{B}_l$ following the Eq. 3.1. It samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_l, \tilde{\mathbf{R}}, \mathbf{G}_{nl}, s_0)$. Last, the challenger sends the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ to the adversary $\mathcal{A}$.
  - The adversary $\mathcal{A}$ outputs a hard commitment pair $(\mathbf{c}, \mathbf{D})$, an index $i \in [\ell]$ and openings $(x, \mathbf{v}, \hat{\mathbf{R}})$, $(x', \mathbf{v}')$.
  - The output of the experiment is 1 if $x \neq x'$ and satisfy the following conditions:

$$\begin{aligned} \|\mathbf{v}\|, \|\mathbf{v}'\| \leq \beta, \qquad \|\hat{\mathbf{R}}\| \leq 1, \qquad \mathbf{A}\hat{\mathbf{R}} = \mathbf{D} \\ \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\mathbf{v} + x\mathbf{e}_1, \qquad \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\mathbf{v}' + x'\mathbf{e}_1 \end{aligned} \tag{3.6}$$

– Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{T} \leftarrow (\mathbf{B}_l)_{s_0}^{-1}(\mathbf{G}_{nl})$ without using the trapdoor $\tilde{\mathbf{R}}$ so the public parameters pp is sampled independently of $\mathbf{R}$.

– Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of execution of experiment Hyb_i with adversary $\mathcal{A}$. We omit the proof of $\mathsf{Hyb}_0(\mathcal{A}) \approx \mathsf{Hyb}_1(\mathcal{A}) \approx \mathsf{Hyb}_2(\mathcal{A})$ because they are given in [28] and same as ours. We now analyze the last step.

Lemma 3.4. *Under the $\mathsf{BASIS_{struct}}$ assumption with parameters $(n-1, m, q, 2(m+m')\beta,\ s_0, l)$, for all efficient adversary $\mathcal{A}$,* $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.

Proof. Suppose there exists an adversary $\mathcal{A}$ where $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \epsilon$ for some non-negligible ϵ. And an algorithm $\mathcal{B}$ will use $\mathcal{A}$ to break the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption.

$\mathcal{B}$ first receives the challenge $\mathbf{A} \in \mathbb{Z}_q^{(n-1)\times m}$, $\mathbf{B}_l \in \mathbb{Z}_q^{nl\times(lm+m')}$, $\mathbf{T} \in \mathbb{Z}_q^{(lm+m')\times lm'}$ and $\mathsf{aux} = (\mathbf{W}_1, ..., \mathbf{W}_l)$, then generate the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ and send it to $\mathcal{A}$. The adversary $\mathcal{A}$ can output a hard commitment $(\mathbf{c}, \mathbf{D})$, a hard opening $(x, \mathbf{v}, \hat{\mathbf{R}})$ and its corresponding soft opening $(x', \mathbf{v}')$ for $x \neq x'$ on some index $i \in [\ell]$, satisfying the Eq. 3.6. Thus, $\|\mathbf{v} - \mathbf{v}'\| \leq 2\beta$ and $[\mathbf{A}|\mathbf{D}](\mathbf{v} - \mathbf{v}') = (x' - x)\mathbf{e}_1$. Since $x \neq x'$, so that $\mathbf{v} - \mathbf{v}' \neq \mathbf{0}$ and we have

$$\begin{bmatrix} \mathbf{a}^{\mathsf{T}} \\ \mathbf{A} \end{bmatrix} [\mathbf{I}_m|\hat{\mathbf{R}}](\mathbf{v} - \mathbf{v}') = \begin{bmatrix} x' - x \\ \mathbf{0}^{n-1} \end{bmatrix}$$

Let $\mathbf{z} = [\mathbf{I}_m|\hat{\mathbf{R}}](\mathbf{v} - \mathbf{v}')$, since $\mathbf{Az} = \mathbf{0}$ and $\|\mathbf{z}\| \leq 2(m+m')\beta$, $\mathbf{z}$ is a valid solution for $\mathcal{B}$ to break the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with non-negligible probability. □

By the lemmas in [28] and Lemma 3.4, we can conclude that for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda)$. Thus, mercurial binding holds. □

Theorem 3.5 (Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(lm^2 \log(ln))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$, then Construction 3.1 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof. The Challenger first sets up the scheme and obtains the public parameter $\mathrm{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ via the real protocol, and $tk = \tilde{\mathbf{R}}$ is the trapdoor. Then we prove the mercurial hiding of our proposed construction from the following aspects.

For Hcom_Hopen Equivocation. Firstly, $\mathbf{D}$ and $\mathbf{R}$ are generated in the same way in fake and hard commitments. Then, by Theorem 2.5, the distribution of of $\{\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}\}$ from $\mathsf{SampPre}(\mathbf{B}'_l, \mathbf{T}', \mathbf{u}, s_1)$ is statistically close to the distribution $(\mathbf{B}'_l)^{-1}_{s_1}(\mathbf{u})$ which the target vector $\mathbf{u}$ is the same as Eq. 3.2.

Let $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{W}_1\mathbf{A}|\mathbf{W}_1\mathbf{D}], ..., [\mathbf{W}_l\mathbf{A}|\mathbf{W}_l\mathbf{D}])$, then $\mathbf{B}'_l = [\bar{\mathbf{A}}| - 1^l \otimes \mathbf{G}]$.Since $s_1 \geq \log(l(m+m'))$, by Lemma 2.4, the distribution of $\{\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}\} \leftarrow (\mathbf{B}'_l)^{-1}_{s_1}(\mathbf{u})$ is statistically close to the distribution

$$\left\{\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}, \{\mathbf{v}_1, ..., \mathbf{v}_l\} \leftarrow \bar{\mathbf{A}}^{-1}_{s_1}\left(\mathbf{u} + (1^l \otimes \mathbf{G}\hat{\mathbf{c}})\right)\right\}$$

where $\hat{\mathbf{c}}$ is generated in the same way as fake commitment and each $\mathbf{v}_i$ is distributed to $([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}])^{-1}_{s_1}(-x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$.

Then extend the trapdoor $\mathbf{R}_i$ to $\mathbf{R}'_i$ by filling in some $\mathbf{0}$. By Theorem 2.5, the distribution of $\mathbf{v}_i \leftarrow ([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}])^{-1}_{s_1}(-x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$ is statistically close to the distribution of $\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}], \mathbf{R}'_i, -x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ in the hard equivocation (since $s_1 \geq \sqrt{(m+m')m'}\|\mathbf{R}'_i\| \cdot \omega(\sqrt{n}) = O(m \log n)$). This leads to fake commitments and hard equivocation having exactly the same distribution as hard commitments and their corresponding hard openings.

For Hcom_Sopen Equivocation. Follow the same arguments as Hcom_Hopen Equivocation.

For Scom_Sopen Equivocation. We note that $\hat{\mathbf{c}}$ are generated in the same way for both fake and soft commitments. By lemma 2.3, the distributions of $\mathbf{D}$ in fake commitment and $\mathbf{D}'$ in soft commitments are

$$\left\{\mathbf{D} = \mathbf{A}\hat{\mathbf{R}} | \hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m \times m'}\right\}, \qquad \left\{\mathbf{D}' = \mathbf{G} - \mathbf{A}\hat{\mathbf{R}}' | \hat{\mathbf{R}}' \xleftarrow{\$} \{0,1\}^{m \times m'}\right\}$$

both statistically close to uniform over $\mathbb{Z}_q^{n \times m'}$. Thus, the adversary's view remains statistically the same if we generate $\mathbf{D}$ in fake commitments from $\mathsf{Soft\_com}$ instead of $\mathsf{Fake\_com}$ in the ideal experiment. Moreover, by Theorem 2.5, the distribution of the soft opening $\mathbf{v}_i \leftarrow \mathsf{SampPre}$ $([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}'], \hat{\mathbf{R}}', -x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ and the distribution of the soft equivocation $\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}'], \mathbf{R}_i, -x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ are both statistically close to $([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{D}'])_{s_1}^{-1}(-x_i\mathbf{W}_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$. This leads to fake commitments and soft equivocation having exactly the same distribution as soft commitments and their corresponding soft openings. □

Remark 3.6 (Succinctness). In Construction 3.1, for $n = \lambda$, $m = O(n \log q)$, $m' = n(\lceil \log q \rceil + 1) \leq m$, Gaussian parameters $s_0 = O(lm^2 \log(nl))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0) = O(l^{5/2}m^{7/2} \log^2(nl))$, bound $\beta = \sqrt{l(m+m') + m'} \cdot s_1 = O(l^3 n^4 \log^2(nl) \log^4 q)$, lattice modulus $q = \beta \cdot \mathsf{poly}(n)$ and $\log q = O(\log \lambda + \log l)$. We have the following parameter sizes:

- Commitment size: A commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^l$ is $(\mathbf{c}, \mathbf{D}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m'}$ where
$$|\mathbf{c}| = O(n \log q) = O(\lambda \cdot (\log \lambda + \log l))$$
$$|\mathbf{D}| = O(nm' \log q) = O(\lambda^2 \cdot (\log^2 \lambda + \log^2 l))$$
- Opening size: A (hard) opening is $(\mathbf{v}, \hat{\mathbf{R}}) \in \mathbb{Z}_q^{m+m'} \times \mathbb{Z}_q^{m \times m'}$ where
$$|\mathbf{v}| = O((m+m') \log \beta) = O(\lambda \cdot (\log^2 \lambda + \log^2 l))$$
$$|\hat{\mathbf{R}}| = O(mm') = O(\lambda^2 \cdot (\log^2 \lambda + \log^2 l))$$
- Public parameters size: The public parameters are $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{W}_i \in \mathbb{Z}_q^{n \times n}$, $\mathbf{T} \in \mathbb{Z}_q^{(lm+m') \times lm'}$ and $|\mathsf{pp}| = l^2 \cdot \mathsf{poly}(\lambda, \log l)$.
- Auxiliary information size: An auxiliary information for (hard) commitment is $\mathsf{aux} = \{\mathbf{x}, \mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{R}}\}$ and $|\mathsf{aux}| = O((\lambda^2 + \lambda l)(\log^2 \lambda + \log^2 l))$.

Therefore, Construction 3.1 is a succinct mercurial vector commitment.

3.1 Updatable Mercurial Vector Commitments

In this section, we describe a variant of Construction 3.1 that supports differential update and satisfies updatable mercurial hiding. The concepts of stateless update and differential update are proposed in the vector commitment [26,28] and we *first* extend them to the mercurial vector commitment.

The definition of updatable mercurial vector commitment was proposed by Catalano et al. [8] and we extend their definition to update both hard and soft commitment to all (multiple) indices. Specifically, the original definition of updatable mercurial commitment [22] requires updating both types of commitment to the hard (updated) commitment. But Catalano's definition and constructions only support updating the commitment on a single index which may break the integrity and consistency of the soft commitment, e.g. it should update the soft commitment to the whole vector for one time instead of one index by one. If some index of the soft commitment fails to update, this commitment cannot be interpreted as either a hard commitment or a soft commitment.

As an additional contribution, there exists a stronger property of updatable mercurial commitment first proposed by Catalano et al. [8], named updatable mercurial hiding and updatable hiding. We *first* formalize them in the mercurial vector commitment and show how our construction achieves updatable mercurial hiding and its extension to achieve updatable hiding.

Definition 3.7 (Updatable Mercurial Vector Commitment). An updatable mercurial vector commitment is defined as a mercurial vector commitment in Definition 2.12 with the following algorithms:

- $\{(\mathbf{c}', \mathbf{D}'), \mathsf{aux}', \mathsf{st}\} \leftarrow \mathsf{Update\_com}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \mathbf{x}, \mathbf{x}')$: This algorithm is run by the committer who produced $(\mathbf{c}, \mathbf{D})$ (and holds aux and flag). It takes old message $\mathbf{x}$, new message $\mathbf{x}'$ as input and outputs an updated commitment $(\mathbf{c}', \mathbf{D}')$, an updated auxiliary information aux' and an statement st. Regardless of the type of $(\mathbf{c}, \mathbf{D})$, the updated commitment $(\mathbf{c}', \mathbf{D}')$ is always a *hard commitment*.
- $U_i \leftarrow \mathsf{Update\_open}(\mathsf{pp}, \mathsf{st}, i)$: This algorithm is run by the committer who holds a statement st. Given the index i, it outputs the update information for the user who holds the opening of index i.
- $\{(\mathbf{c}', \mathbf{D}'), \pi_i'\} \leftarrow \mathsf{User\_update}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), i, \pi_i, U_i)$: This algorithm is run by the users who hold the old commitment $(\mathbf{c}, \mathbf{D})$ and the old opening π_i at index i. Given the update information U_i, it outputs the updated commitment $(\mathbf{c}', \mathbf{D}')$ and the updated opening π_i which will be valid w.r.t $(\mathbf{c}', \mathbf{D}')$ and x_i'. The updated opening π_i' will be of the *same type* of π_i.

The correctness of the updatable mercurial vector commitment is described above. The mercurial binding is defined as usual, namely for any efficient adversary it is computationally infeasible to open a commitment (even an updated one) to two different messages at the same index. The mercurial hiding of the updatable mercurial vector commitment needs not only to satisfy the old commitment but also the updated one, namely even the adversary can see the update information [3][4].

To achieve *global update*, i.e. each user can directly update their holding commitments and openings with the update information, the committer can broadcast all update information $\{U_i\}_{i\in[\ell]}$.

Remark 3.8 (Stateless Updatable MVC). If $\mathsf{Update\_com}$ can be implemented via $\mathsf{Update\_com}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \{x_i, x'_i\}_{i\in[d]})$, the MVC is stateless updatable. Assuming that aux does not consist of vector $\mathbf{x}$, with the same outputs of the original algorithm and the only difference is the inputs only involve the old and new i-th entries x_i, x'_i of the vector $\mathbf{x}$ instead of all entries of $\mathbf{x}$.

Remark 3.9 (Differentially Updatable MVC). If $\mathsf{Update\_com}$ can be implemented via $\mathsf{Update\_com}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \bar{\mathbf{x}})$, the MVC is differentially updatable. Assuming that aux does not consist of vector $\mathbf{x}$, with the same outputs of the original algorithm and the only difference is the inputs only involve the difference between old and new vector $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x}$ instead of all entries of $\mathbf{x}$.

There also exist *more powerful* security properties for the updatable mercurial commitment, named updatable mercurial hiding and updatable hiding introduced by Catalano et al. [8]. Informally, their aims are to guarantee that the message of the old commitment is still hidden even with the update information, i.e. Updatable mercurial hiding requires after the update, the type of *old* commitment is hidden; Updatable hiding says that the adversary cannot extract any information from both the *old* commitment and the updated commitment even given the update information. Although these properties can not make the updatable ZK-EDB more secure [5] , Catalano et al. still think they are an important property for the updatable mercurial commitment. We start by showing the definition of updatable mercurial hiding:

Definition 3.10 (Updatable Mercurial Hiding). Given the public parameter pp, for any $\mathbf{x}$ and $\mathbf{x}'$, and an index i, no PPT adversary can distinguish between hard commitment with its soft commitment and soft commitment with its soft commitment even after the commitment is updated and given the updated commitment and update information. We first define the additional equivocation algorithms for updating:

- $\{(\mathbf{c}', \mathbf{D}'), \mathsf{st}\} \leftarrow \mathsf{Equiv\_Ucom}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}))$: This algorithm is run by the challenger who holds trapdoor key tk and produces $(\mathbf{c}, \mathbf{D})$ and aux. It outputs a *fake updated commitment* $(\mathbf{c}', \mathbf{D}')$, and a statement st.

[3] We observe that the user can learn the type of the updated commitment which may relax the zero-knowledge property in ZK-EDB. This issue has been fully discussed in [22,8] and this paper will not follow it.

[4] Note that since an updated commitment is always a hard commitment, we are interested only in Hcom_Hopen Equivocation and Hcom_Sopen Equivocation for the updated commitment.

[5] For the structure of building the updatable ZK-EDB [22], the committed messages are the commitments itself

- $\{U_i, \mathsf{aux}'\} \leftarrow \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}), i, x'_i, \mathsf{aux}, \mathsf{st})$: This algorithm is run by the challenger who holds trapdoor key tk. It takes the old commitment $(\mathbf{c}, \mathbf{D})$, the index i, the updated message x'_i, the auxiliary information aux, and the statement st as input and outputs the *fake update information* U_i and the updated auxiliary information aux'.

Then, we slightly modify the *equivocation games* for updatable mercurial vector commitment and omit Hcom_Sopen to simply.

- Hcom_Hopen Equivocation: $\mathcal{A}$ picks a vector $\mathbf{x} = (x_1, ..., x_l)$ and an index $i \in [\ell]$. In the real game, $\mathcal{A}$ will receive the hard commitment $\mathbf{c}, \mathbf{D} = \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and the hard opening $\pi_i = \mathsf{Hard\_open}\ (\mathsf{pp}, x_i, i, \mathsf{aux})$, then $\mathcal{A}$ picks a vector $\mathbf{x}'$ to update. And $\mathcal{A}$ will receive the updated commitment $(\mathbf{c}', \mathbf{D}') = \mathsf{Update\_com}(\mathsf{pp}, \mathsf{hard}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \mathbf{x}, \mathbf{x}')$, update information $U_i = \mathsf{Update\_open}(\mathsf{pp}, \mathsf{st}, i)$ and obtain the updated opening $\pi'_i = \mathsf{User\_update}$ $(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), i, \pi_i, U_i)$. While in the ideal game, $\mathcal{A}$ will obtain the fake commitment $(\mathbf{c}, \mathbf{D}) = \mathsf{Fake\_com}(\mathsf{pp}, tk)$ and the hard equivocation $\pi_i = \mathsf{Equiv\_Hopen}$ $(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$, then $\mathcal{A}$ picks a vector $\mathbf{x}'$ to update, then $\mathcal{A}$ will receive the fake updated commitment $(\mathbf{c}', \mathbf{D}') = \mathsf{Equiv\_Ucom}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), tk)$ and fake update information $U_i = \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}), i, x'_i, \mathsf{aux}, \mathsf{st})$ and obtain the updated opening $\pi'_i = \mathsf{User\_update}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), i, \pi_i, U_i)$.
- Scom_Sopen Equivocation: In the real game, $\mathcal{A}$ will get the soft commitment $(\mathbf{c}, \mathbf{D}) = \mathsf{Soft\_com}(\mathsf{pp})$ and choose x_i for some index $i \in [\ell]$, then receive the soft opening $\pi_i = \mathsf{Soft\_open}\ (\mathsf{pp}, \mathsf{soft}, x_i, i, \mathsf{aux})$. After that $\mathcal{A}$ picks a vector $\mathbf{x}'$ to update, then $\mathcal{A}$ will receive the updated commitment $(\mathbf{c}', \mathbf{D}') = \mathsf{Update\_com}(\mathsf{pp}, \mathsf{hard}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \mathbf{x}, \mathbf{x}')$, update information $U_i = \mathsf{Update\_open}(\mathsf{pp}, \mathsf{st}, i)$ and obtain the updated opening $\pi'_i = \mathsf{User\_update}$ $(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), i, \pi_i, U_i)$. While in the ideal game, $\mathcal{A}$ first obtains $(\mathbf{c}, \mathbf{D}) = \mathsf{Fake\_com}$ (pp, tk), and chooses x_i for some index $i \in [\ell]$, then receives $\pi_i = \mathsf{Equiv\_Sopen}$ $(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$. After that, $\mathcal{A}$ picks a vector $\mathbf{x}'$ to update, then $\mathcal{A}$ will receive the fake updated commitment $(\mathbf{c}', \mathbf{D}') = \mathsf{Equiv\_Ucom}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), tk)$ and fake update information $U_i = \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}), i, x'_i, \mathsf{aux}, \mathsf{st})$ and obtain the updated opening $\pi'_i = \mathsf{User\_update}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), i, \pi_i, U_i)$.

We show how to construct a differentially updatable mercurial vector commitment from Construction 3.1 which satisfies updatable mercurial hiding.

Construction 3.11 (Differentially Updatable MVC Based on $\mathsf{BASIS_{struct}}$). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let l be the vector dimension. Let $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x}$ which $\mathbf{x}'$ is the update vector and $\mathbf{x}$ is the old vector. We only present Update_com, Update_open algorithms below, and the other algorithms are the same in Construction 3.1.

- $\{(\mathbf{c}', \mathbf{D}'), \mathsf{aux}', \mathsf{st}\} \leftarrow \mathsf{Update\_com}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \bar{\mathbf{x}})$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$, if $\mathsf{flag} = \mathsf{hard}$ that implies $(\mathbf{c}, \mathbf{D})$ is

a hard commitment which $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$, the auxiliary information $\mathsf{aux} = (\{\mathbf{v}_i\}_{i\in[\ell]}, \hat{\mathbf{R}})$, $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x} = (\bar{x}_1, ..., \bar{x}_l) \in \mathbb{Z}_q^l$;
If $\mathsf{flag} = \mathsf{soft}$ and $(\mathbf{c}, \mathbf{D})$ is a soft commitment which $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = \mathbf{G} - \mathbf{A}\hat{\mathbf{R}}$. And the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}, \{x_i, \mathbf{v}_i\}_{i\in S}\}$ means that the soft commitment $(\mathbf{c}, \mathbf{D})$ has been opened to some message x_i at some indices $i \in S$ ($|S|$ can be 0 which means the commitment have not been opened). Let $\bar{x}_i = x'_i - x_i$ for $i \in S$ and $\bar{x}_i = x'_i - x_i$ where $x_i \xleftarrow{\$} \mathbb{Z}_q$ for $i \in [\ell]/S$. Then, it samples other $\mathbf{v}_i$ for $i \in [\ell]/S$ via $\mathsf{SampPre}([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{G} - \mathbf{W}_i\mathbf{A}\hat{\mathbf{R}}], \hat{\mathbf{R}}, \mathbf{c} - x_i\mathbf{W}_i\mathbf{e}_1, s_1)$.
For both situation, it samples $\hat{\mathbf{R}}' \xleftarrow{\$} \{0,1\}^{m\times m'}$, phases $\mathbf{v}_i = [\mathbf{v}_{i,1} \in \mathbb{Z}_q^m | \mathbf{v}_{i,2} \in \mathbb{Z}_q^{m'}]^\mathsf{T}$ for $i \in [\ell]$ and constructs the target vector $\bar{\mathbf{u}} \in \mathbb{Z}_q^{nl}$, $\bar{\mathbf{B}}'_l \in \mathbb{Z}_q^{nl\times\left(l\left(m+m'\right)+m'\right)}$, $\mathbf{T}' \in \mathbb{Z}_q^{\left(l\left(m+m'\right)+m'\right)\times m'l}$ as follows,

$$\bar{\mathbf{u}} = \begin{bmatrix} -\bar{x}_1\mathbf{W}_1\mathbf{e}_1 + \mathbf{W}_1\mathbf{D}\cdot\mathbf{v}_{1,2} - \mathbf{W}_1\mathbf{A}\hat{\mathbf{R}}'\cdot\mathbf{v}_{1,2} \\ \vdots \\ -\bar{x}_l\mathbf{W}_l\mathbf{e}_l + \mathbf{W}_l\mathbf{D}\cdot\mathbf{v}_{l,2} - \mathbf{W}_l\mathbf{A}\hat{\mathbf{R}}'\cdot\mathbf{v}_{l,2} \end{bmatrix} \tag{3.7}$$

$$\bar{\mathbf{B}}'_l = \left[\begin{array}{ccc|c} [\mathbf{W}_1\mathbf{A}_1|\mathbf{W}_1\mathbf{A}_1\hat{\mathbf{R}}'] & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & [\mathbf{W}_l\mathbf{A}_l|\mathbf{W}_l\mathbf{A}_l\hat{\mathbf{R}}'] & -\mathbf{G} \end{array}\right], \quad \mathbf{T}' = \begin{bmatrix} \frac{\mathbf{T}_1}{\mathbf{0}^{m'\times m'l}} \\ \vdots \\ \frac{\mathbf{T}_l}{\mathbf{0}^{m'\times m'l}} \\ \mathbf{T}_\mathbf{G} \end{bmatrix} \tag{3.8}$$

then, uses $\mathbf{T}'$ to sample the preimage as $[\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l, \bar{\hat{\mathbf{c}}}]^\mathsf{T} \leftarrow \mathsf{SampPre}(\bar{\mathbf{B}}'_l, \mathbf{T}', \bar{\mathbf{u}}, s_1)$. Last, it computes $\bar{\mathbf{c}} = \mathbf{G}\bar{\hat{\mathbf{c}}}$, $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$, $\mathbf{D}' = \mathbf{A}\hat{\mathbf{R}}'$ and $\mathbf{v}'_i = \mathbf{v}_i + \bar{\mathbf{v}}_i$ for all $i \in [\ell]$. It outputs the updated hard commitment $(\mathbf{c}', \mathbf{D}')$, the updated auxiliary information (*updated opening*) $\mathsf{aux}' = (\{\mathbf{v}'_i\}_{i\in[\ell]}, \hat{\mathbf{R}}')$ and the statement $\mathsf{st} = \{\{\bar{\mathbf{v}}_i\}_{i\in[\ell]}, \hat{\mathbf{R}}', \bar{\mathbf{c}}, \mathbf{D}'\}$.

– $U_i \leftarrow \mathsf{Update\_open}(\mathsf{st}, i)$: Input the statement $\mathsf{st} = \{\{\bar{\mathbf{v}}_i\}_{i\in[\ell]}, \hat{\mathbf{R}}', \bar{\mathbf{c}}, \mathbf{D}'\}$ and index $i \in [\ell]$, it outputs $U_i = \{\bar{\mathbf{c}}, \hat{\mathbf{R}}', \bar{\mathbf{v}}_i, \mathbf{D}'\}$.
– $\{\pi'_i, (\mathbf{c}', \mathbf{D}')\} \leftarrow \mathsf{User\_update}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), \pi_i, i, U_i)$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$, the old commitment $(\mathbf{c}, \mathbf{D})$, the opening π_i, the index $i \in [\ell]$, and the update information $U_i = \{\bar{\mathbf{v}}_i, \hat{\mathbf{R}}', \bar{\mathbf{c}}, \mathbf{D}'\}$. It computes $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$, and $\mathbf{v}'_i = \mathbf{v}_i + \bar{\mathbf{v}}_i$. Last it outputs the updated commitment $(\mathbf{c}', \mathbf{D}')$. and the updated hard opening $\pi' = \{\mathbf{v}'_i, \hat{\mathbf{R}}'\}$ if π is a hard opening or the updated soft opening $\pi' = \mathbf{v}'_i$ if π is a soft opening.
– $\{(\mathbf{c}', \mathbf{D}'), \mathsf{st}\} \leftarrow \mathsf{Equiv\_Ucom}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}))$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ and trapdoor key tk, and the old commitment $(\mathbf{c}, \mathbf{D})$, it first samples $\bar{\hat{\mathbf{c}}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}$, $\hat{\mathbf{R}}'_i \xleftarrow{\$} \{0,1\}^{m\times m'}$, then computes $\bar{\mathbf{c}} = \mathbf{G}\bar{\hat{\mathbf{c}}}$, $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$ and $\mathbf{D}' = \mathbf{A}\hat{\mathbf{R}}'$. Finally, it outputs the *fake updated commitment* $(\mathbf{c}', \mathbf{D}')$ and the statement $\mathsf{st} = \{\bar{\mathbf{c}}, \mathbf{c}', \mathbf{D}', \hat{\mathbf{R}}'\}$.

– $\{U_i, \mathsf{aux}'\} \leftarrow \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, i, x'_i, \mathsf{aux}, \mathsf{st})$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$, the trapdoor key tk, the index i, the updated message x'_i, the old commitment $(\mathbf{c}, \mathbf{D})$, the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}, \{x_j, \mathbf{v}_j\}_{j \in S}\}$ which the fake commitment has been opened to some message x_j at some indexes $j \in S$ $(0 \leq |S| \leq l)$,and the statement $\mathsf{st} = \{\bar{\mathbf{c}}, \mathbf{c}', \mathbf{D}', \hat{\mathbf{R}}'\}$. If $i \in [\ell]/S$, it first samples $\mathbf{v}_i \leftarrow D_{\mathbb{Z}^{m+m'}, s_1}$ and then constructs the target vector as

$$\mathbf{u}_i = \mathbf{W}_i \mathbf{c}' - x'_i \mathbf{W}_i \mathbf{e}_1 - [\mathbf{W}_i \mathbf{A}_i | \mathbf{W}_i \mathbf{A}_i \hat{\mathbf{R}}'] \mathbf{v}_i$$

and then phases $\mathbf{R}_i$ from tk to sample the preimage as $\bar{\mathbf{v}}_i = \mathsf{SampPre}$ $([\mathbf{W}_i \mathbf{A}_i | \mathbf{W}_i \mathbf{A}_i \mathbf{R}'], \mathbf{R}_i, \mathbf{u}_i, s_1)$. Next, it computes $\mathbf{v}'_i = \bar{\mathbf{v}}_i + \mathbf{v}_i$. Finally, it outputs the update information $U_i = \{\bar{\mathbf{c}}, \hat{\mathbf{R}}', \bar{\mathbf{v}}_i, \mathbf{D}'\}$ and the updated auxiliary information $\mathsf{aux}' = \{\mathbf{v}'_i, \hat{\mathbf{R}}'\}$.

Theorem 3.12 (Correctness). *For $n = \lambda$, $m = O(n \log q)$, $s_0 = O(lm^2 \log(ln))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$, and $\beta = \sqrt{l(m+m') + m'} \cdot s_1$, then Construction 3.11 is correct.*

Proof. We only show the correctness of $\mathsf{Update\_com}$, $\mathsf{Update\_open}$ and $\mathsf{User\_update}$. Suppose polynomial $l = l(\lambda)$, $\mathbf{x} \in \mathbb{Z}_q^l$, $m \geq m' = O(n \log q)$, for all $\mathbf{x} \in \mathbb{Z}_q^l$ and and index $i \in [\ell]$. Let $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ where $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$ and $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}),$ $\mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, $\pi_i \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$, and $\tau_i \leftarrow \mathsf{Equiv\_Sopen}$ $(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}', \mathbf{D}'), \mathsf{aux}', \mathsf{st}\} \leftarrow \mathsf{Update\_com}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \bar{\mathbf{x}})$ and $U_i \leftarrow \mathsf{Update\_open}(\mathsf{st}, i)$. Let $\{(\mathbf{c}', \mathbf{D}'), \mathsf{st}\} \leftarrow \mathsf{Equiv\_Ucom}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}))$ and $\{U_i, \mathsf{aux}'\} \leftarrow \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, i, x'_i, \mathsf{aux}, \mathsf{st})$. Let $\{\pi'_i, (\mathbf{c}', \mathbf{D}')\} \leftarrow \mathsf{User\_update}$ $(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), \pi_i, i, U_i)$. Consider $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}', \mathbf{D}'), \pi'_i)$:

By Theorem 3.2, for old commitment $(\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}, \mathbf{D})$, for all $i \in [\ell]$, we phase $\mathbf{v}_i = [\mathbf{v}_{i,1} | \mathbf{v}_{i,2}]^\mathsf{T}$ and have

$$\mathbf{W}_i^{-1} \mathbf{G} \hat{\mathbf{c}} - x_i \mathbf{e}_1 = \mathbf{A} \mathbf{v}_{i,1} + \mathbf{D} \cdot \mathbf{v}_{i,2}, \qquad \|\mathbf{v}_i\| \leq \beta \tag{3.9}$$

Suppose $s_1 \geq \sqrt{(l(m+m')+m')lm'} \|\mathbf{T}'\| \cdot \omega(\sqrt{\log(nl)})$, by Theorem 2.5 and invertible matrix $\mathbf{W}_i$, we have

$$\mathbf{W}_i^{-1} \mathbf{G} \bar{\hat{\mathbf{c}}} - \bar{x}_i \mathbf{e}_1 + \mathbf{D} \cdot \mathbf{v}_{i,2} - \mathbf{A} \hat{\mathbf{R}}' \cdot \mathbf{v}_{i,2} = [\mathbf{A} | \mathbf{A} \hat{\mathbf{R}}'] \bar{\mathbf{v}}_i, \quad \|\bar{\mathbf{v}}_i\| \leq \beta \tag{3.10}$$

For $\mathbf{G}\hat{\mathbf{c}}' = \mathbf{G}(\bar{\hat{\mathbf{c}}} + \hat{\mathbf{c}})$, $x'_i = \bar{x}_i + x_i$, $\mathbf{v}'_i = \bar{\mathbf{v}}_i + \mathbf{v}_i$, we add Eq. 3.9 and Eq. 3.10 as

$$\mathbf{W}_i^{-1} \mathbf{G} \hat{\mathbf{c}}' - x'_i \mathbf{e}_1 = \mathbf{A} \mathbf{v}_{i,1} + \mathbf{A} \hat{\mathbf{R}}' \cdot \mathbf{v}_{i,2} + [\mathbf{A} | \mathbf{A} \hat{\mathbf{R}}'] \bar{\mathbf{v}}_i = [\mathbf{A} | \mathbf{A} \hat{\mathbf{R}}'] \mathbf{v}'_i$$

where $\|\mathbf{v}'_i\| \leq 2\beta$. Therefore the verification will accept the update hard commitment and its hard (soft) opening if we set the norm bound on the opening to $k\beta$, which can support up to k updates. Besides, similar to [28], we can set the norm bound and the modulus to be super-polynomial to support an arbitrary polynomial number of updates. ◻

Theorem 3.13 (Mercurial Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n \log q)$, q is prime and $s_0 = O(lm^2 \log(nl))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$. Under the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with parameters $(n-1, m, q, 2k(m+m')\beta, s_0, l)$, the Construction 3.11 is mercurial binding.*

Proof (Sketch). We briefly show that the updated commitment and opening satisfy mercurial binding. The proof of the mercurial binding is basically the same as Theorem 3.3. Namely, given the public parameter pp, if the adversary $\mathcal{A}$ can generate a hard (updated) commitment $(\mathbf{c}, \mathbf{D})$ and two valid (updated) openings $(\mathbf{v}_i, x_i, \hat{\mathbf{R}})$, $(\mathbf{v}'_i, x'_i)$ at same index i to different message which $x_i \neq x'_i$. Then there exist an algorithm $\mathcal{B}$ can use $\|[\mathbf{I}_m|\hat{\mathbf{R}}](\mathbf{v} - \mathbf{v}')\| \leq 2k(m + m')\beta$ as a solution to break the $\mathsf{BASIS}_{\mathsf{struct}}$. □

Theorem 3.14 (Updatable Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(lm^2 \log(ln))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$, then Construction 3.11 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof. The Challenger first sets up the scheme and obtains the public parameter $\mathsf{pp} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}\}$ via the real protocol, and $tk = \tilde{\mathbf{R}} = \mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l)$ is the trapdoor key. Then we prove the updatable mercurial hiding of the construction from the following aspects.

For Hcom_Hopen Equivocation. For any message vector $\mathbf{x}$ and $\mathbf{x}'$, we show that the distribution of fake commitments, hard equivocations, updated fake commitments, and update information is statistically close to that of hard commitments, hard openings, updated commitments, and update information.

Firstly, by Theorem 3.5, we can know that the distribution of fake commitments and hard equivocations is statistically close to the distribution of hard commitments $(\mathbf{c}, \mathbf{D})$ and hard openings $\mathbf{v}$. Then, note that $\hat{\mathbf{R}}$ and $\mathbf{D}$ are generated in the same way in both updated commitments and fake updated commitments. By Theorem 2.5, the distribution of the rest of the update information and updated hard commitment $\{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l, \bar{\hat{\mathbf{c}}}\}$ from $\mathsf{SampPre}$ $(\bar{\mathbf{B}}'_l, \mathbf{T}', \bar{\mathbf{u}}, s_1)$ in Eq. 3.7 is statistically close to the distribution $(\bar{\mathbf{B}}'_l)^{-1}_{s_1}(\bar{\mathbf{u}})$.

Let $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{W}_1\mathbf{A}_1|\mathbf{W}_1\mathbf{D}'], ..., [\mathbf{W}_l\mathbf{A}|\mathbf{W}_l\mathbf{D}'])$, then $\bar{\mathbf{B}}'_l = [\bar{\mathbf{A}}| - 1^l \otimes \mathbf{G}]$. Since $s_1 \geq \log(l(m + m'))$, by Lemma 2.4, the distribution of $\{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l, \bar{\hat{\mathbf{c}}}\} \leftarrow (\bar{\mathbf{B}}'_l)^{-1}_{s_1}(\mathbf{u})$ is statistically close to the distribution

$$\left\{\bar{\hat{\mathbf{c}}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}, \{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l\} \leftarrow \bar{\mathbf{A}}^{-1}_{s_1}\left(\bar{\mathbf{u}} + 1^l \otimes \mathbf{G}\bar{\hat{\mathbf{c}}}\right)\right\}$$

which $\bar{\hat{\mathbf{c}}}$ is the same as fake updated commitment.

Since $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{W}_1\mathbf{A}_1|\mathbf{W}_1\mathbf{D}'], ..., [\mathbf{W}_l\mathbf{A}|\mathbf{W}_l\mathbf{D}'])$, this leads to that each $\bar{\mathbf{v}}_i$ is distributed to $([\mathbf{A}_i|\mathbf{D}'_i])^{-1}_{s_1}(\bar{\mathbf{u}}_i + \mathbf{G}\bar{\hat{\mathbf{c}}})$. For $\bar{\mathbf{u}}_i$ is the same in Eq. 3.7, $\mathbf{c}' = \mathbf{G}\hat{\mathbf{c}} + \mathbf{G}\bar{\hat{\mathbf{c}}}$ and Eq. 3.9 holds in the hard commitment, we have $\mathbf{u}_i$ in fake updated commitment

$$\bar{\mathbf{u}}_i + \mathbf{G}\bar{\hat{\mathbf{c}}} = \mathbf{u}_i = \mathbf{c}' - x'_i\mathbf{e}_1 - [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\mathbf{v}_i$$

And thanks to Theorem 2.5, the distribution of $([\mathbf{A}_i|\mathbf{D}'_i])^{-1}_{s_1}(\bar{\mathbf{u}}_i + \mathbf{G}\bar{\hat{\mathbf{c}}})$ is statistically close to the distribution of $\bar{\mathbf{v}}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}'_i], \mathbf{R}_i, \mathbf{u}_i, s_1)$ in the fake updated information. This leads to fake updated commitments and fake update information having exactly the same distribution as updated commitments and update information.

For Hcom_Sopen Equivocation. Follow the same arguments as Hcom_ Hopen Equivocation.

For Scom_Sopen Equivocation. For any message vector $\mathbf{x}$ and $\mathbf{x}'$, we show that the distribution of fake commitments, soft equivocations, updated fake commitments, and update information is statistically close to that of soft commitments, soft openings, updated commitments, and update information.

The proof is nearly identical to that of the proof of Hcom_Hopen Equivocation. By Theorem 3.5, we can know that the distribution of fake commitments and soft equivocations is statistically close to the distribution of soft commitments $(\mathbf{c}, \mathbf{D})$ and soft openings $\mathbf{v}$. After that, the steps of updating for the soft commitment are the same as the hard commitment. Therefore, the distribution of fake updated commitments and fake update information is statistically close to the distribution of updated commitments and update information. □

Remark 3.15 (Succinctness). In Construction 3.11, if we choose the same parameters in Remark 3.6, after k times update, the sizes of the updated commitment $|(\mathbf{c}', \mathbf{D}')|$ and the updated opening $|\mathbf{v}_i'|$ is $\log k$ times that of the old commitment $|(\mathbf{c}, \mathbf{D})|$ and openings $|\mathbf{v}_i|$ in Remark 3.6. The size of the update information $|U_i| = |(\bar{\mathbf{c}}, \hat{\mathbf{R}}', \bar{\mathbf{v}}_i, \mathbf{D}')|$ is the same as the sum between the size of the old commitment $|(\mathbf{c}, \mathbf{D})|$ and openings $|(\mathbf{v}_i, \hat{\mathbf{R}})|$ in Remark 3.6. Therefore, the Construction 3.11 is a succinct updatable mercurial vector commitment.

Borrowing the idea of [8], we show how to use a standard vector commitment (supporting hiding) to construct an updatable mercurial vector commitment that supports updatable hiding.

Remark 3.16 (Extension to Updatable Hiding). During the update, the update information $\{U_i\}_{i\in[\ell]}$ will leak the information of $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x}$. To solve this problem and achieve updatable hiding, we sample a random vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_l) \in \mathbb{Z}_q^l$ and use $\boldsymbol{\alpha}$ to mask the message $\mathbf{x}$ and $\bar{\mathbf{x}}$. While we mercurial commit to the message $\mathbf{x}$, we also make a standard vector commitment to $\boldsymbol{\alpha}$. So that only holding the old opening π_i users can remove the mask α_i on $\bar{\mathbf{x}}$ and get the updated commitment and openings. The full definitions and details are shown in Appendix C.

3.2 Aggregatable Mercurial Vector Commitment

In this section, we provide a variant of Construction 3.1 that supports aggregating. The existing aggregatable mercurial vector commitment [18] is a pairing-based construction in the AGM model and the ROM model, which restricts the ability of the adversary to perform only the algebraic operation for the group elements, and cannot generate one, so the only way for the adversary to generate the commitment is to run the $\mathsf{Hard\_com}$ algorithm with some message. The restriction in AGM is similar to the notation of *weak* binding introduced by Gorbunov et al. [15].

Construction 3.1 perfectly inherits the property of aggregatable in $\mathsf{BASIS}_{\mathsf{struct}}$ that supports the aggregation of the openings to the *bounded* message and satisfies the same-set binding, which can *break* the limitation of AGM (*weak* binding) and ROM in the existing construction [18]. Additionally, like [28], our construction supports different-set weak binding as well.

We start by defining the notion of aggregatable mercurial vector commitment and leave the proof part in Appendix D.

Definition 3.17 (Aggregatable MVC). An aggregatable mercurial vector commitment is a standard mercurial vector commitment in Definition 2.12 with the additional algorithms as follows:

- $\hat{\Pi} \leftarrow \mathsf{Aggregate}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), S, \{x_i, \pi_i\}_{i \in S})$: Input the public parameter pp, the flag flag, the commitment $(\mathbf{c}, \mathbf{D})$, the index set S, the message x_i and the opening π_i for $i \in S$. It outputs the aggregated opening $\hat{\Pi}$.
- $0/1 \leftarrow \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i \in S}, \hat{\Pi})$: Input the public parameter pp, the flag flag, the commitment $(\mathbf{c}, \mathbf{D})$, the index set S and the message x_i for $i \in S$ and the aggregated opening $\hat{\Pi}$. It outputs $0/1$ to indicate whether $\hat{\Pi}$ is valid or not.

The correctness is that for an honestly generated aggregated opening from $\mathsf{Aggregate}$, $\mathsf{Aggre\_verify}$ should be accepted with overwhelming probability. The succinctness is that for all $\lambda \in \mathbb{N}$, the size of aggregated opening $|\hat{\Pi}| = \mathsf{poly}(\lambda, \log l)$. The mercurial hiding is that no adversary can distinguish between the aggregated hard opening and the aggregated soft opening. The definition of the same-set binding is described as follows.

Definition 3.18 (Mercurial Same-Set Binding). For a *proper* mercurial vector commitment, given the public parameter pp, for any adversary $\mathcal{A}$ outputs a commitment $(\mathbf{c}, \mathbf{D})$, a set S along with the aggregated opening $\hat{\Pi}$ and $\hat{\Pi}'$, the following probability should be $\mathsf{negl}(\lambda)$.

$$\Pr\left[\begin{array}{c|c} \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{hard}, (\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i \in S}, \hat{\Pi}) = 1 & \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda, l); \\ \wedge\ x_i \neq x_i', \text{for some } i \in S\ \wedge & \{(\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i \in S}, \\ \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{soft}, (\mathbf{c}, \mathbf{D}), S, \{x_i'\}_{i \in S}, \hat{\Pi}') = 1 & \hat{\Pi}, \hat{\Pi}'\} \leftarrow \mathcal{A}(1^\lambda, 1^l, \mathsf{pp}) \end{array}\right]$$

Construction 3.19 (Aggregatable MVC based on $\mathsf{BASIS}_{\mathsf{struct}}$). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let l be the vector dimension. Let $\mathcal{M} = \mathbb{Z}_p$ be the message space. The detailed construction is shown below.

- $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$: Input a security parameter λ and the input length l, it first runs the $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ in Construction 3.1. For each $i \in [\ell]$, it randomly samples a target vector $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^n$ and then add all $\{\mathbf{u}_i\}_{i \in [\ell]}$ to pp. It outputs $\mathsf{pp} = \{\mathbf{A}, \{\mathbf{W}_i\}_{i \in [\ell]}, \{\mathbf{u}_i\}_{i \in [\ell]}, \mathbf{T}\}$ and a trapdoor key $tk = \tilde{\mathbf{R}}$ optionally.

- $\{(\mathbf{c},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp},\mathbf{x})$: Input the public parameter pp and a vector $\mathbf{x} \in \mathbb{Z}_p^l$, it constructs $\mathbf{B}'_l$ and $\mathbf{T}'$ like $\mathsf{Hard\_com}$ in Construction 3.1. Next it constructs the target vector $\hat{\mathbf{u}}$ and uses $\mathbf{T}'$ to sample the preimage as follows,

$$\hat{\mathbf{u}} = \begin{bmatrix} -x_1\mathbf{W}_1\mathbf{u}_1 \\ \vdots \\ -x_l\mathbf{W}_l\mathbf{u}_l \end{bmatrix}, \quad \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_l \\ \hat{\mathbf{c}} \end{bmatrix} \leftarrow \mathsf{SampPre}\left(\mathbf{B}'_l, \mathbf{T}', \hat{\mathbf{u}}, s_1\right) \tag{3.11}$$

  Last, it computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}} \in \mathbb{Z}_q^n$, $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \in \mathbb{Z}_q^{n\times m'}$. It outputs the hard commitment $(\mathbf{c},\mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{v}_1,...,\mathbf{v}_l,\hat{\mathbf{R}}\}$.
- $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$: Same as the Construction 3.1, it generates the hard opening $\pi_i = \{\mathbf{v}_i,\hat{\mathbf{R}}\}$.
- $0/1 \leftarrow \mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c},\mathbf{D}), \pi_i)$: Input the public parameter pp, the message x_i, the index i, the commitment pair $(\mathbf{c},\mathbf{D})$, and the hard opening π_i, check if the following conditions hold to verify the opening.

$$\|\mathbf{v}_i\| \leq \beta, \qquad \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\mathbf{v}_i + x_i\mathbf{u}_i \tag{3.12}$$

$$\|\hat{\mathbf{R}}\| \leq 1, \qquad \mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \tag{3.13}$$

  If they all hold, it outputs 1; Otherwise, it outputs 0.
- $\{(\mathbf{c},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$: Same as the Construction 3.1, it outputs the soft commitment $(\mathbf{c},\mathbf{D})$ and $\mathsf{aux} = \{\mathbf{c},\hat{\mathbf{R}}\}$.
- $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp},\mathsf{flag},x,i,\mathsf{aux})$: Input the public parameter pp, the $\mathsf{flag} \in \{\mathsf{hard},\mathsf{soft}\}$ which indicates that the soft opening τ_i is for hard commitment or soft commitment, the message x, the index i and the auxiliary information aux.
  If $\mathsf{flag} = \mathsf{hard}$ and x equals x_i in aux, then it outputs $\mathbf{v}_i$ in aux; Otherwise, it outputs $\perp$.
  And if $\mathsf{flag} = \mathsf{soft}$, it uses $\hat{\mathbf{R}}$ with tag $\mathbf{W}_i$ to sample the preimage as follows,

$$\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{G} - \mathbf{W}_i\mathbf{A}\hat{\mathbf{R}}], \hat{\mathbf{R}}, \mathbf{c} - x_i\mathbf{W}_i\mathbf{u}_i, s_1)$$

  and outputs the soft opening $\tau_i = \mathbf{v}_i$.
- $0/1 \leftarrow \mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c},\mathbf{D}), \tau_i)$: Input the public parameter pp, the commitment pair $(\mathbf{c},\mathbf{D})$, the vector x, the index i, and soft opening τ_i, check if Eq. 3.12 holds. If it holds, it outputs 1; Otherwise, it outputs 0.
- $\hat{\Pi} \leftarrow \mathsf{Aggregate}(\mathsf{pp},\mathsf{flag},(\mathbf{c},\mathbf{D}),S,\{x_i,\pi_i\}_{i\in S})$: Input the public parameter pp, the flag flag, the commitment $(\mathbf{c},\mathbf{D})$, the index set S, and the message x_i and the opening π_i for $i \in S$. It computes

$$\hat{\mathbf{v}} = \sum_{i\in S} \mathbf{v}_i$$

  where $\mathbf{v}_i$ is phased from π_i for $i \in S$. If $\mathsf{flag} = \mathsf{hard}$, it outputs the aggregated opening $\hat{\Pi} = \{\hat{\mathbf{v}},\hat{\mathbf{R}}\}$ which $\hat{\mathbf{R}}$ is phase from π_i for $i \in S$; If $\mathsf{flag} = \mathsf{soft}$, it outputs the aggregated opening $\hat{\Pi} = \hat{\mathbf{v}}$

– $0/1 \leftarrow \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i\in S}, \hat{\Pi})$: Input the public parameter pp, the flag flag, the commitment $(\mathbf{c}, \mathbf{D})$, the index set S, and the message x_i for $i \in S$ and the aggregated opening $\hat{\Pi}$. It first checks

$$\|\hat{\mathbf{v}}\| \leq |S|\beta, \qquad \sum_{i\in S} \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\hat{\mathbf{v}} + \sum_{i\in S} x_i \mathbf{u}_i$$

If $\mathsf{flag} = \mathsf{hard}$, it also needs to check Eq. 3.4. If they hold, it outputs 1; Otherwise, it outputs 0.

– $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$: Same as the Construction 3.1, it generates the fake commitment pair $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}\}$.

– $\pi \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it uses $\mathbf{R}_i$ in tk to sample the preimage as follows,

$$\mathbf{v} \leftarrow \mathsf{SampPre}([\mathbf{W}_i\mathbf{A}|\mathbf{W}_i\mathbf{A}\hat{\mathbf{R}}], \mathbf{R}_i, \mathbf{c} - x_i\mathbf{W}_i\mathbf{u}_i, s_1) \tag{3.14}$$

It generates the equivocation hard opening $\pi = (\mathbf{v}, \hat{\mathbf{R}})$.

– $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it computes the Eq. 3.14 to obtain $\mathbf{v}$. It generates the equivocation soft opening $\tau = \mathbf{v}$.

4 Application: Lattice-Based ZK-EDB

In this section, we show the application of our constructions.

The main application of mercurial commitment is to build the ZKS and ZK-EDB. ZKS was first proposed by Micali [23] and was first built by the mercurial commitment in a structure of binary tree [10] which supports proving the membership of an element x for a set S without leaking any information (knowledge) of the set after committing the set. In ZK-EDB, the data is extended to key-value pairs (x, v) which users can query the key in the elementary database D. If the queried key x belongs to the database D, the committer will return the proof and the corresponding value v where $v = D(x)$; Otherwise, return the proof and $\perp$. Briefly speaking, to commit to a set (database), the structure of ZK-EDB or ZKS is similar to the Merkle tree with commitment instead of the hash value in each node. The proof of the membership consists of the openings of each node in the path from the leaf node of the element to the root node. Thanks to the mercurial property, the subtrees without any elements can be pruned so the size of the tree can be greatly reduced.

l-ary mercurial commitment (mercurial vector commitment) was proposed [11,21] and can be utilized to build the ZK-EDB or ZKS in a l-ary tree in order to reduce the height of the trees as well as the size of the proof. Liskov and Moses [22] proposed the updatable mercurial commitment to build an *updatable* ZK-EDB that supports the owner (committer) changing the element in the ZK-EDB and the users (verifiers) updating their holding commitments and the associated proofs.

And Catalano et al. [8] extended the updatable mercurial commitment to updatable l-ary mercurial commitment. Besides, Li et al. [18] proposed the mercurial subvector commitment (aggregatable mercurial vector commitment), which supports the aggregation of openings that can be utilized to construct the ZK-EDB with *batch verification*. This allows users to verify the aggregated proof once, instead of having to verify multiple proofs of the same commitment.

However, the above constructions are mainly based on the l-DHE assumption and RSA assumption which cannot resist the quantum computer attack. The only lattice-based mercurial commitment proposed by Libert [19] can be built ZK-EDB in a binary tree but cannot support building l-ary, updatable, or aggregatable ZK-EDB.

Following their framework in [22,21,8,18], we'll show how to build the lattice-based l-ary ZK-EDB (ZKS) and its variants, including updatable and batch verification via our proposed MVC at a high level.

In the general case, there are three phases in the ZK-EDB or ZKS: the committing phase, the opening phase, and the verification phase. In the committing phase, the committer will build an l-ary tree and return the root of the tree as the commitment of the database. As we mentioned above, building the tree, or to say the committing phase is made more efficient by pruning subtrees in which all the leaves corresponding to the keys are not in the database. Only the roots of the pruned subtrees are kept in the tree with a soft commitment. For the key x in the database D which $D(x) \neq \perp$, each corresponding leaf contains a hard commitment of the hash value of $D(x)$, and other internal nodes in the tree will contain a hard commitment of its l children (with corresponding hash value); In the opening phase, to prove some key x in the database which $D(x) = v \neq \perp$, the committer generates a proof of membership including all the hard openings for the commitments belonging to the nodes in the path from the root to the leaf x at the corresponding position opening in each commitment. To prove some key x not in the database, i.e. $D(x) = \perp$, the committer first generates the subtree which x lies and is pruned before, and then generates a proof of non-membership including all the soft openings for the commitment belonging to the nodes in the path from the root to the leaf x; In the verification phase, the users will check all the commitments and associated openings of the path from the leaf x to the root. If $D(x) = v \neq \perp$, they run the hard verification algorithm; otherwise, they run the soft verification algorithm.

To update a ZK-EDB, there are two additional phases: the updating ZK-EDB phase and the user updating phase. In the updating ZK-EDB phase, the ZK-EDB owner (committer) is allowed to change the value $D(x)$ of the elements and outputs the updated commitment with some update information for users. During this phase, the owner first needs to update the commitment in the leaf x and then update the commitments in all the nodes of the path from the leaf x to the root. The updated database commitment is the updated commitment of the root, while the update information of ZK-EDB contains the update information for all the nodes involved in the update. In the user updating phase, the users can use the update information from the owner to update their commitments and the associated proofs. In particular, if users hold a proof for the key $x' \neq x$, the updated proof for x' should also be valid.

For batch verification, if the users query multiple keys at one time, the owners (committer) can aggregate the openings for the same commitment in the node and generate the aggregated proof during the opening phase. So, the users only need to check the aggregated proof during the verification phase.

Overall, our constructions of MVC can be used to build the lattice-based ZK-EDB which enables the ZK-EDB owner to commit, open, and update, and allows the users to query, and batch verify without leaking any knowledge except the query result at a post-quantum level.

Acknowledgements. This work is supported by National Natural Science Foundation of China (No. 62202023, No. 62272131), HKU-SCF FinTech Academy, Shenzhen-Hong Kong-Macao Science and Technology Plan Project (Category C Project: SGDX20210823103537030), Theme-based Research Scheme of RGC, Hong Kong (T35-710/20-R), and Shenzhen Science and Technology Major Project (No. KJZD20230923114908017). We would like to thank the anonymous reviewers for their constructive and informative feedback on this work.

References

1. Agrawal, S., Kirshanova, E., Stehlé, D., Yadav, A.: Can round-optimal lattice-based blind signatures be practical? IACR Cryptol. ePrint Arch. **2021**, 1565 (2021)
2. Ajtai, M.: Generating hard instances of lattice problems. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 99–108 (1996)
3. Albrecht, M.R., Cini, V., Lai, R.W., Malavolta, G., Thyagarajan, S.A.: Lattice-based snarks: Publicly verifiable, preprocessing, and recursively composable. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II. pp. 102–132. Springer (2022)
4. Albrecht, M.R., Fenzi, G., Lapiha, O., Nguyen, N.K.: Slap: Succinct lattice-based polynomial commitments from standard assumptions. Cryptology ePrint Archive (2023)
5. Balbás, D., Catalano, D., Fiore, D., Lai, R.W.: Chainable functional commitments for unbounded-depth circuits. In: Theory of Cryptography Conference. pp. 363–393. Springer (2023)
6. de Castro, L., Peikert, C.: Functional commitments for all functions, with transparent setup and from sis. In: Advances in Cryptology–EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III. pp. 287–320. Springer (2023)
7. Catalano, D., Dodis, Y., Visconti, I.: Mercurial commitments: Minimal assumptions and efficient constructions. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 120–144. Springer (2006)
8. Catalano, D., Fiore, D.: Vector commitments and their applications. In: Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in

Public-Key Cryptography, Nara, Japan, February 26–March 1, 2013. Proceedings 16. pp. 55–72. Springer (2013)
9. Catalano, D., Fiore, D., Messina, M.: Zero-knowledge sets with short proofs. In: Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27. pp. 433–450. Springer (2008)
10. Chase, M., Healy, A., Lysyanskaya, A., Malkin, T., Reyzin, L.: Mercurial commitments with applications to zero-knowledge sets. In: Eurocrypt. vol. 5, pp. 422–439. Springer (2005)
11. Chase, M., Healy, A., Lysyanskaya, A., Malkin, T., Reyzin, L.: Mercurial commitments with applications to zero-knowledge sets. Journal of cryptology **26**, 251–279 (2013)
12. Cheon, J.H.: Security analysis of the strong diffie-hellman problem. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 1–11. Springer (2006)
13. Fisch, B., Liu, Z., Vesely, P.: Orbweaver: Succinct linear functional commitments from lattices. In: Annual International Cryptology Conference. pp. 106–131. Springer (2023)
14. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206 (2008)
15. Gorbunov, S., Reyzin, L., Wee, H., Zhang, Z.: Pointproofs: Aggregating proofs for multiple vector commitments. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. pp. 2007–2023 (2020)
16. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
17. Lai, R.W., Malavolta, G.: Subvector commitments with application to succinct arguments. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I 39. pp. 530–560. Springer (2019)
18. Li, Y., Susilo, W., Yang, G., Phuong, T.V.X., Yu, Y., Liu, D.: Concise mercurial subvector commitments: Definitions and constructions. In: Information Security and Privacy: 26th Australasian Conference, ACISP 2021, Virtual Event, December 1–3, 2021, Proceedings 26. pp. 353–371. Springer (2021)
19. Libert, B., Nguyen, K., Tan, B.H.M., Wang, H.: Zero-knowledge elementary databases with more expressive queries. In: IACR International Workshop on Public Key Cryptography. pp. 255–285. Springer (2019)
20. Libert, B., Ramanna, S.C., et al.: Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In: 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016) (2016)
21. Libert, B., Yung, M.: Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In: Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7. pp. 499–517. Springer (2010)
22. Liskov, M.: Updatable zero-knowledge databases. In: Advances in Cryptology-ASIACRYPT 2005: 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005. Proceedings 11. pp. 174–198. Springer (2005)

23. Micali, S., Rabin, M., Kilian, J.: Zero-knowledge sets. In: 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. pp. 80–91. IEEE (2003)
24. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 700–718. Springer (2012)
25. Naor, M.: On cryptographic assumptions and challenges. In: Annual International Cryptology Conference. pp. 96–109. Springer (2003)
26. Peikert, C., Pepin, Z., Sharp, C.: Vector and functional commitments from lattices. In: Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part III 19. pp. 480–511. Springer (2021)
27. Tas, E.N., Boneh, D.: Vector commitments with efficient updates. arXiv preprint arXiv:2307.04085 (2023)
28. Wee, H., Wu, D.J.: Succinct vector, polynomial, and functional commitments from lattices. In: Advances in Cryptology–EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III. pp. 385–416. Springer (2023)

Appendix A (Partially) Succinct Mercurial Vector Commitments Based on Standard SIS

In this section, we demonstrate how to construct a (partially) succinct mercurial vector commitment based on the SIS assumption and show how it can be transformed into a succinct one. Then we describe a variant of our SIS-based mercurial vector commitment that supports updating. We introduce the concept of stateless updates and differential updates for mercurial vector commitment and define a stronger property, named updatable mercurial hiding. We demonstrate how our construction achieves the above properties. As an additional contribution, we find a lattice-based mercurial commitment with transparent setup and provide the analysis in Appendx B.

Construction A.1 (MVC Based on SIS). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let l be the vector dimension. The detailed construction is shown as follows.

– $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$: Input a security parameter λ and the vector dimension l to be committed, $(\mathbf{A}_i, \mathbf{R}_i) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ for each $i \in [\ell]$. Then, it constructs $\mathbf{B}_l \in \mathbb{Z}_q^{nl \times (lm+m')}$ and $\tilde{\mathbf{R}} \in \mathbb{Z}_q^{(lm+m') \times lm'}$ as follows:

$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{A}_1 & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{A}_l & -\mathbf{G} \end{array}\right], \quad \tilde{\mathbf{R}} = \left[\frac{\mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l)}{\mathbf{0}^{m' \times lm'}}\right] \tag{A.1}$$

Finally, it samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_l, \tilde{\mathbf{R}}, \mathbf{G}_{nl}, s_0)$. It outputs $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ and a trapdoor key $tk = \tilde{\mathbf{R}}$ optionally.

– $\{(\mathbf{c},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp},\mathbf{x})$: Input the public parameter pp and a vector message $\mathbf{x} \in \mathbb{Z}_q^l$, it first phases $\mathbf{T}$ as $(\mathbf{T}_1,...,\mathbf{T}_l,\mathbf{T_G})^\mathsf{T}$ where $\mathbf{T}_i \in \mathbb{Z}_q^{m\times m'l}$ for each $i \in [\ell]$ and $\mathbf{T_G} \in \mathbb{Z}_q^{m'\times m'l}$, then samples $\hat{\mathbf{R}}_i \xleftarrow{\$} \{0,1\}^{m\times m'}$ for each $i \in [\ell]$, constructs $\mathbf{B}'_l \in \mathbb{Z}_q^{nl\times(l(m+m')+m')}$, $\mathbf{T}' \in \mathbb{Z}_q^{(l(m+m')+m')\times m'l}$ and uses $\mathbf{T}'$ to sample the preimages as follows.

$$\mathbf{B}'_l = \left[\begin{array}{ccc|c} [\mathbf{A}_1|\mathbf{A}_1\hat{\mathbf{R}}_1] & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & [\mathbf{A}_l|\mathbf{A}_l\hat{\mathbf{R}}_l] & -\mathbf{G} \end{array}\right], \quad \mathbf{T}' = \begin{bmatrix} \frac{\mathbf{T}_1}{\mathbf{0}^{m'\times m'l}} \\ \vdots \\ \frac{\mathbf{T}_l}{\mathbf{0}^{m'\times m'l}} \\ \mathbf{T_G} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_l \\ \hat{\mathbf{c}} \end{bmatrix} \leftarrow \mathsf{SampPre}\left(\mathbf{B}'_l, \mathbf{T}', -\mathbf{x}\otimes\mathbf{e}_1, s_1\right)$$

where $\mathbf{e}_1 = [1,0,...,0]^\mathsf{T} \in \mathbb{Z}_q^n$ is the first standard basis vector. Next, it computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}} \in \mathbb{Z}_q^n$, $\mathbf{D} = (\mathbf{D}_1,...,\mathbf{D}_l)$ where $\mathbf{D}_i = \mathbf{A}_i\hat{\mathbf{R}}_i$ for each $i \in [\ell]$. It outputs the hard commitment $(\mathbf{c},\mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{x}, \{\mathbf{v}_i\}_{i\in[\ell]}, \{\hat{\mathbf{R}}_i\}_{i\in[\ell]}\}$.

– $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$: Input the public parameter pp, the message x_i, the index i, and the auxiliary information $\mathsf{aux} = \{\{\mathbf{v}_i\}_{i\in[\ell]}, \{\hat{\mathbf{R}}_i\}_{i\in[\ell]}\}$. It outputs the hard opening $\pi_i = \{\mathbf{v}_i, \hat{\mathbf{R}}_i\}$.

– $0/1 \leftarrow \mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c},\mathbf{D}), \pi_i)$: Input the public parameter pp, the message x_i, the index i, commitment pair $(\mathbf{c},\mathbf{D})$, and the hard opening π_i, check if the following conditions hold to verify the opening.

$$\|\mathbf{v}_i\| \le \beta, \qquad \mathbf{c} = [\mathbf{A}_i|\mathbf{D}_i]\mathbf{v}_i + x_i\mathbf{e}_1 \tag{A.2}$$

$$\|\hat{\mathbf{R}}_i\| \le 1, \qquad \mathbf{D}_i = \mathbf{A}_i\hat{\mathbf{R}}_i \tag{A.3}$$

If they all hold, it outputs 1; Otherwise, it outputs 0.

– $\{(\mathbf{c},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$: Input the public parameter pp, it first samples $\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'},s_1}$ and $\hat{\mathbf{R}}_i \xleftarrow{\$} \{0,1\}^{m\times m'}$ for each $i \in [\ell]$, then computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = (\mathbf{D}_1,...,\mathbf{D}_l)$ where $\mathbf{D}_i = \mathbf{G} - \mathbf{A}_i\hat{\mathbf{R}}_i$ for each $i \in [\ell]$. It outputs the soft commitment $(\mathbf{c},\mathbf{D})$ and $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}_1,...,\hat{\mathbf{R}}_l\}$.

– $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$: Input the public parameter pp, the $\mathsf{flag} \in \{\mathsf{hard}, \mathsf{soft}\}$ which indicates that the soft opening τ_i is for hard commitment or soft commitment, the message x, the index i and the auxiliary information aux.
If $\mathsf{flag} = \mathsf{hard}$ and x equals x_i in aux, then it outputs $\mathbf{v}_i$ in aux; Otherwise, it outputs $\perp$.

And if $\mathsf{flag} = \mathsf{soft}$, it phases $\hat{\mathbf{R}}_i$ from aux to sample the preimage as follows,

$$\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{G} - \mathbf{A}_i\hat{\mathbf{R}}_i], \hat{\mathbf{R}}_i, \mathbf{c} - x_i\mathbf{e}_1, s_1)$$

It outputs the soft opening $\tau_i = \mathbf{v}_i$.

- $0/1 \leftarrow \mathsf{Soft\_verify}(\mathsf{pp}, x, i, (\mathbf{c}, \mathbf{D}), \tau_i)$: Input the public parameter pp, the commitment pair $(\mathbf{c}, \mathbf{D})$, the message x, the index i, and soft opening τ_i, check if Eq. A.2 holds. If it holds, it outputs 1; Otherwise, it outputs 0.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$: Input the public parameter pp and trapdoor key tk. It first samples $\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}$ and $\hat{\mathbf{R}}_i \xleftarrow{\$} \{0,1\}^{m \times m'}$ for each $i \in [\ell]$, then computes $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$, and $\mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l)$ where $\mathbf{D}_i = \mathbf{A}_i\hat{\mathbf{R}}_i$ for each $i \in [\ell]$. It generates the fake commitment pair $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \hat{\mathbf{R}}_1, ..., \hat{\mathbf{R}}_l\}$.
- $\pi \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it phases $\mathbf{R}_i$ from tk to sample the preimage as follows,

$$\mathbf{v} \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}_i], \mathbf{R}_i, \mathbf{c} - x_i\mathbf{e}_1, s_1) \tag{A.4}$$

It generates the equivocation hard opening $\pi = (\mathbf{v}, \hat{\mathbf{R}}_i)$.

- $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$: Input the public parameter pp and trapdoor key tk, the message x_i, the index i, and the auxiliary information aux, it computes the Eq. A.4 to obtain $\mathbf{v}$. It outputs the equivocation soft opening $\tau = \mathbf{v}$.

Theorem A.2 (Correctness). *For $n = \lambda$, $m = O(n \log q)$, $s_0 = O(lm \log(ln))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$, and $\beta = \sqrt{l(m + m') + m'} \cdot s_1$, then Construction A.1 is correct.*

Proof. Suppose polynomial $l = l(\lambda)$, $m \geq m' = O(n \log q)$, for all $\mathbf{x} \in \mathbb{Z}_q^l$ and index $i \in [\ell]$. Let $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ where $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$ and $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{soft}, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, $\pi_i \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$, and $\tau_i \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$. Consider $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i)$ and $\mathsf{Soft\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \tau_i)$:

Following the same parameters and constructions of $\mathbf{B}_l$ and $\tilde{\mathbf{R}}$ in $\mathsf{BASIS}_{\mathsf{rand}}$, we have $\|\mathbf{T}\| \leq \sqrt{lm + m'} \cdot s_0$.

By the construction and Lemma 2.2, $\|\mathbf{T}'\| = \|\mathbf{T}\| \leq \sqrt{lm + m'} \cdot s_0$ and $\|\hat{\mathbf{R}}_i\| = 1$. Suppose $s_1 \geq \sqrt{(l(m + m') + m')lm'}\|\mathbf{T}'\| \cdot \omega(\sqrt{\log(nl)}) = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$ (opening to hard commitment), $s_1 \geq \sqrt{(m + m')m'}\|\hat{\mathbf{R}}_i\| \cdot \omega(\sqrt{\log(n)}) = O(m \log(n))$ (opening to soft commitment), and $s_1 \geq \sqrt{(m + m')m'}\ \|\mathbf{R}_i\| \cdot \omega(\sqrt{\log(n)}) = O(m \log(n))$ (opening to fake commitment). Then, by Theorem 2.5, if the opening $\mathbf{v}_i$ is generated by $\mathsf{Hard\_open}$, $\mathsf{Soft\_open}$ or $\mathsf{Equiv\_Hopen}$, they must satisfy $\mathbf{c} = [\mathbf{A}_i|\mathbf{D}_i]\mathbf{v}_i + x_i\mathbf{e}_i$ and $\|\mathbf{v}_i\| \leq \sqrt{l(m + m') + m'} \cdot s_1 \leq \beta$ so the verification algorithm accepts with overwhelming probability. $\square$

Theorem A.3 (Mercurial Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n \log q)$, and $s_0 = O(lm \log(nl))$. Under the $\mathsf{BASIS_{rand}}$ assumption with parameters $(n-1, m, q, 2(m+m')\beta, s_0, l)$, the Construction A.1 is mercurial binding.*

Proof. Since our construction is a *proper* mercurial vector commitment in which the hard opening contains its corresponding soft opening as a proper subset. Thus, we only need to consider the hard-soft case. We now define a sequence of hybrid experiments.

- Hyb_0: This is the real mercurial binding experiment:
  - The challenger starts by sampling $(\mathbf{A}_i, \mathbf{R}_i) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ for each $i \in [\ell]$. Then it constructs $\tilde{\mathbf{R}}$ and $\mathbf{B}_l$ following the Eq. A.1. It samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_l, \tilde{\mathbf{R}}, \mathbf{G}_{nl}, s_0)$. Last, the challenger sends the public parameters $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ to the adversary $\mathcal{A}$.
  - The adversary $\mathcal{A}$ outputs a hard commitment $(\mathbf{c}, \mathbf{D})$, an index $i \in [\ell]$ and openings $(x, \mathbf{v}, \hat{\mathbf{R}}_i)$, $(x', \mathbf{v}', \hat{\mathbf{R}}'_i)$.
  - The output of the experiment is 1 if $x \neq x'$ and satisfy the following conditions:

$$\begin{gathered} \|\mathbf{v}\|, \|\mathbf{v}'\| \leq \beta, \quad \|\hat{\mathbf{R}}_i\|, \|\hat{\mathbf{R}}'_i\| \leq 1, \quad \mathbf{A}_i\hat{\mathbf{R}}_i = \mathbf{A}_i\hat{\mathbf{R}}'_i = \mathbf{D}_i \\ \mathbf{c} = [\mathbf{A}_i|\mathbf{D}_i]\mathbf{v} + x\mathbf{e}_1, \qquad \mathbf{c} = [\mathbf{A}_i|\mathbf{D}_i]\mathbf{v}' + x'\mathbf{e}_1 \end{gathered} \tag{A.5}$$

- Hyb_1: Same as Hyb_0 except at the beginning of the game, the challenger samples an index $i^* \xleftarrow{\$} [\ell]$. The output of the experiment is 1 if the conditions in Hyb_0 hold and $i = i^*$.
- Hyb_2: Same as Hyb_1 except the challenger samples $\mathbf{T} \leftarrow (\mathbf{B}_l)^{-1}_{s_0}(\mathbf{G}_{nl})$.
- Hyb_3: Same as Hyb_2 except the challenger samples $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ for each $i \in [\ell]$.

For an adversary $\mathcal{A}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of execution of experiment Hyb_i with adversary $\mathcal{A}$. We omit the proof of $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] = l \cdot \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]$, $\mathsf{Hyb}_1(\mathcal{A}) \approx \mathsf{Hyb}_2(\mathcal{A}) \approx \mathsf{Hyb}_3(\mathcal{A})$ because they are given in [28] and identical to ours. We only analyze the last step.

Lemma A.4. *Under the $\mathsf{BASIS_{rand}}$ assumption with parameters $(n-1, m, q, 2(m+m')\beta, s_0, l)$, for all efficient adversary $\mathcal{A}$, $\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ where $\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] = \epsilon$ for some non-negligible ϵ. An algorithm $\mathcal{B}$ will use $\mathcal{A}$ to break the $\mathsf{BASIS_{rand}}$ assumption.

$\mathcal{B}$ first obtains the challenge $\mathbf{A} \in \mathbb{Z}_q^{(n-1)\times m}$ and $\mathsf{aux} = i^* \in [\ell]$, $\mathbf{B}_l = [\mathsf{diag}(\mathbf{A}_1, ..., \mathbf{A}_l)| - 1^l \otimes \mathbf{G}]$, $\mathbf{T}$, then generates $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ and send pp to $\mathcal{A}$. The adversary $\mathcal{A}$ can output a hard commitment $(\mathbf{c}, \mathbf{D})$, a hard opening $(x, \mathbf{v}, \hat{\mathbf{R}}_i)$ and a soft opening $(x', \mathbf{v}')$ both to index i where $x \neq x'$ and $i = i^*$, satisfying the Eq. A.5.

If $\hat{\mathbf{R}} \neq \hat{\mathbf{R}}'$, we have $|\hat{\mathbf{R}} - \hat{\mathbf{R}}'| \leq 2$ and $\mathbf{A}(\hat{\mathbf{R}} - \hat{\mathbf{R}}') = \mathbf{0}$. Thus, $\hat{\mathbf{R}} - \hat{\mathbf{R}}'$ is a valid solution to break the SIS assumption. If $\hat{\mathbf{R}} = \hat{\mathbf{R}}'$, we have $\|\mathbf{v} - \mathbf{v}'\| \leq 2\beta$ and $[\mathbf{A}_{i^*}|\mathbf{D}_{i^*}](\mathbf{v} - \mathbf{v}') = (x' - x)\mathbf{e}_1$. Since $x \neq x'$, so that $\mathbf{v} - \mathbf{v}' \neq \mathbf{0}$ and we have

$$[\mathbf{A}_{i^*}|\mathbf{A}_{i^*}\hat{\mathbf{R}}_i](\mathbf{v} - \mathbf{v}') = (x' - x)\mathbf{e}_1$$

$$\begin{bmatrix} \mathbf{a}^{\mathsf{T}} \\ \mathbf{A} \end{bmatrix} [\mathbf{I}_m|\hat{\mathbf{R}}_i](\mathbf{v} - \mathbf{v}') = \begin{bmatrix} x' - x \\ \mathbf{0}^{n-1} \end{bmatrix}$$

Let $\mathbf{z} = [\mathbf{I}_m|\hat{\mathbf{R}}_i](\mathbf{v} - \mathbf{v}')$, since $\mathbf{A}\mathbf{z} = \mathbf{0}$ and $\|\mathbf{z}\| \leq 2(m+m')\beta$, $\mathbf{z}$ is a valid solution for $\mathcal{B}$ to break the $\mathsf{BASIS}_{\mathsf{rand}}$ assumption with non-negligible probability. ◻

By the lemmas in [28], we have that for all adversary $\mathcal{A}$, $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \leq l \cdot (\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] + \mathsf{negl}(\lambda))$. Since $l = \mathsf{poly}(\lambda)$, we can conclude via Lemma A.4 that for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \leq \mathsf{negl}(\lambda))$. ◻

Theorem A.5 (Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(lm \log(ln))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$, then Construction A.1 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof. The Challenger first sets up the scheme and obtains the public parameter $\mathrm{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ via the real protocol, and $tk = \tilde{\mathbf{R}} = \mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l)$ is the trapdoor key which $\mathbf{R}_i$ is the trapdoor of $\mathbf{A}_i$ for each $i \in [\ell]$. Then we prove the mercurial hiding of our construction from the following aspects.

For Hcom_Hopen Equivocation. For any message vector $\mathbf{x}$, we show that the distribution of fake commitments and their hard equivocations is statistically close to that of hard commitments and their hard openings. Firstly, note that $\mathbf{D}$ and $\mathbf{R}$ are generated in the same way in both fake and hard commitments and corresponding openings. Then, by Theorem 2.5, the distribution of the rest of hard commitment and openings $\{\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}\}$ from $\mathsf{SampPre}(\mathbf{B}'_l, \mathbf{T}', -\mathbf{x} \otimes \mathbf{e}_1, s_1)$ is statistically close to the distribution $(\mathbf{B}'_l)^{-1}_{s_1}(-\mathbf{x} \otimes \mathbf{e}_1)$.

Let $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{A}_1|\mathbf{D}_1], ..., [\mathbf{A}_l|\mathbf{D}_l])$, then $\mathbf{B}'_l = [\bar{\mathbf{A}}| - 1^l \otimes \mathbf{G}]$.Since $s_1 \geq \log(l(m + m'))$, by Lemma 2.4, the distribution of $\{\mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{c}}\} \leftarrow (\mathbf{B}'_l)^{-1}_{s_1}(-\mathbf{x} \otimes \mathbf{e}_1)$ is statistically close to the distribution

$$\left\{\hat{\mathbf{c}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}, \{\mathbf{v}_1, ..., \mathbf{v}_l\} \leftarrow \bar{\mathbf{A}}^{-1}_{s_1}\left(-(\mathbf{x} \otimes \mathbf{e}_1) + (1^l \otimes \mathbf{G}\hat{\mathbf{c}})\right)\right\}$$

where $\hat{\mathbf{c}}$ is generated in the same way as fake commitment.

Since $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{A}_1|\mathbf{D}_1], ..., [\mathbf{A}_l|\mathbf{D}_l])$, each $\mathbf{v}_i$ is distributed to $([\mathbf{A}_i|\mathbf{D}_i])^{-1}_{s_1}(-x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$. Then extend the trapdoor $\mathbf{R}_i$ of $\mathbf{A}_i$ to the trapdoor $\mathbf{R}'_i$ of $[\mathbf{A}_i|\mathbf{D}_i]$ by filling in $\mathbf{0}$. Thanks to Theorem 2.5 again, the distribution of $\mathbf{v}_i \leftarrow ([\mathbf{A}_i|\mathbf{D}_i])^{-1}_{s_1}(-x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$ is statistically close to the distribution of $\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}_i], \mathbf{R}'_i, -x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ in the hard equivocation (since $s_1 \geq \sqrt{(m + m')m'}\|\mathbf{R}'_i\| \cdot \omega(\sqrt{n}) = O(m \log n)$). This leads to fake commitments and hard equivocations having exactly the same distribution as hard commitments and hard openings.

For Hcom_Sopen Equivocation. Follow the same arguments as Hcom_Hopen Equivocation.

For Scom_Sopen Equivocation. We note that $\hat{\mathbf{c}}$ are generated in the same way in both fake and soft commitments. By Lemma 2.3, the distributions of $\mathbf{D}_i$ in fake commitment and $\mathbf{D}'_i$ in soft commitments are

$$\left\{\mathbf{D}_i = \mathbf{A}_i\hat{\mathbf{R}}_i | \hat{\mathbf{R}}_i \xleftarrow{\$} \{0,1\}^{m\times m'}\right\} \approx \left\{\mathbf{D}'_i = \mathbf{G} - \mathbf{A}_i\hat{\mathbf{R}}'_i | \hat{\mathbf{R}}'_i \xleftarrow{\$} \{0,1\}^{m\times m'}\right\}$$

both statistically close to uniform over $\mathbb{Z}_q^{n\times m'}$. Thus, the adversary's view remains statistically the same if we generate $\mathbf{D}$ in fake commitments from $\mathsf{Soft\_com}$ instead of $\mathsf{Fake\_com}$ in the ideal experiment. Moreover, by Theorem 2.5, the distribution of the soft opening $\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}'_i], \hat{\mathbf{R}}'_i, -x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ and the distribution of the soft equivocation $\mathbf{v}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}'_i], \mathbf{R}_i, -x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}}, s_1)$ are both statistically close to $([\mathbf{A}_i|\mathbf{D}'_i])_{s_1}^{-1}(-x_i\mathbf{e}_1 + \mathbf{G}\hat{\mathbf{c}})$. This leads to fake commitments and soft equivocation having exactly the same distribution as soft commitments and their corresponding soft openings. $\square$

Remark A.6 (Partial Succinctness). For $n = \lambda$, $m = O(n\log q)$, $m' = n(\lceil\log q\rceil + 1) \leq m$, Gaussian parameters $s_0 = O(lm\log(nl))$, $s_1 = O(l^{3/2}m^{3/2}\log(nl)\cdot s_0) = O(l^{5/2}m^{5/2}\log^2(nl))$, bound $\beta = \sqrt{l(m+m')+m'}\cdot s_1 = O(l^3n^3\log^2(nl)\log^3 q)$, lattice modulus $q = \beta\cdot\mathsf{poly}(n)$ and $\log q = O(\log\lambda + \log l)$, in Construction A.1,

- Commitment size: A commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^l$ is $(\mathbf{c}, \mathbf{D}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n\times m'l}$ where
$$|\mathbf{c}| = O(n\log q) = O(\lambda\cdot(\log\lambda + \log l))$$
$$|\mathbf{D}| = O(nm'l\log q) = O(l\cdot\lambda^2\cdot(\log^2\lambda + \log^2 l))$$
- Opening size: A (hard) opening is $(\mathbf{v}_i, \hat{\mathbf{R}}_i) \in \mathbb{Z}_q^{m+m'} \times \mathbb{Z}_q^{m\times m'}$ for any $i \in [\ell]$ where
$$|\mathbf{v}_i| = O((m+m')\log\beta) = O(\lambda\cdot(\log^2\lambda + \log^2 l))$$
$$|\hat{\mathbf{R}}_i| = O(mm') = O(\lambda^2\cdot(\log^2\lambda + \log^2 l))$$
- Public parameters size: The public parameters are $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ where $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$, $\mathbf{T} \in \mathbb{Z}_q^{(lm+m')\times lm'}$ and $|\mathsf{pp}| = l^2\cdot\mathsf{poly}(\lambda, \log l)$.
- Auxiliary information size: An auxiliary information for (hard) commitment is $\mathsf{aux} = \{\mathbf{x}, \mathbf{v}_1, ..., \mathbf{v}_l, \hat{\mathbf{R}}_1, ..., \hat{\mathbf{R}}_l\}$ and $|\mathsf{aux}| = O(\lambda^2 l(\log^2\lambda + \log^2 l))$.

Observe that the commitment size and opening size is $\mathsf{poly}(\lambda, \log l)$ except $\mathbf{D}$ is $\mathsf{poly}(\lambda, l)$ in commitment. Therefore the Construction A.1 is a partially succinct mercurial vector commitment based on standard SIS.

Although Construction A.1 is partially succinct, we still want to argue that it suffices to build a lattice-based ZK-EDB. Following the general construction of ZK-EDB [9], informally speaking, the message of the commitment in each (internal) node is always a *hash value* of the commitment in its child node. It leads to such a partially succinct commitment that can also be hashed and stored in the father node when building the ZK-EDB. What's more, inspired by [8], Construction A.1 can be transformed into a fully succinct mercurial vector commitment by utilizing a standard lattice-based vector commitment e.g. [26,28].

Remark A.7 (Extension to Succinctness). We observe that when verifying an opening π_i or τ_i, it only involves $\mathbf{D}_i$ in $\mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l)$ for any $i \in [\ell]$. Therefore, we can use a vector commitment to commit to $(\mathbf{D}_1, ..., \mathbf{D}_l)$ so that to extend Construction A.1 to succinctness. Define a standard (lattice-based) vector commitment $\mathsf{VC} = (\mathsf{VC.KeyGen}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Verify})$. We modify the main four phases in Construction A.1 as follows:

- Setup: It outputs $(\mathsf{pp}, \mathsf{pp}_{\mathsf{VC}})$ that are from $\mathsf{VC.KeyGen}$ and Setup.
- Commit: Including hard commitment, soft commitment, and fake commitment, it obtains $(\mathbf{c}, \mathbf{D})$ from $\mathsf{Hard\_com}$, $\mathsf{Soft\_com}$ or $\mathsf{Fake\_com}$, then computes $(C_{\mathsf{VC}}, \mathsf{aux}_{\mathsf{VC}}) \leftarrow \mathsf{VC.Com}(\mathsf{pp}_{\mathsf{VC}}, \mathbf{D}_1, ..., \mathbf{D}_l)$. It outputs $(\mathbf{c}, C_{\mathsf{VC}})$.
- Open: Including hard opening, soft opening, hard equivocation, and soft equivocation, it obtains π_i or τ_i from $\mathsf{Hard\_open}$, $\mathsf{Soft\_open}$, $\mathsf{Equiv\_Hopen}$ or $\mathsf{Equiv\_Sopen}$, then computes $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}_{\mathsf{VC}}, i, \mathbf{D}_i, \mathsf{aux}_{\mathsf{VC}})$. It outputs (π_i, Λ_i) or (τ_i, Λ_i).
- Verify: Including hard verification and soft verification, it first runs $\mathsf{Hard\_verify}$ or $\mathsf{Soft\_verify}$, then computes $\mathsf{VC.Verify}(C_{\mathsf{VC}}, \mathbf{D}_i, i, \Lambda_i)$. It outputs 1 for both algorithms accept; Otherwise, it outputs 0.

After our modification, we can observe that the size of commitment and opening is $\mathsf{poly}(\lambda, \log l)$ if VC is succinct.

Remark A.8 (Transparent Setup in [19]). As an additional contribution, we observe that a lattice-based mercurial commitment with a trusted setup proposed by Libert et al. [19] can be transformed into a transparent setup in a straight way. The matrix $\mathbf{A}_1$ sampled by $\mathsf{TrapGen}$ in Setup algorithm actually can be replaced by a randomly sampled matrix. Therefore, there will not be a trapdoor in the setup phase, and can be run by the untrusted party e.g. using a hash function to generate the (pseudo) random matrix. The revisited construction and security analysis are shown in Appendix B. What's more, following the structure in [8], we can utilize a lattice-based vector commitment with transparent setup e.g. [6] and our mercurial commitment to build the *first* ZK-EDB with transparent setup.

A.1 Updatable Mercurial Vector Commitment Based on SIS

We show how to construct a differentially updatable mercurial vector commitment from Construction A.1 which satisfies updatable mercurial hiding.

Construction A.9 (Differentially Updatable MVC Based on SIS). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$ be Gaussian width parameters. Let l be the vector dimension. Let $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x}$ which $\mathbf{x}'$ is the update vector and $\mathbf{x}$ is the old vector. We only present $\mathsf{Update\_com}$, $\mathsf{Update\_open}$ algorithms below, and the other algorithms are the same in Construction A.1.

– $\{(\mathbf{c}',\mathbf{D}'),\mathsf{aux}',\mathsf{st}\} \leftarrow \mathsf{Update\_com}(\mathsf{pp},\mathsf{flag},(\mathbf{c},\mathbf{D}),\mathsf{aux},\bar{\mathbf{x}})$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}_1,...,\mathbf{A}_l,\mathbf{T}\}$, if $\mathsf{flag} = \mathsf{hard}$ that implies $(\mathbf{c},\mathbf{D})$ is a hard commitment which $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = (\mathbf{D}_1,...,\mathbf{D}_l)$, $\mathbf{D}_i = \mathbf{A}_i\hat{\mathbf{R}}_i$ for all $i \in [\ell]$, the auxiliary information $\mathsf{aux} = \{\{\mathbf{v}_i\}_{i\in[\ell]},\{\hat{\mathbf{R}}_i\}_{i\in[\ell]}\}$, $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x} = (\bar{x}_1,...,\bar{x}_l) \in \mathbb{Z}_q^l$;
If $\mathsf{flag} = \mathsf{soft}$ and $(\mathbf{c},\mathbf{D})$ is a soft commitment which $\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}$ and $\mathbf{D} = (\mathbf{D}_1,...,\mathbf{D}_l)$, $\mathbf{D}_i = \mathbf{G} - \mathbf{A}_i\hat{\mathbf{R}}_i$ for $i \in [\ell]$. And the auxiliary information $\mathsf{aux} = \{\mathbf{c},\{x_i,\hat{\mathbf{R}}_i,\mathbf{v}_i\}_{i\in S}\}$ means that the soft commitment $(\mathbf{c},\mathbf{D})$ has been opened to some message x_i at some indices $i \in S$ ($|S|$ can be 0 which means the commitment have not been opened). Let $\bar{x}_i = x_i' - x_i$ for $i \in S$ and $\bar{x}_i = x_i' - x_i$ which $x_i \xleftarrow{\$} \mathbb{Z}_q$ for $i \in [\ell]/S$. Then, it samples other $\mathbf{v}_i$ for $i \in [\ell]/S$ via $\mathsf{SampPre}([\mathbf{A}_i|\mathbf{G} - \mathbf{A}_i\hat{\mathbf{R}}_i],\hat{\mathbf{R}}_i,\mathbf{c} - x_i\mathbf{e}_1,s_1)$.
For both situation, it samples $\hat{\mathbf{R}}_i' \xleftarrow{\$} \{0,1\}^{m\times m'}$, phases $\mathbf{v}_i = [\mathbf{v}_{i,1} \in \mathbb{Z}_q^m|\mathbf{v}_{i,2} \in \mathbb{Z}_q^{m'}]^\mathsf{T}$ for $i \in [\ell]$ and constructs the target vector $\bar{\mathbf{u}} \in \mathbb{Z}_q^{nl}$, $\bar{\mathbf{B}}_l' \in \mathbb{Z}_q^{nl\times\left(l\left(m+m'\right)+m'\right)}$, $\mathbf{T}' \in \mathbb{Z}_q^{\left(l\left(m+m'\right)+m'\right)\times m'l}$ as follows,

$$\bar{\mathbf{u}} = \begin{bmatrix} -\bar{x}_1\mathbf{e}_1 + \mathbf{D}_1 \cdot \mathbf{v}_{1,2} - \mathbf{A}_1\hat{\mathbf{R}}_1' \cdot \mathbf{v}_{1,2} \\ \vdots \\ -\bar{x}_l\mathbf{e}_l + \mathbf{D}_l \cdot \mathbf{v}_{l,2} - \mathbf{A}_l\hat{\mathbf{R}}_l' \cdot \mathbf{v}_{l,2} \end{bmatrix} \tag{A.6}$$

$$\bar{\mathbf{B}}_l' = \left[\begin{array}{ccc|c} [\mathbf{A}_1|\mathbf{A}_1\hat{\mathbf{R}}_1'] & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & [\mathbf{A}_l|\mathbf{A}_l\hat{\mathbf{R}}_l'] & -\mathbf{G} \end{array}\right], \quad \mathbf{T}' = \begin{bmatrix} \frac{\mathbf{T}_1}{\mathbf{0}^{m'\times m'l}} \\ \vdots \\ \frac{\mathbf{T}_l}{\mathbf{0}^{m'\times m'l}} \\ \mathbf{T}_\mathbf{G} \end{bmatrix} \tag{A.7}$$

then, uses $\mathbf{T}'$ to sample the preimage as $[\bar{\mathbf{v}}_1,...,\bar{\mathbf{v}}_l,\bar{\hat{\mathbf{c}}}]^\mathsf{T} \leftarrow \mathsf{SampPre}(\bar{\mathbf{B}}_l',\mathbf{T}',\bar{\mathbf{u}},s_1)$. Last, it computes $\bar{\mathbf{c}} = \mathbf{G}\bar{\hat{\mathbf{c}}}$, $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$, $\mathbf{D}' = (\mathbf{D}_1',...,\mathbf{D}_l')$ which $\mathbf{D}_i' = \mathbf{A}_i\hat{\mathbf{R}}_i'$ for $i \in [\ell]$ and $\mathbf{v}_i' = \mathbf{v}_i + \bar{\mathbf{v}}_i$ for all $i \in [\ell]$. It outputs the updated hard commitment $(\mathbf{c}',\mathbf{D}')$, the updated auxiliary information (*updated opening*) $\mathsf{aux}' = (\{\mathbf{v}_i'\}_{i\in[\ell]},\{\hat{\mathbf{R}}_i'\}_{i\in[\ell]})$ and the statement $\mathsf{st} = \{\{\bar{\mathbf{v}}_i\}_{i\in[\ell]},\{\hat{\mathbf{R}}_i'\}_{i\in[\ell]},\bar{\mathbf{c}},\mathbf{D}'\}$.

– $U_i \leftarrow \mathsf{Update\_open}(\mathsf{st},i)$: Input the statement $\mathsf{st} = \{\{\bar{\mathbf{v}}_i\}_{i\in[\ell]},\{\hat{\mathbf{R}}_i'\}_{i\in[\ell]},\bar{\mathbf{c}},\mathbf{D}'\}$ and index $i \in [\ell]$, it outputs $U_i = \{\bar{\mathbf{c}},\hat{\mathbf{R}}_i',\bar{\mathbf{v}}_i,\mathbf{D}'\}$.

– $\{\pi_i',(\mathbf{c}',\mathbf{D}')\} \leftarrow \mathsf{User\_update}(\mathsf{pp},(\mathbf{c},\mathbf{D}),\pi_i,i,U_i)$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}_1,...,\mathbf{A}_l,\mathbf{T}\}$, the old commitment $(\mathbf{c},\mathbf{D})$, the opening π_i, the index $i \in [\ell]$, and the update information $U_i = \{\bar{\mathbf{v}}_i,\hat{\mathbf{R}}_i',\bar{\mathbf{c}},\mathbf{D}'\}$. It computes $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$, and $\mathbf{v}_i' = \mathbf{v}_i + \bar{\mathbf{v}}_i$. Last it outputs the updated commitment $(\mathbf{c}',\mathbf{D}')$. and the updated hard opening $\pi' = \{\mathbf{v}_i',\hat{\mathbf{R}}_i'\}$ if π is a hard opening or the updated soft opening $\pi' = \mathbf{v}_i'$ if π is a soft opening.

– $\{(\mathbf{c}',\mathbf{D}'),\mathsf{st}\} \leftarrow \mathsf{Equiv\_Ucom}(\mathsf{pp},tk,(\mathbf{c},\mathbf{D}))$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}_1,...,\mathbf{A}_l,\mathbf{T}\}$ and trapdoor key tk, and the old commitment $(\mathbf{c},\mathbf{D})$, it first samples $\bar{\bar{\mathbf{c}}} \leftarrow D_{\mathbb{Z}^{m'},s_1}$, $\hat{\mathbf{R}}'_i \xleftarrow{\$} \{0,1\}^{m \times m'}$ for all $i \in [\ell]$, then computes $\bar{\mathbf{c}} = \mathbf{G}\bar{\bar{\mathbf{c}}}$, $\mathbf{c}' = \mathbf{c} + \bar{\mathbf{c}}$ and $\mathbf{D}' = (\mathbf{D}'_1, ..., \mathbf{D}'_l)$ which $\mathbf{D}'_i = \mathbf{A}_i\hat{\mathbf{R}}'_i$ for $i \in [\ell]$. Finally, it outputs the *fake updated commitment* $(\mathbf{c}', \mathbf{D}')$ and the statement $\mathsf{st} = \{\bar{\mathbf{c}}, \mathbf{c}', \mathbf{D}', \hat{\mathbf{R}}'_1, ..., \hat{\mathbf{R}}'_l\}$.

– $\{U_i, \mathsf{aux}'\} \leftarrow \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, i, x'_i, \mathsf{aux}, \mathsf{st})$: Input the public parameters $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$, the trapdoor key tk, the index i, the updated message x'_i, the old commitment $(\mathbf{c}, \mathbf{D})$, the auxiliary information $\mathsf{aux} = \{\mathbf{c}, \{x_j, \hat{\mathbf{R}}_j, \mathbf{v}_j\}_{j \in S}\}$ which the fake commitment has been opened to some message x_j at some indexes $j \in S$ $(0 \le |S| \le l)$,and the statement $\mathsf{st} = \{\bar{\mathbf{c}}, \mathbf{c}', \mathbf{D}'\hat{\mathbf{R}}'_1, ..., \hat{\mathbf{R}}'_l\}$. If $i \in [\ell]/S$, it first samples $\mathbf{v}_i \leftarrow D_{\mathbb{Z}^{m+m'},s_1}$ and then constructs the target vector as

$$\mathbf{u}_i = \mathbf{c}' - x'_i\mathbf{e}_1 - [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\mathbf{v}_i$$

and then phases $\mathbf{R}_i$ from tk to sample the preimage as $\bar{\mathbf{v}}_i = \mathsf{SampPre}([\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i], \mathbf{R}_i, \mathbf{u}_i, s_1)$. Next, it computes $\mathbf{v}'_i = \bar{\mathbf{v}}_i + \mathbf{v}_i$. Finally, it outputs the update information $U_i = \{\bar{\mathbf{c}}, \hat{\mathbf{R}}'_i, \bar{\mathbf{v}}_i, \mathbf{D}'\}$ and the updated auxiliary information $\mathsf{aux}' = \{\mathbf{v}'_i, \hat{\mathbf{R}}'_1, ..., \hat{\mathbf{R}}'_l\}$.

Theorem A.10 (Correctness). *For $n = \lambda$, $m = O(n \log q)$, $s_0 = O(lm \log(ln))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$, and $\beta = \sqrt{l(m+m') + m'} \cdot s_1$, then Construction A.9 is correct.*

Proof. We only show the correctness of $\mathsf{Update\_com}$, $\mathsf{Update\_open}$ and $\mathsf{User\_update}$. Suppose polynomial $l = l(\lambda)$, $\mathbf{x} \in \mathbb{Z}_q^l$, $m \ge m' = O(n \log q)$, for all $\mathbf{x} \in \mathbb{Z}_q^l$ and and index $i \in [\ell]$. Let $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ where $\mathsf{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$ and $\pi_i \leftarrow \mathsf{Hard\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$ and $\tau_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, x, i, \mathsf{aux})$. Let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp}, tk)$, $\pi_i \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, tk, x, i, \mathsf{aux})$, and $\tau_i \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, tk, x_i, i, \mathsf{aux})$. Let $\{(\mathbf{c}', \mathbf{D}'), \mathsf{aux}', \mathsf{st}\} \leftarrow \mathsf{Update\_com}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \bar{\mathbf{x}})$ and $U_i \leftarrow \mathsf{Update\_open}(\mathsf{st}, i)$. Let $\{(\mathbf{c}', \mathbf{D}'), \mathsf{st}\} \leftarrow \mathsf{Equiv\_Ucom}(\mathsf{pp}, tk, (\mathbf{c}, \mathbf{D}))$ and $\{U_i, \mathsf{aux}'\} \leftarrow \mathsf{Equiv\_Uopen}(\mathsf{pp}, tk, i, x'_i, \mathsf{aux}, \mathsf{st})$. Let $\{\pi'_i, (\mathbf{c}', \mathbf{D}')\} \leftarrow \mathsf{User\_update}(\mathsf{pp}, (\mathbf{c}, \mathbf{D}), \pi_i, i, U_i)$. Consider $\mathsf{Hard\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}', \mathbf{D}'), \pi'_i)$:

By Theorem A.2, for old commitment $(\mathbf{c} = \mathbf{G}\hat{\mathbf{c}}, \mathbf{D} = (\mathbf{D}_1, ..., \mathbf{D}_l))$, for all $i \in [\ell]$, we phase $\mathbf{v}_i = [\mathbf{v}_{i,1}|\mathbf{v}_{i,2}]^\mathsf{T}$ and have

$$\mathbf{G}\hat{\mathbf{c}} - x_i\mathbf{e}_1 = \mathbf{A}_i\mathbf{v}_{i,1} + \mathbf{D}_i \cdot \mathbf{v}_{i,2}, \qquad \|\mathbf{v}_i\| \le \beta \tag{A.8}$$

Suppose $s_1 \ge \sqrt{(l(m+m') + m')lm'}\|\mathbf{T}'\| \cdot \omega(\sqrt{\log(nl)})$, by Theorem 2.5, we have

$$\mathbf{G}\bar{\bar{\mathbf{c}}} - \bar{x}_i\mathbf{e}_1 + \mathbf{D}_i \cdot \mathbf{v}_{i,2} - \mathbf{A}_i\hat{\mathbf{R}}'_i \cdot \mathbf{v}_{i,2} = [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\bar{\mathbf{v}}_i, \quad \|\bar{\mathbf{v}}_i\| \le \beta \tag{A.9}$$

For $\mathbf{G}\hat{\mathbf{c}}' = \mathbf{G}(\bar{\bar{\mathbf{c}}} + \hat{\mathbf{c}})$, $x'_i = \bar{x}_i + x_i$, $\mathbf{v}'_i = \bar{\mathbf{v}}_i + \mathbf{v}_i$, we add Eq. A.8 and Eq. A.9 to obtain

$$\mathbf{G}\hat{\mathbf{c}}' - x'_i\mathbf{e}_1 = \mathbf{A}_i\mathbf{v}_{i,1} + \mathbf{A}_i\hat{\mathbf{R}}'_i \cdot \mathbf{v}_{i,2} + [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\bar{\mathbf{v}}_i = [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\mathbf{v}'_i$$

which $\|\mathbf{v}_i'\| \leq 2\beta$. Therefore the verification will accept the update hard commitment and its hard (soft) opening if we set the norm bound on the opening to $k\beta$, which can support up to k updates. Besides, similar to [28], we can set the norm bound and the modulus to be super-polynomial to support an arbitrary polynomial number of updates. $\square$

Theorem A.11 (Mercurial Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n \log q)$, q is prime and $s_0 = O(lm \log(nl))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$. Under the BASIS_{rand} assumption with parameters $(n-1, m, q, 2(m+m')k\beta, s_0, l)$, the Construction A.9 is mercurial binding.*

Proof (Sketch). We briefly show that the updated commitment and opening satisfy mercurial binding and mercurial hiding. The proof of the mercurial binding is basically the same as Theorem A.3. Namely, given the public parameter pp, if the adversary $\mathcal{A}$ can generate a hard (updated) commitment $(\mathbf{c}, \mathbf{D})$ and two valid (updated) openings $(\mathbf{v}_i, x_i, \hat{\mathbf{R}}_i)$, $(\mathbf{v}_i', x_i')$ at same index i to different message which $x_i \neq x_i'$. Then there exist an algorithm $\mathcal{B}$ can use $\|[\mathbf{I}_m|\hat{\mathbf{R}}_i](\mathbf{v} - \mathbf{v}')\| \leq 2(m + m')k\beta$ as a solution to break the $\mathsf{BASIS}_{\mathsf{rand}}$. $\square$

Theorem A.12 (Updatable Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(lm \log(ln))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$, then Construction A.9 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof. The Challenger first sets up the scheme and obtains the public parameter $\mathrm{pp} = \{\mathbf{A}_1, ..., \mathbf{A}_l, \mathbf{T}\}$ via the real protocol, and $tk = \tilde{\mathbf{R}} = \mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l)$ is the trapdoor key which $\mathbf{R}_i$ is the trapdoor of $\mathbf{A}_i$. Then we prove the updatable mercurial hiding of the construction from the following aspects.

For Hcom_Hopen Equivocation. For any message vector $\mathbf{x}$ and $\mathbf{x}'$, we show that the distribution of fake commitments, hard equivocations, updated fake commitments, and update information is statistically close to that of hard commitments, hard openings, updated commitments, and update information.

Firstly, by Theorem A.5, we can know that the distribution of fake commitments and hard equivocations is statistically close to the distribution of hard commitments $(\mathbf{c}, \mathbf{D})$ and hard openings $\mathbf{v}$. Then, note that $\hat{\mathbf{R}}_i'$ and $\mathbf{D}' = (\mathbf{D}_1', ..., \mathbf{D}_l')$ are generated in the same way in both updated commitments and fake updated commitments. By Theorem 2.5, the distribution of the rest of the update information and updated hard commitment $\{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l, \bar{\bar{\mathbf{c}}}\}$ from $\mathsf{SampPre}$ $(\bar{\mathbf{B}}_l', \mathbf{T}', \bar{\mathbf{u}}, s_1)$ in Eq. A.7 is statistically close to the distribution $(\bar{\mathbf{B}}_l')_{s_1}^{-1}(\bar{\mathbf{u}})$.

Let $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{A}_1|\mathbf{D}_1'], ..., [\mathbf{A}_l|\mathbf{D}_l'])$, then $\bar{\mathbf{B}}_l' = [\bar{\mathbf{A}}| - 1^l \otimes \mathbf{G}]$. Since $s_1 \geq \log(l(m + m'))$, by Lemma 2.4, the distribution of $\{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l, \bar{\bar{\mathbf{c}}}\} \leftarrow (\bar{\mathbf{B}}_l')_{s_1}^{-1}(\mathbf{u})$ is statistically close to the distribution

$$\left\{\bar{\bar{\mathbf{c}}} \leftarrow D_{\mathbb{Z}^{m'}, s_1}, \{\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l\} \leftarrow \bar{\mathbf{A}}_{s_1}^{-1}\left(\bar{\mathbf{u}} + 1^l \otimes \mathbf{G}\bar{\bar{\mathbf{c}}}\right)\right\}$$

which $\bar{\bar{\mathbf{c}}}$ is the same as fake updated commitment.

Since $\bar{\mathbf{A}} = \mathsf{diag}([\mathbf{A}_1|\mathbf{D}'_1], ..., [\mathbf{A}_l|\mathbf{D}'_l])$, this leads to that each $\bar{\mathbf{v}}_i$ is distributed to $([\mathbf{A}_i|\mathbf{D}'_i])_{s_1}^{-1}(\bar{\mathbf{u}}_i + \mathbf{G}\bar{\bar{\mathbf{c}}})$. For $\bar{\mathbf{u}}_i$ is the same in Eq. A.6, $\mathbf{c}' = \mathbf{G}\hat{\mathbf{c}} + \mathbf{G}\bar{\bar{\mathbf{c}}}$ and Eq. 3.9 holds in the hard commitment, we have $\mathbf{u}_i$ in fake updated commitment

$$\bar{\mathbf{u}}_i + \mathbf{G}\bar{\bar{\mathbf{c}}} = \mathbf{u}_i = \mathbf{c}' - x'_i\mathbf{e}_1 - [\mathbf{A}_i|\mathbf{A}_i\hat{\mathbf{R}}'_i]\mathbf{v}_i$$

And thanks to Theorem 2.5, the distribution of $([\mathbf{A}_i|\mathbf{D}'_i])_{s_1}^{-1}(\bar{\mathbf{u}}_i+\mathbf{G}\bar{\bar{\mathbf{c}}})$ is statistically close to the distribution of $\bar{\mathbf{v}}_i \leftarrow \mathsf{SampPre}([\mathbf{A}_i|\mathbf{D}'_i], \mathbf{R}_i, \mathbf{u}_i, s_1)$ in the fake updated information. This leads to fake updated commitments and fake update information having exactly the same distribution as updated commitments and update information.

For Hcom_Sopen Equivocation. Follow the same arguments as Hcom_Hopen Equivocation.

For Scom_Sopen Equivocation. For any message vector $\mathbf{x}$ and $\mathbf{x}'$, we show that the distribution of fake commitments, soft equivocations, updated fake commitments, and update information is statistically close to that of soft commitments, soft openings, updated commitments, and update information.

The proof is nearly identical to that of the proof of Hcom_Hopen Equivocation. By Theorem A.5, we can know that the distribution of fake commitments and soft equivocations is statistically close to the distribution of soft commitments $(\mathbf{c}, \mathbf{D})$ and soft openings $\mathbf{v}$. After that, the steps of updating for the soft commitment are the same as the hard commitment. Therefore, the distribution of fake updated commitments and fake update information is statistically close to the distribution of updated commitments and update information. □

What's more, as we mentioned before, the partially succinct updatable MVC can be used to build the updatable ZK-EDB following the generic framework [22,8].

Appendix B Lattice-Based Mercurial Commitment with Transparent Setup

In this section, we revisit [19] and provide the construction of lattice-based mercurial commitment with transparent setup.

Construction B.1 (Mercurial Commitment with Transparent Setup). Let λ be a security parameter, $l = l(\lambda)$ be a dimension of the message space $\mathcal{M} = \{0,1\}^l$, a dimension $n = n(\lambda)$, a prime modulus $q = q(\lambda)$. Let $m' = n\lceil\log q\rceil$, $\bar{m} = 2n\lceil\log q\rceil$, $m = \bar{m} + m'$. Let $\sigma = \Omega(\sqrt{n\log q\log n})$ be a Gaussian parameter.

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$: Input the security parameter λ, and the dimension of the message space l, it randomly sample $\mathbf{A}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times l}$, and $\mathbf{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and output the public parameters $\mathsf{pp} = (\mathbf{A}_0, \mathbf{A}_1)$.
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$: Input the public parameter pp and a message $\mathbf{x} \in 0, 1^l$. It first randomly samples $\mathbf{R} \xleftarrow{\$} \{0,1\}^{n\times m}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}_q^{m+m'},\sigma}$. Then it computes $\mathbf{D} = \mathbf{A}_1\mathbf{R} \in \mathbb{Z}_q^{n\times m'}$, $\mathbf{c} = \mathbf{A}_0\mathbf{x} + [\mathbf{A}_1|\mathbf{D}]\mathbf{r} \in \mathbb{Z}_q^n$. It outputs the hard commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = (\mathbf{R}, \mathbf{r})$.

- $\pi \leftarrow \mathsf{Hard\_open}(\mathsf{aux})$: Input the auxiliary information aux, it outputs the hard opening $\pi = (\mathbf{R}, \mathbf{r})$.
- $0/1 \leftarrow \mathsf{Hard\_verify}(\mathsf{pp}, \mathbf{x}, (\mathbf{c}, \mathbf{D}), \pi)$: Input the public parameter pp, the message $\mathbf{x}$, the hard commitment $(\mathbf{c}, \mathbf{D})$, the hard opening π. It phases $\pi = (\mathbf{R}, \mathbf{r})$ and checks the following conditions:

$$\|\mathbf{r}\| \leq \sigma\sqrt{m+m'}, \qquad \mathbf{c} = \mathbf{A}_0\mathbf{x} + [\mathbf{A}_1|\mathbf{D}]\mathbf{r} \tag{B.1}$$

$$\|\mathbf{R}\| \leq 1, \qquad \mathbf{D} = \mathbf{A}_1\mathbf{R} \tag{B.2}$$

  If they all hold, it output 1; Otherwise, it outputs 0;
- $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Soft\_com}(\mathsf{pp})$: Input the public parameter pp. It first randomly samples $\mathbf{R} \xleftarrow{\$} \{0,1\}^{n\times m}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}_q^{m+m'},\sigma}$. Then it computes $\mathbf{D} = \mathbf{G} - \mathbf{A}_1\mathbf{R} \in \mathbb{Z}_q^{n\times m'}$, $\mathbf{c} = [\mathbf{A}_1|\mathbf{D}]\mathbf{r} \in \mathbb{Z}_q^n$. It outputs the hard commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = (\mathbf{R}, \mathbf{r})$.
- $\tau \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, \mathsf{flag}, \mathbf{x})$: Input the public parameters pp, the flag flag, the message $\mathbf{x}$, the auxiliary information aux, it phases $\mathsf{aux} = (\mathbf{R}, \mathbf{r})$. If $\mathsf{flag} = \mathsf{soft}$, it uses $\mathbf{R}$ as trapdoor to sample the preimage $\mathbf{r}' \leftarrow \mathsf{SampPre}([\mathbf{A}_1|\mathbf{D}], \mathbf{R}, \mathbf{c} - \mathbf{A}_0\mathbf{x}, \sigma)$. It outputs the soft opening $\tau = \mathbf{r}'$; If $\mathsf{flag} = \mathsf{hard}$, it output the soft opening $\tau = \mathbf{r}$.
- $0/1 \leftarrow \mathsf{Soft\_verify}(\mathsf{pp}, \mathbf{x}, (\mathbf{c}, \mathbf{D}), \tau)$: Input the public parameter pp, the message $\mathbf{x}$, the commitment $(\mathbf{c}, \mathbf{D})$, the soft opening τ. It phases $\tau = \mathbf{r}$ and checks Eq. B.1. If they hold, output 1; Otherwise, output 0.

The correctness and mercurial binding of Construction B.1 are obvious and the same as [19]. We use a hybrid game to prove the mercurial hiding.

Theorem B.2. *For $n = \lambda$, q is prime, $m' = n\lceil \log q\rceil$, $\bar{m} = 2n\lceil \log q\rceil$, $m = \bar{m} + m'$,$\sigma = \Omega(\sqrt{n \log q \log n})$, then Construction B.1 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof (Sketch). We define the following hybrid games to prove the mercurial hiding of Construction B.1.

- Hyb_0: This is the real mercurial hiding experiment: the challenger stats by randomly sampling the public parameters $\mathsf{pp} = (\mathbf{A}_0, \mathbf{A}_1)$, then run the equivocation games to let the adversary $\mathcal{A}$ distinguish the real output and ideal output. The real outputs are honestly generated from the above algorithms in Construction B.1 and the ideal output is simulated without any message. And the distribution between the real outputs and ideal outputs is statistically close. The output of the experiment is 1 if $\mathcal{A}$ succeeds.
- Hyb_1: Same as Hyb_0 except the challenger samples $(\mathbf{A}_1, \mathbf{T}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$.
- Hyb_2: Same as Hyb_1 except the ideal output is from the following algorithms:
  - $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Fake\_com}(\mathsf{pp})$: Input the public parameters $\mathsf{pp} = (\mathbf{A}_0, \mathbf{A}_1)$, it first randomly sample $\mathbf{R} \xleftarrow{\$} \{0,1\}^{n\times m}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}_q^{m+m'},\sigma}$. Then it computes $\mathbf{D} = \mathbf{A}_1\mathbf{R} \in \mathbb{Z}_q^{n\times m'}$, $\mathbf{c} = [\mathbf{A}_1|\mathbf{D}]\mathbf{r}$. It outputs the fake commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = (\mathbf{R}, \mathbf{r})$.

- $\pi \leftarrow \mathsf{Equiv\_Hopen}(\mathsf{pp}, \mathbf{T}, \mathbf{x}, (\mathbf{c}, \mathbf{D}), \mathsf{aux})$: Input the public parameters pp, the trapdoor $\mathbf{T}$, the message $\mathbf{x}$, the fake commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information aux. It uses $\mathbf{T}$ to sample the preimage $\mathbf{r}' \leftarrow \mathsf{SampPre}$ $([\mathbf{A}_1|\mathbf{D}], \mathbf{T}, \mathbf{c} - \mathbf{A}_0\mathbf{x}, \sigma)$. It outputs the hard equivocation $\pi = (\mathbf{R}, \mathbf{r}')$.
- $\tau \leftarrow \mathsf{Equiv\_Sopen}(\mathsf{pp}, \mathbf{T}, \mathbf{x}, (\mathbf{c}, \mathbf{D}), \mathsf{aux})$: Input the public parameters pp, the trapdoor $\mathbf{T}$, the message $\mathbf{x}$, the fake commitment $(\mathbf{c}, \mathbf{D})$ and the auxiliary information aux. It uses $\mathbf{T}$ to sample the preimage $\mathbf{r}' \leftarrow \mathsf{SampPre}$ $([\mathbf{A}_1|\mathbf{D}], \mathbf{T}, \mathbf{c} - \mathbf{A}_0\mathbf{x}, \sigma)$. It outputs the soft equivocation $\tau = \mathbf{r}'$.

By Theorem 2.5, $\mathsf{Hyb}_0(\mathsf{A}) \approx \mathsf{Hyb}_1(\mathsf{A})$. By construction (same as [19]), $\mathsf{Hyb}_1(\mathsf{A}) \approx \mathsf{Hyb}_2(\mathsf{A})$ and $\Pr[\mathsf{Hyb}_2(\mathcal{A}) = 1] = \mathsf{negl}(\lambda)$. Therefore, Construction B.1 satisfies mercurial hiding. □

Appendix C Updatable Hiding Mercurial Vector Commitment

We first extend the definition of updatable hiding in mercurial commitment [8] to mercurial vector commitment and then provide the variant of Constructions 3.11 and Constructions A.9 that supports updatable hiding.

Definition C.1 (Updatable Hiding). An updatable mercurial vector commitment satisfies updatable hiding if any efficient adversary $\mathcal{A}$ can win the following game with negligible probability. The adversary $\mathcal{A}$ is allowed to choose two messages $\mathbf{x}_0$, $\mathbf{x}_1$ and then receives $(\mathbf{c}, \mathbf{D})$ which is a commitment to $\mathbf{x}_b$ for a randomly chosen $b \xleftarrow{\$} \{0, 1\}$. Then $\mathcal{A}$ is allowed to choose other two messages $\mathbf{x}'_0$, $\mathbf{x}'_1$ and gets the updated commitment $(\mathbf{c}', \mathbf{D}')$ and update information U for $\mathbf{x}'_d$ where $d \xleftarrow{\$} \{0, 1\}$ is chosen at random. Finally, $\mathcal{A}$ outputs two bits b', d' to guess b, d. If $b = b'$ and $d = d'$, the adversary wins.

Construction C.2 (Updatable Hiding MVC). Define a standard (lattice-based) vector commitment $\mathsf{VC} = (\mathsf{VC.KeyGen}, \mathsf{VC.Com}, \mathsf{VC.Open}, \mathsf{VC.Verify})$. We modify the main five phases in Construction A.9 as follows:

- Setup: It additionally generates $\mathsf{pp}_{\mathsf{VC}} \leftarrow \mathsf{VC.KeyGen}(1^\lambda, 1^l)$, outputs $(\mathsf{pp}, \mathsf{pp}_{\mathsf{VC}})$ where pp is from Setup.
- Commit: It additionally samples $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^l$ and computes $(C_{\mathsf{VC}}, \mathsf{aux}_{\mathsf{VC}}) \leftarrow \mathsf{VC.Com}(\mathsf{pp}_{\mathsf{VC}}, \boldsymbol{\alpha})$. It outputs $\{(\mathbf{c}, \mathbf{D}), C_{\mathsf{VC}}\}$ where $(\mathbf{c}, \mathbf{D})$ is from $\mathsf{Hard\_com}$ or $\mathsf{Soft\_com}$.
- Open: It additionally computes $\Lambda_i \leftarrow \mathsf{VC.Open}(\mathsf{pp}_{\mathsf{VC}}, i, \alpha_i, \mathsf{aux}_{\mathsf{VC}})$. It outputs (π_i, Λ_i) or (τ_i, Λ_i) where π_i or τ_i is from $\mathsf{Hard\_open}$ or $\mathsf{Soft\_open}$.
- Update: It sets $\bar{\mathbf{x}} = \boldsymbol{\alpha} + \mathbf{x}' - \mathbf{x}$ and computes $\mathsf{Update\_com}(\mathsf{pp}, \mathsf{flag}, (\mathbf{c}, \mathbf{D}), \mathsf{aux}, \bar{\mathbf{x}})$ to obtain $\{(\mathbf{c}', \mathbf{D}'), \mathsf{aux}', \mathsf{st}\}$. It outputs the updated commitment $(\mathbf{c}', \mathbf{D}')$ and the *global* update information $U = \mathsf{st}$ for the users holding any openings to update.
- Verify: It first runs $\mathsf{Hard\_verify}$ or $\mathsf{Soft\_verify}$ to check the message $x'_i + \alpha_i$ on the index $i \in [\ell]$, then computes $\mathsf{VC.Verify}(C_{\mathsf{VC}}, \alpha_i, i, \Lambda_i)$. It outputs 1 for both algorithms accept; Otherwise, it outputs 0.

The correctness, mercurial binding, and mercurial hiding are the same as Construction A.9, so we omit them.

We briefly show the proof of updatable hiding. The view of the adversary in this game is as follows,

$$\mathcal{D}_{b,d} = \{(\mathbf{c}, \mathbf{D}), C_{\mathsf{VC}}, (\mathbf{c}', \mathbf{D}'), U = \{\bar{\mathbf{c}}, \mathbf{D}', (\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l)\}\}$$

We can observe that for any choice of the massage $\mathbf{x}_b$, $\mathbf{x}'_d$, there exists $\boldsymbol{\alpha}$ that satisfies the equation above. Therefore, without C_{VC}, the distribution $\mathcal{D}_{b,d}$ for $b, d \in \{0, 1\}$ are statistically indistinguishable, even given the global update information U, i.e. the openings $(\bar{\mathbf{v}}_1, ..., \bar{\mathbf{v}}_l)$ of $\bar{\mathbf{x}} = \boldsymbol{\alpha} + \mathbf{x}' - \mathbf{x}$. Therefore, if $\boldsymbol{\alpha}$ is hidden by C_{VC}, the updatable hiding holds.

Appendix D Proof for Construction 3.19

First, we give the correctness, mercurial same-set binding, and mercurial hiding of Construction 3.19, then provide the definition of the mercurial different-set weak binding and show how Construction 3.19 achieves it.

Theorem D.1 (Correctness of Aggregation). *For $n = \lambda$, $m = O(n \log q)$, $s_0 = O(lm^2 \log(nl))$, $s_1 = O(l^{3/2} m^{3/2} \log(nl) \cdot s_0)$, and $\beta = \sqrt{l(m + m') + m'} \cdot s_1$, then Construction 3.19 is correct.*

Proof. To simplify, we only show the correctness of aggregation. Suppose polynomial $l = l(\lambda)$, $m \geq m' = O(n \log q)$, for any $\mathbf{x} \in \mathbb{Z}_p^l$ and set $S \subseteq [\ell]$. Let $\{\mathsf{pp}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^l)$ where $\mathsf{pp} = \{\mathbf{A}, \{\mathbf{W}_i\}_{i\in[\ell]}, \{\mathbf{u}_i\}_{i\in[\ell]}, \mathbf{T}\}$. Wlog, let $\{(\mathbf{c}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{Hard\_com}(\mathsf{pp}, \mathbf{x})$. For $i \in S$, $\pi_i \leftarrow \mathsf{Soft\_open}(\mathsf{pp}, x_i, i, \mathsf{aux})$ and $\mathsf{Soft\_verify}(\mathsf{pp}, x_i, i, (\mathbf{c}, \mathbf{D}), \pi_i) = 1$. Let $\hat{\Pi} \leftarrow \mathsf{Aggregate}(\mathsf{pp}, \mathsf{soft}, (\mathbf{c}, \mathbf{D}), S, \{x_i, \pi_i\}_{i\in S})$. Consider $\mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{soft}, (\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i\in S}, \hat{\Pi})$:

Since π_i is a valid opening to $(\mathbf{c}, \mathbf{D})$, we have $\|\mathbf{v}_i\| \leq \beta$ and $\mathbf{W}_i^{-1}\mathbf{c} = \mathbf{A}\mathbf{v}_i + x_i\mathbf{u}_i$. By construction, since $\hat{\mathbf{v}} = \sum_{i\in S} t_i \mathbf{v}_i$, $\|\hat{\mathbf{v}}\| \leq |S|\beta$. We have

$$\sum_{i\in S} \mathbf{W}_i^{-1}\mathbf{c} = [\mathbf{A}|\mathbf{D}]\hat{\mathbf{v}} + \sum_{i\in S} x_i \mathbf{u}_i$$

and the aggregate verification algorithm accepts. □

Theorem D.2 (Mercurial Same-Set Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n \log q)$, $m' = n(\lceil \log q \rceil + 1) \leq m$ and $s_0 = O(lm^2 \log(nl))$. Under the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with parameters $(n, m, q, 2(m+m')\beta + 2lp, s_0, l)$, Construction 3.19 satisfies (mercurial) same-set binding.*

Proof (Sketch). Suppose there exists an adversary $\mathcal{A}$ that can break the mercurial same-set binding. It means that $\mathcal{A}$ can generate two valid aggregated openings $\hat{\mathbf{v}}$, $\hat{\mathbf{v}}'$ for $\mathbf{x}[S]$ and $\mathbf{x}'[S]$ on the same set $S \in [\ell]$ which $\mathbf{x}[S] \neq \mathbf{x}'[S]$. So that

$$[\mathbf{A}|\mathbf{A}\hat{\mathbf{R}}](\hat{\mathbf{v}} - \hat{\mathbf{v}}') + \sum_{i\in S} (x_i - x'_i)\mathbf{u}_i = 0$$

By the leftover hash lemma and min-entropy (mentioned in [28]), $\mathbf{u}_i$ can be replaced by $\mathbf{A}\mathbf{r}_i$ which $\|\mathbf{r}_i\| = 1$ and the probability of $\mathbf{z} = [\mathbf{I}_m|\hat{\mathbf{R}}](\hat{\mathbf{v}} - \hat{\mathbf{v}}') + \sum_{i\in S}(x_i - x_i')\mathbf{r}_i \neq 0$ is overwhelming. Moreover, $\|\mathbf{z}\| \leq 2(m+m')\beta + 2lp$ so that $\mathbf{z}$ is a valid solution to break the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption. □

Theorem D.3 (Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(lm^2 \log(ln))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0)$, then Construction 3.19 satisfies statistical Hcom_Hopen Equivocation, Hcom_Sopen Equivocation, and Scom_Sopen Equivocation.*

Proof (Sketch). By Theorem 3.5, we know that the distributions of the commitment $(\mathbf{c}, \mathbf{D})$ and the opening $\mathbf{v}$ from Hard_com, Hard_open, Soft_com, Soft_open, Fake_com, Equiv_Hopen and Equiv_Sopen are statistically close. Thus, for any set $S \in [\ell]$, the distributions of each aggregated opening are still statistically close. It leads to the mercurial hiding hold. □

Remark D.4 (Succinctness). In the Construction 3.19, for $n = \lambda$, $m = O(n \log q)$, $m' = n(\lceil \log q \rceil + 1) \leq m$, Gaussian parameters $s_0 = O(lm^2 \log(nl))$, $s_1 = O(l^{3/2}m^{3/2} \log(nl) \cdot s_0) = O(l^{5/2}m^{7/2} \log^2(nl))$, bound $\beta = \sqrt{l(m+m') + m'} \cdot s_1 = O(l^3 n^4 \log^2(nl) \log^4 q)$, lattice modulus $q = (2(m+m')\beta + 2lp) \cdot \mathsf{poly}(n)$ and $\log q = O(\log \lambda + \log l + \log p)$. We have the following parameter sizes:

- Commitment size: A commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^l$ is $(\mathbf{c}, \mathbf{D}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^{n \times m'}$ where
$$|\mathbf{c}| = O(n \log q) = O(\lambda \cdot (\log \lambda + \log l + \log p))$$
$$|\mathbf{D}| = O(nm' \log q) = O(\lambda^2 \cdot (\log^2 \lambda + \log^2 l + \log^2 p))$$
- Opening size: A (hard) opening is $(\mathbf{v}, \hat{\mathbf{R}}) \in \mathbb{Z}_q^{m+m'} \times \mathbb{Z}_q^{m \times m'}$ where
$$|\mathbf{v}| = O((m+m') \log \beta) = O(\lambda \cdot (\log^2 \lambda + \log^2 l + \log^2 p))$$
$$|\hat{\mathbf{R}}| = O(mm') = O(\lambda^2 \cdot (\log^2 \lambda + \log^2 l + \log^2 p))$$
- Aggregated opening size: A (soft) aggregated opening is $\hat{\mathbf{v}} \in \mathbb{Z}_q^{m+m'}$ where
$$|\hat{\mathbf{v}}| = O((m+m') \log(l\beta)) = O(\lambda \cdot (\log^2 \lambda + \log^2 l + \log^2 p))$$

The scale of the public parameter and the auxiliary information is the same as in Remark 3.6. Therefore, Construction 3.19 is a succinct aggregatable mercurial vector commitment.

Definition D.5 (Mercurial Different-Set Weak Binding). For a *proper* mercurial vector commitment, given the public parameter pp, for any adversary $\mathcal{A}$ outputs a commitment $(\mathbf{c}, \mathbf{D})$, two *different* set S and T along with the aggregated opening $\hat{\Pi}_S$ and $\hat{\Pi}_T$, the following probability should be $\mathsf{negl}(\lambda)$.

$$\Pr\left[\begin{array}{c|c} \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{hard}, (\mathbf{c}, \mathbf{D}), S, \{x_i\}_{i\in S}, \hat{\Pi}_S) = 1 & \mathsf{pp} \leftarrow \mathsf{Setup}(\lambda, l); \\ \wedge\ x_i \neq x_i', \text{for some } i \in S \cap T\ \wedge & \{(\mathbf{c}, \mathbf{D}), S, T, \{x_i\}_{i\in S}, \\ \mathsf{Aggre\_verify}(\mathsf{pp}, \mathsf{soft}, (\mathbf{c}, \mathbf{D}), T, \{x_i'\}_{i\in T}, \hat{\Pi}_T) = 1 & \{x_i'\}_{i\in[T]}, \hat{\Pi}_S, \hat{\Pi}_T\} \leftarrow \mathcal{A}(1^\lambda, 1^l, \mathsf{pp}) \end{array}\right]$$

The *weak binding* means that the commitment $(\mathbf{c}, \mathbf{D})$ must be honestly generated by $\mathsf{Hard\_com}$ (on some possibly adversarial chosen messages) rather than chosen arbitrarily by the adversary.

Theorem D.6 (Mercurial Different-Set Weak Binding). *For any polynomial $l = l(\lambda)$, $n = \lambda$, $m = O(n \log q)$, $m' = n(\lceil \log q \rceil + 1) \le m$ and $s_0 = O(lm^2 \log(nl))$. Under the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption with parameters $(n, m, q, 2(m+m')\beta + 2lp, s_0, l)$, Construction 3.19 satisfies (mercurial) same-set binding.*

Proof (Sketch). Suppose there exists an adversary $\mathcal{A}$ that can break the mercurial different-set weak binding, i.e. $\mathcal{A}$ can output a hard commitment $(\mathbf{c}, \mathbf{D})$ from $\mathsf{Hard\_com}(\mathsf{pp}, \mathbf{z})$, two valid aggregated openings $\hat{\mathbf{v}}_S$ and $\hat{\mathbf{v}}_T$ for sets $S, T \subseteq [\ell]$ and messages $\mathbf{x}$, $\mathbf{x}'$ which for some $i \in S \cap T$, $x_i \neq x'_i$. Thus, it must be the case that (both) $x_i \neq z_i$ or (and) $x'_i \neq z_i$. Wlog, assume $x_i \neq z_i$, and it leads to break the mercurial same-set binding. Overall, mercurial different-set weak binding holds.

$\square$