Lattice-Based Succinct Mercurial Functional Commitment for Boolean Circuits: Definitions, and Constructions

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

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

Abstract. Vector commitments (VC) have gained significant attention due to their extensive use in applications such as blockchain and accumulators. Mercurial vector commitments (MVC) and mercurial functional commitments (MFC), as variants of VC, are central techniques for constructing more advanced cryptographic primitives, such as zero-knowledge sets and zero-knowledge functional elementary databases (ZK-FEDB). However, existing MFCs *only support linear functions*, which limits their applicability—for instance, in building ZK-FEDBs that support only linear function queries. Moreover, to the best of our knowledge, the current MFCs and ZK-FEDBs, including the state-of-the-art proposed by Zhang and Deng (ASIACRYPT '23) using RSA accumulators, are all based on group-based assumptions and *cannot resist quantum computer attacks.*

To address these limitations, we *first* formalize the system and security models of MFC to support Boolean circuits. Then, we target specific properties of a new falsifiable assumption, namely the BASIS assumption proposed by Wee and Wu (EUROCRYPT '23), to construct the *first* lattice-based succinct mercurial functional commitment for Boolean circuits. As an application of our construction, we demonstrate how it can be used to build the *first* lattice-based ZK-FEDB within the existing generic framework.

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

1 Introduction

Vector commitment (VC) [17,6] supports one to commit to a vector of messages and later fine-grained opens the commitment at a specific index. Generally, a standard VC has three properties:

– *Succinctness*: the sizes of the commitment and the opening are polylogarithmic with the length of the vector.

– *Binding*: the adversary cannot open the commitment to different values at the same index.
– *Hiding*: the adversary can only learn the message at the index given the corresponding opening and cannot learn the messages at other indices.

Later, VCs have been extended to subvector commitment (SVC) [13,23] that allow aggregating the openings to a subvector of the committed vector instead of one index. And functional commitment (FC) supports opening to a linear map [13], constant degree polynomial [2], or Boolean circuit [20,23,4,3,22] of the committed input.

Besides, mercurial vector commitment (MVC) [17,6], as one of the most interesting variants of VCs, satisfies the *mercurial* property additionally. The *mercurial* property was first proposed by Chase et al. [7] in the mercurial commitment (MC) which allows the committer to make two kinds of opening: In the *soft* opening, the committer can claim that "If I have committed to anything at all, then the committed value is m", i.e. it implies that he *may* commit to the value m or nothing. While in the *hard* opening, the committer can declare that "Yes, I really have committed to the value m". It means that he *must* commit to m. In particular, the commitment c can either be both *soft* and *hard* opened only to the unique value m (if c is *hard* commitment), or can only be *soft* opened to arbitrary values, but cannot be *hard* opened at all (if c is *soft* commitment). Moreover, the committer must decide before generating the commitment which one of the two cases suits him better: the *hard* commitment of only one value, or the *soft* commitment of nothing at all.

Correspondingly, the MVC allows one to commit a *hard* commitment to the input vector or a *soft* commitment to nothing at all. The *hard* commitment can be both *hard* and *soft* opened to the unique value at each index, while the *soft* commitment can only be *soft* opened to *arbitrary* value at every index. Furthermore, the security of MVC, named *mercurial hiding* requires that the adversary cannot distinguish between the *soft* commitment and *hard* commitment even given their associated soft openings. *One can find that the property of mercurial hiding in MVC implies the property of hiding in VC.* Subsequently, mercurial subvector commitment (MSVC) [14] was proposed to open to the subvector, while the existing mercurial functional commitment (MFC) [24] *only* supports opening to a *linear* function of the committed vector.

Applications: MVC and MFC apply to many cryptographic building blocks such as zero-knowledge set (ZKS) [7,18,5,15], zero-knowledge elementary database (ZK-EDB) [8,6,14], and zero-knowledge functional elementary database (ZK-FEDB) [24] in which all utilize the mercurial property in MVC and MFC, i.e. using the hard commitment (and soft commitment) to denote the existent (and non-existent) elements and then using the hard opening (and soft opening) to compose the proof of membership, key-value or function value (and non-membership) in the database. It guarantees that the generated proof does not leak any knowledge about the database except the result itself.

Overall, there is neither MFC that supports opening to Boolean circuits nor lattice-based construction of MFC or ZK-FEDB.

To fill these gaps at one time, roughly speaking, we observe that the property of hiding in functional commitment is a *subset* of mercurial hiding property in mercurial functional commitment, and the remaining challenge for achieving mercurial hiding property is how to generate an indistinguishable valid (soft) opening from the soft commitment. Meanwhile, we notice that the functional commitment for Boolean circuits based on the BASIS assumption proposed by Wee and Wu [23] supports hiding the committed messages. Thus, we intend to solve the remaining challenges based on their constructions in this paper.

We refer to Table 1 for a comparison among the state of the art.

Scheme	**AS**	**MC**	**Functions**	$\lvert\mathsf{crs}\rvert$	$\lvert C\rvert$	$\lvert\pi\rvert$	T_c	T_o	T_v
[24]	DDH	✓	linear maps	ℓ	1	1	ℓ	$\lvert f\rvert$	$\lvert f\rvert$
[2]	k-M-ISIS	✗	d-degree polynomial	ℓ^{2d}	1	1	ℓ^{2d}	$\lvert f\rvert$	1^*
[23]	BASIS	✗	d-depth circuit	ℓ^2	1	1	ℓ	$\lvert f\rvert$	$\lvert f\rvert$
[4]	SIS	✗	d-depth circuit†	ℓ	1	ℓ	$\lvert f\rvert$	$\lvert f\rvert$	ℓ
[22]	ℓ-succinct SIS	✗	d-depth circuit	ℓ^2	1	1	ℓ	$\lvert f\rvert$	$\lvert f\rvert$
Cons. 4.1	BASIS	✓	d-depth circuit	ℓ^2	1	1	ℓ	$\lvert f\rvert$	$\lvert f\rvert$

* It needs additional pre-procession before the verification.

† It is a dual functional commitment where one commits to a function f and opens to an input $\mathbf{x}$, while other schemes in this comparison are standard functional commitments where one commits to an input $\mathbf{x}$ and opens to a function f.

Table 1. Comparison to current works on (mercurial) functional commitments. For each scheme, we report the assumption (**AS**) it is based on, whether it satisfies the mercurial property (**MC**), the class of functions it supports, the size of common reference string crs, commitment C, opening π, and the running times T_c, T_o, T_v of the commit, opening, and verification algorithms in terms of the input length ℓ and the size of the associated function $|f|$, i.e., the input number of linear maps, the degree of the polynomial, and the depth of Boolean circuits. We assume functions with a single output. For simplicity, we suppress $\mathsf{poly}(\lambda, d, \log \ell)$ terms throughout the comparison (where λ denotes the security parameter and d refers to either the fixed degree of polynomials or the fixed depth of Boolean circuits).

1.1 Our Contribution

We *first* formalize the definition of succinct mercurial functional commitment for Boolean circuits and propose the *first* lattice-based construction that supports opening to a general (Boolean) circuit, achieves succinctness, and satisfies the security requirements of mercurial (target) binding and mercurial hiding. Furthermore, we show how to utilize our construction to build the *first* lattice-based

ZK-FEDB directly within the existing generic framework that allows users to make Boolean circuit queries.

1.2 Technical Overview

We first recall the construction of succinct functional commitment for Boolean circuits based on the BASIS assumption that supports *private opening* proposed by Wee and Wu [23]. To simplify, we omit some details.

In the setup, it first generates a random target vector $\mathbf{u}$ and a random matrix $\mathbf{A}$ with its trapdoor $\mathbf{R}$. There exists a PPT algorithm $\mathbf{x} \leftarrow \mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{t}, s)$ that input the random matrix $\mathbf{A}$, its trapdoor $\mathbf{R}$, any target vector $\mathbf{t}$, and some Gaussian parameter s, it can output a short vector $\mathbf{x}$ over the distribution of discrete Gaussian $\mathbf{A}^{-1}(\mathbf{t})$ conditioned on $\mathbf{A}\mathbf{x} = \mathbf{t}$. It publishes $\mathbf{u}$, $\mathbf{A}$, and other public parameters as the common reference string and keeps $\mathbf{R}$ as a secret.

During the commitment phase, due to the property of BASIS assumption, the commitment $\mathbf{C}$ of the input $\mathbf{x} \in \{0,1\}^\ell$ can be sampled by $\mathsf{SampPre}(\mathbf{x}, \cdot)$ via a public matrix composed of $\mathbf{A}$ and some public parameters and its public trapdoor. This mechanism can guarantee to *hide* the committed values $\mathbf{x}$, even given the trapdoor of a public matrix composed of $\mathbf{A}$. The full analysis can be found in Definition 2.5 and [23].

Then, we show the opening and verification phases in more detail:

– In the opening phase, it constructs the matrix $\mathbf{D}_f$ and its associated trapdoor $\mathbf{R}_f$ as below:

$$\mathbf{D}_f = [\mathbf{A} | \tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}], \qquad \mathbf{R}_f = \begin{bmatrix} -\mathbf{V}_f \\ \mathbf{I} \end{bmatrix}$$

  where $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^\mathsf{T}$ is the gadget matrix, $\tilde{\mathbf{C}}_f$, $\mathbf{V}_f$ are generated by the homomorphic encoding described in Theorem 2.6 taking commitment $\mathbf{C}$, Boolean circuit $f : \{0,1\}^\ell \rightarrow \{0,1\}$, and input $\mathbf{x} \in \{0,1\}^\ell$ (only for $\mathbf{V}_f$) as input. Thus, we have $\mathbf{D}_f \mathbf{R}_f = \{-\mathbf{G}, \mathbf{G}\}$ (by Theorem 2.6) so that $\mathbf{R}_f$ is the gadget trapdoor of $\mathbf{D}_f$ (by Theorem 2.1). Then it samples the preimage of the public random target vector $\mathbf{u}$ as the opening:

$$\mathbf{v}_f \leftarrow \mathsf{SampPre}(\mathbf{D}_f, \mathbf{R}_f, \mathbf{u}, s)$$

  where s is the Gaussian parameter.

– In the verification phase, it accepts that $\mathbf{v}_f$ is the valid opening to (f, y), i.e. $y = f(\mathbf{x})$, for the commitment $\mathbf{C}$ if

$$\|\mathbf{v}_f\| \leq \beta \quad \wedge \quad [\mathbf{A} | \tilde{\mathbf{C}}_f + (y - 1) \cdot \mathbf{G}] \mathbf{v}_f = \mathbf{u} \tag{1.1}$$

We omit the analysis of correctness and binding and would like to emphasize the property of *private opening*: since the opening $\mathbf{v}_f \leftarrow \mathsf{SampPre}(\mathbf{D}_f, \mathbf{R}_f, \mathbf{u}, s)$ is over the distribution of $\mathbf{D}_f^{-1}(\mathbf{u})$, it means that there exist some *simulating* algorithms that can randomly sample a *fake* commitment $\mathbf{C}$ without any input

$\mathbf{x}$ and generate its valid *equivocation* opening $\mathbf{v}_f$ to any function f at any value y only with the trapdoor $\mathbf{R}$ of $\mathbf{A}$ over the same distribution.

We observe that *private opening* meets the part of the mercurial hiding property, and the rest of this property requires generating the *indistinguishable* soft opening for hard commitment and soft commitment without the trapdoor $\mathbf{R}$ of $\mathbf{A}$. To achieve it, inspired by [15,21], we secretly insert a "trapdoor" into the soft commitment. Here, the difference between [21] is that due to the different phases of generating the opening between functional commitment and vector commitment, we need to modify the opening phase instead of the commitment phase. This is non-trivial work because we need to guarantee it *indistinguishable*, *valid*, and *checkable* without compromising the *private opening*. We sketch as follows:

We first provide two algorithms to generate the *indistinguishable* $\mathbf{D}$ in hard commitment and soft commitment respectively:

$$\mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \quad \text{and} \quad \mathbf{D} = \mathbf{G} - \mathbf{A}\hat{\mathbf{R}}$$

where $\hat{\mathbf{R}}$ is short and sampled randomly. Then we extend the matrix $\mathbf{D}_f$ as follows:

$$\mathbf{D}_f = [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}]$$

After that, to generate an opening for the hard commitment, the trapdoor $\mathbf{R}_f$ of $\mathbf{D}_f$ can be extended naturally by $\mathbf{R}_f = [-\mathbf{V}_f, \mathbf{0}, \mathbf{I}]^\mathsf{T}$; To generate an opening for the soft commitment, the trapdoor $\mathbf{R}_f$ can be constructed by $\hat{\mathbf{R}}$ instead of $\mathbf{V}_f$, i.e. $\mathbf{R}_f = [\hat{\mathbf{R}}, \mathbf{I}, \mathbf{0}]^\mathsf{T}$. It means that without the trapdoor $\mathbf{R}$ of $\mathbf{A}$, it can still generate a *valid* and *indistinguishable* opening that satisfies Eq. 1.1 for the soft commitment which does not contain any input messages and the *private opening* still holds. Therefore, we need $\hat{\mathbf{R}}$ as the additional opening for the hard commitment and add a *check* for $\mathbf{D} \stackrel{?}{=} \mathbf{A}\hat{\mathbf{R}}$ during the verification for hard opening. We provide the formal definition in Section 3 and full constructions and analysis in Section 4.

1.3 Related Work

There are a number of breakthroughs in the academic research of MCs. The first MC was proposed by Chase et al. [7] based on a variant of the Diffie-Hellman (DH) assumption. Catalano et al. [5] presented a trapdoor mercurial commitments (TMC) based on a one-way function. Libert et al. [15] propose the first lattice-based construction of MC. In addition, Libert and Yung [17] proposed the concept of MVC and provided two different constructions based on q-DH assumption and RSA assumption, respectively, which support mercurially commit on a q-length vector. Subsequently, Wang et al. [21] propose a lattice-based construction of MVC that satisfies updatability and aggregatability. Wu et al. [24] put forward the concept of MFC and gave a pairing-based construction that supports opening the commitment to a linear function. Then, as the follow-up of [10,17], Li et al. [14] proposed the first definitions of MSVC and provided a construction based on Computational-DH (CDH) assumption in random oracle (RO) model.

Another line of work is to construct the vector commitments and functional commitments. The concept of VC was first proposed by Catalano and Fiore in [6] and provided two different constructions of VC based on CDH assumptions and RSA assumptions, respectively. Then, Libert et al. [16] generalized the concept of the VC to FC that can open the commitment to a linear function. Besides, there are numerous works in lattice-based constructions of VC [20,23] and FC [20,2,23,4,22,3]. Among them, only the constructions of FC for Boolean circuits proposed by Wee and Wu [23] using a new falsifiable family of basis-augmented SIS assumption (BASIS) satisfy *private opening* which implies *hiding* property. Therefore, our work is based on the BASIS assumption as well.

Overall, there is no work on an MFC that supports opening to a Boolean circuit or a lattice-based construction.

2 Preliminaries

2.1 Notation

Let $\lambda \in \mathbb{N}$ denote the security parameter. For a positive integer ℓ, denote the set $(1, ..., \ell)$ 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 $\mathbf{x}$. $\|\mathbf{x}\|$ is denoted as the infinity norm of vector $\mathbf{x}$. When $\mathbf{X}$ is a matrix, $\|\mathbf{X}\| := \max_{i,j} |X_{i,j}|$. For matrices $\mathbf{A}_1, ..., \mathbf{A}_\ell \in \mathbb{Z}_q^{n\times m}$, we use $\mathsf{diag}(\mathbf{A}_1, ..., \mathbf{A}_\ell) \in \mathbb{Z}_q^{n\ell\times m\ell}$ to be the block diagonal matrix with blocks $\mathbf{A}_1, ..., \mathbf{A}_\ell$ along the main diagonal (and $\mathbf{0}$ elsewhere). We let $\mathsf{poly}(\lambda)$ be a fixed function $O(\lambda^c)$ for some $c \in \mathbb{N}$ and $\mathsf{negl}(\lambda)$ as a function $o(\lambda^{-c})$ for all $c \in \mathbb{N}$. We use $\mathbf{R} \xleftarrow{\$} \{0,1\}^{m\times m'}$ to denote a uniformly randomly sampled 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 $i \in [m']$. For any positive integer k, we denote $\mathbf{I}_k$ as the identity matrix of order k. Let n be a positive integer, $q \in \mathsf{poly}(n)$ be a modulus. Define the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, ..., 2^{\lceil \log q\rceil}) \in \mathbb{Z}_q^{n\times m'}$ where $m' = n(\lceil \log q\rceil + 1)$ and $\otimes$ denotes Kronecker product.

Min-entropy. According to [9,11,23], for a discrete random variable X, let $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$ denote its min-entropy. For two (possibly correlated) discrete random variables X and Y, the average min-entropy of X given Y is denoted as $\mathbf{H}_\infty(X \mid Y) = -\log(\mathbb{E}_{y\to Y} \max_x \Pr[X = x \mid Y = y])$. The optimal probability of an unbounded adversary guessing X given the correlated value Y is $2^{-\mathbf{H}_\infty(X|Y)}$.

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 $\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{Ax} = \mathbf{0} \mod q\}$.

Discrete Gaussian over Lattice. For integer $m \in \mathbb{N}$, we denote $D_{\mathbb{Z}^m,s}$ as the discrete Gaussian distribution centered at 0 over $\mathbb{Z}^m$ with width parameter $s \in \mathbb{R}^+$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times\ell}$ and a vector $\mathbf{v} \in \mathbb{Z}_q^n$, let $\mathbf{A}_s^{-1}(\mathbf{v})$ be the preimage distributed on $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$ conditioned on $\mathbf{A}\mathbf{x} = \mathbf{v} \mod q$. $\mathbf{A}_s^{-1}$ can be extended to matrices by applying $\mathbf{A}_s^{-1}$ to each column of the input.

Theorem 2.1 (Gadget Trapdoor [23,19]). *Let n, m, q, m' be lattice parameters. There exist efficient algorithms (*$\mathsf{TrapGen}$*,* $\mathsf{SampPre}$*):*

- $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m, q)$*: On input the lattice dimension n, the modulus q, and the number of samples m, the trapdoor-generation algorithm outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{R} \in \mathbb{Z}_q^{m\times m'}$ which $\mathbf{A}\mathbf{R} = \mathbf{G}$ and $\|\mathbf{R}\| = 1$.*
- $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, s)$*: 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. If $s \geq \sqrt{mm'}\|\mathbf{R}\|\omega(\sqrt{\log n})$), the preimage sampling algorithm outputs a vector $\mathbf{u} \in \mathbb{Z}_q^m$ satisfying $\mathbf{A}\mathbf{u} = \mathbf{v}$ and the distribution of $\mathbf{u}$ is statistically close to $\mathbf{A}_s^{-1}(\mathbf{v})$.*

Remark 2.2. *Denote $\mathbf{H}$ as a tag if $\mathbf{A}\mathbf{R} = \mathbf{H}\mathbf{G}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$.*

Remark 2.3. *To sample the preimage of a matrix $\mathbf{V} \in \mathbb{Z}_q^{n\times\ell}$, we denote* $\mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{V}, s)$ *as the algorithms that outputs the matrix where the i^{th} column is* $\mathsf{SampPre}(\mathbf{A}, \mathbf{R}, \mathbf{v}_i, s)$ *and $\mathbf{v}_i$ is the i^{th} column of $\mathbf{V}$.*

Definition 2.4 (SIS Assumption [1]). Let λ be a security parameter, and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $\beta = \beta(\lambda)$ be lattice parameters. The short integer solution assumption $\mathsf{SIS}_{n,m,q,\beta}$ holds if for all efficient adversaries $\mathcal{A}$,

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

Definition 2.5 (BASIS Assumption [23]). Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, $\beta = \beta(\lambda)$ be lattice parameters, s be a Gaussian width parameter, 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. The basis-augmented SIS (BASIS) assumption holds with respect to Samp if for all efficient adversaries $\mathcal{A}$,

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

Informally, BASIS assumption requires that SIS assumption is hard towards $\mathbf{A}$ even given a trapdoor $\mathbf{T}$ for its related matrix $\mathbf{B}$.

The instantiation of the BASIS assumption with structured matrices ($\mathsf{BASIS}_{\mathsf{struct}}$) is that: algorithm $\mathsf{Samp}(\lambda, \mathbf{A})$ samples $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times n}$ for all $i \in [\ell]$ and outputs

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

Note that the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption is conceptually similar to k-R-ISIS assumption [2] in which some instances are as hard as standard SIS. However, for now, there is no analogous reduction for the $\mathsf{BASIS}_{\mathsf{struct}}$ assumption or k-R-ISIS assumption from the standard lattice assumption.

To simplify, we use BASIS to represent $\mathsf{BASIS}_{\mathsf{struct}}$ in the following, unless otherwise noted.

Theorem 2.6 (Homomorphic Encoding [11,23]). *Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$. Let $\ell = \ell(\lambda)$ be an input length. Let $\mathcal{F}_\lambda$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ that can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. Then, there exists a pair of efficient algorithms (*EvalF*,* EvalFX*) with the following properties:*

- $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$*: Input a matrix $\tilde{\mathbf{C}} \in \mathbb{Z}_q^{n \times \ell m'}$ and a function $f \in \mathcal{F}_\lambda$, the input-independent evaluation algorithm outputs a matrix $\tilde{\mathbf{C}}_f \in \mathbb{Z}_q^{n \times m'}$.*
- $\mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$*: Input a matrix $\tilde{\mathbf{C}} \in \mathbb{Z}_q^{n \times \ell m'}$ and a function $f \in \mathcal{F}_\lambda$, and an input $\mathbf{x} \in \{0,1\}^\ell$, the input-independent evaluation algorithm outputs a matrix $\mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}} \in \mathbb{Z}_q^{lm' \times m'}$.*

Moreover, for all security parameter $\lambda \in \mathbb{N}$, matrix $\tilde{\mathbf{C}} \in \mathbb{Z}_q^{n \times \ell m'}$, all functions $f \in \mathcal{F}_\lambda$, and all inputs $\mathbf{x} \in \{0,1\}^\ell$, the matrix $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$ and $\mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}} \leftarrow$ $\mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$ *satisfy the following properties:*

- $\|\mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}}\| \leq (n \log q)^{O(d)}$.
- $(\tilde{\mathbf{C}} - \mathbf{x}^\mathsf{T} \otimes \mathbf{G}) \cdot \mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}} = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G}$.

3 System Model and Security Model

In this section, we show the definition of our mercurial functional commitment for Boolean circuits and the security properties required to satisfy.

Definition 3.1 (Mercurial Functional Commitment). Let λ be the security parameter. Let $\mathcal{F}_\lambda$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ that can be computed by Boolean circuits of depth at most $d = d(\lambda)$. A succinct (trapdoor) mercurial functional commitment for $\mathcal{F}_\lambda$ comprises the following algorithms:

– $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$: Input a security parameter λ and an input length ℓ, and a circuit depth d, it outputs common reference string crs and a trapdoor key tk optionally.
– $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs}, \mathbf{x})$: Input the common reference string crs and an input $\mathbf{x} \in \{0,1\}^\ell$, it outputs a hard commitment $(\mathbf{C}, \mathbf{D})$ and auxiliary information aux.
– $\pi \leftarrow \mathsf{HOpen}(\mathsf{crs}, f, \mathsf{aux})$: Input the common reference string crs, a function $f \in \mathcal{F}_\lambda$, and the auxiliary information aux, it outputs a hard opening π.
– $\{0,1\} \leftarrow \mathsf{HVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, y, \pi)$: Input the common reference string crs, a hard commitment $(\mathbf{C}, \mathbf{D})$, a function $f \in \mathcal{F}_\lambda$, a value $y \in \{0,1\}$, and a hard opening π, it outputs 0 or 1 to indicate whether π is a valid hard opening.
– $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{SCom}(\mathsf{crs})$: Input the common reference string crs, it outputs a soft commitment $(\mathbf{C}, \mathbf{D})$, and auxiliary information aux.
– $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{F}, f, y, \mathsf{aux})$: Input the common reference string crs, a flag $\mathbb{F} \in \{\mathbb{H}, \mathbb{S}\}$ which indicates that the soft opening τ is for hard commitment or soft commitment, a function $f \in \mathcal{F}_\lambda$, a value $y \in \{0,1\}$ and the auxiliary information aux, it outputs the soft opening τ. If $\mathbb{F} = \mathbb{H}$ and $y \neq f(\mathbf{x})$, it aborts and outputs $\perp$.
– $\{0,1\} \leftarrow \mathsf{SVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, y, \tau)$: Input the common reference string crs, the commitment $(\mathbf{C}, \mathbf{D})$, a function $f \in \mathcal{F}_\lambda$, a value $y \in \{0,1\}$, and soft opening τ, it outputs 0 or 1 to indicate whether τ is a valid soft opening.
– $\{\mathbf{C}, \mathbf{D}, \mathsf{aux}\} \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$: Input the common reference string crs and trapdoor key tk, it outputs a *fake commitment* $(\mathbf{C}, \mathbf{D})$ and auxiliary information aux.
– $\pi \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$: Input the common reference string crs and the trapdoor key tk, a function $f \in \mathcal{F}_\lambda$, a value $y \in \{0,1\}$, and auxiliary information aux, it outputs a *hard equivocation* π.
– $\tau \leftarrow \mathsf{ESOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$: Input the common reference string crs and the trapdoor key tk, a function $f \in \mathcal{F}_\lambda$, a value $y \in \{0,1\}$, and auxiliary information aux, it outputs a *soft equivocation* τ.

Remark 3.2 (Proper Mercurial Commitment [15]). Generally, for all existing constructions, the soft opening of a hard commitment is a proper part of the hard opening to the same message, so are $\mathsf{SVerify}$ and $\mathsf{HVerify}$. Such mercurial (functional) commitments are called *proper* mercurial (functional) commitments.

Correctness. The correctness of a trapdoor mercurial functional commitment is as follows. Specifically, for all security parameters λ, all functions $f \in \mathcal{F}_\lambda$, all input $\mathbf{x} \in \{0,1\}^\ell$, and the common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d)$, the following conditions must hold with an overwhelming probability.

– For a hard commitment $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs}, \mathbf{x})$, a hard opening $\pi \leftarrow \mathsf{HOpen}(\mathsf{crs}, f, \mathsf{aux})$ and a soft opening $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{H}, f, f(\mathbf{x}), \mathsf{aux})$ to the hard commitment, they must have $\mathsf{HVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, f(\mathbf{x}), \pi) = 1$ and $\mathsf{SVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, f(\mathbf{x}), \tau) = 1$.
– For a soft commitment $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{SCom}(\mathsf{crs})$, a soft opening $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{S}, f, y, \mathsf{aux})$ to the soft commitment, there must have $\mathsf{SVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, y, \tau) = 1$.

- For a fake commitment $\{(\mathbf{C},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$ where tk is the trapdoor key for the construction, a hard equivocation $\pi \leftarrow \mathsf{EHOpen}\ (\mathsf{crs}, tk, f, y, \mathsf{aux})$ and a soft equivocation $\tau \leftarrow \mathsf{ESOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$ to the fake commitment, there must have $\mathsf{HVerify}(\mathsf{crs},(\mathbf{C},\mathbf{D}), f, y, \pi) = 1$ and $\mathsf{SVerify}(\mathsf{crs},(\mathbf{C},\mathbf{D})$ $, f, y, \tau) = 1$.

Mercurial binding. A *proper* mercurial functional commitment satisfies mercurial *target* binding if given the common reference string crs, for any adversary $\mathcal{A}$ outputs a hard commitment $(\mathbf{C},\mathbf{D})$ which is *honestly-generated* from $\mathsf{HCom}(\mathsf{crs},\mathbf{x})$ with some input $\mathbf{x} \in \{0,1\}^\ell$ (possibly adversarially chosen), a function $f \in \mathcal{F}_\lambda$ and a hard opening π (or soft opening τ) to the value $1 - f(\mathbf{x})$, the following probability should be $\mathsf{negl}(\lambda)$. [3]

$$\Pr\left[\mathsf{HVerify}(\mathsf{crs},(\mathbf{C},\mathbf{D}), f, 1-f(\mathbf{x}), \pi) = 1 \;\middle|\; \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell, 1^d); \\ \mathbf{x} \leftarrow \mathcal{A}(\mathsf{crs}); \\ (\mathbf{C},\mathbf{D}) \leftarrow \mathsf{HCom}(\mathsf{crs},\mathbf{x}); \\ \{f,\pi\} \leftarrow \mathcal{A}((\mathbf{C},\mathbf{D}),\mathsf{crs}) \end{array}\right]$$

Mercurial hiding. Given the common reference string crs, for any function $f \in \mathcal{F}_\lambda$, any input $\mathbf{x} \in \{0,1\}^\ell$, no efficient adversary can distinguish between hard commitment with its soft opening $\{\mathbf{x}, (\mathbf{C},\mathbf{D}) \leftarrow \mathsf{HCom}(\mathsf{crs},\mathbf{x}), \tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{H}, f, f(\mathbf{x}), \mathsf{aux})\}$ and soft commitment with its soft opening $\{\mathbf{x}, (\mathbf{C},\mathbf{D}) \leftarrow \mathsf{SCom}(\mathsf{crs}), \tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{S}, f, f(\mathbf{x}), \mathsf{aux})\}$. It uses an equivocation game to prove this.

Equivocation game. There are three sub-games composed of a pair of *real* scenario and *ideal* scenario. Given the common reference string crs and the trapdoor tk, no adversary $\mathcal{A}$ can distinguish between the two scenarios in each sub-game.

- $\mathsf{HHEquivocation}$: $\mathcal{A}$ picks an input $\mathbf{x} \in \{0,1\}^\ell$ and a function $f \in \mathcal{F}_\lambda$. In the real game, $\mathcal{A}$ will receive $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{HCom}(\mathsf{crs},\mathbf{x})$, and $\pi \leftarrow \mathsf{HOpen}$ $(\mathsf{crs}, f, \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ will obtain $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$, and $\pi \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, f(\mathbf{x}), \mathsf{aux})$.
- $\mathsf{HSEquivocation}$: $\mathcal{A}$ picks an input $\mathbf{x} \in \{0,1\}^\ell$ and a function $f \in \mathcal{F}$. In the real game, $\mathcal{A}$ will receive $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{HCom}(\mathsf{crs},\mathbf{x})$, and $\tau \leftarrow \mathsf{SOpen}$ $(\mathsf{crs}, \mathbb{H}, f, f(\mathbf{x}), \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ will obtain $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{FCom}(\mathsf{crs},$ $tk)$, and $\tau \leftarrow \mathsf{ESOpen}(\mathsf{crs}, tk, f, f(\mathbf{x}), \mathsf{aux})$.
- $\mathsf{SSEquivocation}$: In the real game, $\mathcal{A}$ will first get $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{SCom}\ (\mathsf{crs})$, then choose a function $f \in \mathcal{F}_\lambda$ and a value $y \in \{0,1\}$, and finally receive $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{S}, f, y, \mathsf{aux})$. While in the ideal game, $\mathcal{A}$ first obtains $(\mathbf{C},\mathbf{D}) \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$, then chooses a function $f \in \mathcal{F}_\lambda$ and a value $y \in \{0,1\}$, and finally receives $\tau \leftarrow \mathsf{ESOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$.

[3] There exists a stronger notion of mercurial binding where the commitment from the adversary can be chosen arbitrarily, and there is no need to contain any input message. However, like existing lattice-based functional commitments for circuits that satisfy private opening [23] and pairing-based constructions in Algebraic Group Model (AGM) [14,10], our constructions achieve the *weak* (*target*) binding.

Succinctness. A mercurial functional commitment is succinct if there exists a universal polynomial $\mathsf{poly}(\cdot,\cdot,\cdot)$ such that for all $\lambda \in \mathbb{N}$, the size of the commitment has $|(\mathbf{C}, \mathbf{D})| = \mathsf{poly}(\lambda, d, \log \ell)$, and the size of the opening has $|\pi| = \mathsf{poly}(\lambda, d, \log \ell)$.

4 Our MFC Construction

In this section, we put forward the detailed constructions of succinct mercurial functional commitments for Boolean circuits based on BASIS assumption. Then we show the correctness, mercurial binding, mercurial hiding, and succinctness of our constructions.

Construction 4.1 (MFC Based on BASIS). Let λ be a security parameter and $\mathcal{F}_\lambda$ be a family of functions f where each function $f : \{0,1\}^\ell \to \{0,1\}$ is on inputs of length $\ell = \ell(\lambda)$ and can be computed by a Boolean circuit of depth at most $d = d(\lambda)$. Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ be lattice parameters. Let $m' = n(\lceil \log q \rceil + 1)$, and $\beta = \beta(\lambda)$ be the bound. Let $s_0 = s_0(\lambda)$, $s_1 = s_1(\lambda)$, $s_2 = s_2(\lambda)$ be Gaussian width parameters. Denote $\mathbf{G}$ as the gadget matrix. The detailed construction is shown as follows:

- $\{\mathsf{crs}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$: Input a security parameter λ and a input length 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 $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ and a random vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. 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 $\mathbf{B}_\ell \in \mathbb{Z}_q^{n\ell\times(\ell m+m')}$ and $\tilde{\mathbf{R}} \in \mathbb{Z}_q^{(\ell m+m')\times \ell m'}$ as follows:
$$\mathbf{B}_l = \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}_l\mathbf{A} & -\mathbf{G} \end{array}\right], \qquad \tilde{\mathbf{R}} = \left[\begin{array}{c} \mathsf{diag}(\mathbf{R}_1, ..., \mathbf{R}_l) \\ \hline \mathbf{0}^{m'\times lm'} \end{array}\right] \tag{4.1}$$
After that, it samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_\ell, \tilde{\mathbf{R}}, \mathbf{G}_{n\ell}, s_0)$. It outputs the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ and the trapdoor key $tk = \mathbf{R}$ optionally.
- $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs}, \mathbf{x})$: Input the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ and a vector $\mathbf{x} \in \{0,1\}^l$, it first samples $\hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m\times m'}$ and computes $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \in \mathbb{Z}_q^{n\times m'}$. Next, it constructs $\mathbf{B}_\ell$ as in Eq. 4.1 and the target matrix $\mathbf{U}_\mathbf{x} \in \mathbb{Z}_q^{n\ell\times m'}$ and then uses $\mathbf{T}$ to sample the preimage as follows,
$$\mathbf{U}_\mathbf{x} = \begin{bmatrix} -x_1\mathbf{W}_1\mathbf{G} \\ \vdots \\ -x_\ell\mathbf{W}_\ell\mathbf{G} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_\ell \\ \hat{\mathbf{C}} \end{bmatrix} \leftarrow \mathsf{SampPre}\left(\mathbf{B}_l, \mathbf{T}, \mathbf{U}_\mathbf{x}, s_1\right) \tag{4.2}$$
Last, it computes $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}} \in \mathbb{Z}_q^{n\times m'}$. It outputs the hard commitment $(\mathbf{C}, \mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{\mathbf{x}, \mathbf{V}_1, ..., \mathbf{V}_\ell, (\mathbf{C}, \mathbf{D}), \hat{\mathbf{R}}\}$.

– $\pi \leftarrow \mathsf{HOpen}(\mathsf{crs}, f, \mathsf{aux})$: Input the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$, a function $f : \{0,1\}^l \to \{0,1\}$, and the auxiliary information $\mathsf{aux} = \{\mathbf{x}, \mathbf{V}_1, ..., \mathbf{V}_\ell, (\mathbf{C}, \mathbf{D}), \hat{\mathbf{R}}\}$. It first constructs $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}| \cdots |\mathbf{W}_l^{-1}\mathbf{C}] \in \mathbb{Z}_q^{n \times lm'}$, and computes $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$ and $\mathbf{V}_f = [\mathbf{V}_1| \cdots |\mathbf{V}_\ell] \cdot \mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}}$ where $\mathbf{H}_{\tilde{\mathbf{C}}, f, \mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$. Then, it constructs the trapdoor $\mathbf{R}_f = [-\mathbf{V}_f | \mathbf{0}^{m' \times m'} | \mathbf{I}_{m'}]^\mathsf{T}$ to sample the preimage as follows,

$$\mathbf{v}_f \leftarrow \mathsf{SampPre}([\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}], \mathbf{R}_f, \mathbf{u}, s_2)$$

where $\mathbf{D}$ actually equals $\mathbf{A}\hat{\mathbf{R}}$. It outputs the hard opening $\pi = \{\mathbf{v}_f, \hat{\mathbf{R}}\}$.

– $\{0,1\} \leftarrow \mathsf{HVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, y, \pi)$: Input the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$, the hard commitment $(\mathbf{C}, \mathbf{D})$, the function $f : \{0,1\}^\ell \to \{0,1\}$, the value $y \in \{0,1\}$ and the hard opening π. It first computes $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}| \cdots |\mathbf{W}_\ell^{-1}\mathbf{C}] \in \mathbb{Z}_q^{n \times \ell m'}$ and $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$. Then, it checks if the following conditions hold to verify the opening.

$$\|\mathbf{v}_f\| \leq \beta, \qquad \mathbf{u} = [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (y - 1) \cdot \mathbf{G}]\mathbf{v}_f \tag{4.3}$$

$$\|\hat{\mathbf{R}}\| \leq 1, \qquad \mathbf{D} = \mathbf{A}\hat{\mathbf{R}} \tag{4.4}$$

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

– $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{SCom}(\mathsf{crs})$: Input the common reference string crs, it first samples $\hat{\mathbf{C}} \leftarrow D_{\mathbb{Z}^{m' \times m'}, s_1}$ 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 the auxiliary information $\mathsf{aux} = \{(\mathbf{C}, \mathbf{D}), \hat{\mathbf{R}}\}$.

– $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{F}, f, y, \mathsf{aux})$: Input the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$, a flag $\mathbb{F} \in \{\mathbb{H}, \mathbb{S}\}$ which indicates that the soft opening τ is for hard commitment or soft commitment, a function $f : \{0,1\}^\ell \to \{0,1\}$, a value $y \in \{0,1\}$, and the auxiliary information aux.
If $\mathbb{F} = \mathbb{H}$ and y equals $f(\mathbf{x})$ where $\mathbf{x}$ is phased from aux, then it computes $\{\mathbf{v}_f, \hat{\mathbf{R}}\} \leftarrow \mathsf{HOpen}(\mathsf{crs}, f, \mathsf{aux})$ and outputs $\tau = \mathbf{v}_f$; If $y \neq f(\mathbf{x})$, it aborts and outputs $\perp$.
If $\mathbb{F} = \mathbb{S}$, it first computes $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}| \cdots |\mathbf{W}_\ell^{-1}\mathbf{C}] \in \mathbb{Z}_q^{n \times lm'}$ and $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$. Then, it constructs the trapdoor $\mathbf{R}_f = [\hat{\mathbf{R}} | \mathbf{I}_{m'} | \mathbf{0}^{m' \times m'}]^\mathsf{T}$ to sample the preimage as follows,

$$\mathbf{v}_f \leftarrow \mathsf{SampPre}([\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (y - 1) \cdot \mathbf{G}], \mathbf{R}_f, \mathbf{u}, s_2)$$

where $\mathbf{D}$ is phased from aux and actually equals $\mathbf{G} - \mathbf{A}\hat{\mathbf{R}}$. It outputs the soft opening $\tau = \mathbf{v}_f$.

– $\{0,1\} \leftarrow \mathsf{SVerify}(\mathsf{crs}, (\mathbf{C}, \mathbf{D}), f, y, \tau)$: Input the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$, the commitment $(\mathbf{C}, \mathbf{D})$, the function $f : \{0,1\}^\ell \to \{0,1\}$, the value $y \in \{0,1\}$, and soft opening τ. It first computes $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}| \cdots |\mathbf{W}_\ell^{-1}\mathbf{C}] \in \mathbb{Z}_q^{n \times \ell m'}$ and $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$, then check if Eq. 4.3 holds. If it holds, it outputs 1; Otherwise, it outputs 0.

– $\{(\mathbf{C},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$: Input the common reference string crs and trapdoor key tk. It first samples $\hat{\mathbf{C}} \leftarrow D_{\mathbb{Z}^{m'\times m'},s_1}$, $\hat{\mathbf{R}} \xleftarrow{\$} \{0,1\}^{m\times m'}$ and then computes $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}}$, $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$. It generates the fake commitment $(\mathbf{C},\mathbf{D})$ and the auxiliary information $\mathsf{aux} = \{(\mathbf{C},\mathbf{D}), \hat{\mathbf{R}}\}$.
– $\pi \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$: Input the common reference string crs, trapdoor key $tk = \mathbf{R}$, a function $f : \{0,1\}^\ell \to \{0,1\}$, a value $y \in \{0,1\}$, and the auxiliary information aux. It first computes $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}|\cdots|\mathbf{W}_\ell^{-1}\mathbf{C}] \in \mathbb{Z}_q^{n\times lm'}$ and $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$. Then, it constructs the trapdoor $\mathbf{R}_f = [\mathbf{R}|\mathbf{0}^{m'\times m'}|\mathbf{0}^{m'\times m'}]^\mathsf{T}$ to sample the preimage as follows,

$$\mathbf{v}_f \leftarrow \mathsf{SampPre}([\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (y-1)\cdot\mathbf{G}], \mathbf{R}_f, \mathbf{u}, s_2)$$

where $\mathbf{D}$ actually equals $\mathbf{A}\hat{\mathbf{R}}$. It outputs the hard equivocation $\pi = \{\mathbf{v}_f, \hat{\mathbf{R}}\}$.
– $\tau \leftarrow \mathsf{ESOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$: Input the common reference string crs and trapdoor key tk, the function $f : \{0,1\}^\ell \to \{0,1\}$, the value $y \in \{0,1\}$, and the auxiliary information aux, it computes $\mathbf{v}_f \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$. It outputs the soft equivocation $\tau = \mathbf{v}_f$.

Theorem 4.2 (Correctness). *For $n = \lambda$, $m = O(n\log q)$, $s_0 = O(\ell m^2 \log(ln))$, $s_1 = O(\ell^{3/2}m^{3/2}\log(n\ell)\cdot s_0)$, $s_2 = s_1\cdot m^{5/2}\ell^{3/2}\cdot(n\log q)^{O(d)}$, and $\beta = \sqrt{m+2m'}\cdot s_2$, then the Construction 4.1 is correct.*

Proof. Take a security parameter λ, a function $f \in \mathcal{F}_\lambda$ and an input $\mathbf{x} \in \{0,1\}^\ell$. Let $\{\mathsf{crs}, tk\} \leftarrow \mathsf{Setup}(1^\lambda, 1^\ell)$ where $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_l, \mathbf{T}, \mathbf{u}\}$. Let $\{(\mathbf{C},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs},\mathbf{x})$ and $\pi \leftarrow \mathsf{HOpen}(\mathsf{crs}, f, \mathsf{aux})$. Let $\{(\mathbf{C},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{SCom}(\mathsf{crs})$ and $\tau \leftarrow \mathsf{SOpen}(\mathsf{crs}, \mathbb{F}, f, y, \mathsf{aux})$. Let $\{(\mathbf{C},\mathbf{D}),\mathsf{aux}\} \leftarrow \mathsf{FCom}(\mathsf{crs}, tk)$, $\pi \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$, and $\tau \leftarrow \mathsf{EHOpen}(\mathsf{crs}, tk, f, y, \mathsf{aux})$. Consider $\mathsf{HVerify}(\mathsf{crs}, (\mathbf{C},\mathbf{D}), f, y, \pi)$ and $\mathsf{SVerify}(\mathsf{crs}, (\mathbf{C},\mathbf{D}), f, y, \tau)$:

Following the same parameters and constructions of $\mathbf{B}_\ell$ and $\tilde{\mathbf{R}}$ in BASIS [23], we have $\|\mathbf{T}\| \leq \sqrt{\ell m + m'}\cdot s_0$. By our construction and Theorem 2.1, we also have $\|\hat{\mathbf{R}}\| = 1$ and $\|\mathbf{R}\| = 1$. We prove the correctness of our proposed construction from the following aspects.

For hard commitment. Suppose $s_1 \geq \sqrt{(\ell m + m')\ell m'}\cdot\|\mathbf{T}\|\cdot\omega(\sqrt{\log(n\ell)}) = O(l^{3/2}m^{3/2}\log(n\ell)\cdot s_0)$, by Theorem 2.1 and construction of $(\mathbf{V}_1, ..., \mathbf{V}_\ell, \mathbf{C})$ in Eq. 4.2, for each $i \in [\ell]$, we have

$$\mathbf{W}_i\mathbf{A}\mathbf{V}_i - \mathbf{C} = -x_i\mathbf{W}_i\mathbf{G}$$

where $\mathbf{C} = \mathbf{G}\hat{\mathbf{C}}$. As well as $\mathbf{A}\mathbf{V}_i - \mathbf{W}_i^{-1}\mathbf{C} = -x_i\mathbf{G}$ for each $i \in [\ell]$. Let $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}|\cdots|\mathbf{W}_\ell^{-1}\mathbf{C}]$ and $\tilde{\mathbf{V}} = [\mathbf{V}_1|\cdots|\mathbf{V}_\ell]$. We have

$$\tilde{\mathbf{C}} - \mathbf{x}^\mathsf{T} \otimes \mathbf{G} = \mathbf{A}\cdot[\mathbf{V}_1|\cdots|\mathbf{V}_\ell] = \mathbf{A}\cdot\tilde{\mathbf{V}} \tag{4.5}$$

Let $\beta_0 = \sqrt{\ell m + m'}\cdot s_1$ be the "initial" noise bound. So $\|\mathbf{V}_i\| \leq \sqrt{\ell m + m'}\cdot s_1 = \beta_0$ (by Lemma 1 in [15]), and thus $\|\tilde{\mathbf{V}}\| \leq \beta_0$.

By construction, we have $\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} \leftarrow \mathsf{EvalFX}(\tilde{\mathbf{C}}, f, \mathbf{x})$ and $\mathbf{V}_f = \tilde{\mathbf{V}}\cdot\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}$ where by Theorem 2.6, $\|\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}}\| \leq (n\log q)^{O(d)}$. By our notation of norm of matrix i.e. $\|\mathbf{X}\| := \max_{i,j}|X_{i,j}|$, so that we have $\|\mathbf{V}_f\| \le \ell m' \cdot \beta_0 \cdot (n\log q)^{O(d)} \le \ell m' \cdot \sqrt{\ell m + m'} \cdot s_1 \cdot (n \log q)^{O(d)}$. Thanks to Theorem 2.6 again and according to Eq. 4.5, we have

$$\mathbf{A}\mathbf{V}_f = \mathbf{A}\tilde{\mathbf{V}}\mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = (\tilde{\mathbf{C}} - \mathbf{x}^\mathsf{T} \otimes \mathbf{G}) \cdot \mathbf{H}_{\tilde{\mathbf{C}},f,\mathbf{x}} = \tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G} \tag{4.6}$$

where $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$.

Let $\mathbf{D}_f = [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (f(\mathbf{x})-1)\cdot\mathbf{G}] \in \mathbb{Z}_q^{n\times(m+2m')}$ where $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$, and $\mathbf{R}_f = [-\mathbf{V}_f|\mathbf{0}^{m'\times m'}|\mathbf{I}_{m'}]^\mathsf{T} \in \mathbb{Z}_q^{(m+2m')\times m'}$. Thus, $\|\mathbf{R}_f\| = \|\mathbf{V}_f\| \le \ell m' \cdot \sqrt{\ell m + m'} \cdot s_1 \cdot (n\log q)^{O(d)}$ and by Eq. 4.6, we have

$$\mathbf{D}_f\mathbf{R}_f = -\mathbf{A}\mathbf{V}_f + \tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G} = (2f(\mathbf{x}) - 1)\mathbf{G} \in \{-\mathbf{G}, \mathbf{G}\}$$

Thus, $\mathbf{R}_f$ is a gadget trapdoor for $\mathbf{D}_f$ (with tag $\mathbf{I}_n$ or $-\mathbf{I}_n$, depending on the value of $f(\mathbf{x}) \in \{0, 1\}$). Suppose $m \ge m' = O(n \log q)$ and

$$s_2 \ge \sqrt{(m + 2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = s_1 \cdot m^{5/2} \cdot \ell^{3/2} \cdot (n\log q)^{O(d)}$$

For soft commitment. By our construction, $\mathbf{D}_f = [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n\times(m+2m')}$ where $\mathbf{D} = \mathbf{G} - \mathbf{A}\hat{\mathbf{R}}$, and $\mathbf{R}_f = [\hat{\mathbf{R}}|\mathbf{I}_{m'}|\mathbf{0}^{m'\times m'}]^\mathsf{T} \in \mathbb{Z}_q^{(m+2m')\times m'}$. Then, we have $\|\mathbf{R}_f\| = 1$ and $\mathbf{D}_f\mathbf{R}_f = \mathbf{G}$. Thus, $\mathbf{R}_f$ is a gadget trapdoor for $\mathbf{D}_f$. Suppose $m \ge m' = O(n \log q)$ and

$$s_2 \ge \sqrt{(m + 2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = O(m \log n)$$

For fake commitment. By our construction, $\mathbf{D}_f = [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f + (f(\mathbf{x}) - 1) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n\times(m+2m')}$ where $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$, and $\mathbf{R}_f = [\mathbf{R}|\mathbf{0}^{m'\times m'}|\mathbf{0}^{m'\times m'}]^\mathsf{T} \in \mathbb{Z}_q^{(m+2m')\times m'}$. Then, we have $\|\mathbf{R}_f\| = 1$ and $\mathbf{D}_f\mathbf{R}_f = \mathbf{G}$. Thus, $\mathbf{R}_f$ is a gadget trapdoor for $\mathbf{D}_f$. Suppose $m \ge m' = O(n \log q)$ and

$$s_2 \ge \sqrt{(m + 2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = O(m \log n)$$

Overall, for each opening $\mathbf{v}_f \leftarrow \mathsf{SampPre}(\mathbf{D}_f, \mathbf{R}_f, \mathbf{u}, s_2)$ from hard commitment, soft commitment, and fake commitment, by Theorem 2.1, it must satisfy $\mathbf{D}_f\mathbf{v}_f = \mathbf{u}$ and $\|\mathbf{v}_f\| \le \sqrt{m + 2m'} \cdot s_2 \le \beta$ so that the verification algorithms will accept with overwhelming probability. □

Theorem 4.3 (Mercurial Binding). *For any polynomial $\ell = \ell(\lambda)$, $n = \lambda$, $m = O(n \log q)$, $s_0 = O(\ell m^2 \log(nl))$, $s_1 = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot s_0)$. Under the* BASIS *assumption with parameters $(n, m, q, \beta', s_0, \ell)$ where $\beta' = s_1 \cdot m^{5/2}\ell^{3/2} \cdot \beta \cdot (n \log q)^{O(d)}$, Construction 4.1 satisfies mercurial (target) binding.*

Proof. Considering that our construction is a *proper* MFC where the hard opening contains its corresponding soft opening as a proper subset. Thus, we only focus on 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 $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(1^n, q, m)$ and $\mathbf{W}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times n}$ for each $i \in [\ell]$. Then it constructs $\tilde{\mathbf{R}}$ and $\mathbf{B}_l$ following the Eq. 4.1.

  It samples $\mathbf{T} \leftarrow \mathsf{SampPre}(\mathbf{B}_\ell, \tilde{\mathbf{R}}, \mathbf{G}_{n\ell}, s_0)$ and $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. Last, the challenger sends the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ to the adversary $\mathcal{A}$.
- The adversary $\mathcal{A}$ chooses an input vector $\mathbf{x} \in \{0,1\}^\ell$.
- The challenger gives $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs}, \mathbf{x})$ to $\mathcal{A}$.
- The adversary $\mathcal{A}$ outputs a function $f \in \mathcal{F}_\lambda$ and an openings $\mathbf{v}_f$ to the value $1 - f(\mathbf{x})$.
- The output of the experiments is 1 if it satisfies the following conditions:

$$\|\mathbf{v}_f\| \leq \beta, \qquad [\mathbf{A}|\mathbf{D}|\tilde{\mathbf{C}}_f - f(\mathbf{x}) \cdot \mathbf{G}]\mathbf{v}_f = \mathbf{u} \tag{4.7}$$

  where $\mathbf{D} = \mathbf{A}\hat{\mathbf{R}}$, $\|\hat{\mathbf{R}}\| \leq 1$, $\tilde{\mathbf{C}}_f \leftarrow \mathsf{EvalF}(\tilde{\mathbf{C}}, f)$, and $\tilde{\mathbf{C}} = [\mathbf{W}_1^{-1}\mathbf{C}| \cdots |\mathbf{W}_l^{-1}\mathbf{C}]$; Otherwise, the experiments output 0.

– Hyb_1: Same as Hyb_0 except the challenger samples $\mathbf{T} \leftarrow (\mathbf{B}_\ell)_{s_0}^{-1}(\mathbf{G}_{n\ell})$ without using the trapdoor $\tilde{\mathbf{R}}$ so the common reference string crs is sampled independently of $\mathbf{R}$.

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

– Hyb_3: Same as Hyb_2 except the challenger samples $\mathbf{u} \leftarrow \mathbf{A}\mathbf{r}$ where $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$.

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 $\mathsf{Hyb}_0(\mathcal{A}) \approx \mathsf{Hyb}_1(\mathcal{A}) \approx \mathsf{Hyb}_2(\mathcal{A}) \approx \mathsf{Hyb}_3(\mathcal{A})$ because they are given in [23] (Lemma 4.26∼4.28) and same as ours. We now analyze the last step.

Lemma 4.4. *Suppose the conditions on n, m, s_0, s_1 in Theorem 4.3 hold and $m \geq n \log q + \lambda$. Let $\beta' = s_1 \cdot m^{5/2}\ell^{3/2} \cdot \beta \cdot (n \log q)^{O(d)}$. Then, under the* BASIS *assumption with parameters $(n, m, q, \beta', s_0, \ell)$, 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. And an algorithm $\mathcal{B}$ will use $\mathcal{A}$ to break the BASIS assumption.

Algorithm $\mathcal{B}$ first receives the challenge $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{B}_\ell \in \mathbb{Z}_q^{nl \times (\ell m + m')}$, $\mathbf{T} \in \mathbb{Z}_q^{(\ell m + m') \times \ell m'}$ and $\mathsf{aux} = (\mathbf{W}_1, ..., \mathbf{W}_\ell)$, Then $\mathcal{B}$ samples $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$, computes $\mathbf{u} = \mathbf{A}\mathbf{r}$, and sends the common reference string $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ and to $\mathcal{A}$. The adversary $\mathcal{A}$ outputs a vector $\mathbf{x} \in \{0,1\}^\ell$ to $\mathcal{B}$. Algorithm $\mathcal{B}$ computes $\{(\mathbf{C}, \mathbf{D}), \mathsf{aux}\} \leftarrow \mathsf{HCom}(\mathsf{crs}, \mathbf{x})$ and sends $(\mathbf{C}, \mathbf{D})$ and aux to $\mathcal{A}$. The adversary $\mathcal{A}$ can output a function $f \in \mathcal{F}_\lambda$ and an opening $\mathbf{v}_f \in \mathbb{Z}_q^{m+2m'}$ satisfying Eq. 4.7. By Eq. 4.6 and $\mathbf{u} = \mathbf{A}\mathbf{r}$, we have

$$\mathbf{u} = \mathbf{A}\mathbf{r} = [\mathbf{A}|\mathbf{A}\hat{\mathbf{R}}|\tilde{\mathbf{C}}_f - f(x) \cdot \mathbf{G}]\mathbf{v}_f = [\mathbf{A}|\mathbf{A}\hat{\mathbf{R}}|\mathbf{A}\mathbf{V}_f]\mathbf{v}_f$$

Let $\mathbf{z} = [\mathbf{I}_m|\hat{\mathbf{R}}|\mathbf{V}_f]\mathbf{v}_f - \mathbf{r}$ so that we have $\mathbf{A}\mathbf{z} = \mathbf{0}$. We now show $0 < \|\mathbf{z}\| \leq \beta'$ in the following two aspects:

– We show $\|\mathbf{z}\| \le \beta'$. Since $\|\hat{\mathbf{R}}\| = 1$, $\|\mathbf{V}_f\| \le \ell m' \cdot \sqrt{\ell m + m'} \cdot s_1 \cdot (n \log q)^{O(d)}$, $\|\mathbf{v}_f\| \le \beta$, $\|\mathbf{r}\| = 1$, and $m > m'$, we have that

$$\|\mathbf{z}\| \le \ell m' \cdot \sqrt{\ell m + m'} \cdot s_1 \cdot (n \log q)^{O(d)} \cdot \beta \cdot (m + 2m') + 1 \le s_1 \cdot m^{5/2} \ell^{3/2} \cdot \beta \cdot (n \log q)^{O(d)}$$

where $s_1 \cdot m^{5/2} \ell^{3/2} \cdot \beta \cdot (n \log q)^{O(d)} = \beta'$.

– We show $\|\mathbf{z}\| \ne 0$, i.e. $\mathbf{r} \ne [\mathbf{I}_m|\hat{\mathbf{R}}|\mathbf{V}_f]\mathbf{v}_f$. Following the same entropy argument as in [11] (Theorem 3.1), by our construction, $[\mathbf{I}_m|\hat{\mathbf{R}}|\mathbf{V}_f]\mathbf{v}_f$ is a function of $\mathbf{u} \in \mathbb{Z}_q^n$ (and other quantities that are independent of $\mathbf{r}$). By construction, $\mathbf{u}$ contains at most $n \log q$ bits of information about $\mathbf{r}$. It leads that

$$\mathbf{H}_\infty(\mathbf{r} \mid [\mathbf{I}_m|\hat{\mathbf{R}}|\mathbf{V}_f]\mathbf{v}_f) \ge \mathbf{H}_\infty(\mathbf{r} \mid \mathbf{u}) \ge m - n \log q \ge \lambda$$

It means that $\Pr[\mathbf{r} = [\mathbf{I}_m|\hat{\mathbf{R}}|\mathbf{V}_f]\mathbf{v}_f] \le 2^{-\lambda}$.

Overall, $\mathbf{z}$ is a valid solution for $\mathcal{B}$ to break the BASIS assumption with non-negligible probability $\epsilon - 2^{-\lambda}$. □

By lemmas in [23] and Lemma 4.4, we conclude that for all efficient adversaries $\mathcal{A}$, $\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] \le \mathsf{negl}(\lambda)$. Therefore, mercurial (target) binding holds. □

Theorem 4.5 (Mercurial Hiding). *For $n = \lambda$, $m = O(n \log q)$, q is prime, $s_0 = O(\ell m^2 \log(\ell n))$, $s_1 = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot s_0)$, $s_2 = s_1 \cdot m^{5/2} \ell^{3/2} \cdot (n \log q)^{O(d)}$, then Construction 4.1 satisfies statistical* HHEquivocation*,* HSEquivocation*, and* SSEquivocation*.*

Proof. The Challenger first sets up the scheme and obtains the common reference string crs $= \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ via the real protocol, and $tk = \mathbf{R}$ is the trapdoor. Then we prove the mercurial hiding of our proposed construction in the equivocation games.

For HHEquivocation. Firstly, $\mathbf{D}$ and $\mathbf{R}$ are generated in the same way for both fake and hard commitments. By Theorem 4.2 and Theorem 2.1, since $s_2 \ge \sqrt{(m + 2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = s_1 \cdot m^{5/2} \cdot \ell^{3/2} \cdot (n \log q)^{O(d)}$ in hard opening and $s_2 \ge \sqrt{(m + 2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = O(m \log n)$ in hard equivocation, the distributions of $\mathbf{v}_f \leftarrow \mathsf{SampPre}(\mathbf{D}_f, \mathbf{R}_f, \mathbf{u}, s_2)$ from both hard opening and hard equivocation are statistically close to the distribution of $\mathbf{v}_f \leftarrow (\mathbf{D}_f)_{s_2}^{-1}(\mathbf{u})$.

Then, by Theorem 2.1, if $s_1 \ge \sqrt{(\ell m + m')\ell m'} \|\mathbf{T}\| \cdot \omega(\sqrt{\log(n\ell)}) = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot s_0)$, the distribution of of $\{\mathbf{V}_1, ..., \mathbf{V}_\ell, \hat{\mathbf{C}}\} \leftarrow \mathsf{SampPre}(\mathbf{B}_\ell, \mathbf{T}, \mathbf{U}_\mathbf{x}, s_1)$ in hard commitment is statistically close to the distribution $(\mathbf{B}_\ell)_{s_1}^{-1}(\mathbf{U}_\mathbf{x})$ where the target vector $\mathbf{U}_\mathbf{x}$ is the same as Eq. 4.2.

Let $\bar{\mathbf{A}} = \mathsf{diag}(\mathbf{W}_1\mathbf{A}, ..., \mathbf{W}_\ell\mathbf{A})$, then $\mathbf{B}_\ell = [\bar{\mathbf{A}}| - 1^\ell \otimes \mathbf{G}]$.Since $s_1 \ge \log(\ell m)$, by the distribution of discrete Gaussian preimages (Lemma 2.4 [21]), the distribution of $\{\mathbf{V}_1, ..., \mathbf{V}_\ell, \hat{\mathbf{C}}\} \leftarrow (\mathbf{B}_\ell)_{s_1}^{-1}(\mathbf{U}_\mathbf{x})$ is statistically close to the distribution

$$\left\{\hat{\mathbf{C}} \leftarrow D_{\mathbb{Z}^{m' \times m'}, s_1}, \{\mathbf{V}_1, ..., \mathbf{V}_\ell\} \leftarrow \bar{\mathbf{A}}_{s_1}^{-1}\left(\mathbf{U}_\mathbf{x} + (1^\ell \otimes \mathbf{G}\hat{\mathbf{C}})\right)\right\}$$

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

Overall, these lead to fake commitments and hard equivocation having exactly the same distribution as hard commitments and hard openings.

For HSEquivocation. Follow the same arguments as $\mathsf{HHEquivocation}$.

For SSEquivocation. We note that $\hat{\mathbf{C}}$ are generated in the same way for both fake and soft commitments. By the well-known Leftover Hash Lemma [12], 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 $\mathbb{Z}_q^{n \times m'}$ (Lemma 2.3 in [21]).

Thus, the adversary's view remains statistically the same if we generate $\mathbf{D}$ in fake commitments from SCom instead of FCom in the ideal experiment. Moreover, by Theorem 2.1, since $s_2 \geq \sqrt{(m+2m')m'} \cdot \|\mathbf{R}_f\| \cdot \omega(\sqrt{\log n}) = O(m \log n)$ in both soft commitment and fake commitment, the distribution of $\mathbf{v}_f \leftarrow \mathsf{SampPre}\ ([\mathbf{A}|\mathbf{D}'|\tilde{\mathbf{C}}_f + (y-1) \cdot \mathbf{G}], \mathbf{R}_f, \mathbf{u}, s_2)$ in the soft opening and the distribution of $\mathbf{v}_f \leftarrow \mathsf{SampPre}([\mathbf{A}|\mathbf{D}'|\tilde{\mathbf{C}}_f + (y-1) \cdot \mathbf{G}], \mathbf{R}_f, \mathbf{u}, s_2)$ in the soft equivocation are both statistically close to $([\mathbf{A}|\mathbf{D}'|\tilde{\mathbf{C}}_f + (y-1) \cdot \mathbf{G}])_{s_2}^{-1}(\mathbf{u})$. These lead to fake commitments and soft equivocation having exactly the same distribution as soft commitments and their corresponding soft openings. □

Remark 4.6 (Parameter Instantiation). Let λ be the security parameter and $\mathcal{F}_\lambda$ be a family of functions $f : \{0,1\}^\ell \to \{0,1\}$ on inputs of length $\ell = \ell(\lambda)$ which can be computed by Boolean circuits of depth at most $d = d(\lambda)$. We provide the parameter instance of Construction 4.1.

- Let $\epsilon > 0$ be a constant, the lattice dimension be $n = d^{1/\epsilon} \cdot \mathsf{poly}(\lambda)$ and $m = O(n \log q)$.
- Let the Gaussian parameters be $s_0 = O(\ell m^2 \log(n\ell))$, $s_1 = O(\ell^{3/2} m^{3/2} \log(n\ell) \cdot s_0) = O(\ell^{5/2} m^{7/2} \log^2(n\ell))$, and $s_2 = s_1 \cdot m^{5/2} \ell^{3/2} \cdot (n \log q)^{O(d)} = \ell^4 \log^2 \ell \cdot (n \log q)^{O(d)}$
- Let the bound be $\beta = s_2 \cdot \sqrt{m + 2m'} = \ell^4 \log^2 \ell \cdot (n \log q)^{O(d)}$, $\beta' = s_1 \cdot m^{5/2} \ell^{3/2} \cdot \beta \cdot (n \log q)^{O(d)} = 2^{\tilde{O}(d)} = 2^{\tilde{O}(n^\epsilon)}$ where $\tilde{O}(\cdot)$ is denoted to suppress polylogarithmic factors in λ, d, ℓ.
- Let the modulus be $q = \beta' \cdot \mathsf{poly}(n)$ in the BASIS assumption with parameters $(n, m, q, \beta', s_0, \ell)$. Then $\log q = \mathsf{poly}(d, \log \lambda, \log \ell)$. Note that the BASIS assumption as well as SIS assumption relies on a *sub-exponential* noise bound.

Remark 4.7 (Succinctness). Following the parameter instance in Remark 4.6, we show the succinctness of Construction 4.1.

- Commitment size: A commitment to a vector $\mathbf{x} \in \{0,1\}^\ell$ is $(\mathbf{C}, \mathbf{D}) \in \mathbb{Z}_q^{n \times m'} \times \mathbb{Z}_q^{n \times m'}$ where
$$|\mathbf{C}| = |\mathbf{D}| = nm' \log q = O(n^2 \log^2 q) = \mathsf{poly}(\lambda, d, \log \ell)$$
- Opening size: A (hard) opening is $(\mathbf{v}_f, \hat{\mathbf{R}}) \in \mathbb{Z}_q^{m+2m'} \times \mathbb{Z}_q^{m \times m'}$ where
$$|\mathbf{v}_f| = (m + 2m') \log \beta = O(nd \cdot \log q \cdot \log \ell \cdot \log \lambda) = \mathsf{poly}(\lambda, d, \log \ell)$$
$$|\hat{\mathbf{R}}| = mm' = O(n^2 \cdot \log^2 q) = \mathsf{poly}(\lambda, d, \log \ell)$$

– Common reference string size: The common reference string is $\mathsf{crs} = \{\mathbf{A}, \mathbf{W}_1, ..., \mathbf{W}_\ell, \mathbf{T}, \mathbf{u}\}$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{W}_i \in \mathbb{Z}_q^{n\times n}$, $\mathbf{T} \in \mathbb{Z}_q^{(\ell m+m')\times \ell m'}$, and $\mathbf{u} \in \mathbb{Z}_q^n$, where

$$|\mathsf{crs}| = nm\log q + \ell n^2 \log q + (\ell m+m')(\ell m')\log q + n\log q = \ell^2 \cdot \mathsf{poly}(\lambda, d, \log \ell)$$

Therefore, Construction 4.1 is succinct.

5 Application: Lattice-Based ZK-FEDB

The main application of MCs is to build ZKS, ZK-EDB, and ZK-FEDB. ZKS allows a set owner to prove the membership (and non-membership) of an element x for a set S and ZK-EDB extends the set to an elementary database D containing key-value pairs (x, v) which others can query the key x. Different from them, ZK-FEDB [24,25] allows the database owner to provide the proof to the function value $f(x, v)$ or non-membership after the users query the key x with some function f to the elementary database. Due to the limitation of existing MFC, the ZK-FEDB [24] constructed by MFC only supports linear function queries. The most general ZK-FEDB was first proposed by Zhang and Deng [25] using an RSA accumulator and set-operation instead of MFC which allows the user to query the key with Boolean circuits.

However, all existing constructions of ZK-FEDB cannot resist the quantum computer attack. The current lattice-based constructions of MC and MVC [15,21] can only be used to build the lattice-based ZKS and ZK-EDB and does not suffice to construct the ZK-FEDB, i.e. allowing users to make function queries, especially for Boolean circuit queries.

In this section, we illustrate how to use our construction to build the *first* lattice-based ZK-FEDB in the generic framework [24] at a high level.

Normally, there are three phases in the ZK-FEDB: the *committing* phase, the *opening* phase, and the *verification* phase: (1) In the *committing* phase, the committer will build a binary (or N-ary) tree where the leaf nodes are indexed by the keys in the elementary database and the root as the database's commitment. Thanks to the *mercurial* property, it can use a soft commitment to prune (replace) the subtrees without any leaves (keys) in the database to enhance efficiency. After that, only the subtrees with at least one leaf node in the database are kept. For the leaf node whose level equals the height of the whole tree, and if its index (key) is in the database, i.e. $D(x) \neq \perp$, the leaf node contains a hard commitment of input $(x, D(x)) \in \{0,1\}^\ell$ generated by our MFC, otherwise it contains a soft commitment produced by our MFC; for other leaf nodes, i.e. their level is less than the height of the tree, they contain soft commitments generated by the standard lattice-based MVC [21] or MC [15]. The remaining nodes in the tree, i.e. internal node, will contain a hard commitment to their children nodes generated by the same lattice-based MVC or MV as above. The commitment in the root node is the final commitment to the database. (2) In the *opening* phase, to prove that some key x is in the database and the output of a Boolean circuit $f \in \mathcal{F}_\lambda$ is $f(x, D(x))$, the committer generates a proof of membership, including all the hard openings for the commitments in the internal nodes on the path from the root to the leaf x and the hard opening for the commitment in the leaf node x to the Boolean circuit f; To prove the non-membership, i.e. $D(x) = \perp$ (we can treat $\perp$ as 0 in this case), the committer first generates the subtree which x lies and is pruned before. Then it generates the proof, including all the soft openings for the commitment in the internal nodes on the path from the root to the leaf x and the soft opening for the soft commitment in the leaf node x to the function f and value $f(x, \perp)$. (3) In the *verification* phase, the users will check all the commitments and openings of internal nodes and the leaf node on the path from the leaf x to the root.

Overall, our constructions of MFC can be used to build the *first* lattice-based ZK-FEDB. Compared to the existing ZK-FEDBs, our construction not only enables the database owner to commit the elementary database, generates a convinced answer to the query of a Boolean circuit on some key, and allows the users to verify the answer without leaking any knowledge except the query result, but also can achieve the security at a *post-quantum* level.

References

1. 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)
2. 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)
3. 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)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)

9. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23. pp. 523–540. Springer (2004)
10. 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)
11. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: Proceedings of the forty-seventh annual ACM symposium on Theory of computing. pp. 469–477 (2015)
12. 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)
13. 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)
14. 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)
15. 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)
16. 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)
17. 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)
18. 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)
19. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Eurocrypt. vol. 7237, pp. 700–718. Springer (2012)
20. 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)
21. Wang, H., Yiu, S.M., Zhao, Y., Jiang, Z.L.: Updatable, aggregatable, succinct mercurial vector commitment from lattice. In: Tang, Q., Teague, V. (eds.) Public-Key Cryptography – PKC 2024. pp. 3–35. Springer Nature Switzerland, Cham (2024)
22. Wee, H., Wu, D.J.: Lattice-based functional commitments: Fast verification and cryptanalysis. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 201–235. Springer Nature Singapore, Singapore (2023)

23. 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)
24. Wu, C., Chen, X., Susilo, W.: Concise id-based mercurial functional commitments and applications to zero-knowledge sets. International Journal of Information Security **19**(4), 453–464 (2020)
25. Zhang, X., Deng, Y.: Zero-knowledge functional elementary databases. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 269–303. Springer Nature Singapore, Singapore (2023)