How to Hash and Commit Real Values with Dynamic Precision

Binyi Chen* Yilei Chen† Zhenghao Guan‡ Wenjie Li§

November 26, 2026

Abstract

When hashing or committing real values (represented by floating-point numbers), traditional cryptographic hash functions and commitment schemes have to first convert the real values to integers or finite ring elements. Such solutions suffer from several drawbacks. The schemes do not support dynamic choices of precisions. More generally, hashing or committing algorithms lose the efficiency and flexibility of directly working on real numbers.

In this paper, we define and construct cryptographic hash functions and commitment schemes with dynamic precision choices and native real number arithmetic. Our main contributions are:

1. Providing the definitions of the real value collision resistant hash function (RVCRH) and commitment for real numbers, formalizing the notions of dynamic precision choices in the hashing, committing, and opening algorithms.
2. Defining the continuous analogs of the Short Integer Solution Problem (CSIS) and proving their hardness based on the worst-case hardness of lattice problems. We then construct RVCRHs and compressing commitment schemes with dynamic precision and native real arithmetic from CSIS problems.
3. Constructing a statistically binding additively homomorphic commitment scheme that supports real number arithmetic from the Continuous Learning With Errors Problem (CLWE). As an application, we provide a Sigma protocol for proving the knowledge of real numbers from CLWE.

We also show that additively homomorphic commitment schemes for real values are unlikely to be compressing by showing a partial impossibility result for achieving such schemes.

*IIIS, Tsinghua University by-chen@mail.tsinghua.edu.cn.

†IIIS, Tsinghua University and Shanghai Qi Zhi Institute. chenyilei@mail.tsinghua.edu.cn. Supported by Shanghai Qi Zhi Institute Innovation Program.

‡IIIS, Tsinghua University guanzh24@mails.tsinghua.edu.cn.

§IIIS, Tsinghua University and Shanghai Qi Zhi Institute. liwj22@mails.tsinghua.edu.cn.

Contents

1 Introduction **1**
- 1.1 Our Contributions . . . 2
- 1.2 Further Related Work . . . 4

2 Technical Overview **4**
- 2.1 Hashing and Commitments for Real Values: Definitions . . . 4
- 2.2 Hashing and Commitments for Real Values: Constructions . . . 5
- 2.3 Homomorphic Commitments from Continuous LWE . . . 8
- 2.4 On the Existence of Homomorphic and Compressing Commitments . . . 10
- 2.5 AI Disclosure . . . 11

3 Hash and Commitment for Real Values: Definitions **11**
- 3.1 Notations . . . 11
- 3.2 Real Value Arithmetic . . . 12
- 3.3 Real Value Commitments . . . 12
- 3.4 Real Value Collision Resistant Hash . . . 13

4 Average-case Hard Problems on Real Values **14**
- 4.1 Lattice Problems . . . 14
- 4.2 CLWE . . . 14
- 4.3 CSIS: Standard worst-case to average-case reduction . . . 15
- 4.4 CSIS′: A variant of CSIS without modulo 1 . . . 18
- 4.5 From CLWE to CSIS′ . . . 21

5 Real Value Collision Resistant Hash Functions: Constructions **22**
- 5.1 Base Expansion . . . 24
- 5.2 Hash Functions with Iterative Compression . . . 25
- 5.3 Hash Functions with Decaying Weights . . . 27
- 5.4 Hash Functions with Base Shifting . . . 29

6 Constructing RVCs from RVCRHs **33**
- 6.1 Commitment Schemes from Hash Functions with Iterative Compression . . . 33
- 6.2 Commitment Schemes from Hash Functions with Decaying Weights . . . 34
- 6.3 Commitment Schemes from Hash Functions Natively on Real Values . . . 36

7 Real Value Commitment from Continuous LWE **41**
- 7.1 Commitment Scheme . . . 41
- 7.2 Computationally Hiding . . . 42
- 7.3 Statistically Binding . . . 43

8 Sigma Protocol from CLWE Commitment **46**

9 A Generic Method of Finding Collisions **50**
- 9.1 Smooth Functions . . . 51
- 9.2 Additively Homomorphic Functions . . . 52

A Proof of Theorem 6.11 **56**

B Handling computation error when arguing the hiding property of the hash-based commitment scheme **58**

C Handling computation error when arguing the hiding property of the CLWE-based commitment scheme **59**

1 Introduction

A cryptographic commitment scheme allows a sender to commit to a value while keeping it hidden, and later to reveal that value without changing it after the commitment phase. Commitments are fundamental in cryptography and are widely used in zero-knowledge proofs [GMR85, Sch90, CDS94], secure multi-party computation [GMW87], and blockchain protocols [BSCG+14].

However, existing commitment schemes are designed for **discrete values** such as integers, bits, or finite field elements. In many applications, particularly in machine learning and scientific computing, we need to commit to vectors of **real** numbers. For example, a model owner may wish to commit to trained weights and later prove properties such as differential privacy [BTFM+26, PGL+26], fairness [ZDK+25, YCBC24], or other model guarantees [TGM+25]. How can we commit to a vector of real numbers and prove meaningful statements about it?

State of the Art: Quantization. Many cryptographic constructions handle real-valued data by first quantizing them and then applying commitment schemes designed for discrete domains. For example, zero-knowledge proof systems for machine-learning inference [KHSS22, GJJZ22, CWSK24, GGJ+23, APKP24] encode quantized values in a cryptographically large finite field or ring and implement numerical computations through arithmetic in that domain. This approach reuses well-established cryptographic tools, but it also couples the precision of numerical data with the specification of the cryptographic schemes.

This coupling complicates applications in which precision requirements evolve over time or differ across platforms. Although a sufficiently large field can accommodate multiple fixed-point formats, changing precision is not simply a matter of retaining more digits: it may require changing the encoding and modifying the arithmetic constraints used to verify computations. Moreover, *a commitment to a value at one precision does not automatically provide a way to open that same commitment at another precision*. What is missing is a precision-flexible interface in which commitments remain meaningful as the numerical precision changes.

Quantization also limits what binding can guarantee about the original real-valued input. If a scheme commits to a value with precision δ, then distinct inputs in the same quantization cell have identical encodings. Binding of the underlying discrete commitment therefore only binds the committer to the quantized value. Hence, a commitment created for a fixed precision δ provides no binding guarantee for inputs at a finer precision.

A naive solution is to choose a maximum supported precision in advance and commit to the full representation of every value. Lower-precision outputs can then be derived from that representation. However, this approach can force low-precision tasks to incur the costs of high-precision representations and arithmetic. Moreover, the discrete commitments introduce costs unrelated to the precision actually needed by the application: For instance, computing a Pedersen commitment to a vector of length n requires an n-term multi-scalar multiplication in an elliptic curve group of order $\approx 2^{256}$. These operations are more expensive than the native floating-point units on modern CPUs and GPUs.

These limitations motivate the following question:

> *Can we design a commitment scheme for real-valued data that supports dynamic precision and distance-sensitive binding, while avoiding processing every value at a fixed maximum precision in a large algebraic domain?*

1.1 Our Contributions

We answer this question in the affirmative by developing definitions, hardness foundations, and constructions for commitments to real-valued data. Our results support dynamically chosen precision and precision-sensitive security guarantees. Our contributions are as follows.

Definitions of real value hashes and commitments. In Section 3, we introduce **real value commitments** (RVCs) (Definition 3.1) and **real value collision-resistant hash functions** (RVCRHs) (Definition 3.3). These definitions make numerical precision an explicit part of the functionality and security guarantees, rather than fixing it implicitly through a discrete encoding.

Average-Case Hard Problems over Real-Valued Domains. To build a computational foundation for our constructions, Section 4 formalizes two continuous variants of the Small Integer Solution problem: Continuous SIS (CSIS) and Continuous SIS without modulo 1 (CSIS'). Both ask for a short, nonzero integer vector whose image under a real-valued matrix is close to zero; they differ in whether the image is measured modulo 1.

Definition 1.1 (Continuous Small Integer Solution (CSIS))**.** *The CSIS problem is: given a matrix* $\mathbf{A} \in \mathbb{T}^{n\times m}$ *and two real numbers* β, ζ*, find a non-zero integer vector* $\mathbf{z} \in \mathbb{Z}^m \setminus \{0\}$ *such that* $\|\mathbf{z}\|_2 \leq \beta$ *and* $\|\mathbf{Az}\|_{\mathbb{T}} \leq \zeta$*.*

Definition 1.2 (Continuous Small Integer Solution without modulo 1 (CSIS'))**.** *The CSIS' problem is: given a matrix* $\mathbf{A} \in \mathbb{R}^{n\times m}$ *and two real numbers* β, ζ*, find a non-zero integer vector* $\mathbf{z} \in \mathbb{Z}^m \setminus \{0\}$ *such that* $\|\mathbf{z}\|_2 \leq \beta$ *and* $\|\mathbf{Az}\| \leq \zeta$*.*

Here and throughout this paper, for $p \in [1, \infty]$, we use $\|\mathbf{x}\|_p$ to denote the usual ℓ_p norm; when the subscript is omitted, $\|\mathbf{x}\| = \|\mathbf{x}\|_\infty$. For a matrix, $\|\mathbf{A}\|_p$ denotes the maximum of the ℓ_p norms of its columns; when the matrix subscript is omitted, $\|\mathbf{A}\|$ is the maximum of the infinity norms of its columns. We define the norm on $\mathbb{T}$ as $\|x\|_{\mathbb{T}} = \min_{k\in\mathbb{Z}} |x - k|$.

Let us remark that problems similar to CSIS' have already been studied in [VV25, BRVV26] for the purpose of studying symmetric perceptrons and Johnson-Lindenstrauss. They have proven the hardness of those problems from the worst-case hardness of lattice problems by generalizing the reduction in [MR07], or from the continuous learning with errors problem (CLWE) [BRST21]. However, the formulations of their problems differ than ours in the bound of the solution set. In [VV25], they require $\mathbf{z}$ to be in $\{-1, 1\}^m$; in [BRVV26], they require $\frac{\|\mathbf{Az}\|_2}{\|\mathbf{z}\|_2}$ to be bounded.

In this paper, we prove the hardness of both CSIS and CSIS' based on worst-case lattice assumptions using a high-level abstraction of the standard approach in [MR07]. Our result is

Theorem 1.3 (Summary of Theorem 4.16 and Corollary 4.17)**.** *For any* $n, m \in \mathbb{N}^+$, $\beta, \zeta \in \mathbb{R}^+$*, such that* $\beta, m \in \mathsf{poly}(n)$*,* $\beta \geq \sqrt{m}\zeta^{-n/m}$*,* $\zeta \leq \frac{1}{8n}$*, for any* $\gamma = \omega(\beta\sqrt{n \log n})$*, solving* $CSIS_{m,\beta,\zeta}$ *on average is at least as hard as solving the short independent vector problem (SIVP) with approximation factor* γ*.*

Dropping in $\zeta = \frac{1}{8n}$, $m = \Theta(n \log n)$, $\beta = O(\sqrt{m})$ gives the smallest approximation factor $\gamma = \omega(n \log n)$.

Also, we build reductions relating CSIS, CSIS', and CLWE. Reductions among those problems are summarized in Figure 1.

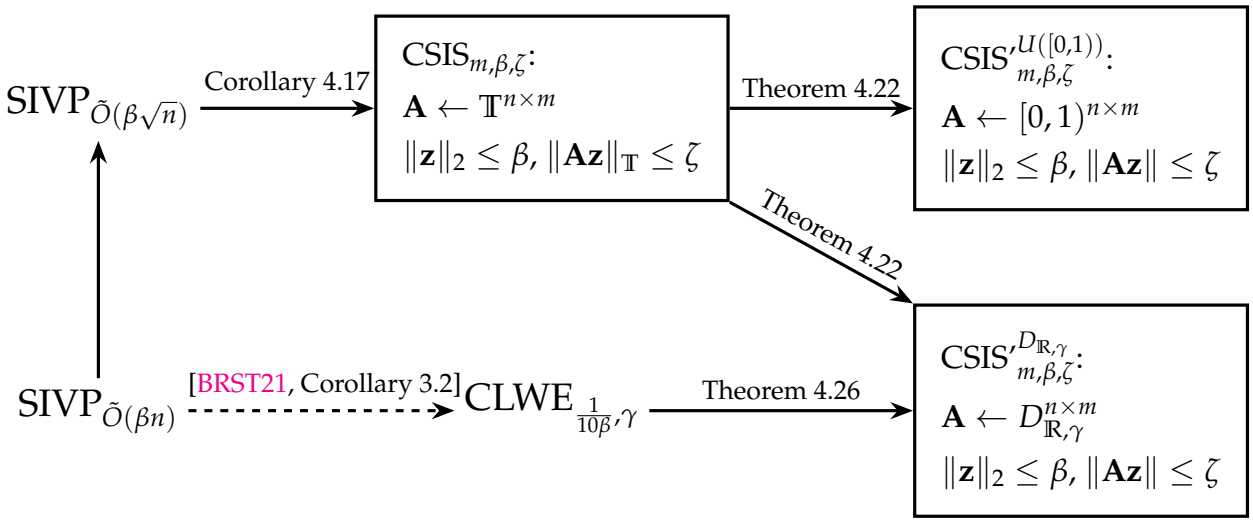


Figure 1: Reductions among CSIS, CSIS', CLWE, and lattice problems. Here A $\rightarrow$ B means B is at least as hard as A, and dashed arrows denote quantum reductions.

Hashes and Commitments with Dynamic Precisions. In Section 5 and Section 6, we construct RVCRHs and RVCs based on SIS [Ajt96] and CSIS'. Our last two hash constructions support dynamically chosen precision and satisfy a key property called **consistency**: for each fixed input, the finite-precision hash outputs converge quickly to a common ideal value as precision increases.

The first construction (Scheme 5.15) has a precision-independent setup and relies on the hardness of SIS. The second construction (Scheme 5.22) operates directly over real values, enabling the use of fast floating-point hardware, and relies on the hardness of CSIS' and a new technique called **base shifting**.

We then transform these real value hash functions to real-valued commitment schemes. Our approach to achieving hiding follows the high-level strategy of [KTX08], but requires a different analysis to deal with real-valued parameters and approximation errors. Also, consistency of the underlying hashes is necessary to allow the committing and verification algorithms to operate at different precision levels.

Homomorphic Real-Valued Commitments from CLWE. The constructions in Section 5 and Section 6 support compression but are not homomorphic. Homomorphism is useful in practice. For example, classical Sigma protocols [Sch90, CDS94] use homomorphic commitments to prove knowledge of committed messages. In Section 7, we construct an additively homomorphic RVC based on CLWE. The scheme is statistically binding and computationally hiding. In Section 8, we use it to obtain a Sigma protocol for proving knowledge of a committed real-valued vector, which may be of independent interest. As a limitation, due to the introduction of noise, the CLWE-based commitment has binding guarantees only for inputs with distance more than $1/\mathsf{poly}(\lambda)$.

On the Existence of Homomorphic and Compressing RVCs. The (C)SIS-based schemes are compressing[1] but not homomorphic, whereas the CLWE-based scheme is homomorphic but not compressing. This raises a natural question: can a real-valued commitment scheme provide both properties? Section 9 gives a partial negative answer. We show that if a compressing real-valued

[1] A scheme is compressing if the dimension of its commitment output is smaller than that of its input.

function f is additively homomorphic and continuous in a neighborhood of a point $\mathbf{x}_0$, then there is another point x_1, at a nontrivial distance from $\mathbf{x}_0$, for which $f(\mathbf{x}_1) = f(\mathbf{x}_0)$. Such a pair of points is efficiently computable and, therefore, violates the binding property.

1.2 Further Related Work

Cryptography for Real-Valued Computations. Most cryptographic systems for numerical computation fix a precision and encode real-valued data in discrete domains. Besides the previously mentioned ZKML proof systems, media provenance systems (e.g., VerITAS [DCB25] and zk-Cinema [FGZ+26]) also express computations as finite-field constraints to verify transformations of discretely represented media; homomorphic encryption schemes, such as CKKS [CKKS17], support approximate arithmetic on encrypted complex vectors via scaling and rounding into an integer cyclotomic ring; MPC systems such as SecureML [MZ17] and ABY3 [MR18] use fixed-point encodings and truncation protocols to support real-valued arithmetic. These systems do not use a continuous algebraic domain for their cryptographic operations.

Cryptography from Continuous LWE. Bruna, Regev, Song, and Tang [BRST21] introduced continuous learning with errors (CLWE), whose samples lie in a continuous space. They also provide a quantum reduction from worst-case lattice problems to CLWE. Later, Gupte, Vafa and Vaikuntanathan [GVV22] provide a classical reduction from LWE to CLWE. CLWE has been used to construct public-key encryption schemes with *binary* plaintexts [BCHR22] and plant undetectable backdoors in binary classifiers [GKVZ22] and neural networks [BRV26].

Sum-Check for Approximate Computations. Bitan et al. [BDG+26] developed an information-theoretic protocol for approximate claims of the form $\sum_{\mathbf{x}\in\{0,1\}^v} g(\mathbf{x}) \approx s$, where g is a low-degree polynomial over the reals. Similar to our work, the prover and the verifier directly operate over the real space. As in a classical sum-check protocol, their protocol reduces the summation claim to an evaluation claim of g at a random point. However, it does not cryptographically bind the prover to a real-valued polynomial; the authors leave polynomial commitments as an open problem. Our commitments provide binding and hiding for real-valued data, but polynomial commitments additionally require *succinct* evaluation proofs without revealing the full polynomial, which are beyond the scope of this work. We leave the construction of a full-fledged polynomial commitment to real values as an interesting open problem.

2 Technical Overview

This section provides an overview of our technical contributions. We first define real-valued commitments and hash functions, and then describe our constructions.

2.1 Hashing and Commitments for Real Values: Definitions

We formalize hashing and commitments for real-valued inputs with explicit support for dynamically chosen precision.

Hashing real values. We first define hash functions for real values (Definition 3.3), which serve as a key building block for our commitment schemes. A scheme Π_{crh} consists of two algorithms,

Gen and Hash. The setup algorithm Gen generates a public parameter h. Given h, an input $\mathbf{x} \in [0,1)^m$, and a precision length ℓ, the hashing algorithm outputs $y = \mathsf{Hash}(h, \mathbf{x}, \ell)$. The precision length ℓ specifies the input resolution: collision resistance is required only for pairs of inputs whose distance exceeds $2^{-\ell}$.

Specifically, Π_h satisfies **f-approximate collision resistance** (Definition 3.5) if for all polynomially bounded precision lengths ℓ, it is computationally hard to find inputs $\mathbf{x}, \mathbf{x}'$ at distance greater than $2^{-\ell}$ whose hash outputs $\mathsf{Hash}(h, \mathbf{x}, \ell)$, $\mathsf{Hash}(h, \mathbf{x}', \ell)$ are within distance $f(\ell)$.

Supporting dynamic precision levels also requires compatibility across those levels. We capture this through the **g-consistency** property (Definition 3.4): as ℓ increases, $\mathsf{Hash}(h, \mathbf{x}, \ell)$ converges to an ideal hash value $h(x)$, with approximation error controlled by a monotone function $g(\ell)$.

We want $f(\cdot)$ to be as large as possible, and $g(\cdot)$ to be as small as possible.

Commitments to real values. We next define real value commitments (Definition 3.1). A scheme Π_{rvc} consists of algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyOpen})$. $\mathsf{Setup}(1^\lambda, 1^m)$ on input the security parameter λ and vector length m, generates a public parameter pp. Given pp, a real vector $\mathbf{x}$, and a committing precision length ℓ_P, the randomized algorithm $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$ outputs a commitment cm and an opening randomness $\mathbf{r}$ (more abstractly, an opening proof π_o). The verification algorithm $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \mathbf{r}, \ell_V)$ checks whether cm is consistent with the opening $(\mathbf{x}, \mathbf{r})$ at the verification precision ℓ_V.

The committing and verification precisions do not need to be the same. A committer may create a high-precision commitment that is later checked by a verifier requiring only a coarser approximation. Likewise, several verifiers with different precision requirements may check the same commitment without requiring the committer to generate a new one.

This flexibility requirement leads to our definitions of correctness and security. **Correctness** requires that a commitment generated at precision ℓ_P accept its honest opening at every verification precision $\ell_V \leq \ell_P$. **Binding** must hold for dynamically chosen verifier precision: an efficient adversary cannot find two accepted openings at verifier precision ℓ_V, whose messages differ by more than $2^{-\ell_V}$. Crucially, the verification cost depends on ℓ_V, rather than the potentially much larger ℓ_P. Finally, **hiding** is defined with respect to the committing precision ℓ_P: commitments to different messages generated at the same precision must be computationally indistinguishable.

For short, ℓ_P controls commitment generation and hiding, whereas ℓ_V controls the binding resolution and the verification cost. Looking ahead, the consistency property of hash functions is a key to supporting independently chosen ℓ_P, ℓ_V in our commitment schemes.

2.2 Hashing and Commitments for Real Values: Constructions

We present our constructions of hash functions for real-valued inputs and then show how to transform them into commitment schemes. We first obtain approximate collision resistance via **adaptive discretization**, then achieve consistency via **decaying weights**, and finally support efficient real-arithmetic evaluation by relying on **base-shifting** and the hardness of CSIS'.

Some naive attempts. In the discrete setting, Ajtai's hash function [Ajt96] has a simple form: sample a random matrix $\mathbf{A} \in \mathbb{Z}_B^{n \times m}$ and hash a low-norm integer vector $\mathbf{x}$ to $\mathbf{A}\mathbf{x}$ mod B. A natural first attempt is to replace the discrete matrix and input with their real-valued counterparts. For inputs normalized to $[0,1)^m$, one could sample $\mathbf{A} \in [0,1)^{n \times m}$ and output $\mathbf{A}\mathbf{x}$. This construction,

however, is insecure: real linearity allows an adversary to find collisions efficiently, as shown by our attack in Section 9.

Another natural approach is to discretize the input before applying Ajtai's hash. Specifically, fix a base b, such as a power of two, set $\mathbf{x}' = \lfloor b\mathbf{x} \rfloor$, and output $\mathbf{A}\mathbf{x}' \bmod B$, where $\mathbf{A} \in \mathbb{Z}_B^{n \times m}$ and $B \gg b$. However, this construction discards all information below the scale $1/b$. Consequently, once the precision is finer than this fixed scale, the construction provides no binding guarantee and fails to meet our requirement of approximate collision resistance.

Collision resistance from adaptive discretization. The problem with quantization is that its precision is determined once and for all during setup. To support dynamic precision, the hashing algorithm must adaptively incorporate finer information about the input. This observation leads to Scheme 5.12: Write the base-b expansion of $\mathbf{x} \in [0,1)^m$ as $dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots) \in \mathbb{Z}_b^{m \times *}$ so that $\mathbf{x} = \sum_{i=1}^{\infty} \mathbf{x}^{(i)} \cdot b^{-i}$. Given a precision length ℓ, set $k = \lceil \ell / \log_2 b \rceil$, the scheme applies a variable-input-length hash function to the prefix $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}) \in \mathbb{Z}_b^{m \times k}$.

We instantiate the variable-input-length hash using the Merkle-Damgård paradigm, with Ajtai hash as the underlying compression function. Under the SIS assumption, this scheme achieves approximate collision resistance for all polynomially bounded precision lengths, without requiring precision-dependent parameters.

However, Scheme 5.12 does not satisfy *consistency*: for the same input $\mathbf{x}$, the outputs at precisions ℓ and $\ell + 1$ differ substantially, and the hash values need not converge as ℓ increases.

Consistency via decaying weights. To achieve consistency, we must ensure that less significant input digits have a smaller effect on the hash output. Scheme 5.15 applies this idea by hashing each digit vector separately and combining the results with geometrically decaying weights.

Specifically, sample a random matrix $\mathbf{A} \in \mathbb{Z}_B^{n \times m}$, where B is sufficiently large. At precision length ℓ, let $k = \lceil \ell / \log_2 b \rceil$. Given the first k base-b digit vectors of $\mathbf{x}$, compute $\mathbf{y}^{(i)} = \mathbf{A}\mathbf{x}^{(i)} \bmod B$ for $i \in [k]$, and output $\mathbf{z}_k = \sum_{i=1}^{k} \mathbf{y}^{(i)} B^{-i}$.

Because each additional term has geometrically decreasing magnitude, the outputs converge as ℓ increases, giving consistency. With appropriate parameters, approximate collision resistance can again be based on SIS: if two hash outputs are close while the inputs are far, it means that they have identical $\mathbf{y}$-values at an index i, while the underlying input vectors $\mathbf{x}^{(i)}$, $\mathbf{x}'^{(i)}$ are different, which gives a SIS solution $\mathbf{x}^{(i)} - \mathbf{x}'^{(i)}$ for $\mathbf{A}$.

However, computing the hash requires a matrix-vector multiplication over $\mathbb{Z}_B$ for every input digit vector, and it is difficult to exploit the fast floating-point units on modern CPUs and GPUs.

Real-arithmetic hashing via base shifting. The above limitation motivates a construction whose main computation is a *single* matrix-vector multiplication over *real values*. Our final construction (Scheme 5.22) achieves this via a non-linear encoding technique called **base shifting**:

- Fix b, B with $B > b$. For $\mathbf{x} \in [0,1)^m$ with base-b expansion $dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots)$, define

$$\phi_{b \to B}(\mathbf{x}) = \sum_{i=1}^{\infty} \mathbf{x}^{(i)} \cdot B^{-i}.$$

  Here $\phi_{b \to B}$ reinterprets the base-b digits of $\mathbf{x}$ as base-B digits. E.g., when $b = 2$, $B = 4$, the binary input $\mathbf{x} = 0.11_2$ is mapped to $\phi_{b \to B}(\mathbf{x}) = 0.11_4 = 0.0101_2$, which inserts a zero before each input bit.

- The public parameter is a randomly sampled matrix $\mathbf{A} \in [0,1)^{n \times m}$. At precision length ℓ, set $k = \lceil \ell / \log_2 b \rceil$, the output is $h_{\mathbf{A},k}(\mathbf{x}) = \mathbf{A} \times \phi^{(k)}_{b \to B}(\mathbf{x})$ where $\phi^{(k)}_{b \to B}(\mathbf{x}) = \sum_{i=1}^{k} \mathbf{x}^{(i)} \cdot B^{-i}$.

Unlike Scheme 5.15, this construction performs no modular reduction to individual digit vectors. Once the base-shifted input has been prepared, evaluation consists of a single real matrix-vector multiplication. Consistency follows directly from the geometric decay of the weights $(B^{-1}, B^{-2}, \ldots)$.

Collision-resistance is based on the hardness of CSIS′ (Proposition 5.25). To see this, suppose an efficient adversary finds inputs $\mathbf{x}, \mathbf{x}'$ whose distance exceeds $2^{-\ell}$, but whose precision-ℓ hash outputs are ϵ-close. Let $dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots)$ and $dig_b(\mathbf{x}') = (\mathbf{x}'^{(1)}, \mathbf{x}'^{(2)}, \ldots)$ be their base-b expansions. Set $k = \lceil \ell / \log_2 b \rceil$ and let $i \le k$ be the first index at which $\mathbf{x}^{(i)} \neq \mathbf{x}'^{(i)}$. For simplicity, we ignore the approximate evaluation errors, and for sufficiently large B, we have

$$\epsilon = \|\mathbf{A}[\phi_{b \to B}(\mathbf{x}) - \phi_{b \to B}(\mathbf{x}')]\| \approx \|\mathbf{A}(\mathbf{x}^{(i)} - \mathbf{x}'^{(i)}) \cdot B^{-i}\|,$$

as the differences $(\mathbf{x}^{(i+1)} - \mathbf{x}'^{(i+1)}, \mathbf{x}^{(i+2)} - \mathbf{x}'^{(i+2)}, \ldots)$ have a sufficiently small tail effect on ϵ due to the exponentially decaying weights $(B^{-(i+1)}, B^{-(i+2)}, \ldots)$. With appropriate parameters, $\mathbf{x}^{(i)} - \mathbf{x}'^{(i)}$ is therefore a low-norm, nonzero solution to the CSIS′ problem defined by $\mathbf{A}$. See Section 5.4 for the full argument.

In Table 1, we provide a comparison of the four real-valued collision resistant hash functions.

Scheme	**Assumption**	**Collision Resistance** $f(\cdot)$	**Output**	**Consistency** $g(\cdot)$
Simple Rounding (5.2)	SIS	1 $\quad (\ell \le \log b)$	$\mathbb{Z}_B^n$	0
Iterative Compression (5.12)	SIS	1	$\mathbb{Z}_B^n$	–
Decaying Weights (5.15)	SIS	$B^{-\lceil \frac{\ell}{\log b} \rceil}$	$\mathbb{R}^n$	$B^{-\lceil \frac{\ell}{\log b} \rceil}$
Base Shifting (5.22)	CSIS′	$B^{-\frac{\ell}{\log b}}/40nB$	$\mathbb{R}^n$	$\epsilon_\ell \le B^{-\frac{\ell}{\log b}}/80nB$

Table 1: Summary of the four hash function constructions in terms of their assumptions, collision resistance properties, output spaces, and consistency properties. The parameter B is much larger than b in the Simple Rounding scheme (5.2), with $\log b$ a fixed upper bound on the precision lengths. B is independent of the dynamically chosen precision lengths in all other schemes.

Transforming hashes to hiding commitments. We now show how to add hiding to our real-valued hash functions. For Scheme 5.12 and Scheme 5.15, we follow the approach of [KTX08]: append a uniformly random vector $\mathbf{r} \in [0,1)^{m_r}$ to the input $\mathbf{x}$, and commit by hashing $(\mathbf{x}, \mathbf{r})$. Binding follows directly from collision resistance. To prove hiding, we interpret the base-b expansion of $\mathbf{r}$ as a sequence of random integer vectors and apply the standard Leftover Hash Lemma.

For Scheme 5.22, the transformation is more subtle because the hashing parameter $\mathbf{A}$ is a real matrix. The standard Leftover Hash Lemma requires a universal hash family, but obtaining a suitable notion of universality in this real-valued setting is nontrivial. Thus, the usual argument does not apply here.

Our final construction is still similar to its discrete counterpart in [KTX08]. Let $\mathbf{A}_r \in \mathbb{T}^{n \times m_r}$ be a public parameter, and sample a vector $\mathbf{r}$ uniformly from $\{0, \ldots, b-1\}^{m_r}$, for suitable parameters b, m_r. We mask a scaled hash output $h(\mathbf{x})$ by setting $h'(\mathbf{x}, \mathbf{r}) = h(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r} \bmod 1$, followed by a final rounding step. Security analysis, however, is quite different:

- **Hiding:** We directly analyze the distribution of $\mathbf{A}_r \cdot \mathbf{r} \bmod 1$, obtaining a real-valued analogue of the Leftover Hash Lemma without universality.

- **Binding:** Consider two openings $(\mathbf{x}, \mathbf{r})$ and $(\mathbf{x}', \mathbf{r}')$. If $\mathbf{r} = \mathbf{r}'$, binding follows from the approximate collision resistance of h. If $\mathbf{r} \neq \mathbf{r}'$, the CSIS assumption rules out the possibility of efficiently finding such openings with $\|\mathbf{A}_r(\mathbf{r} - \mathbf{r}') \bmod 1\|_{\mathbb{T}}$ below a threshold c. We scale $h(\mathbf{x})$ so that the difference between any two scaled hash values is much smaller than c. This gap prevents the hash difference from canceling the mask difference.

Correctness requires care for both types of transformations. To allow the committing and verification algorithms to use different precision lengths, the underlying hash families must be *consistent*. Hence, only the transformations for Scheme 5.15 and Scheme 5.22 fully support dynamic precisions ℓ_V, ℓ_P where $\ell_V \leq \ell_P$. Here is a special case of our main theorem.

Theorem 2.1 (Real Value Commitment, a special case of Theorem 6.4)**.** *Assuming the worst-case hardness of $SIVP_{\tilde{O}(n^{1.5})}$ and the existence of RVCRHs with $f(\cdot)$-approximate collision resistance and $g(\cdot)$-consistency, where f, g satisfies $f(\ell) > 12g(\ell)$ for all $\ell \in \mathbb{N}$, then there is a computationally binding and statistical hiding real value commitment scheme that works for any $\ell_V \leq \ell_P \leq \Theta(n)$.*

See Section 6 for the details of general statements and proofs.

2.3 Homomorphic Commitments from Continuous LWE

The (C)SIS-based commitment schemes described above lose additive homomorphism because of their nonlinear discretization layers. Their binding guarantees are also only computational. In Section 7, we construct a CLWE-based commitment scheme that is statistically binding, computationally hiding, and additively homomorphic.

We begin by recalling the CLWE problem. Fix parameters $\beta, \gamma > 0$ and sample a secret unit vector $\mathbf{w} \leftarrow \mathcal{S}^{n-1}$. A CLWE sample is a pair $(\mathbf{a}, y)$ where

$$\mathbf{a} \leftarrow D^n_{\mathbb{R},1}, \quad e \leftarrow D_{\mathbb{R},\beta}, \quad y = (\gamma\langle \mathbf{a}, \mathbf{w}\rangle + e) \bmod 1\,.$$

The decisional CLWE problem asks an adversary to distinguish samples of this form generated using the same secret $\mathbf{w}$ from samples $(\mathbf{a}, u)$ where $u \leftarrow U(\mathbb{T})$ is independent of $\mathbf{a}$. [BRST21] gives a polynomial-time quantum reduction from worst-case lattice problems to average-case CLWE.

The main theorem is as follows.

Theorem 2.2 (Real-Valued Commitment from CLWE, informal)**.** *Assume the hardness of $CLWE_{\beta,\gamma}$ where $\gamma \geq 2\sqrt{n}$, $\beta \in (0, \frac{1}{100\sqrt{n}})$, $\gamma/\beta \in \mathsf{poly}(n)$. Then there is a real-valued commitment scheme that is computationally hiding and statistically binding.*

Our construction uses CLWE samples to mask a linear encoding of the committed message. The setup algorithm chooses $k > n + m$ and samples $\mathbf{A} \leftarrow D^{(n+m)\times k}_{\mathbb{R},1}$. To commit to a vector $\mathbf{x} \in [-M, M]^m$, where M is polynomially bounded, the committer samples $\mathbf{w} \leftarrow \mathcal{S}^{n-1}$ and $\mathbf{e} \leftarrow D^k_{\mathbb{R},\beta}$, and outputs

$$\mathbf{c} = L(\mathbf{x}, \mathbf{w}, \mathbf{e}) = \left(\gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e}\right) \bmod 1 \in \mathbb{T}^k\,.$$

Unlike the constructions based on nonlinear encoding, the commitment map L is additive over the $\mathbb{R}$-vector space: for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n+m+k}$,

$$L(\mathbf{u} + \mathbf{v}) = [L(\mathbf{u}) + L(\mathbf{v})] \bmod 1\,.$$

Thus, adding commitments adds their messages and opening randomness, subject to the corresponding norm bounds. In other words, the scheme satisfies bounded additive homomorphism.

Computational hiding follows from the CLWE assumption. To see this, write

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_x \\ \mathbf{A}_w \end{pmatrix}, \qquad \mathbf{c} = \left(\gamma \mathbf{A}_x^{\mathbb{T}} \mathbf{x} + \left(\gamma \mathbf{A}_w^{\mathbb{T}} \mathbf{w} + \mathbf{e}\right)\right) \bmod 1.$$

Under CLWE, the second term is computationally indistinguishable from a uniform vector in $\mathbb{T}^k$, even given the public matrix.

Statistical binding relies on a geometric property: with overwhelming probability over $\mathbf{A}$, sufficiently separated inputs cannot be mapped to close points. We prove this in two steps: First, Lemma 7.6 bounds the probability that a fixed vector of sufficiently large norm has a small image under the random map $\mathbf{v} \mapsto \mathbf{A}^T\mathbf{v} \bmod 1$. Here, distance on the torus is measured by

$$\|\mathbf{x}\|_{\mathbb{T}} := \min_{\mathbf{z} \in \mathbb{Z}^k} \|\mathbf{x} - \mathbf{z}\|_\infty.$$

Second, Lemma 7.7 upgrades this fixed-vector bound and argues that for all inputs over $[-2M, 2M]^{m+n}$ with a large norm, the corresponding random map images also have large norms. The proof discretizes the input set, applies a union bound over the resulting net, and controls the approximation error between net points.

Consequently, with overwhelming probability over $\mathbf{A}$, for all $(\mathbf{x}, \mathbf{w}), (\mathbf{x}', \mathbf{w}') \in [-M, M]^{m+n}$ that are far apart, by setting $\mathbf{v} = (\mathbf{x}, \mathbf{w}) - (\mathbf{x}', \mathbf{w}')$, we have that the quantity $\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}}$ exceeds the amount that the noise difference $\mathbf{e} - \mathbf{e}'$ can cancel. Hence, their commitments $\mathbf{c}$ and $\mathbf{c}'$ are distant, giving the required statistical binding property. However, since the noise difference $\mathbf{e} - \mathbf{e}'$ is an inverse polynomial, there is no way to detect negligible deviation of the input, which means that the binding only holds for inputs with distance more than $1/\mathsf{poly}(n)$.

Let us remark that the real value commitment scheme from CLWE as stated only supports dynamic precision lengths of $\ell_V \leq \ell_P \leq O(\log n)$, since the CLWE error is chosen to be $1/\mathsf{poly}(n)$. One could get a larger bound by using a standard lattice cryptographic trick: set the error bound to be $O(1/2^{n^c})$ for some $c \in (0, 1)$ and rely on the subexponential hardness of CLWE with subexponentially small error. Then let the CLWE dimension parameter n to be a large polynomial in the security parameter λ. For example, by letting $c = 1/2$, $n = O(\lambda^2)$, we can set $\ell_V \leq \ell_P \leq O(\lambda)$, linear in the security parameter.

Application: Zero-Knowledge Proof of Knowledge for Committed Real Values. The additive structure of the commitment also enables a Sigma-style protocol for proving knowledge of the committed real values. Our scheme follows the Fiat-Shamir with aborts (FSwA) paradigm [Lyu09, Lyu12], using a careful abort rule to prevent the prover's response from revealing its *low-norm* witness. We achieve a stronger honest-verifier zero-knowledge property introduced by [DFPS23], which requires simulation of the verifier's view not only in successful executions but also in aborted ones. As emphasized in [DFPS23], this stronger guarantee is important for establishing the security of the Fiat-Shamir transformation in the quantum random oracle model (QROM).

Let pp denote the public parameters and $x = (\mathbf{y}, \mathsf{pp})$ the public input. Define the set:

$$\mathcal{D}_\ell = \left\{(s, \mathbf{w}, \mathbf{e}) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^k : |s| \leq \ell B, \quad \|\mathbf{w}\|_\infty \leq \ell B, \quad \|\mathbf{e}\|_\infty < \ell b\right\}.$$

To prove knowledge of a witness $\omega = (s, \mathbf{w}, \mathbf{e}) \in \mathcal{D}_1$ satisfying $\mathbf{y} = L(\omega)$, the prover first sends

$$\mathbf{t} = L(\omega_0 + \mathbf{v}),$$

where $\omega_0 = (0, \mathbf{w}_0, \mathbf{e}_0)$ is sampled as an opening of a commitment to zero, with $\mathbf{w}_0$ uniform on the unit sphere in $\mathbb{R}^n$ and $\mathbf{e}_0 \leftarrow \widetilde{D}^k_{\mathbb{R},\sigma}$. Independently, it samples $\mathbf{v} \leftarrow U(\mathcal{D}_K)$. Upon receiving a challenge $c \in \{0,1\}$ from the verifier, the prover first checks whether $\mathbf{h} = \mathbf{v} + c\omega$ belongs to $\mathcal{D}_{K-1}$. If not, it sends $\perp$; otherwise, it responds with

$$\mathbf{z} = \omega_0 + \mathbf{h} = \omega_0 + \mathbf{v} + c\omega.$$

The verifier rejects $\perp$ and otherwise checks

$$\mathbf{z} \in \mathcal{D}_K \qquad \text{and} \qquad L(\mathbf{z}) = \mathbf{t} + c\mathbf{y} \pmod 1.$$

Theorem 2.3 (CLWE-based Σ-protocol for real values, informal)**.** *Assume the hardness of $\mathrm{CLWE}_{\sigma,\gamma}$ where $\gamma \geq 2\sqrt{n}$, $\sigma \in (0, \frac{1}{100\sqrt{n}})$, $\gamma/\sigma \in \mathsf{poly}(n)$. Then the protocol described above is a Σ-protocol for real values. The protocol satisfies special soundness and computational strong honest-verifier zero knowledge, including aborting executions.*

Correctness holds whenever the honest prover does not abort, which occurs with probability greater than $3/4$. Special soundness follows from the additivity of L, with the witness bounds relaxed by a polynomial factor. For HVZK, observe that conditioned on non-abort, $\mathbf{h}$ is uniform on $\mathcal{D}_{K-1}$ and independent of ω, giving statistical HVZK for accepting executions.

We need ω_0 in the first message to simulate the aborting view. In the real aborting view, the distribution of $\mathbf{v}$ is biased by the abort condition. To simulate $\mathbf{t}$ conditioned on abort, we mask $L(\mathbf{v})$ with the independent term $L(\omega_0)$, which is computationally indistinguishable from uniform on $\mathbb{T}^k$ under CLWE. For a prescribed challenge, the abort condition depends only on $\mathbf{v}$ and leaves the distribution of ω_0 unchanged. Thus, $\mathbf{t} = L(\omega_0) + L(\mathbf{v})$ remains computationally indistinguishable from uniform even conditioned on abort, allowing us to simulate the aborting view.

2.4 On the Existence of Homomorphic and Compressing Commitments

The CLWE-based commitment scheme is not compressing: the commitment dimension is always larger than the sum of the input and masking dimensions. Is it possible to construct a real value commitment scheme that is both additively homomorphic and compressing? In Section 9, we give a partial negative answer. For simplicity, we ignore randomness here and focus on collision attacks.

Theorem 2.4 (Informal)**.** *Fix any $m > n$. If $f : \mathbb{R}^m \to \mathbb{R}^n$ is additively homomorphic and continuous in a neighborhood of $\mathbf{0}$, then f cannot be collision resistant.*

The attack works in two steps:

1. In Proposition 9.5, we show that if f is additively homomorphic and continuous in a neighborhood $\mathcal{B}_0$ of $\mathbf{0}$, then $f(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ for all $\mathbf{x} \in \mathcal{B}_0$, for some matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$.

2. Let $\mathbf{x}_0 = \mathbf{0}$. Since $f(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ within $\mathcal{B}_0$, f is twice differentiable there and the second-order derivative is always zero. Thus, for all $\mathbf{x}_0 + \Delta \in \mathcal{B}_0$, we have

   $$g(\Delta) := f(\mathbf{x}_0 + \Delta) - f(\mathbf{x}_0) = \mathbf{J}(\mathbf{x}_0) \cdot \Delta,$$

   where $\mathbf{J}(\mathbf{x}_0) \in \mathbb{R}^{n \times m}$ is the Jacobian matrix at $\mathbf{x}_0$. Since $m > n$, there is a unit vector Δ_1 such that $\mathbf{J}(\mathbf{x}_0)\Delta_1 = \mathbf{0}$. By choosing a small enough parameter $\alpha > 0$ so that $\mathbf{x}_1 = \mathbf{x}_0 + \alpha \cdot \Delta_1$ lies in $\mathcal{B}_0$, we have that $\|\mathbf{x}_1 - \mathbf{x}_0\| = \alpha$ while $\|f(\mathbf{x}_1) - f(\mathbf{x}_0)\| = \|\mathbf{J}(\mathbf{x}_0) \cdot \Delta_1 \cdot \alpha\| = 0$.

In particular, our attack rules out constructions of the form $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ or $f(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod 1$. In Section 9, we also account for the hiding randomness and extend the attack to a broader class of functions that need only be sufficiently smooth in a neighborhood of some point $\mathbf{x}_0$.

2.5 AI Disclosure

The main motivations, definitions, technical ideas, and security proofs are all generated by humans. We only use AI for language revision and for checking mathematical errors.

3 Hash and Commitment for Real Values: Definitions

3.1 Notations

The symbol λ denotes the security parameter. We use $\mathsf{poly}(\lambda)$ and $\mathsf{negl}(\lambda)$ for an arbitrary polynomially bounded function and an arbitrary negligible function, respectively. An algorithm is efficient if it runs in probabilistic polynomial time (p.p.t.). The notation $U(S)$ denotes the uniform distribution over a set S. We write $X \approx_s Y$ and $X \approx_c Y$ when two distribution ensembles are statistically close and computationally indistinguishable, respectively.

We write $\mathbb{R}$, $\mathbb{Q}$, and $\mathbb{Z}$ for the real, rational, and integer numbers, respectively. $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denotes the quotient group of $\mathbb{R}$ modulo 1, with default representative set $[0, 1)$. For $q \in \mathbb{N}^*$, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ with default representative set $\{0, 1, \ldots, q-1\}$.

By default, bold lowercase letters denote vectors and bold uppercase letters denote matrices. For vectors, $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\mathsf{T}\mathbf{y}$ denotes the Euclidean inner product, $\mathbf{x}; \mathbf{y}$ denotes vertical concatenation, and $\mathbf{x}^T$ denotes transpose. For $p \in [1, \infty]$, let $\|\mathbf{x}\|_p$ be the usual ℓ_p norm; when the subscript is omitted, $\|\mathbf{x}\| = \|\mathbf{x}\|_\infty$. For a matrix, $\|\mathbf{A}\|_p$ denotes the maximum of the ℓ_p norms of its columns; when the matrix subscript is omitted, $\|\mathbf{A}\|$ is the maximum of the infinity norms of its columns. Define the norm on $\mathbb{T}$ as $\|x\|_\mathbb{T} = \min_{k\in\mathbb{Z}} |x - k|$. For a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{T}^n$, define $\|\mathbf{x}\|_\mathbb{T} = \max_{i\in[n]} \|x_i\|_\mathbb{T}$. We use $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ for the floor and ceiling functions, respectively, and $\lfloor \cdot \rceil$ for rounding to the nearest integer. By default, operations on vectors and matrices are performed entrywise.

The symbol $Vol(\cdot)$ denotes Lebesgue volume. For $r > 0$, let $\mathcal{B}_p(\mathbf{x}, r) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{x}\|_p \le r\}$ and $\mathcal{S}^{n-1}(r) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 = r\}$; write $\mathcal{S}^{n-1} = \mathcal{S}^{n-1}(1)$. For a full-rank basis $\mathbf{B}$, $\mathcal{L}(\mathbf{B}) = \{\mathbf{B}\mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$ denotes the lattice generated by $\mathbf{B}$, and $\mathcal{P}(\mathbf{B}) = \mathbf{B}[0, 1)^n$ denotes its fundamental parallelepiped. We write $\lambda_i(\mathcal{L})$ for the i-th successive minimum.

For $r \in \mathbb{R}^+$ and any real vector $\mathbf{x}$, define

$$\rho_r(\mathbf{x}) = \exp\left(-\pi \frac{\|\mathbf{x}\|_2^2}{r^2}\right).$$

For a subset S, either discrete or continuous, of any real vector space, let $D_{S,r}$ denote the distribution over S with probability or probability density proportional to $\rho_r(\cdot)$. In particular, $\mathcal{N}(0, \sigma^2) = D_{\mathbb{R},\sqrt{2\pi}\sigma}$ is a centered normal distribution.
Sometimes we want to bound the magnitude of a Gaussian sample, so we define the probability ensemble

$$\tilde{D}_{\mathbb{R},r} = \{D_{(-r\sqrt{\lambda}, r\sqrt{\lambda}),r}\}_{\lambda=1}^{\infty},$$

which is the distribution $D_{\mathbb{R},r}$ conditioned on $|x| < r\sqrt{\lambda}$. Clearly $\tilde{D}_{\mathbb{R},r}$ is statistically close to $\{D_{\mathbb{R},r}\}_\lambda$. We also abbreviate $\{D_{\mathbb{R},r}\}_\lambda$ to $D_{\mathbb{R},r}$.

3.2 Real Value Arithmetic

In practice, there is no way to represent real numbers exactly with finite bits. Storage, transmission and computation of real numbers are all done with finite precision and hence incur errors. To avoid excessive implementation details, we only consider computation errors in this paper. One can show that this is without loss of generality. When the computation error is worth noting, we write $\mathbf{y} =_\epsilon C(\mathbf{x})$ to denote that $\mathbf{y}$ is computed from $C(\mathbf{x})$ with error $\|\mathbf{y} - C(\mathbf{x})\| \leq \epsilon$. If ϵ is omitted, it is assumed to be exponentially small. At first reading, one can ignore the computation error and treat $\mathbf{y} =_\epsilon C(\mathbf{x})$ as $\mathbf{y} = C(\mathbf{x})$.

Both addition and multiplication of two polynomially bounded real numbers can be computed with error at most ϵ with overhead quasi-linear in $\log(1/\epsilon)$. When every value involved in the computation is polynomially bounded, which is usually the case, any efficient algorithm with constant multiplication depth only leads to a polynomially bounded blowup of the computation error. For these algorithms, we can safely use the end-to-end error bound without worrying about the intermediate errors.

Some protocols in this paper also involve digit operations. To avoid strange compatibility issues, we further assume that in actual implementation all the real numbers are represented on a consistent grid, say $\bigcup_{i=1}^{\infty} \mathbb{Z}/b^i$ for some integer $b > 1$.

3.3 Real Value Commitments

Imagine a prover committing to a message and then a verifier verifying the commitment. Before this interaction, they need to declare a precision length ℓ. The verifier only cares about whether the committed message is within distance $2^{-\ell}$ of the claimed message. In other words, a deviation bounded by $2^{-\ell}$ is acceptable.

However, in many practical scenarios, it is desirable to have more flexibility in the precision length, as sometimes it is inconvenient for the prover and verifier to agree on a common precision length before the commitment. For example, imagine there is a single prover committing to a message, while multiple verifiers verify the commitment. Due to limited resources, some verifiers may not be able to afford as high a precision as the prover uses, and they may want to verify the commitment with a lower precision length more efficiently.

In this case, we want to allow the prover and verifier to choose precision lengths ℓ_P, ℓ_V independently. ℓ_P serves as the upper bound of the precision length that the verifier can verify. The prover performs the committing procedure according to ℓ_P, while the verifier can dynamically choose $\ell_V \leq \ell_P$, perform the verification according to ℓ_V, and detect any deviation larger than $2^{-\ell_V}$, hopefully more efficiently.

Formally, we have the following definition of real value commitment schemes.

Definition 3.1 (Real value commitments)**.** *A real value commitment scheme over message space $\mathcal{M} \subseteq \mathbb{R}$ consists of a tuple of efficient algorithms $\Pi_{\mathsf{rvc}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyOpen})$:*

- $\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{pp}$*: On input the security parameter λ and vector length m, outputs a public parameter* pp*.*
- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P) \to (\mathsf{cm}, \pi_o)$*: On input the public parameter* pp*, a vector $\mathbf{x} \in \mathcal{M}^m$, a precision length $\ell_P \in \mathsf{poly}(\lambda)$, outputs a commitment* cm *and an opening π_o.*
- $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) \to 0/1$*: On input the public parameter* pp*, a commitment* cm*, a vector $\mathbf{x} \in \mathcal{M}^m$, an opening proof π_o, and a precision length $\ell_V \in \mathsf{poly}(\lambda)$, outputs a bit $b \in \{0, 1\}$.*

Π_{rvc} *satisfies the following properties:*

- ***Correctness:*** *For all security parameters λ, vector lengths $m = m(\lambda)$, precision lengths $\ell_P = \ell_P(\lambda), \ell_V = \ell_V(\lambda)$ where $\ell_P \geq \ell_V$, and vectors $\mathbf{x} \in \mathcal{M}^m$,*

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m); \\ (\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)\end{array} : \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) = 1\right] = 1 - \mathsf{negl}(\lambda)\,.$$

- ***Binding:*** Π_{rvc} *is statistically (resp. computationally) binding if for all security parameters λ, vector lengths $m = m(\lambda)$, precision lengths $\ell_V = \ell_V(\lambda)$, and all unbounded (resp. efficient) adversaries $\mathcal{A}$,*

$$\Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m); \\ (\mathsf{cm}, \mathbf{x}, \pi_o, \mathbf{x}', \pi_o') \leftarrow \mathcal{A}(\mathsf{pp})\end{array} : \begin{array}{c}\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) = 1 \wedge \\ \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}', \pi_o', \ell_V) = 1 \wedge \\ \|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell_V}\end{array}\right] = \mathsf{negl}(\lambda)\,.$$

- ***Hiding:*** Π_{rvc} *is statistically (resp. computationally) hiding if there exists an efficient simulator $\mathcal{S}$ such that for all vector lengths $m = m(\lambda)$, precision lengths $\ell_P = \ell_P(\lambda)$, and vectors $\mathbf{x} \in \mathcal{M}^m$, the following two distributions are statistically (resp. computationally) indistinguishable:*

$$\mathsf{Real}(1^\lambda, 1^m, 1^{\ell_P}, \mathbf{x}) = \{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m),\ (\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P) \ :\ (\mathsf{pp}, \mathsf{cm})\}\,;$$
$$\mathsf{Ideal}(1^\lambda, 1^m, 1^{\ell_P}) = \{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m),\ \mathsf{cm} \leftarrow \mathcal{S}(1^\lambda, 1^m, 1^{\ell_P}, \mathsf{pp}) \ :\ (\mathsf{pp}, \mathsf{cm})\}\,.$$

Remark 3.2. *Looking ahead, Scheme 6.2 achieves all required properties but does not fully support native real arithmetic. On the other hand, some of our constructions with native real arithmetic (Scheme 6.12 and Scheme 7.1) require precision lengths to be $\tilde{O}(\lambda)$ and $\tilde{O}(\log \lambda)$ to achieve hiding. Constructing a real value commitment scheme with native real arithmetic that requires no a priori upper bound on precision is still an open problem.*

3.4 Real Value Collision Resistant Hash

A common approach to constructing commitment schemes is to use collision resistant hash functions. Thus, we also need a notion of real value hash functions:

Definition 3.3 (Real value hash functions)**.** *A real value hashing scheme over message space $\mathcal{M} \subseteq \mathbb{R}$ consists of a pair of efficient algorithms $\Pi_{\mathsf{crh}} = (\mathsf{Gen}, \mathsf{Hash})$:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n) \to h$*: On input the security parameter λ, input length $m = m(\lambda)$, output length $n = n(\lambda)$ where $n < m$, outputs a hashing parameter h.*
- $\mathsf{Hash}(h, \mathbf{x}, \ell) \to y$*: On input the hashing parameter h, input $\mathbf{x} \in \mathcal{M}^m$, precision length $\ell \in \mathsf{poly}(\lambda)$, outputs a value y.* Hash *is a deterministic polynomial-time algorithm, and we write the output value as $h_\ell(\mathbf{x})$ in short.*

Here, ℓ is again the precision length. When using $\mathsf{Hash}(h, \cdot, \ell)$, we do not distinguish between two inputs $\mathbf{x}$ and $\mathbf{x}'$ if they are within distance $2^{-\ell}$.

Ideally, ℓ does not explicitly appear in the description of the Hash algorithm, but only provides a suggestion to the precision needed. The following definition of consistency captures the property that the output of Hash is consistent with an ideal hash function IdealHash, which is independent of the precision length ℓ. As ℓ increases, the output of Hash should get closer to the output of IdealHash. The rate of convergence is given by a monotone decreasing function g.

Definition 3.4 (consistency). *For monotone decreasing function $g : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we say that Π_{crh} satisfies $g(\cdot)$-consistency if for all $\lambda \in \mathbb{N}$, input lengths $m = m(\lambda)$, output lengths $n = n(\lambda)$ where $n < m$, there exists a function $\mathsf{IdealHash}(h, \mathbf{x})$ such that for any hashing parameter h generated by Gen, any input $\mathbf{x} \in \mathcal{M}^m$, and any precision length ℓ,*

$$\|\mathsf{Hash}(h, \mathbf{x}, \ell) - \mathsf{IdealHash}(h, \mathbf{x})\| \leq g(\ell)\,.$$

We generalize the collision resistance property to real value hash functions in a natural way. Any efficient adversary cannot find two inputs that are distinguishable under the given precision length ℓ but have outputs that are close to each other. The distance between the outputs may depend on the precision length ℓ, and is captured by a function f.

Definition 3.5 (approximate collision resistance). *For function $f : \mathbb{N} \to \mathbb{R}^+$, we say Π_{crh} satisfies $f(\cdot)$-approximate collision resistance if for all $\lambda \in \mathbb{N}$, input lengths $m = m(\lambda)$, output lengths $n = n(\lambda)$ where $n < m$, precision lengths $\ell = \ell(\lambda)$, and all efficient adversaries $\mathcal{A}$,*

$$\Pr\left[\begin{array}{l} h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n) \\ (\mathbf{x}, \mathbf{x}') \leftarrow \mathcal{A}(1^\lambda, h, \ell) \end{array} : \mathbf{x}, \mathbf{x}' \in \mathcal{M}^m \wedge \|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell} \wedge \|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < f(\ell)\right] \leq \mathsf{negl}(\lambda)\,.$$

We want $f(\cdot)$ to be as large as possible, and $g(\cdot)$ to be as small as possible.

4 Average-case Hard Problems on Real Values

4.1 Lattice Problems

We first recall the definitions of some standard lattice problems that will be used in our reductions.

Definition 4.1 (Shortest Vector Problem). *An instance of $GapSVP_\gamma$ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is an n-dimensional lattice basis and d is a number. In YES instances, $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$, while in NO instances, $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.*

Definition 4.2 (Shortest Independent Vectors Problem). *An instance of $SIVP_\gamma$ is an n-dimensional lattice basis $\mathbf{B}$. The goal is to output a set of n linearly independent lattice vectors (written as a matrix) $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{S}\|_2 \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.*

Definition 4.3 (Guaranteed Distance Decoding). *An instance of GDD^ϕ_γ is an n-dimensional lattice basis $\mathbf{B}$ and a target vector $\mathbf{t}$. The goal is to output a lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x} - \mathbf{t}\|_2 \leq \gamma(n) \cdot \phi(\mathcal{L}(\mathbf{B}))$.*

Definition 4.4 (Covering Radius Problem). *An instance of $GapCRP_\gamma$ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is an n-dimensional lattice basis and d is a number. In YES instances, $\nu(\mathcal{L}(\mathbf{B})) \leq d$, while in NO instances, $\nu(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$. Here ν is the covering radius of a lattice.*

4.2 CLWE

[BRST21] introduced the Continuous Learning with Errors (CLWE) problem, an analog of LWE on real values, and gave a polynomial-time quantum reduction from lattice problems to average-case CLWE. In the following, we review some of their core definitions and results.

Definition 4.5 (CLWE distribution, Definition 2.18, [BRST21]). *For a unit vector $\mathbf{w} \in \mathcal{S}^{n-1}$ and parameters $\beta, \gamma > 0$, the CLWE distribution $A_{\mathbf{w},\beta,\gamma}$ is over $\mathbb{R}^n \times \mathbb{T}$. To sample $(\mathbf{a}, y)$ from $A_{\mathbf{w},\beta,\gamma}$, first sample $\mathbf{a} \leftarrow D^n_{\mathbb{R},1}$, then sample $e \leftarrow D_{\mathbb{R},\beta}$ and set $y = (\gamma\langle \mathbf{a}, \mathbf{w}\rangle + e) \bmod 1$.*

Definition 4.6 (Definition 2.20, [BRST21]). *For parameters $\beta, \gamma > 0$, the average-case decision problem $CLWE_{\beta,\gamma}$ is to distinguish the following two distributions, from which the distinguisher can obtain polynomially many samples:*

(1) The CLWE distribution $A_{\mathbf{w},\beta,\gamma}$ for some uniformly random unit vector $\mathbf{w} \in \mathcal{S}^{n-1}$ (fixed for all samples),

(2) $D^n_{\mathbb{R},1} \times U(\mathbb{T})$.

They base the hardness of CLWE in a framework similar to that for LWE, by iteratively applying a classical reduction from the bounded distance decoding problem (BDD) to the discrete Gaussian sampling problem (DGS) with the help of the CLWE oracle, and a quantum reduction from DGS back to BDD with a better approximation factor. Their final result is stated below.

Theorem 4.7 (Corollary 3.2, [BRST21]). *Let $\beta \in (0,1)$ and $\gamma \geq 2\sqrt{n}$ such that γ/β is polynomially bounded. Then there is a polynomial-time quantum reduction from $SIVP_\alpha$ and $GapSVP_\alpha$ to $CLWE_{\beta,\gamma}$ for some $\alpha = \tilde{O}(n/\beta)$.*

In other words, if there is no quantum polynomial-time algorithm that can solve $SIVP_\alpha$ or $GapSVP_\alpha$, then for any polynomially bounded m, we have

$$\left\{\mathbf{A} \leftarrow D^{n\times m}_{\mathbb{R},1}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D^m_{\mathbb{R},\beta} : \left(\mathbf{A},\ \gamma\mathbf{A}^T\mathbf{w} + \mathbf{e} \mod 1\right)\right\} \approx_c D^{n\times m}_{\mathbb{R},1} \times U(\mathbb{T}^m).$$

Let us remark that Gupte, Vafa and Vaikuntanathan [GVV22] have provided a classical reduction from LWE to CLWE. So the hardness of CLWE can also be based on the classical hardness of worst-case lattice problems.

4.3 CSIS: Standard worst-case to average-case reduction

We also want an analog of SIS on real values. We briefly review the core part of the standard worst-case to average-case reduction for SIS given by [MR07]. First recall the definition of SIS.

Definition 4.8 (Small Integer Solution (SIS)). *The SIS problem is: given an integer q, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a real number β, find a non-zero integer vector $\mathbf{z} \in \mathbb{Z}^m \backslash \{0\}$ such that $\|\mathbf{z}\|_2 \leq \beta$ and $\mathbf{Az} = 0 \bmod q$.*

Definition 4.9 (Average-case SIS). *For functions $q(n), m(n), \beta(n)$, let*

$$SIS_{q,m,\beta} = \{(q, U(\mathbb{Z}_q^{n\times m}), \beta)\}_n$$

be the probability ensemble over SIS instances $(q, \mathbf{A}, \beta)$ where $\mathbf{A}$ is uniformly sampled from $\mathbb{Z}_q^{n\times m}$.

The intuition of solving lattice problems (like GapSVP or SIVP) with the help of an SIS oracle is as follows:

- Given a basis $\mathbf{B}$ of a lattice, sample many pairs of vectors $\{(\mathbf{c}_i, \mathbf{y}_i)\}_{i=1}^m$ such that $\mathbf{c}_i$ is uniformly distributed in the parallelepiped $\mathcal{P}(\mathbf{B})$ and $\mathbf{y}_i$ is a lattice vector close to $\mathbf{c}_i$.

- Equally partition $\mathcal{P}(\mathbf{B})$ into q^n small parallelepipeds. Let $\tilde{\mathbf{c}}_i$ be the "lower-left" corner of the parallelepiped in which $\mathbf{c}_i$ lies. Assume q is sufficiently large to make $\tilde{\mathbf{c}}_i$ close to $\mathbf{c}_i$.
- Map $\tilde{\mathbf{c}}_i$ to $\mathbf{a}_i \in \mathbb{Z}_q^n$ in the canonical way. Use the SIS oracle (note that $\mathbf{a}_i$ is uniform since $\mathbf{c}_i$ is uniform) to find a short $\mathbf{z} \in \mathbb{Z}^m$ such that $\begin{pmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_m\end{pmatrix}\mathbf{z} = 0$. Then $\begin{pmatrix}\tilde{\mathbf{c}}_1 & \tilde{\mathbf{c}}_2 & \cdots & \tilde{\mathbf{c}}_m\end{pmatrix}\mathbf{z}$ will be a lattice vector. Notice that $\begin{pmatrix}\mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m\end{pmatrix}\mathbf{z}$ is also a lattice vector and $\mathbf{y}_i$ is close to $\tilde{\mathbf{c}}_i$, which results in a short lattice vector $\begin{pmatrix}\tilde{\mathbf{c}}_1 - \mathbf{y}_1 & \tilde{\mathbf{c}}_2 - \mathbf{y}_2 & \cdots & \tilde{\mathbf{c}}_m - \mathbf{y}_m\end{pmatrix}\mathbf{z}$.

If we are comfortable working with real numbers, it is more straightforward to directly map $\mathbf{c}_i$ to elements in $\mathbb{T}^n$ without discretizing into $\mathbb{Z}_q^n$. Now we may want the oracle to, given uniformly random $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m \in \mathbb{T}^n$, find small $\mathbf{z} \in \mathbb{Z}^m$ such that $\begin{pmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_m\end{pmatrix}\mathbf{z} \approx 0$. Note that the integer-coefficient linear combination is not required to be exactly 0 (almost surely unsolvable if exact equality is required), but within a bounded error from 0. The reduction tolerates this error, because originally there is an error between $\mathbf{c}_i$ and $\tilde{\mathbf{c}}_i$ as well. The above demand is formalized by:

Definition 4.10 (Continuous Small Integer Solution (CSIS))**.** *The CSIS problem is: given a matrix $\mathbf{A} \in \mathbb{T}^{n\times m}$ and two positive real numbers β, ζ, find a non-zero integer vector $\mathbf{z} \in \mathbb{Z}^m \backslash \{0\}$ such that $\|\mathbf{z}\|_2 \leq \beta$ and $\|\mathbf{A}\mathbf{z}\|_{\mathbb{T}} \leq \zeta$.*

Note that "continuous" means that the output values of the linear transformation are continuous, while "integer" means that the input values to the linear transformation are integers.

Similar to SIS, the existence of solutions is guaranteed for appropriate parameters.

Lemma 4.11. *For any $\zeta \in (0,1)$, $\mathbf{A} \in \mathbb{T}^{n\times m}$, and $\beta \geq \sqrt{m}\zeta^{-n/m}$, the CSIS instance $(\mathbf{A}, \beta, \zeta)$ admits a solution.*

Proof. Consider the set

$$S = \{\mathbf{z} = (z_1, z_2, \ldots, z_m) : z_i \in [0, \zeta^{-n/m}] \cap \mathbb{Z}\}.$$

Observe that $|S| > (\zeta^{-n/m})^m = \zeta^{-n}$. For each $\mathbf{z} \in S$, define the n-dimensional cube in $\mathbb{T}^n$:

$$\mathcal{B}_{\mathbf{z}} = \{\mathbf{u} \in \mathbb{T}^n : \|\mathbf{u} - \mathbf{A}\mathbf{z}\|_{\mathbb{T}} \leq \zeta/2\}.$$

The volume of each cube is ζ^n. Thus,

$$\sum_{\mathbf{z}\in S} Vol(\mathcal{B}_{\mathbf{z}}) = |S| \cdot \zeta^n > 1 = Vol(\mathbb{T}^n) \geq Vol\left(\cup_{\mathbf{z}\in S}\mathcal{B}_{\mathbf{z}}\right),$$

which implies that there exist $\mathbf{z}_1, \mathbf{z}_2 \in S, \mathbf{z}_1 \neq \mathbf{z}_2$ such that $\mathcal{B}_{\mathbf{z}_1} \cap \mathcal{B}_{\mathbf{z}_2} \neq \emptyset$. Let $\mathbf{u} \in \mathcal{B}_{\mathbf{z}_1} \cap \mathcal{B}_{\mathbf{z}_2}$, then

$$\|\mathbf{A}(\mathbf{z}_1 - \mathbf{z}_2)\|_{\mathbb{T}} \leq \|\mathbf{u} - \mathbf{A}\mathbf{z}_1\|_{\mathbb{T}} + \|\mathbf{u} - \mathbf{A}\mathbf{z}_2\|_{\mathbb{T}} \leq \frac{\zeta}{2} + \frac{\zeta}{2} = \zeta.$$

We also have $\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \leq \sqrt{m}\zeta^{-n/m} \leq \beta$ since $\mathbf{z}_1, \mathbf{z}_2 \in S$. Therefore, $\mathbf{z}_1 - \mathbf{z}_2$ is a solution to the CSIS instance $(\mathbf{A}, \beta, \zeta)$. □

Definition 4.12 (Average-case CSIS)**.** *For functions $m(n), \beta(n), \zeta(n)$, let*

$$CSIS_{m,\beta,\zeta} = \{(U(\mathbb{T}^{n\times m}), \beta, \zeta)\}_n$$

be the probability ensemble over CSIS instances $(\mathbf{A}, \beta, \zeta)$ where $\mathbf{A}$ is uniformly sampled from $\mathbb{T}^{n\times m}$.

Now we formally prove the average-case hardness of CSIS. Note that the following reduction shares similar ideas with the reduction for SBP in [VV25], but in CSIS the solution $\mathbf{z}$ can be any non-zero small integer vector (rather than in $\{-1,1\}^m$) and we only require $\|\mathbf{Az} \bmod 1\|_{\mathbb{T}}$ to be small (rather than $\|\mathbf{Az}\|$), which makes CSIS more general and potentially easier than SBP.

We first recall the definition of IncGDD, an intermediate lattice problem that is used in the reduction.

Definition 4.13 (Incremental Guaranteed Distance Decoding)**.** *An instance of IncGDD$^{\phi}_{\gamma,g}$ is an n-dimensional lattice basis $\mathbf{B}$, a set of n linearly independent lattice vectors $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$, a target point $\mathbf{t}$, and a number $r > \gamma(n) \cdot \phi(\mathcal{L}(\mathbf{B}))$. The goal is to output a lattice vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x} - \mathbf{t}\|_2 \leq (\|\mathbf{S}\|_2/g) + r$.*

We point out that the standard reduction from the worst-case IncGDD to the average-case SIS in [MR07] does not rely much on the specific structure of SIS. As long as the oracle in [MR07, Lemma 5.8] helps to construct the combining procedure as required, the reduction can be established. Below we state the abstraction of Lemma 5.8 and Theorem 5.9 in [MR07], with every specification of SIS replaced by a general average-case problem P. One may refer to [MR07] for the complete proof.

Theorem 4.14 (Abstraction of Lemma 5.8 & Theorem 5.9, [MR07])**.** *Consider a search problem P together with an instance distribution (ensemble) D.*
For any function $g(n)$, polynomially bounded functions $m(n), \beta(n)$, negligible function $\epsilon(n)$, if there is a p.p.t. oracle algorithm $\mathcal{A}^{\mathcal{F}}(\mathbf{B}, \mathbf{S}, \mathbf{C})$ that on input an n-dimensional lattice basis $\mathbf{B}$, a full-rank sub-lattice $\mathcal{L}(\mathbf{S}) \subset \mathcal{L}(\mathbf{B})$, m vectors $\mathbf{C} = \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_m\end{pmatrix} \in \mathcal{P}(\mathbf{B})^m$, makes a single oracle call $\mathcal{F}(\cdot) = \mathbf{z}$ and outputs $\mathbf{x} \in \mathbb{R}^n$ such that

- *If $\mathbf{C} \in \mathcal{P}(\mathbf{B})^m$ is uniformly distributed, then the query to $\mathcal{F}$ made by $\mathcal{A}$ follows the distribution D;*
- *If $\mathcal{F}$ successfully solves problem P, then $\mathbf{z} \in \mathbb{Z}^m \setminus \{0\}$, $\|\mathbf{z}\|_2 \leq \beta$, $\mathbf{x} \in \mathcal{L}(\mathbf{B})$, $\|\mathbf{x} - \mathbf{Cz}\|_2 \leq \|\mathbf{S}\|_2/g$,*

then there is a p.p.t. reduction from worst-case IncGDD$^{\eta_\epsilon}_{\gamma,g}$ for $\gamma(n) = \beta(n)\sqrt{n}$ to average-case P w.r.t. random instance from D. Here η_ϵ denotes the smoothing parameter of a lattice.

It remains to construct such an oracle algorithm $\mathcal{A}^{\mathcal{F}}$ for CSIS.

Lemma 4.15. *For any $n, m \in \mathbb{N}^+$, $\beta, \zeta \in \mathbb{R}^+$ such that $\beta, m \in \mathsf{poly}(n)$, $\beta \geq \sqrt{m}\zeta^{-n/m}$, there is a p.p.t. oracle algorithm $\mathcal{A}^{\mathcal{F}}(\mathbf{B}, \mathbf{S}, \mathbf{C})$ that on input an n-dimensional lattice basis $\mathbf{B}$, a full-rank sub-lattice $\mathcal{L}(\mathbf{S}) \subset \mathcal{L}(\mathbf{B})$, m vectors $\mathbf{C} = \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_m\end{pmatrix} \in \mathcal{P}(\mathbf{B})^m$, makes a single oracle call to $\mathcal{F}(\mathbf{A}, \beta, \zeta) = \mathbf{z}$ and outputs a vector $\mathbf{x} \in \mathbb{R}^n$ such that*

- *If $\mathbf{C} \in \mathcal{P}(\mathbf{B})^m$ is uniformly distributed, then the query matrix $\mathbf{A} \in \mathbb{T}^{n \times m}$ is also uniformly distributed.*
- *If $\mathcal{F}$ successfully solves the CSIS instance $(\mathbf{A}, \beta, \zeta)$, then $\|\mathbf{z}\|_2 \leq \beta$, $\mathbf{x} \in \mathcal{L}(\mathbf{B})$, $\|\mathbf{x} - \mathbf{Cz}\|_2 \leq n\zeta\|\mathbf{S}\|_2$.*

Proof. $\mathcal{A}^{\mathcal{F}}(\mathbf{B}, \mathbf{S}, \mathbf{C})$ works as follows:

1. Generate m uniformly random lattice vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B}) \bmod \mathcal{P}(\mathbf{S})$.
2. Let $\mathbf{W} = \begin{pmatrix}\mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_m\end{pmatrix}$ where $\mathbf{w}_i = \mathbf{v}_i + \mathbf{c}_i \bmod \mathcal{P}(\mathbf{S})$.

3. Let $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_m \end{pmatrix}$ where $\mathbf{a}_i = \mathbf{S}^{-1}\mathbf{w}_i \in [0,1)^n$.
4. View $\mathbf{A}$ as an element in $\mathbb{T}^{n\times m}$. Invoke $\mathcal{F}$ on input $\mathbf{A}$ and obtain $\mathbf{z} \in \mathbb{Z}^m$.
5. Output $\mathbf{x} = \mathbf{C}\mathbf{z} - \mathbf{W}\mathbf{z} + \mathbf{S}\lfloor \mathbf{A}\mathbf{z} \rceil$, where $\lfloor \cdot \rceil$ rounds each component to the nearest integer.

Now verify the two requirements:

- If $\mathbf{c}_i$ is uniformly distributed on $\mathcal{P}(\mathbf{B})$, then $\mathbf{w}_i = \mathbf{v}_i + \mathbf{c}_i \bmod \mathcal{P}(\mathbf{S})$ is uniformly distributed on $\mathcal{P}(\mathbf{S})$ because $\mathbf{v}_i$ is uniformly distributed on $\mathcal{L}(\mathbf{B}) \bmod \mathcal{P}(\mathbf{S})$. Thus, $\mathbf{a}_i = \mathbf{S}^{-1}\mathbf{w}_i$ is uniformly distributed on $\mathbb{T}^n$.
- If $\mathcal{F}$ successfully solves the CSIS instance $(\mathbf{A}, \beta, \zeta)$, then $\|\mathbf{z}\|_2 \le \beta$, $\|\mathbf{A}\mathbf{z} \bmod 1\|_{\mathbb{T}} \le \zeta$. Since $(\mathbf{C}-\mathbf{W})\mathbf{z} \in \mathcal{L}(\mathbf{B})$, $\mathbf{S}\lfloor \mathbf{A}\mathbf{z} \rceil \in \mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$, we have $\mathbf{x} \in \mathcal{L}(\mathbf{B})$. Finally,

$$\|\mathbf{x} - \mathbf{C}\mathbf{z}\|_2 = \|\mathbf{W}\mathbf{z} - \mathbf{S}\lfloor \mathbf{A}\mathbf{z} \rceil\|_2 = \|\mathbf{S}(\mathbf{A}\mathbf{z} - \lfloor \mathbf{A}\mathbf{z} \rceil)\|_2 \le n\|\mathbf{S}\|_2 \cdot \|\mathbf{A}\mathbf{z} \bmod 1\|_{\mathbb{T}} \le n\zeta\|\mathbf{S}\|_2.$$

□

By combining Theorem 4.14 and Lemma 4.15, we get our hardness result of CSIS as follows:

Theorem 4.16. *For any $n, m \in \mathbb{N}^+$, $g, \beta, \zeta, \epsilon \in \mathbb{R}^+$, such that $\beta, m \in \mathsf{poly}(n)$, $\beta \ge \sqrt{m}\zeta^{-n/m}$, $\zeta \le \frac{1}{ng}$, and ϵ is negligible in n, there is a p.p.t. reduction from worst-case $\mathrm{IncGDD}^{\eta_\epsilon}_{\gamma,g}$ for $\gamma = \beta\sqrt{n}$ to average-case $\mathrm{CSIS}_{m,\beta,\zeta}$.*

Finally, we derive the average-case hardness of CSIS based on lattice assumptions, with the help of standard reductions between worst-case lattice problems (see [MR07], section 5.3).

Corollary 4.17. *For any $n, m \in \mathbb{N}^+$, $\beta, \zeta \in \mathbb{R}^+$, such that $\beta, m \in \mathsf{poly}(n)$, $\beta \ge \sqrt{m}\zeta^{-n/m}$, $\zeta \le \frac{1}{8n}$, for any $\gamma = \omega(\beta\sqrt{n\log n})$, solving $\mathrm{CSIS}_{m,\beta,\zeta}$ on average is at least as hard as solving any of the following worst-case problems:*

- SIVP_γ,
- $\mathrm{GDD}^{\lambda_n}_\gamma$,
- GapCRP_γ.

Note that the above reductions hold for any $\beta > 0$, while $\beta \ge \sqrt{m}\zeta^{-n/m}$ is only needed to ensure that CSIS solutions exist. Dropping in $\zeta = \frac{1}{8n}$, $m = \Theta(n\log n)$, $\beta = \Theta(\sqrt{m})$ gives $\gamma = \omega(n\log n)$.

4.4 CSIS′: A variant of CSIS without modulo 1

Sometimes it is inconvenient to work on $\mathbb{T}$ as we need to frequently take the fractional part of a real number. Hence we naturally want a variant of CSIS with numbers and operations over the plain real number field $\mathbb{R}$.

Definition 4.18 (Continuous Small Integer Solution without modulo 1)**.** *The CSIS′ problem is: given a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ and two real numbers β, ζ, find a non-zero integer vector $\mathbf{z} \in \mathbb{Z}^m \backslash \{0\}$ such that $\|\mathbf{z}\|_2 \le \beta$ and $\|\mathbf{A}\mathbf{z}\| \le \zeta$.*

Intuitively, $\|\mathbf{Az}\| \leq \zeta$ is a stronger requirement than $\|\mathbf{Az}\|_{\mathbb{T}} \leq \zeta$, so one can imagine a rather straightforward reduction. Before that, we first derive the guarantee for the existence of a solution, which holds for stricter parameter restrictions.

Lemma 4.19. *For any $\zeta > 0$, $m > n$, $\mathbf{A} \in \mathbb{R}^{n \times m} \setminus \{0\}$, and $\beta \geq \sqrt{m}\lceil(2m\|\mathbf{A}\|/\zeta)^{\frac{n}{m-n}}\rceil$, the CSIS' instance $(\mathbf{A}, \beta, \zeta)$ admits a solution.*

Proof. Assume $\|\mathbf{A}\| > \zeta$, otherwise a solution $\mathbf{z} = (1, 0, \ldots, 0)$ trivially exists.
Let $b = \lceil(2m\|\mathbf{A}\|/\zeta)^{\frac{n}{m-n}}\rceil$. Consider the set

$$S = \{\mathbf{z} = (z_1, z_2, \ldots, z_m) : z_i \in \{0, 1, \ldots, b-1\}\}.$$

Observe that $|S| = b^m$. For each $\mathbf{z} \in S$, define the n-dimensional cube in $\mathbb{R}^n$:

$$\mathcal{B}_{\mathbf{z}} = \{\mathbf{u} \in \mathbb{R}^n : \|\mathbf{u} - \mathbf{Az}\| \leq \zeta/2\}.$$

The volume of each cube is ζ^n. Thus, $\sum_{\mathbf{z} \in S} Vol(\mathcal{B}_{\mathbf{z}}) = |S| \cdot \zeta^n = b^m \cdot \zeta^n$. On the other hand, since $\|\mathbf{Az}\| \leq m\|\mathbf{A}\| \cdot \|\mathbf{z}\| \leq (b-1)m\|\mathbf{A}\|$,

$$\mathcal{B}_{\mathbf{z}} \subseteq \mathcal{B}_\infty^n((b-1)m\|\mathbf{A}\| + \zeta/2) \subsetneq \mathcal{B}_\infty^n(bm\|\mathbf{A}\|).$$

Therefore,

$$Vol(\cup_{\mathbf{z} \in S}\mathcal{B}_{\mathbf{z}}) < Vol(\mathcal{B}_\infty^n(bm\|\mathbf{A}\|)) = (2bm\|\mathbf{A}\|)^n \leq b^m \cdot \zeta^n = \sum_{\mathbf{z} \in S} Vol(\mathcal{B}_{\mathbf{z}}),$$

which implies that there exist $\mathbf{z}_1, \mathbf{z}_2 \in S, \mathbf{z}_1 \neq \mathbf{z}_2$ such that $\mathcal{B}_{\mathbf{z}_1} \cap \mathcal{B}_{\mathbf{z}_2} \neq \emptyset$. Let $\mathbf{u} \in \mathcal{B}_{\mathbf{z}_1} \cap \mathcal{B}_{\mathbf{z}_2}$, then

$$\|\mathbf{A}(\mathbf{z}_1 - \mathbf{z}_2)\| \leq \|\mathbf{u} - \mathbf{Az}_1\| + \|\mathbf{u} - \mathbf{Az}_2\| \leq \frac{\zeta}{2} + \frac{\zeta}{2} = \zeta.$$

We also have $\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \leq \sqrt{m}b \leq \beta$ since $\mathbf{z}_1, \mathbf{z}_2 \in S$. Therefore, $\mathbf{z}_1 - \mathbf{z}_2$ is a solution to the CSIS' instance $(\mathbf{A}, \beta, \zeta)$. □

Regarding the average-case problem, we consider general distributions over $\mathbb{R}$ from which each entry of input matrix $\mathbf{A}$ is sampled.

Definition 4.20 (Average-case CSIS')**.** *For distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_n$ over $\mathbb{R}$ and functions $m(n), \beta(n), \zeta(n)$, let*

$$CSIS'^{\mathcal{D}}_{m,\beta,\zeta} = \{(\mathcal{D}_n^{n \times m}, \beta, \zeta)\}_n$$

be the probability ensemble over CSIS' instances $(\mathbf{A}, \beta, \zeta)$ where each entry of $\mathbf{A}$ is sampled from $\mathcal{D}$.

We will reduce CSIS to CSIS' for distributions satisfying the following definition:

Definition 4.21 (Sampleable fractional-uniform distribution)**.** *A distribution ensemble $\mathcal{D} = \{\mathcal{D}_n\}_n$ over $\mathbb{R}$ is said to be sampleable fractional-uniform if:*

- *The fractional part of a sample from $\mathcal{D}$ is statistically close to uniform, i.e.,*

$$\{x \leftarrow \mathcal{D}_n : x \bmod 1\} \approx_s U(\mathbb{T}).$$

- *There is a p.p.t. algorithm $\mathcal{S}$ that on input a $U(\mathbb{T})$ sample, outputs a sample statistically close to $\mathcal{D}$ while preserving the fractional part, i.e.,*

$$\{x \leftarrow U(\mathbb{T}) : \mathcal{S}(1^n, x)\} \approx_s \mathcal{D}_n,$$

$$\forall x \in \mathbb{T}, \mathcal{S}(1^n, x) \equiv x \bmod 1.$$

Theorem 4.22. *For polynomially bounded $m(n)$, if $\mathcal{D}$ is sampleable fractional-uniform, then there is a p.p.t. reduction from average-case $\mathrm{CSIS}_{m,\beta,\zeta}$ to average-case $\mathrm{CSIS}'^{\mathcal{D}}_{m,\beta,\zeta}$.*

Proof. Let $\mathcal{S}$ be the sampling algorithm for $\mathcal{D}$. Suppose $\mathcal{A}'$ solves $\mathrm{CSIS}'^{D}_{m,\beta,\zeta}$ on average with non-negligible probability. We construct $\mathcal{A}$ solving $\mathrm{CSIS}_{m,\beta,\zeta}$ as follows:

1. On input $(\mathbf{A}, \beta, \zeta)$ where $\mathbf{A} \leftarrow U(\mathbb{T}^{n\times m})$, $\mathcal{A}$ invokes $\mathcal{S}$ on each entry of $\mathbf{A}$ and obtains $\mathbf{A}' \in \mathbb{R}^{n\times m}$ with $\mathbf{A} \equiv \mathbf{A}' \bmod 1$.
2. $\mathcal{A}$ runs $\mathcal{A}'$ on input $(\mathbf{A}', \beta, \zeta)$ and outputs whatever $\mathcal{A}'$ outputs.

By the definition of $\mathcal{S}$, the input $\mathbf{A}'$ to $\mathcal{A}'$ is statistically close to $\mathcal{D}_n^{n\times m}$, so $\mathcal{A}'$, with non-negligible probability, outputs $\mathbf{z} \in \mathbb{Z}^m \backslash \{0\}$ with $\|\mathbf{z}\|_2 \le \beta$ and $\|\mathbf{A}'\mathbf{z}\| \le \zeta$. Notice that $\mathbf{A} \equiv \mathbf{A}' \bmod 1$ implies $\mathbf{A}\mathbf{z} \equiv \mathbf{A}'\mathbf{z} \bmod 1$, so $\|\mathbf{A}\mathbf{z}\|_{\mathbb{T}} \le \|\mathbf{A}'\mathbf{z}\| \le \zeta$. Thus, $\mathbf{z}$ is a solution to the CSIS instance $(\mathbf{A}, \beta, \zeta)$, which means that $\mathcal{A}$ solves $\mathrm{CSIS}_{m,\beta,\zeta}$ on average with non-negligible probability. □

We mainly care about two kinds of distributions, uniform and Gaussian, which are both sampleable fractional-uniform under appropriate parameters.

Proposition 4.23. *The distribution ensemble $\mathcal{D} = \{\mathcal{D}_n = U([0,1))\}_n$ is sampleable fractional-uniform.*

Proposition 4.24. *For any polynomially bounded function $s(n) \ge \sqrt{n}$, the distribution ensembles*

$$\mathcal{D} = \{\mathcal{D}_n = D_{\mathbb{R},s}\}_n, \quad \tilde{\mathcal{D}} = \{\tilde{\mathcal{D}}_n = D_{(-s\sqrt{n}, s\sqrt{n}),s}\}_n$$

are both sampleable fractional-uniform.

Proof sketch. Since s is greater than the smoothing parameter of $\mathbb{Z}$ w.r.t. some negligible function, $\mathcal{D}_n \bmod 1$ is statistically close to $U(\mathbb{T})$, and so is $\tilde{\mathcal{D}}_n \bmod 1$. Given $x \in \mathbb{T}$, the sampler samples from the restricted discrete Gaussian $D_{(x+\mathbb{Z})\cap(-s\sqrt{n}, s\sqrt{n}),s}$ via rejection sampling. □

Corollary 4.25. *For polynomially bounded functions $m(n), \beta(n), s(n) \ge \sqrt{n}$, and $\zeta(n) \le \frac{1}{8n}$ satisfying the requirement in Lemma 4.19 for solutions to exist, for any $\gamma(n) = \omega(\beta\sqrt{n \log n})$, solving any of $\mathrm{CSIS}'^{U([0,1))}_{m,\beta,\zeta}, \mathrm{CSIS}'^{D_{\mathbb{R},s}}_{m,\beta,\zeta}, \mathrm{CSIS}'^{\tilde{D}_{\mathbb{R},s}}_{m,\beta,\zeta}$ on average is at least as hard as solving any of the following worst-case problems:*

- SIVP_γ ,
- $\mathrm{GDD}^{\lambda_n}_\gamma$,
- GapCRP_γ .

For example, for $\mathrm{CSIS}'^{U([0,1))}_{m,\beta,\zeta}$, by Lemma 4.19, solutions exist when $\beta \ge \sqrt{m}\lceil(2m/\zeta)^{\frac{n}{m-n}}\rceil$. Dropping in $\zeta = \frac{1}{8n}$, $m = \Theta(n \log n)$, $\beta = O(\sqrt{m})$ gives $\gamma = \omega(n \log n)$.

4.5 From CLWE to CSIS'

In this section, we complete our discussions about CSIS/CSIS' by relating them to CLWE. We want to show that the hardness of CSIS can be based on the decisional CLWE assumption (see Theorem 4.7) by distinguishing the two ensembles with the help of a CSIS oracle. Following the standard idea, on input $(\mathbf{A}, \mathbf{y}^T)$, we invoke the CSIS oracle to obtain a small $\mathbf{z}$ with small $\|\mathbf{Az}\|_{\mathbb{T}}$ and expect $\mathbf{y}^T\mathbf{z}$ to be small as well when $(\mathbf{A}, \mathbf{y}^T = \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T \bmod 1)$ is sampled from the CLWE distribution. Unfortunately, this does not work since in general

$$\gamma\mathbf{w}^T\mathbf{Az} \not\equiv \gamma\mathbf{w}^T(\mathbf{Az} \bmod 1) \mod 1,$$

which is determined by the fundamental difference between $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ and $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$: $\mathbb{Z}$ is not an ideal of $\mathbb{R}$, so real multiplication does not descend to a well-defined multiplication on $\mathbb{T}$.

Consequently, we require a stronger CSIS' oracle to give $\mathbf{z}$ with small $\|\mathbf{Az}\|$, rather than merely small $\|\mathbf{Az}\|_{\mathbb{T}}$. Moreover, each entry of $\mathbf{A}$ is sampled from the Gaussian distribution $D_{\mathbb{R},1}$, hence we consider CSIS' w.r.t. Gaussian distributions.

We remark that a similar reduction was given in [BRVV26] from CLWE to CHV (Contracting Hypergrid Vector), a problem that has the same structure as CSIS' but with a different requirement on the solution.

Theorem 4.26. *For polynomially bounded functions $m(n), \beta(n), \gamma(n) \geq 2\sqrt{n}$ and $\zeta(n) \leq \frac{1}{8n}$, assume that*

$$\left\{\mathbf{A} \leftarrow D_{\mathbb{R},1}^{n\times m}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D_{\mathbb{R},\frac{1}{10\beta}}^{m} : \left(\mathbf{A}, \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T \bmod 1\right)\right\} \approx_c D_{\mathbb{R},1}^{n\times m} \times U(\mathbb{T}^m),$$

then there is no p.p.t. algorithm that can solve average-case CSIS$'^{D_{\mathbb{R},\gamma}}_{m,\beta,\zeta}$ with non-negligible probability.

Proof. Assume $n \geq 2$. Suppose $\mathcal{A}'$ solves average-case CSIS$'^{D_{\mathbb{R},\gamma}}_{m,\beta,\zeta}$ with non-negligible probability δ. We construct a distinguisher $\mathcal{D}$ as follows:

1. On input $(\mathbf{A}, \mathbf{y}^T)$, invoke $\mathcal{A}'$ on input $(\gamma\mathbf{A}, \beta, \zeta)$ and obtain $\mathbf{z} \in \mathbb{Z}^m\backslash\{0\}$. Check whether $\mathbf{z}$ is a solution to the CSIS' instance $(\gamma\mathbf{A}, \beta, \zeta)$.
2. Compute $\mathbf{y}^T\mathbf{z} \bmod 1$.
3. If $\mathbf{z}$ is indeed a solution and $\|\mathbf{y}^T\mathbf{z} \bmod 1\|_{\mathbb{T}} \leq 0.2$, then output 1. Otherwise, output 0.

When $\mathbf{A}$ follows $D_{\mathbb{R},1}^{n\times m}$, $\gamma\mathbf{A}$ follows $D_{\mathbb{R},\gamma}^{n\times m}$. The input to $\mathcal{A}'$ follows the distribution specified by CSIS$'^{D_{\mathbb{R},\gamma}}_{m,\beta,\zeta}$, hence $\mathcal{A}'$ successfully outputs a solution with probability δ.

When $(\mathbf{A}, \mathbf{y}^T)$ is sampled from $D_{\mathbb{R},1}^{n\times m} \times U(\mathbb{T}^m)$, $\mathbf{y}^T$ is independent of $\mathbf{z}$. Notice that for any fixed $\mathbf{z} \in \mathbb{Z}^m\backslash\{0\}$, $\mathbf{y}^T\mathbf{z} \bmod 1$ follows $U(\mathbb{T})$ when $\mathbf{y}^T$ follows $U(\mathbb{T}^m)$, so $\|\mathbf{y}^T\mathbf{z} \bmod 1\|_{\mathbb{T}} \leq 0.2$ happens with probability 0.4. Thus,

$$\Pr[\mathbf{A} \leftarrow D_{\mathbb{R},1}^{n\times m}, \mathbf{y}^T \leftarrow U(\mathbb{T}^m) : \mathcal{D}(\mathbf{A}, \mathbf{y}^T) = 1] = 0.4\delta.$$

On the other hand, when $\mathbf{y}^T = \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T \bmod 1$ where $\mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D_{\mathbb{R},\frac{1}{10\beta}}^{m}$, we claim that $\mathcal{D}$ outputs 1 with a much higher probability. Conditioned on $\mathbf{z}$ being a solution to the CSIS' instance $(\gamma\mathbf{A}, \beta, \zeta)$, we have $\|\mathbf{z}\|_2 \leq \beta, \|\gamma\mathbf{Az}\| \leq \zeta$. Thus,

$$|\gamma\mathbf{w}^T\mathbf{Az}| \leq \|\mathbf{w}\|_2 \cdot \|\gamma\mathbf{Az}\|_2 \leq \zeta\sqrt{n} < 0.1.$$

When $\mathbf{e}$ follows $D^m_{\mathbb{R},\frac{1}{10\beta}}$, $\mathbf{e}^T\mathbf{z}$ follows $D_{\mathbb{R},\frac{\|\mathbf{z}\|_2}{10\beta}}$, where $\frac{\|\mathbf{z}\|_2}{10\beta} \leq 0.1$. One can estimate that a $D_{\mathbb{R},s}$ sample falls in $[-s,s]$ with probability larger than 0.9. Thus, for any fixed $\mathbf{A}$ and solution $\mathbf{z}$,

$$\Pr[\mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D^m_{\mathbb{R},\frac{1}{10\beta}} : |\gamma \mathbf{w}^T\mathbf{A}\mathbf{z} + \mathbf{e}^T\mathbf{z}| \leq 0.2] > 0.9.$$

Since $\mathbf{y}^T = \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T \bmod 1 \in \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T + \mathbb{Z}^m$, we have $\mathbf{y}^T\mathbf{z} \in \gamma\mathbf{w}^T\mathbf{A}\mathbf{z} + \mathbf{e}^T\mathbf{z} + \mathbb{Z}$. As a result, $|\gamma\mathbf{w}^T\mathbf{A}\mathbf{z} + \mathbf{e}^T\mathbf{z}| \leq 0.2$ implies $\|\mathbf{y}^T\mathbf{z} \bmod 1\|_{\mathbb{T}} \leq 0.2$. Therefore,

$$\Pr[\mathbf{A} \leftarrow D^{n\times m}_{\mathbb{R},1}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D^m_{\mathbb{R},\frac{1}{10\beta}} : \mathcal{D}(\mathbf{A}, \gamma\mathbf{w}^T\mathbf{A} + \mathbf{e}^T \bmod 1) = 1] > 0.9\delta,$$

which means $\mathcal{D}$ distinguishes the above two ensembles with distinguishing advantage at least 0.5δ. □

Together with Theorem 4.7, the average-case hardness of CSIS' can be based on worst-case lattice problems in another way, but with a larger approximation factor. Figure 1 summarizes the relations among CSIS, CSIS', CLWE and lattice problems.

5 Real Value Collision Resistant Hash Functions: Constructions

In this section, we aim to design hashing schemes that admit a continuous message space $\mathcal{M}$ and achieve approximate collision resistance. Since any bounded message space in $\mathbb{R}$ can be embedded into $[0,1)$ via an affine map, we assume $\mathcal{M} = [0,1)$ for simplicity.

Our constructions are built on the widely used framework proposed by Ajtai [Ajt96], which describes the following hash function family over discrete message space $\mathcal{M} = \{0,1,\ldots,b-1\}$:

Scheme 5.1. *The discrete hash function family for input space $\{0,1,\ldots,b-1\}$ is defined as:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$*: On input the security parameter λ, input length m, output length n, choose a modulus B appropriately. Uniformly sample a matrix $\mathbf{A}$ from $\mathbb{Z}_B^{n\times m}$ and output $h = (B, \mathbf{A})$.*
- $\mathsf{Hash}(h, \mathbf{x})$*: On input the hashing parameter $h = (B, \mathbf{A})$ and a message $\mathbf{x}$, output*

$$\mathbf{y} = \mathbf{A}\mathbf{x} \bmod B.$$

Observe that a colliding pair $(\mathbf{x}, \mathbf{x}')$ with $\mathsf{Hash}(h, \mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}')$ directly leads to a solution $\mathbf{z} = \mathbf{x} - \mathbf{x}'$ with $\mathbf{A}\mathbf{z} = 0 \bmod B$ for the SIS problem instance $(B, \mathbf{A}, \beta = b\sqrt{m})$. Thus, the collision resistance property of this family of hash functions can be based on the average-case hardness of $\mathrm{SIS}_{B,m,\beta}$.

The major challenge of constructing Ajtai-style collision resistant hash functions on real inputs comes from the wonderful analytical properties of linear transformations in the field $\mathbb{R}$. In particular, for any $n \times m$ ($n < m$) matrix $\mathbf{A}$, the linear transformation $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ defined on $\mathbb{R}^m$ has a nontrivial null space, which contains vectors of arbitrary length. If we directly apply this linear transformation on input messages, then one can easily find collisions by solving the linear system $\mathbf{A}\mathbf{z} = 0$, as long as the message space contains an open set (see section 9 for further discussions).

Consequently, we believe that it is necessary to discretize the messages. By cutting $\mathcal{M} = [0,1)$ into segments of length $1/b$ and rounding the messages according to the segmentation, we get the following hash function family:

Scheme 5.2. *The hash function family with simple rounding as discretization is defined as:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$*: On input the security parameter* λ*, input length* m*, output length* n*, choose a rounding parameter* $b \in \mathbb{N}^*$ *and a modulus* $B > b$ *appropriately. Uniformly sample a matrix* $\mathbf{A}$ *from* $\mathbb{Z}_B^{n\times m}$ *and output* $h = (b, B, \mathbf{A})$*.*
- $\mathsf{Hash}(h, \mathbf{x}, \ell)$*: On input the hashing parameter* $h = (b, B, \mathbf{A})$*, a message* $\mathbf{x} \in \mathcal{M}^m$*, and a precision length* ℓ *(which is useless here), output*

$$\mathbf{y} = \mathbf{A}\lfloor b\mathbf{x} \rfloor \bmod B,$$

where $\lfloor \cdot \rfloor$ *is the component-wise floor function.*

The precision length ℓ does not appear in the hashing algorithm since all the operations are in discrete space after the rounding. This means that it trivially satisfies perfect consistency with $\mathsf{IdealHash} = \mathsf{Hash}$. Below we omit the precision length and write $h(\mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}, \ell)$.

The above hashing can be viewed as adding an encoding phase before applying Scheme 5.1. Thus, the collision resistance property of this scheme inherits from that of the original one, unless the encodings are identical.

Proposition 5.3. *Based on the average-case hardness of* $SIS_{B,m,\beta}$ *where* $\beta = b\sqrt{m}$*, Scheme 5.2 satisfies*

$$\Pr\left[\begin{array}{c} h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n) \\ (\mathbf{x}, \mathbf{x}') \leftarrow \mathcal{A}(1^\lambda, h) \end{array} \;:\; \|\mathbf{x} - \mathbf{x}'\| \geq \frac{1}{b} \wedge h(\mathbf{x}) = h(\mathbf{x}')\right] \leq \mathsf{negl}(\lambda)$$

for any efficient adversary $\mathcal{A}$*.*

Proof. If there exists an adversary $\mathcal{A}$ which gives $(\mathbf{x}, \mathbf{x}')$ with $\|\mathbf{x} - \mathbf{x}'\| \geq \frac{1}{b}$ and $h(\mathbf{x}) = h(\mathbf{x}')$ with non-negligible probability, then one can use $\mathcal{A}$ to solve average-case $\mathrm{SIS}_{B,m,\beta}$ as follows:

Given an SIS instance $(B, \mathbf{A}, \beta)$ where $\mathbf{A} \leftarrow U(\mathbb{Z}_B^{n\times m})$, run $\mathcal{A}$ on input $(1^\lambda, (b = \beta/\sqrt{m}, B, \mathbf{A}))$. If $\mathcal{A}$ successfully outputs $(\mathbf{x}, \mathbf{x}')$ with $\|\mathbf{x} - \mathbf{x}'\| \geq \frac{1}{b}$ and $h(\mathbf{x}) = h(\mathbf{x}')$, output $\mathbf{z} = \lfloor b\mathbf{x} \rfloor - \lfloor b\mathbf{x}' \rfloor$. Otherwise fail.

$(b, B, \mathbf{A})$ follows the same distribution as $h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n)$. As long as $\mathcal{A}$ succeeds, $\mathbf{z}$ satisfies $\mathbf{z} = \lfloor b\mathbf{x} \rfloor - \lfloor b\mathbf{x}' \rfloor \neq 0$ since $\|b\mathbf{x} - b\mathbf{x}'\| \geq 1$, $\|\mathbf{z}\|_2 < b\sqrt{m} = \beta$, and $\mathbf{A}\mathbf{z} = h(\mathbf{x}) - h(\mathbf{x}') = 0 \bmod B$, which means $\mathbf{z}$ is a solution to the SIS problem. Thus, the above algorithm solves $\mathrm{SIS}_{B,m,\beta}$ on average. □

Corollary 5.4. *Based on the average-case hardness of* $SIS_{B,m,\beta}$ *where* $\beta = b\sqrt{m}$*, Scheme 5.2 satisfies* $f(\cdot)$*-approximate collision resistance where*

$$f(\ell) = \begin{cases} 1 & (\ell \leq \log b) \\ 0 & (\ell > \log b) \end{cases}.$$

One can easily find collisions with distance $\|\mathbf{x} - \mathbf{x}'\| < \frac{1}{b}$, so this $f(\cdot)$ is tight. In other words, Scheme 5.2 provides no collision resistance guarantee when the precision length exceeds the threshold $\log b$. Note that the hardness of $\mathrm{SIS}_{B,m,\beta}$ needs b to be polynomially bounded, which results in an unsatisfactory threshold of $O(\log \lambda)$.

In the following subsections, we propose three different hashing schemes that achieve some desirable properties. We first define a more fine-grained discretization method in section 5.1.

Then, in section 5.2, we utilize it and perform the hashing in a discrete space. This scheme enjoys a wonderful collision resistance guarantee but no consistency. In section 5.3, we propose a hashing scheme that converts the discrete outputs back to real numbers and achieves both consistency and collision resistance, but the consistency error and collision resistance threshold are coupled, which is not ideal for constructing commitment schemes. Finally, in section 5.4, we propose a new discretization method that works natively on real numbers. Together with the average-case hard problems we developed in section 4, this construction avoids modular arithmetic and decouples the consistency error and collision resistance threshold. See Table 1 for a comparison.

5.1 Base Expansion

We point out that although those discontinuity points of the rounding function $\lfloor bx \rfloor$ break the overall linear structure, the function is still too simple (actually constant) below the scale of $\frac{1}{b}$, which gives adversaries the chance to find collisions.

Notice that the rounding function $\lfloor bx \rfloor$ can be viewed as taking the first digit of the base-b expansion of x and ignoring the rest. This inspires us to also take the trailing digits into account in order to achieve collision resistance at smaller scales. We first define the base-b expansion of a real number and derive some useful properties.

Definition 5.5 (Base-b expansion)**.** *For $b \in \mathbb{Z}_{\geq 2}$ and $x \in [0,1)$, let*

$$dig_b(x) = (x^{(1)}, x^{(2)}, x^{(3)}, \dots) \in \{0, 1, \dots, b-1\}^{\mathbb{N}^*}$$

be the base-b expansion of x such that

$$x = (0.x^{(1)}x^{(2)}x^{(3)}\dots)_b = \sum_{k=1}^{+\infty} x^{(k)} \cdot b^{-k}.$$

To make it well-defined, numbers are always expressed in terminating form with trailing zeros if possible. The infinite sequence $(x^{(1)}, x^{(2)}, x^{(3)}, \dots)$ can also be viewed as a map from $\mathbb{N}^*$ to $\{0, 1, \dots, b-1\}$, so sometimes we use $dig_b(x)(k)$ to denote $x^{(k)}$.

The inverse of dig_b is defined as

$$dig_b^{-1}((x^{(1)}, x^{(2)}, x^{(3)}, \dots)) = \sum_{k=1}^{+\infty} x^{(k)} \cdot b^{-k} = \begin{pmatrix} x^{(1)} & x^{(2)} & x^{(3)} & \cdots \end{pmatrix} \begin{pmatrix} b^{-1} \\ b^{-2} \\ b^{-3} \\ \vdots \end{pmatrix}.$$

To assist subsequent discussions, we measure the scale at which two real numbers are differentiable:

Definition 5.6 (Differentiable scale)**.** *For $b \in \mathbb{Z}_{\geq 2}$, $x \neq y \in [0,1)$, suppose $dig_b(x) = (x^{(1)}, x^{(2)}, x^{(3)}, \dots)$, $dig_b(y) = (y^{(1)}, y^{(2)}, y^{(3)}, \dots)$, let*

$$dif_b(x, y) = \min\{k | x^{(k)} \neq y^{(k)}\}.$$

The relation between distance and differentiable scale is described by:

Proposition 5.7. *For $b \in \mathbb{Z}_{\geq 2}$, $x \neq y \in [0,1)$,*

$$dif_b(x, y) \leq \lceil -\log_b |x - y| \rceil.$$

Proof. Suppose $dig_b(x) = (x^{(1)}, x^{(2)}, x^{(3)}, \dots)$, $dig_b(y) = (y^{(1)}, y^{(2)}, y^{(3)}, \dots)$, $k = \lceil -\log_b |x-y| \rceil$. If $dif_b(x,y) > k$, then by definition $x^{(i)} = y^{(i)}$ for every $1 \le i \le k$. Thus,

$$|x-y| = \left| \sum_{i=k+1}^{+\infty} (x^{(i)} - y^{(i)}) \cdot b^{-i} \right| \le \sum_{i=k+1}^{+\infty} \left| x^{(i)} - y^{(i)} \right| \cdot b^{-i} < \sum_{i=k+1}^{+\infty} (b-1) \cdot b^{-i} = b^{-k}.$$

which implies $k < -\log_b |x-y|$, a contradiction. □

Note that we have no bounds on the reverse direction. For example, $|(b^{-1} - b^{-k}) - b^{-1}| = b^{-k}$ can be arbitrarily small when k goes to infinity, but $dif_b(b^{-1} - b^{-k}, b^{-1}) = 1$ always holds.

The above definitions and propositions can readily be extended to vectors.

Definition 5.8 (Base-b expansion for vectors)**.** *For $b \in \mathbb{Z}_{\ge 2}$ and $\mathbf{x} = (x_1, x_2, \dots, x_m)^T \in [0,1)^m$, let*

$$dig_b(\mathbf{x}) = \begin{pmatrix} dig_b(x_1) \\ dig_b(x_2) \\ \vdots \\ dig_b(x_m) \end{pmatrix} \in \{0, 1, \dots, b-1\}^{m \times \mathbb{N}^*}$$

be the component-wise base-b expansion of $\mathbf{x}$. Sometimes we use $dig_b(\mathbf{x})(k)$ to denote the k-th column of $dig_b(\mathbf{x})$. The inverse of dig_b is defined as

$$dig_b^{-1} \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & \cdots \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ x_m^{(1)} & x_m^{(2)} & x_m^{(3)} & \cdots \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & \cdots \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ x_m^{(1)} & x_m^{(2)} & x_m^{(3)} & \cdots \end{pmatrix} \begin{pmatrix} b^{-1} \\ b^{-2} \\ b^{-3} \\ \vdots \end{pmatrix}.$$

Definition 5.9 (Differentiable scale for vectors)**.** *For $b \in \mathbb{Z}_{\ge 2}, \mathbf{x} \ne \mathbf{y} \in [0,1)^m$, suppose $dig_b(\mathbf{x}) = \begin{pmatrix} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & \mathbf{x}^{(3)} & \cdots \end{pmatrix}$, $dig_b(\mathbf{y}) = \begin{pmatrix} \mathbf{y}^{(1)} & \mathbf{y}^{(2)} & \mathbf{y}^{(3)} & \cdots \end{pmatrix}$, let*

$$dif_b(\mathbf{x}, \mathbf{y}) = \min\{k | \mathbf{x}^{(k)} \ne \mathbf{y}^{(k)}\}.$$

Proposition 5.10. *For $b \in \mathbb{Z}_{\ge 2}, \mathbf{x} \ne \mathbf{y} \in [0,1)^m$,*

$$dif_b(\mathbf{x}, \mathbf{y}) \le \lceil -\log_b \|\mathbf{x} - \mathbf{y}\| \rceil.$$

Proof. Suppose $\mathbf{x} = (x_1, x_2, \dots, x_m)^T, \mathbf{y} = (y_1, y_2, \dots, y_m)^T$. Let $i^* = \arg\max_i |x_i - y_i|$, then $|x_{i^*} - y_{i^*}| = \max_i |x_i - y_i| = \|\mathbf{x} - \mathbf{y}\|$. By definition, $dif_b(\mathbf{x}, \mathbf{y}) \le dif_b(x_{i^*}, y_{i^*})$. By Proposition 5.7, $dif_b(x_{i^*}, y_{i^*}) \le \lceil -\log_b |x_{i^*} - y_{i^*}| \rceil$. Therefore,

$$dif_b(\mathbf{x}, \mathbf{y}) \le \lceil -\log_b |x_{i^*} - y_{i^*}| \rceil = \lceil -\log_b \|\mathbf{x} - \mathbf{y}\| \rceil.$$

□

5.2 Hash Functions with Iterative Compression

In this section, we present a hash function that achieves approximate collision resistance for all $\mathsf{poly}(\lambda)$-sized precision lengths. The idea is to reinterpret the real input vector $\mathbf{x}$ as a stream of base-b expansion vectors $dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots)$ and compress (a finite prefix of) $dig_b(\mathbf{x})$ using the Merkle-Damgård framework with the Ajtai hash as the compression function.

We first review the definition of gadget decomposition.

Definition 5.11 (Gadget decomposition)**.** *Let $b \geq 2$, $B > b$ be integers. For any $x \in \mathbb{Z}_B$, the* gadget decomposition *of x with respect to the base b is defined as $G_b^{-1}(x) = (x_0, x_1, \ldots, x_{k-1}) \in \{0, 1, \ldots, b-1\}^k$ such that $x = \sum_{i=0}^{k-1} x_i b^i$ and $k = \lceil \log_b B \rceil$. For a vector $\mathbf{x} \in \mathbb{Z}_B^n$, the gadget decomposition is defined as $G_b^{-1}(\mathbf{x}) = (G_b^{-1}(\mathbf{x}_1), G_b^{-1}(\mathbf{x}_2), \ldots, G_b^{-1}(\mathbf{x}_n)) \in \{0, 1, \ldots, b-1\}^{kn}$.*

Now we are ready to present the scheme.

Scheme 5.12. *The hash function family with iterative compression is defined as:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$*: On input the security parameter λ, input length m, output length n, choose a base b, a modulus B, and set $m' = m + n\lceil \log_b B \rceil$. Uniformly sample a matrix $\mathbf{A}$ from $\mathbb{Z}_B^{n \times m'}$ and output $h = (b, B, \mathbf{A})$.*
- $\mathsf{Hash}(h, \mathbf{x}, \ell)$*: On input the hashing parameter $h = (b, B, \mathbf{A})$, a message $\mathbf{x} \in [0,1)^m$ and the precision length ℓ, let $dig_b(\mathbf{x}) = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots] \in \{0, 1, \ldots, b-1\}^{m \times \mathbb{N}^*}$ be the base-b expansion of $\mathbf{x}$. The algorithm does the following:*
  - *Set $\mathbf{z}^{(0)} = \mathbf{0}^{n\lceil \log_b B \rceil} \in \{0, 1, \ldots, b-1\}^{n\lceil \log_b B \rceil}$.*
  - *For $i = 1, 2, \ldots, \ell$,*
    - *Set $\mathbf{y}^{(i)} = \mathbf{A} \begin{bmatrix} \mathbf{x}^{(i)} \\ \mathbf{z}^{(i-1)} \end{bmatrix} \in \mathbb{Z}_B^n$.*
    - *Set $\mathbf{z}^{(i)} = G_b^{-1}(\mathbf{y}^{(i)}) \in \{0, 1, \ldots, b-1\}^{n\lceil \log_b B \rceil}$.*
  - *Output $\mathbf{y}^{(\ell)} \in \mathbb{Z}_B^n$.*

Proposition 5.13. *Based on the hardness of $SIS_{B,m',\beta}$ where $\beta = b\sqrt{m'}$, Scheme 5.12 satisfies that for all $\ell = \ell(\lambda) \in \mathsf{poly}(\lambda)$ and any efficient adversary $\mathcal{A}$,*

$$\Pr\left[\begin{array}{l} h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n) \\ (\mathbf{x}, \mathbf{x}') \leftarrow \mathcal{A}(1^\lambda, h, \ell) \end{array} \; : \; \|\mathbf{x} - \mathbf{x}'\| \geq b^{-\ell} \wedge \mathsf{Hash}(h, \mathbf{x}, \ell) = \mathsf{Hash}(h, \mathbf{x}', \ell) \right] \leq \mathsf{negl}(\lambda)\,.$$

Proof. If there exists an adversary $\mathcal{A}$ which gives $(\mathbf{x}, \mathbf{x}')$ with $\|\mathbf{x} - \mathbf{x}'\| \geq b^{-\ell}$ and $\mathsf{Hash}(h, \mathbf{x}, \ell) = \mathsf{Hash}(h, \mathbf{x}', \ell)$ with non-negligible probability, then one can use $\mathcal{A}$ to solve $\mathrm{SIS}_{B,m',\beta}$ as follows:

Given an SIS instance $(B, \mathbf{A}, \beta)$ where $\mathbf{A} \leftarrow U(\mathbb{Z}_B^{n \times m'})$, run $\mathcal{A}$ on input $(1^\lambda, h = (b, B, \mathbf{A}), \ell)$ where $b = \beta/\sqrt{m'}$. If $\mathcal{A}$ successfully outputs $(\mathbf{x}, \mathbf{x}')$ with $\|\mathbf{x} - \mathbf{x}'\| \geq b^{-\ell}$ and $\mathsf{Hash}(h, \mathbf{x}, \ell) = \mathsf{Hash}(h, \mathbf{x}', \ell)$, it means the following holds:

- Let $[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\ell)}]$ be the first ℓ vectors in $dig_b(\mathbf{x})$ and $[\mathbf{x}'^{(1)}, \ldots, \mathbf{x}'^{(\ell)}]$ be the first ℓ vectors in $dig_b(\mathbf{x}')$. Since $\|\mathbf{x} - \mathbf{x}'\| \geq b^{-\ell}$, we have
$$[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\ell)}] \neq [\mathbf{x}'^{(1)}, \ldots, \mathbf{x}'^{(\ell)}]. \tag{1}$$
- Let $(\mathbf{y}^{(i)}, \mathbf{z}^{(i)})_{i=1}^{\ell}$ and $(\mathbf{y}'^{(i)}, \mathbf{z}'^{(i)})_{i=1}^{\ell}$ be the outputs of the i-th iteration of the hash function for $\mathbf{x}$ and $\mathbf{x}'$ respectively. Since $\mathbf{y}^{(\ell)} = \mathsf{Hash}(h, \mathbf{x}, \ell) = \mathsf{Hash}(h, \mathbf{x}', \ell) = \mathbf{y}'^{(\ell)}$, by Equation 1, there exists an index $i \in [\ell]$ such that $\mathbf{y}^{(i)} = \mathbf{y}'^{(i)}$ but $(\mathbf{x}^{(i)}, \mathbf{z}^{(i-1)}) \neq (\mathbf{x}'^{(i)}, \mathbf{z}'^{(i-1)})$.

Thus, if $\mathcal{A}$ successfully outputs $(\mathbf{x}, \mathbf{x}')$ with $\|\mathbf{x} - \mathbf{x}'\| \geq b^{-\ell}$ and $\mathsf{Hash}(h, \mathbf{x}, \ell) = \mathsf{Hash}(h, \mathbf{x}', \ell)$, the SIS adversary outputs

$$\mathbf{z} = (\mathbf{x}^{(i)}, \mathbf{z}^{(i-1)}) - (\mathbf{x}'^{(i)}, \mathbf{z}'^{(i-1)}) \neq 0^{m'}\,,$$

otherwise it fails.

$h = (b, B, \mathbf{A})$ follows the same distribution as $h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n)$. As long as $\mathcal{A}$ succeeds, $\mathbf{z}$ satisfies $\mathbf{z} \neq 0^{m'}$ and $\|\mathbf{z}\|_2 \leq (b-1)\sqrt{m'} < \beta$. Moreover, since $\mathbf{y}^{(i)} = \mathbf{y}'^{(i)}$,

$$\mathbf{A}\mathbf{z} = \mathbf{A}\begin{bmatrix}\mathbf{x}^{(i)} \\ \mathbf{z}^{(i-1)}\end{bmatrix} - \mathbf{A}\begin{bmatrix}\mathbf{x}'^{(i)} \\ \mathbf{z}'^{(i-1)}\end{bmatrix} = \mathbf{y}^{(i)} - \mathbf{y}'^{(i)} = 0 \bmod B,$$

which means $\mathbf{z}$ is a solution to the SIS problem. Thus, the above algorithm solves $\mathrm{SIS}_{B,m',\beta}$. □

Corollary 5.14. *For all $b \geq 2$, based on the average-case hardness of $SIS_{B,m',\beta}$ where $\beta = b\sqrt{m'}$, Scheme 5.12 satisfies $f(\cdot)$-approximate collision resistance where $f(\ell) = 1$ for all $\ell \in \mathsf{poly}(\lambda)$.*

Scheme 5.12 overcomes the limitations of Scheme 5.2 and achieves collision resistance for all $\mathsf{poly}(\lambda)$-sized precision lengths. However, the hash outcome explicitly depends on the precision length ℓ, which means that it has no consistency guarantee.

5.3 Hash Functions with Decaying Weights

To achieve collision resistance and consistency at the same time, we want the output of the hash function to be real numbers and converge to a limit as the precision length goes to infinity. We still make use of the base-b expansion of a real vector

$$dig_b(\mathbf{x}) = \begin{pmatrix} dig_b(x_1) \\ dig_b(x_2) \\ \vdots \\ dig_b(x_m) \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & x_1^{(3)} & \cdots \\ x_2^{(1)} & x_2^{(2)} & x_2^{(3)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ x_m^{(1)} & x_m^{(2)} & x_m^{(3)} & \cdots \end{pmatrix},$$

and use a random matrix $\mathbf{A} \in \mathbb{Z}_B^{n\times m}$ to compress the dimension of each column:

$$\mathbf{y}^{(1)} = \mathbf{A}\begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_m^{(1)} \end{pmatrix}, \quad \mathbf{y}^{(2)} = \mathbf{A}\begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \\ \vdots \\ x_m^{(2)} \end{pmatrix}, \quad \cdots$$

We get a sequence of n-dimensional vectors $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots \in \mathbb{Z}_B^n$. We combine them together into a single real vector by assigning each $\mathbf{y}^{(k)}$ a weight and taking the weighted sum. Since each component of $\mathbf{y}^{(k)}$ is in $\{0, 1, \ldots, B-1\}$, a natural weight assignment is an exponential decay sequence $(B^{-1}, B^{-2}, B^{-3}, \ldots)$, which guarantees the convergence of the weighted sum.

$$\mathbf{y} = \frac{\mathbf{y}^{(1)}}{B} + \frac{\mathbf{y}^{(2)}}{B^2} + \frac{\mathbf{y}^{(3)}}{B^3} + \cdots$$

Observe that this weighted sum is exactly dig_B^{-1}. Therefore, one can imagine a clear expression of $\mathsf{IdealHash}$:

$$\mathsf{IdealHash}(\cdot, \mathbf{x}) = dig_B^{-1}\left(\mathbf{A} \cdot dig_b(\mathbf{x})\right).$$

When instantiating the hash function with a precision length ℓ, we only take the first few columns of $dig_b(\mathbf{x})$ and compute the weighted sum accordingly.

Scheme 5.15. *The hash function family with decaying weights is defined as:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$*: On input the security parameter* λ*, input length* m*, output length* n*, choose a base* $b \in \mathbb{N}_{\geq 2}$ *and a modulus* $B > b$ *appropriately. Uniformly sample a matrix* $\mathbf{A}$ *from* $\mathbb{Z}_B^{n \times m}$ *and output* $h = (b, B, \mathbf{A})$*.*
- $\mathsf{Hash}(h, \mathbf{x}, \ell)$*: On input the hashing parameter* $h = (b, B, \mathbf{A})$*, a message* $\mathbf{x} \in \mathcal{M}^m$*, and a precision length* ℓ*, let* $dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots)$ *be the base-*b *expansion of* $\mathbf{x}$*, and* $k = \lceil \frac{\ell}{\log b} \rceil$*. Output*

$$\mathbf{y} = \frac{\mathbf{A}\mathbf{x}^{(1)} \bmod B}{B} + \frac{\mathbf{A}\mathbf{x}^{(2)} \bmod B}{B^2} + \cdots + \frac{\mathbf{A}\mathbf{x}^{(k)} \bmod B}{B^k}.$$

Proposition 5.16. *Let* $g(\ell) = B^{-\lceil \frac{\ell}{\log b} \rceil}$*. Scheme 5.15 satisfies* $g(\cdot)$*-consistency.*

Proof. For any hashing parameter $h = (b, B, \mathbf{A})$ and message $\mathbf{x} \in \mathcal{M}^m$, recall that

$$h(\mathbf{x}) = \mathsf{IdealHash}(h, \mathbf{x}) = dig_B^{-1}\left(\mathbf{A} \cdot dig_b(\mathbf{x}) \bmod B\right) = \sum_{i=1}^{\infty} \frac{1}{B^i}\left(\mathbf{A}\mathbf{x}^{(i)} \bmod B\right).$$

For any precision length ℓ, we have

$$h_\ell(\mathbf{x}) = \sum_{i=1}^{k} \frac{1}{B^i}\left(\mathbf{A}\mathbf{x}^{(i)} \bmod B\right),$$

where $k = \lceil \frac{\ell}{\log b} \rceil$. Together,

$$\|h(\mathbf{x}) - h_\ell(\mathbf{x})\| = \left\| \sum_{i=k+1}^{\infty} \frac{1}{B^i}\left(\mathbf{A}\mathbf{x}^{(i)} \bmod B\right) \right\| \leq \sum_{i=k+1}^{\infty} \frac{B-1}{B^i} = B^{-k} = g(\ell).$$

□

Proposition 5.17. *Let* $f(\ell) = B^{-\lceil \frac{\ell}{\log b} \rceil}$*. Assuming the average-case hardness of* $SIS_{B,m,\beta}$ *where* $\beta = b\sqrt{m}$*, Scheme 5.15 satisfies* $f(\cdot)$*-approximate collision resistance.*

Proof. Suppose there exists a p.p.t. adversary $\mathcal{A}$ that, given $h = (b, B, \mathbf{A}), \ell$ where $\mathbf{A} \leftarrow U(\mathbb{Z}_B^{n \times m})$, finds $\mathbf{x}, \mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell}$ and $\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < f(\ell)$ with non-negligible probability. We construct $\mathcal{A}'$ solving average-case $\mathrm{SIS}_{B,m,\beta}$ as follows:

1. Given an SIS instance $(B, \mathbf{A}, \beta)$, invoke $\mathcal{A}$ on input $(h = (b, B, \mathbf{A}), \ell)$ where $b = \beta/\sqrt{m}$ and obtain $\mathbf{x}, \mathbf{x}'$.
2. If $\|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell}$ and $\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < f(\ell)$, compute $t = dif_b(\mathbf{x}, \mathbf{x}')$ and output
$$\mathbf{z} = dig_b(\mathbf{x})(t) - dig_b(\mathbf{x}')(t).$$

Assume $\mathcal{A}$ succeeds. By Proposition 5.10, $t \leq \lceil -\log_b \|\mathbf{x} - \mathbf{x}'\| \rceil \leq \lceil \frac{\ell}{\log b} \rceil = k$.

Notice that $h_\ell(\mathbf{x}), h_\ell(\mathbf{x}') \in \mathbb{Z}^n/B^k$, so $\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < B^{-k}$ implies $h_\ell(\mathbf{x}) = h_\ell(\mathbf{x}')$. Thus,

$$\mathbf{A}\mathbf{z} = \mathbf{A}dig_b(\mathbf{x})(t) - \mathbf{A}dig_b(\mathbf{x}')(t) = 0 \bmod B,$$

since $dig_B(h_\ell(\mathbf{x})) = dig_B(h_\ell(\mathbf{x}'))$ and $t \leq k$. Also, we have $\mathbf{z} \neq 0$ and $\|\mathbf{z}\|_2 \leq \beta$ by definition. Therefore, $\mathbf{z}$ is a solution to the SIS instance $(B, \mathbf{A}, \beta)$, and hence $\mathcal{A}'$ solves average-case $\mathrm{SIS}_{B,m,\beta}$ with non-negligible probability. □

Scheme 5.15 achieves consistency and collision resistance at the same time for all $\mathsf{poly}(\lambda)$-sized precision lengths. However, both the consistency error g and the collision resistance threshold f depend on the number of digits k, which means that g cannot be smaller than f. This results in zero tolerance for computation errors in the construction of commitment schemes, as we will see in the next section. Moreover, the hashing algorithm requires a large number of arithmetic operations over $\mathbb{Z}_B$, making it hard to leverage the fast floating point operation units of GPUs.

5.4 Hash Functions with Base Shifting

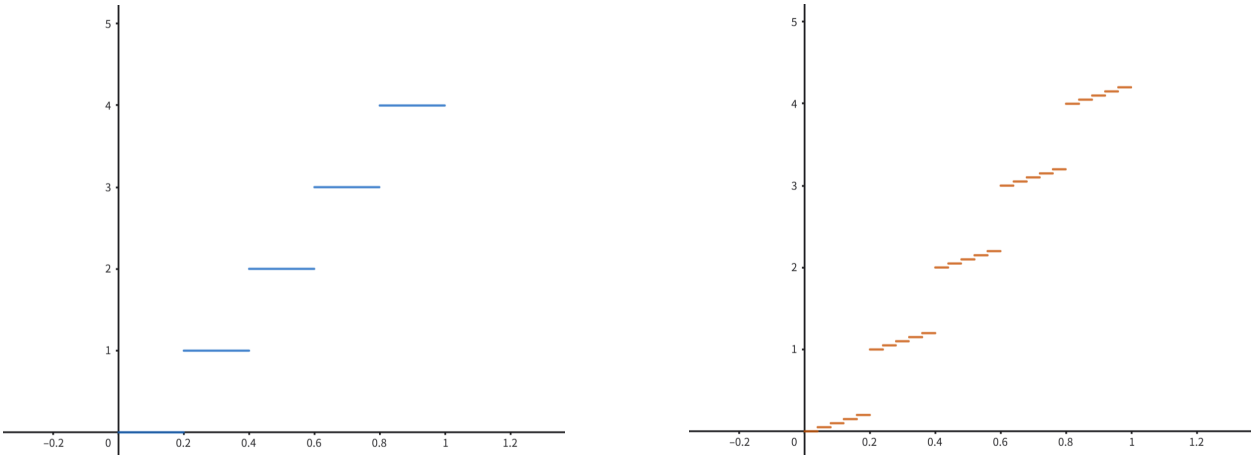
We still want to follow the original discretization-then-linear-transformation framework which provides a clear expression of the hash function on real numbers, independent of the precision length. Therefore, we try to design a new discretization function which works natively on real numbers, independent of the precision length.

Figure 2a shows the graph of the rounding function $R_1(x) = \lfloor 5x \rfloor$. We know that such a stair-like pattern can serve as a discretization encoding before the linear transformation and provides collision resistance above the scale of the length of each step.

A natural generalization is to introduce stair-like patterns at smaller scales as well, as illustrated by the graph of the following function in Figure 2b

$$R_2(x) = R_1(x) + \frac{R_1(\{5x\})}{20} = \lfloor 5x \rfloor + \frac{\lfloor 5\{5x\} \rfloor}{20},$$

where $\{x\} = x - \lfloor x \rfloor$ takes the fractional part of a real number. As a result, R_2 looks similar to R_1 at both macroscopic (above $1/5$) and microscopic (between $1/25$ and $1/5$) scales, which can potentially be the prerequisite for a stronger collision resistance.



(a) R_1: one-level rounding

(b) R_2: two-level rounding

Figure 2: Rounding at different levels

Notice that it is more straightforward to express R_2 in base 5:

$$x = \sum_{k=1}^{+\infty} x^{(k)} \cdot 5^{-k} = (0.x^{(1)}x^{(2)}x^{(3)}\dots)_5,$$

$$R_2(x) = x^{(1)} + \frac{x^{(2)}}{20}.$$

Following this idea, we can assign each digit a weight according to the magnitude and add them together, so that it exhibits a stair-like pattern at any specific scale.

Again, we use the exponential decay sequence $(B^{-1}, B^{-2}, B^{-3}, \dots)$ where the decay factor B is reasonably larger than b. Observe that with this weight assignment, the linear combination is exactly dig_B^{-1}. Thus, the function can be formalized by the following base-shifting function:

Definition 5.18 (Base-shifting function)**.** *For $b, B \in \mathbb{N}$, $1 < b < B$, let the base-shifting function $\phi_{b\to B} : [0,1) \to [0,1)$ be*

$$\phi_{b\to B} = dig_B^{-1} \circ dig_b.$$

In other words,

$$\phi_{b\to B}((0.x^{(1)}x^{(2)}x^{(3)}\dots)_b) = (0.x^{(1)}x^{(2)}x^{(3)}\dots)_B.$$

The intuition that $\phi_{b\to B}$ exhibits a stair-like pattern at any scale when B is sufficiently large compared to b is formalized by the proposition below. Basically it says the difference between $\phi(x)$ and $\phi(y)$ almost entirely depends on their digits at the differentiable scale.

Proposition 5.19. *For $b, B \in \mathbb{N}$, $1 < b < B$, $x \neq y \in [0,1)$, let $k = dif_b(x,y)$, then*

$$\phi_{b\to B}(x) - \phi_{b\to B}(y) \ \in \ B^{-k}\left(dig_b(x)(k) - dig_b(y)(k) \pm \frac{b}{B}\right).$$

Proof. Suppose $dig_b(x) = (x^{(1)}, x^{(2)}, x^{(3)}, \dots)$, $dig_b(y) = (y^{(1)}, y^{(2)}, y^{(3)}, \dots)$. By definition, $x^{(i)} = y^{(i)}$ holds for $1 \le i \le k-1$, so

$$\begin{aligned}\phi_{b\to B}(x) - \phi_{b\to B}(y) &= \sum_{i=1}^{+\infty} B^{-i}(x^{(i)} - y^{(i)}) \\ &= B^{-k}(x^{(k)} - y^{(k)}) + \sum_{i=k+1}^{+\infty} B^{-i}(x^{(i)} - y^{(i)}).\end{aligned}$$

Since $x^{(i)}, y^{(i)} \in \{0, 1, \dots, b-1\}$, the second part is bounded by

$$\begin{aligned}\left|\sum_{i=k+1}^{+\infty} B^{-i}(x^{(i)} - y^{(i)})\right| &\le \sum_{i=k+1}^{+\infty} B^{-i}\left|(x^{(i)} - y^{(i)})\right| \\ &\le \sum_{i=k+1}^{+\infty} B^{-i}(b-1) = B^{-k}\cdot\frac{b-1}{B-1} < B^{-k}\cdot\frac{b}{B}\end{aligned}$$

Therefore,

$$\phi_{b\to B}(x) - \phi_{b\to B}(y) \ \in \ B^{-k}(x^{(k)} - y^{(k)}) \pm B^{-k}\cdot\frac{b}{B} = B^{-k}\left(dig_b(x)(k) - dig_b(y)(k) \pm \frac{b}{B}\right).$$

□

The above definitions and propositions can readily be extended to vectors.

Definition 5.20 (Base-shifting function for vectors)**.** *For $b, B \in \mathbb{N}$, $1 < b < B$, let the base-shifting function $\phi_{b\to B} : [0,1)^m \to [0,1)^m$ be*

$$\phi_{b\to B} = dig_B^{-1} \circ dig_b,$$

or equivalently,

$$\phi_{b\to B}(\mathbf{x}) = \begin{pmatrix} \phi_{b\to B}(x_1) \\ \phi_{b\to B}(x_2) \\ \vdots \\ \phi_{b\to B}(x_m) \end{pmatrix}.$$

Proposition 5.21. *For $b, B \in \mathbb{N}$, $1 < b < B$, $\mathbf{x} \neq \mathbf{y} \in [0,1)^m$, let $k = dif_b(\mathbf{x}, \mathbf{y})$, then*

$$\phi_{b\to B}(\mathbf{x}) - \phi_{b\to B}(\mathbf{y}) \ \in \ B^{-k}\left(dig_b(\mathbf{x})(k) - dig_b(\mathbf{y})(k) + \mathcal{B}_\infty^m\left(\frac{b}{B}\right)\right),$$

where $\mathcal{B}_\infty^m(r) = \{\mathbf{u} \in \mathbb{R}^m : \|\mathbf{u}\| \leq r\}$ is the m-dimensional ball of radius r with respect to the infinity norm.

Proof. Similar to Proposition 5.19, where every absolute value is replaced with the infinity norm. □

Now we are ready to propose our hash function family with base-shifting function as discretization. One may expect the hash function of the form $\mathbf{A} \cdot \phi_{b\to B}(\mathbf{x})$, where $\mathbf{A}$ is an $n \times m$ random matrix. We point out that although $\phi_{b\to B}$ serves as the discretization function and indeed shows many discrete properties, outputs of $\phi_{b\to B}$ still need to be viewed as normal real numbers for further operations, as they do not fit in any discrete algebraic structures like $\mathbb{Z}_B$. Thus, the linear transformation $\mathbf{A}$ has to at least work on $\mathbb{R}$. Fortunately, CSIS' satisfies this requirement, while enjoying average-case hardness that we have already shown in section 4.4.

Scheme 5.22. *The hash function family with base-shifting as discretization is defined as:*

- $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$*: On input the security parameter λ, input length m, output length n, choose two polynomially-bounded bases b, B. Uniformly sample a matrix $\mathbf{A}$ from $[0,1)^{n\times m}$ and output $h = (b, B, \mathbf{A})$.*
- $\mathsf{Hash}(h = (b, B, \mathbf{A}), \mathbf{x}, \ell)$*: On input the bases b, B, the matrix $\mathbf{A} \in [0,1)^{n\times m}$, a message $\mathbf{x}$ and the precision length ℓ, output*

  $$\mathbf{y} =_{\epsilon_\ell} \mathbf{A} \cdot \phi_{b\to B}(\mathbf{x}),$$

  where the computation error is bounded by ϵ_ℓ.

Remark 5.23. *Essentially, Scheme 5.22 follows the same framework as Scheme 5.15:*

$$\mathbf{y} = dig_B^{-1}(\mathbf{A} \cdot dig_b(\mathbf{x})).$$

The difference is that here $\mathbf{A}$ is a left matrix multiplication over $\mathbb{R}$. Since dig_B^{-1} is a right matrix multiplication over $\mathbb{R}$, they commute with each other:

$$dig_B^{-1}(\mathbf{A} \cdot dig_b(\mathbf{x})) = \mathbf{A} \cdot dig_B^{-1}(dig_b(\mathbf{x})) = \mathbf{A} \cdot \phi_{b\to B}(\mathbf{x}).$$

Proposition 5.24. *Let $g(\ell) = \epsilon_\ell$. Scheme 5.22 satisfies $g(\cdot)$-consistency.*

Proof. Let $\mathsf{IdealHash}(h = (b, B, \mathbf{A}), \mathbf{x}) = \mathbf{A} \cdot \phi_{b\to B}(\mathbf{x})$. The scheme naturally meets our expectation of consistency. □

Proposition 5.25. *Let $f(\ell) = B^{-\frac{\ell}{\log b}}/40nB$. Assuming the hardness of $SIVP_{\tilde{O}(b\sqrt{mn})}$, when $B > 20mnb$, $\epsilon_\ell \le B^{-\frac{\ell}{\log b}}/80nB$, Scheme 5.22 satisfies $f(\cdot)$-approximate collision resistance.*

Proof. Let $\beta = b\sqrt{m}, \zeta = \frac{1}{8n}$, denote $h(\mathbf{x}) = \mathsf{IdealHash}(h, \mathbf{x})$.

Suppose there exists a p.p.t. adversary $\mathcal{A}$ that, given $b, B, \mathbf{A}, \ell$ where $\mathbf{A} \leftarrow [0,1)^{n\times m}$, finds $\mathbf{x}, \mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\| \ge 2^{-\ell}$ and $\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < f(\ell)$ with non-negligible probability. We construct $\mathcal{A}'$ solving $\mathrm{CSIS}'^{U([0,1))}_{m,\beta,\zeta}$ on average as follows:

1. Given $(\mathbf{A}, \beta, \zeta)$, invoke $\mathcal{A}$ on input $((b, B, \mathbf{A}), \ell)$ and obtain $\mathbf{x}, \mathbf{x}'$.
2. If $\|\mathbf{x} - \mathbf{x}'\| \ge 2^{-\ell}$ and $\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| < f(\ell)$, compute $k = dif_b(\mathbf{x}, \mathbf{x}')$ and output
$$\mathbf{z} = dig_b(\mathbf{x})(k) - dig_b(\mathbf{x}')(k).$$

Assume $\mathcal{A}$ succeeds. By Proposition 5.10, $k \le \lceil -\log_b \|\mathbf{x} - \mathbf{x}'\| \rceil < \frac{\ell}{\log b} + 1$. By Proposition 5.21,
$$\phi_{b\to B}(\mathbf{x}) - \phi_{b\to B}(\mathbf{x}') \in B^{-k}(\mathbf{z} + \mathcal{B}^m_\infty(b/B)).$$
If $\|\mathbf{Az}\| > \zeta$, then, since
$$h(\mathbf{x}) - h(\mathbf{x}') = \mathbf{A}(\phi_{b\to B}(\mathbf{x}) - \phi_{b\to B}(\mathbf{x}')) \in B^{-k}(\mathbf{Az} + \mathcal{B}^n_\infty(mb/B)),$$
this implies that
$$\begin{aligned}\|h_\ell(\mathbf{x}) - h_\ell(\mathbf{x}')\| &\ge \|h(\mathbf{x}) - h(\mathbf{x}')\| - \|h_\ell(\mathbf{x}) - h(\mathbf{x})\| - \|h_\ell(\mathbf{x}') - h(\mathbf{x}')\| \\ &\ge B^{-k}\left(\zeta - \frac{mb}{B}\right) - \epsilon_\ell - \epsilon_\ell \\ &\ge \frac{B^{-k}}{20n} - 2\epsilon_\ell \ge \frac{B^{-\frac{\ell}{\log b}}}{20nB} - \frac{B^{-\frac{\ell}{\log b}}}{40nB} = f(\ell),\end{aligned}$$
a contradiction. Thus, $\|\mathbf{Az}\| \le \zeta$. We also have $\mathbf{z} \ne 0$ and $\|\mathbf{z}\|_2 \le b\sqrt{m} = \beta$ by definition. Therefore, $\mathbf{z}$ is a solution to the CSIS′ instance $(\mathbf{A}, \beta, \zeta)$, which means that $\mathcal{A}'$ solves $\mathrm{CSIS}'^{U([0,1))}_{m,\beta,\zeta}$ on average with non-negligible probability. This breaks the hardness of $\mathrm{SIVP}_{\tilde{O}(b\sqrt{mn})}$, according to Corollary 4.25. □

Combining the above two propositions, we give the following theorem:

Theorem 5.26. *Assuming the hardness of $SIVP_{\tilde{O}(b\sqrt{mn})}$, when $B > 20mnb$ and $\epsilon_\ell \le B^{-\frac{\ell}{\log b}}/80nB$, Scheme 5.22 satisfies $f(\cdot)$-approximate collision resistance and $g(\cdot)$-consistency, where $g(\ell) = \epsilon_\ell$ and $f(\ell) = B^{-\frac{\ell}{\log b}}/40nB$.*

For $m = \Theta(n)$, $b = O(1)$, we can get a hashing scheme with $f(\ell) = 2^{-\Theta(\ell \log n)}$-approximate collision resistance based on the hardness of $\mathrm{SIVP}_{\tilde{O}(n)}$. For a better f, one may set $b = \Theta(\sqrt{n})$, $B = \Theta(n^{2.5})$ to get a hashing scheme with $f(\ell) = 2^{-\Theta(\ell + \log n)}$-approximate collision resistance based on the hardness of $\mathrm{SIVP}_{\tilde{O}(n^{1.5})}$.

Scheme 5.22 only involves digit operations and matrix multiplication over $\mathbb{R}$ and avoids any modular arithmetic. Also, it decouples the consistency error g and collision resistance threshold f, since g naturally depends on the computation error ϵ_ℓ while f depends on the differentiable scale. This allows g to be reasonably smaller than f, which is crucial for constructing commitment schemes which can tolerate computation errors.

6 Constructing RVCs from RVCRHs

First, let us review the common approach, first proposed in [KTX08], of constructing a commitment scheme from the discrete Ajtai-style hash functions, $h_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} \bmod q$, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{x} \in \{0, 1, \ldots, b-1\}^m = [b]^m$. To commit a message $\mathbf{x}$, one can sample a random salt $\mathbf{r}$, append it to the message, and feed them into the hash function together. Formally,

$$\mathsf{Commit}_{\mathbf{A}}(\mathbf{x}; \mathbf{r}) = h_{\mathbf{A}}(\mathbf{x}; \mathbf{r}) = \mathbf{A} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} \bmod q.$$

The binding property of the commitment scheme follows from the collision resistance of the hash function, while the hiding property relies on the Leftover Hash Lemma (LHL). This approach works for Scheme 5.12 and Scheme 5.15, as we will show in section 6.1 and section 6.2. Unfortunately, hashing schemes that work natively on real values, such as Scheme 5.22, do not admit this simple conversion, so we give another generic framework of constructing commitment schemes from them in section 6.3.

6.1 Commitment Schemes from Hash Functions with Iterative Compression

In this section, we construct a commitment scheme from the hashing scheme 5.12. Since this hashing scheme has no consistency guarantee, we can only obtain a weaker version of the commitment scheme, where the prover and the verifier need to agree on the same precision length $\ell_P = \ell_V = \ell$.

Scheme 6.1. *The real value commitment scheme based on Scheme 5.12 is defined as follows:*

- $\mathsf{Setup}(1^\lambda, 1^{m_m})$*: On input the security parameter* λ *and message vector length* m_m*, set* $n = \lambda$*, choose appropriate parameter* m_r*, set* $m = m_m + m_r$*, and invoke* $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$ *to obtain the hashing parameter* $h = (b, B, \mathbf{A})$*. Output the public parameter* $\mathsf{pp} = h$*.*

- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell)$*: On input the public parameter* $\mathsf{pp} = h$*, a message vector* $\mathbf{x} \in [0, 1)^{m_m}$*, and a precision length* ℓ*, uniformly sample a vector* $\mathbf{r} \leftarrow [0, 1)^{m_r}$ *and compute*

  $$\mathbf{c}_P = h_\ell(\mathbf{x}; \mathbf{r}) \in \mathbb{Z}_B^n.$$

  Output $(\mathsf{cm} = \mathbf{c}_P, \pi_o = \mathbf{r})$*.*

- $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell)$*: On input the public parameter* $\mathsf{pp} = h$*, a commitment* $\mathsf{cm} \in \mathbb{Z}_B^n$*, a message vector* $\mathbf{x} \in [0, 1)^{m_m}$*, an opening proof* $\pi_o = \mathbf{r} \in [0, 1)^{m_r}$ *and a precision length* ℓ*, compute*

  $$\mathbf{c}_V = h_\ell(\mathbf{x}; \mathbf{r}) \in \mathbb{Z}_B^n.$$

  Then check whether $\mathsf{cm} = \mathbf{c}_V$*. If so, output* 1*; otherwise, output* 0*.*

Correctness. Note that in Scheme 5.12, all the computation is done in the discrete domain after base expansion, so no computation error is introduced. If both Commit and VerifyOpen are honestly executed, we have $\mathsf{cm} = \mathbf{c}_P = h_\ell(\mathbf{x}; \mathbf{r}) = \mathbf{c}_V$, and therefore VerifyOpen always outputs 1.

Computationally Binding. The binding property of the commitment scheme follows from the approximate collision resistance of the hashing scheme. If $\mathcal{A}(\mathsf{pp})$ finds $(\mathsf{cm}, \mathbf{x}, \pi_o = \mathbf{r}, \mathbf{x}', \pi_o' = \mathbf{r}')$ such that

$$\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell) = 1 \wedge \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}', \pi_o', \ell) = 1 \wedge \|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell},$$

then $\|(\mathbf{x}; \mathbf{r}) - (\mathbf{x}'; \mathbf{r}')\| \geq \|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell}$ and $h_\ell(\mathbf{x}; \mathbf{r}) = \mathsf{cm} = h_\ell(\mathbf{x}'; \mathbf{r}')$. Thus, $(\mathbf{x}; \mathbf{r})$ and $(\mathbf{x}'; \mathbf{r}')$ form a pair of collisions for the hashing scheme, which contradicts the $f(\cdot)$-approximate collision resistance of Scheme 5.12 where $f(\ell) = 1$.

Statistically Hiding. Taking the random salt into account, the Hash algorithm computes

$$\mathbf{y}^{(i)} = \mathbf{A} \begin{bmatrix} \mathbf{x}^{(i)} \\ \mathbf{r}^{(i)} \\ \mathbf{z}^{(i-1)} \end{bmatrix} \in \mathbb{Z}_B^n,$$
$$\mathbf{z}^{(i)} = G_b^{-1}(\mathbf{y}^{(i)}) \in \{0, 1, \ldots, b-1\}^{n\lceil \log_b B \rceil}$$

for $i = 1, 2, \ldots, \ell$, where $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots] = dig_b(\mathbf{x})$, $[\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots] = dig_b(\mathbf{r})$ and $\mathbf{z}^{(0)} = 0^{n\lceil \log_b B \rceil}$. It outputs $h_\ell(\mathbf{x}; \mathbf{r}) = \mathbf{y}^{(\ell)}$.

Notice that when $\mathbf{r} \leftarrow [0, 1)^{m_r}$, each $\mathbf{r}^{(i)}$ is uniformly distributed over $\{0, 1, \ldots, b-1\}^{m_r}$. Write $\mathbf{A} = [\mathbf{A}_x \mid \mathbf{A}_r \mid \mathbf{A}_z]$. By a discrete LHL argument with composite modulus, we have

$$\left\{ \mathbf{A}_r, \mathbf{r}^{(\ell)} : \left(\mathbf{A}_r, \mathbf{A}_r \mathbf{r}^{(\ell)} \right) \right\} \approx_s (U(\mathbb{Z}_B^{n \times m_r}), U(\mathbb{Z}_B^n))$$

when $m_r > O(n \log B)$. Therefore, conditioned on any fixed $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots, \mathbf{r}^{(\ell-1)}$,

$$\{\mathbf{A}, \mathbf{r}^{(\ell)} : (\mathbf{A}, \mathbf{y}^{(\ell)} = \mathbf{A}_x \mathbf{x}^{(\ell)} + \mathbf{A}_r \mathbf{r}^{(\ell)} + \mathbf{A}_z \mathbf{z}^{(\ell-1)})\} \approx_s (U(\mathbb{Z}_B^{n \times m'}), U(\mathbb{Z}_B^n)).$$

This implies

$$\{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^{m_m}), (\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell) : (\mathsf{pp}, \mathsf{cm})\} \approx_s \mathsf{Setup}(1^\lambda, 1^{m_m}) \times U(\mathbb{Z}_B^n).$$

The simulator $\mathcal{S}$ simply samples $\mathbf{u} \leftarrow U(\mathbb{Z}_B^n)$ and outputs $\mathbf{u}$.

6.2 Commitment Schemes from Hash Functions with Decaying Weights

In this section, we construct a commitment scheme from the hashing scheme 5.15.

Scheme 6.2. *The real value commitment scheme based on Scheme 5.15 is defined as follows:*

- $\mathsf{Setup}(1^\lambda, 1^{m_m})$*: On input the security parameter* λ *and message vector length* m_m*, set* $n = \lambda$*, choose appropriate parameter* m_r*, set* $m = m_m + m_r$*, and invoke* $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$ *to obtain the hashing parameter* $h = (b, B, \mathbf{A})$*. Output the public parameter* $\mathsf{pp} = h$*.*
- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$*: On input the public parameter* $\mathsf{pp} = h$*, a message vector* $\mathbf{x} \in [0, 1)^{m_m}$*, and a precision length* ℓ_P*, uniformly sample a vector* $\mathbf{r} \leftarrow [0, 1)^{m_r}$ *and compute*

$$\mathbf{c}_P = h_{\ell_P}(\mathbf{x}; \mathbf{r}) \in [0, 1)^n.$$

Output $(\mathsf{cm} = \mathbf{c}_P, \pi_o = \mathbf{r})$.

- $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V)$: *On input the public parameter* $\mathsf{pp} = h = (b, B, \mathbf{A})$, *a commitment* $\mathsf{cm} \in [0,1)^n$, *a message vector* $\mathbf{x} \in [0,1)^{m_m}$, *an opening proof* $\pi_o = \mathbf{r} \in [0,1)^{m_r}$ *and a precision length* ℓ_V, *compute*

$$\mathbf{c}_V = h_{\ell_V}(\mathbf{x}; \mathbf{r}) \in [0,1)^n.$$

Let $k_V = \lceil \frac{\ell_V}{\log b} \rceil$. *Check whether* $\lfloor B^{k_V} \mathbf{c}_V \rfloor = \lfloor B^{k_V} \mathsf{cm} \rfloor$. *If so, output* 1; *otherwise, output* 0.

Assume that no computation error is introduced after base expansion.

Correctness. Suppose $k_P = \lceil \frac{\ell_P}{\log b} \rceil, k_V = \lceil \frac{\ell_V}{\log b} \rceil, dig_b(\mathbf{x}) = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots)$ and $dig_b(\mathbf{r}) = (\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots)$. If both Commit and VerifyOpen are honestly executed, by definition, (omitting $\bmod B$)

$$\mathsf{cm} = \mathbf{c}_P = \frac{1}{B}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(1)}\\ \mathbf{r}^{(1)}\end{pmatrix} + \frac{1}{B^2}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(2)}\\ \mathbf{r}^{(2)}\end{pmatrix} + \cdots + \frac{1}{B^{k_P}}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(k_P)}\\ \mathbf{r}^{(k_P)}\end{pmatrix},$$

$$\mathbf{c}_V = \frac{1}{B}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(1)}\\ \mathbf{r}^{(1)}\end{pmatrix} + \frac{1}{B^2}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(2)}\\ \mathbf{r}^{(2)}\end{pmatrix} + \cdots + \frac{1}{B^{k_V}}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(k_V)}\\ \mathbf{r}^{(k_V)}\end{pmatrix}.$$

Since $k_V \le k_P$, we have

$$\lfloor B^{k_V} \mathsf{cm} \rfloor = \sum_{i=1}^{k_V} B^{k_V - i}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(i)}\\ \mathbf{r}^{(i)}\end{pmatrix} = \lfloor B^{k_V}\mathbf{c}_V \rfloor.$$

Therefore, VerifyOpen always outputs 1.

Computationally Binding. The binding property of the commitment scheme follows from the approximate collision resistance of the hashing scheme. If $\mathcal{A}(\mathsf{pp})$ finds $(\mathsf{cm}, \mathbf{x}, \pi_o = \mathbf{r}, \mathbf{x}', \pi_o' = \mathbf{r}')$ such that

$$\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) = 1 \wedge \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}', \pi_o', \ell_V) = 1 \wedge \|\mathbf{x} - \mathbf{x}'\| \ge 2^{-\ell_V},$$

then $\|(\mathbf{x}; \mathbf{r}) - (\mathbf{x}'; \mathbf{r}')\| \ge \|\mathbf{x} - \mathbf{x}'\| \ge 2^{-\ell_V}$ and $\lfloor B^{k_V} h_{\ell_V}(\mathbf{x}; \mathbf{r}) \rfloor = \lfloor B^{k_V} \mathsf{cm} \rfloor = \lfloor B^{k_V} h_{\ell_V}(\mathbf{x}'; \mathbf{r}') \rfloor$. Since $B^{k_V} h_{\ell_V}(\mathbf{x}; \mathbf{r})$ and $B^{k_V} h_{\ell_V}(\mathbf{x}'; \mathbf{r}')$ are both in $\mathbb{Z}^n$, we have $h_{\ell_V}(\mathbf{x}; \mathbf{r}) = h_{\ell_V}(\mathbf{x}'; \mathbf{r}')$. Thus, $(\mathbf{x}; \mathbf{r})$ and $(\mathbf{x}'; \mathbf{r}')$ form a pair of collisions for the hashing scheme, which contradicts the $f(\cdot)$-approximate collision resistance of Scheme 5.15 where $f(\ell) = B^{-\lceil \frac{\ell}{\log b} \rceil}$.

Statistically Hiding. The hiding property of the commitment scheme still follows from the discrete LHL argument with composite modulus. When $\mathbf{r} \leftarrow [0,1)^{m_r}$, each $\mathbf{r}^{(i)}$ is uniformly distributed over $\{0, 1, \dots, b-1\}^{m_r}$. When $m_r > O(n \log B)$, for each $1 \le i \le k_P$,

$$\left\{\mathbf{A}, \mathbf{r}^{(i)} : \left(\mathbf{A}, \mathbf{A}\begin{pmatrix}\mathbf{x}^{(i)}\\ \mathbf{r}^{(i)}\end{pmatrix}\right)\right\} \approx_s (U(\mathbb{Z}_B^{n \times m}), U(\mathbb{Z}_B^n)).$$

Since k_P is polynomially bounded and $\mathbf{r}^{(1)}, \dots \mathbf{r}^{(k_P)}$ are mutually independent conditioned on $\mathbf{A}$, by a hybrid argument, we have

$$\left\{\mathbf{A}, \mathbf{r} : \left(\mathbf{A}, \sum_{i=1}^{k_P} \frac{1}{B^i}\mathbf{A}\begin{pmatrix}\mathbf{x}^{(i)}\\ \mathbf{r}^{(i)}\end{pmatrix}\right)\right\} \approx_s \left(U(\mathbb{Z}_B^{n \times m}), \sum_{i=1}^{k_P} \frac{1}{B^i} U(\mathbb{Z}_B^n)\right).$$

Note that $\sum_{i=1}^{k_P} \frac{1}{B^i} U(\mathbb{Z}_B^n) = \frac{1}{B^{k_P}} U(\mathbb{Z}_{B^{k_P}}^n)$. Therefore,

$$\{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^{m_m}), (\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P) : (\mathsf{pp}, \mathsf{cm})\} \approx_s \mathsf{Setup}(1^\lambda, 1^{m_m}) \times \frac{1}{B^{k_P}} U(\mathbb{Z}_{B^{k_P}}^n).$$

The simulator $\mathcal{S}$ simply samples $\mathbf{u} \leftarrow \frac{1}{B^{k_P}} U(\mathbb{Z}_{B^{k_P}}^n)$ and outputs $\mathbf{u}$.

6.3 Commitment Schemes from Hash Functions Natively on Real Values

In this section, we design commitment schemes from hashing schemes that work natively on real values, such as Scheme 5.22.

Note that under discrete settings, LHL requires two conditions:

- h is a universal hash function family, i.e., for arbitrary $\mathbf{x} \neq \mathbf{y}$,

$$\Pr_{\mathbf{A}}[h_{\mathbf{A}}(\mathbf{x}) = h_{\mathbf{A}}(\mathbf{y})] \leq \frac{1}{|H|},$$

  where H is the output space of h, in this case $\mathbb{Z}_q^n$.

- The distribution of the input to the hash function has enough collision entropy, which is lower-bounded by min-entropy. This is used to bound $\Pr_{\mathbf{x},\mathbf{y}}[\mathbf{x} = \mathbf{y}]$, where $\mathbf{x}, \mathbf{y}$ are independently sampled from the input distribution.

However, under the real value setting, if we view the hash function family as a black box and do not assume any universality property of it (which turns out to be difficult to define and achieve), then the argument does not work.

Thus, we may adopt another interpretation of the original commitment scheme, which separates the message hash and the random mask:

$$\mathsf{Commit}_{\mathbf{A},\mathbf{A}_r}(\mathbf{x};\mathbf{r}) = h_{\mathbf{A}}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r} \bmod q. \tag{2}$$

Similarly, for a hash function family h on real values, we can imagine a commitment scheme of the form

$$\mathsf{Commit}(\mathbf{x};\mathbf{r}) = h(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r} \bmod 1. \tag{3}$$

Now when arguing the hiding property, we can fix $h(\mathbf{x})$ and focus on analyzing the distribution of $\mathbf{A}_r \cdot \mathbf{r} \bmod 1$. Note that here $\mathbf{r}$ cannot be chosen from a continuous domain, otherwise it will be trivial to find different openings by manipulating $\mathbf{r}$. Nonetheless, we claim that it suffices to have $\mathbf{A}_r$ uniformly sampled from $\mathbb{T}^{n \times m_r}$ and $\mathbf{r}$ uniformly sampled from $\{0, 1, \ldots, b-1\}^{m_r}$, a discrete domain. An analog of LHL (Theorem 6.11) can still be established to show that the output is statistically close to some uniform distribution.

On the other hand, some careful consideration is needed for the binding property. We point out that with the interpretation in (2), the binding property is not directly implied by the collision resistance of h, since it is possible to find a tuple $(\mathbf{x}, \mathbf{r}, \mathbf{x}', \mathbf{r}')$ such that $\mathbf{A}_r\mathbf{r} - \mathbf{A}_r\mathbf{r}' = h_{\mathbf{A}}(\mathbf{x}') - h_{\mathbf{A}}(\mathbf{x})$ cancels the difference of the message-hash. Luckily, when working with real values as in (3), we can put the randomness and the message-hash on different scales so that they do not interfere with each other. Intuitively, if the adversary finds two different openings $(\mathbf{x}, \mathbf{r}, \mathbf{x}', \mathbf{r}')$ with $\mathbf{r} \neq \mathbf{r}'$, then there must be a noticeable gap ζ (typically $\frac{1}{8n}$) between $\mathbf{A}_r\mathbf{r}$ and $\mathbf{A}_r\mathbf{r}'$, otherwise CSIS (Definition 4.10) can be solved. If the hash function works on a scale smaller than this gap, then the difference of the message-hash cannot be canceled by $\mathbf{A}_r\mathbf{r} - \mathbf{A}_r\mathbf{r}'$, which is much larger than $h(\mathbf{x}') - h(\mathbf{x})$. At the same time, the adversary is not able to find a pair of different openings with $\mathbf{r} = \mathbf{r}'$, since that would violate the collision resistance of the hash function.

Formally, we construct a commitment scheme from a hash function family $\Pi_{\mathsf{crh}} = (\mathsf{Gen}, \mathsf{Hash})$ as follows:

Scheme 6.3. *Let $q(\cdot)$ be a monotone decreasing function upper-bounded by $\frac{1}{100\lambda}$. The real value commitment scheme $\Pi^q_{\mathsf{rvc}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyOpen})$ based on Π_{crh} with verification threshold q is defined as follows:*

- $\mathsf{Setup}(1^\lambda, 1^m)$*: On input the security parameter λ and vector length m, set $n = \lambda$, choose appropriate parameters m_r, b and invoke $\mathsf{Gen}(1^\lambda, 1^m, 1^n)$ to obtain a hashing parameter h. Uniformly sample a matrix $\mathbf{A}_r \leftarrow \mathbb{T}^{n \times m_r}$. Output the public parameter $\mathsf{pp} = (h, \mathbf{A}_r)$.*
- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$*: On input the public parameter $\mathsf{pp} = (h, \mathbf{A}_r)$, a vector $\mathbf{x} \in \mathcal{M}^m$, a precision length ℓ_P, first compute $h_{\ell_P}(\mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}, \ell_P)$. Then uniformly sample a vector $\mathbf{r} \leftarrow \{0, 1, \ldots, b-1\}^{m_r}$ and compute*
$$\mathbf{c}_P =_{\epsilon_P} (h_{\ell_P}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \in \mathbb{T}^n,$$
where the computation error is bounded by $\epsilon_P = \frac{1}{4} q(\ell_P)$. Output $(\mathsf{cm} = \mathbf{c}_P, \pi_o = \mathbf{r})$.
- $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V)$*: On input the public parameter $\mathsf{pp} = (h, \mathbf{A}_r)$, a commitment $\mathsf{cm} \in \mathbb{T}^n$, a vector $\mathbf{x} \in \mathcal{M}^m$, an opening proof $\pi_o = \mathbf{r} \in \{0, 1, \ldots, b-1\}^{m_r}$ and a precision length ℓ_V, first compute $h_{\ell_V}(\mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}, \ell_V)$ and*
$$\mathbf{c}_V =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1,$$
where the computation error is bounded by $\epsilon_V = \frac{1}{4} q(\ell_V)$. Then check whether
$$\|\mathsf{cm} - \mathbf{c}_V\|_{\mathbb{T}} < q(\ell_V).$$
If so, output 1; otherwise, output 0.

Theorem 6.4. *Assuming the average-case hardness of $CSIS_{m_r, \beta, \zeta}$, where $\beta = b\sqrt{m_r}$, $\zeta = \frac{1}{8n}$ (which can be based on the worst-case hardness of $SIVP_{\tilde{O}(b\sqrt{m_r n})}$ due to Theorem 4.17). If Π_{crh} satisfies $f(\cdot)$-approximate collision resistance and $g(\cdot)$-consistency, where f, g satisfies $\frac{1}{3} f(\ell) \geq q(\ell) > 4g(\ell)$ for all $\ell \in \mathbb{N}^+$, then Π^q_{rvc} in Scheme 6.3 is a computationally binding real value commitment scheme with verification threshold $q(\ell)$.*

Moreover, with a post-processing rounding step in the commit algorithm described in Scheme 6.12, and $m_r > 10n \log \frac{b}{q(\ell_P)}$, the real value commitment scheme is also statistically hiding.

The proof of Theorem 6.4 consists of three parts: correctness (Proposition 6.6), computationally binding (Theorem 6.8), and statistical hiding (Corollary 6.13).

Remark 6.5. *By using the updated commit algorithm in Scheme 6.12 (which can be omitted for certain commitment implementations), the precision length ℓ_P has an upper bound depending on the vector length m_r, which is determined in* Setup*. Nonetheless, considering that the consistency error g and the collision resistance threshold f of Scheme 5.22 are both $2^{-\tilde{\Theta}(\ell)}$, $\ell = n$, we can choose $m_r = \Theta(n^2)$ and $q(\ell) = 2^{-\tilde{\Theta}(\ell)}$ so that the upper bound of ℓ_P is $\tilde{\Theta}(n)$, which is sufficient for most applications. By choosing $b \in O(1)$, the computation assumption of the scheme is then $SIVP_{\tilde{O}(n^{1.5})}$ plus any real value hash function with approximate collision resistance f and consistency g set above.*

Correctness. For the correctness property to hold, we need the consistency of the hash function family to bound the gap between $h_{\ell_P}(\mathbf{x})$ and $h_{\ell_V}(\mathbf{x})$.

Proposition 6.6. *If Π_{crh} satisfies $g(\cdot)$-consistency, then for $q(\ell) > 4g(\ell)$, Π^q_{rvc} satisfies correctness.*

Proof. When both Commit and VerifyOpen are honestly executed, we have

$$\mathbf{c}_P =_{\epsilon_P} (h_{\ell_P}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1, \quad \mathbf{c}_V =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1,$$

which implies

$$\|\mathsf{cm} - \mathbf{c}_V\|_{\mathbb{T}} = \|\mathbf{c}_P - \mathbf{c}_V\|_{\mathbb{T}} \le \epsilon_P + \epsilon_V + \|h_{\ell_P}(\mathbf{x}) - h_{\ell_V}(\mathbf{x})\|_{\mathbb{T}} \le \frac{1}{2}q(\ell_V) + \|h_{\ell_P}(\mathbf{x}) - h_{\ell_V}(\mathbf{x})\|_{\mathbb{T}}$$

since $\ell_V \le \ell_P$. By the $g(\cdot)$-consistency of Π_{crh}, we have

$$\begin{aligned}\|h_{\ell_P}(\mathbf{x}) - h_{\ell_V}(\mathbf{x})\|_{\mathbb{T}} &\le \|\mathsf{Hash}(h, \mathbf{x}, \ell_P) - \mathsf{IdealHash}(h, \mathbf{x})\| + \|\mathsf{Hash}(h, \mathbf{x}, \ell_V) - \mathsf{IdealHash}(h, \mathbf{x})\| \\ &\le g(\ell_P) + g(\ell_V) < \frac{1}{2}q(\ell_V).\end{aligned}$$

Therefore, $\|\mathsf{cm} - \mathbf{c}_V\|_{\mathbb{T}} < \frac{1}{2}q(\ell_V) + \frac{1}{2}q(\ell_V) = q(\ell_V)$ and therefore VerifyOpen always outputs 1. □

Computationally Binding. As we have mentioned, the hash function should work on a smaller scale than $\zeta = \frac{1}{8n}$. This can be achieved by rescaling the output.

Proposition 6.7. *Suppose the hash function family $\Pi_{\mathsf{crh}} = (\mathsf{Gen}, \mathsf{Hash})$ satisfies $f(\cdot)$-approximate collision resistance and $g(\cdot)$-consistency, and the output of Hash is within $(-M, M)^n$ for some polynomially bounded M. Then the hash function family $\Pi'_{\mathsf{crh}} = (\mathsf{Gen}', \mathsf{Hash}')$ defined by*

$$\mathsf{Gen}' = \mathsf{Gen}, \quad \mathsf{Hash}' = \frac{1}{100Mn}\mathsf{Hash}$$

satisfies $f'(\cdot)$-approximate collision resistance and $g'(\cdot)$-consistency, where $f'(\ell) = f(\ell)/100Mn$ and $g'(\ell) = g(\ell)/100Mn$. More importantly, the output of Hash' is within $\left(-\frac{1}{100n}, \frac{1}{100n}\right)^n$.

In the following, we assume that the hash output is within $\left(-\frac{1}{100n}, \frac{1}{100n}\right)^n$.

We derive the computational binding property of Π^q_{rvc} based on the approximate collision resistance of Π_{crh} and the average-case hardness of CSIS.

Theorem 6.8. *Assuming the average-case hardness of $\mathrm{CSIS}_{m_r,\beta,\zeta}$, where $\beta = b\sqrt{m_r}, \zeta = \frac{1}{8n}$ (which, by Theorem 4.17, can be based on the worst-case hardness of $\mathrm{SIVP}_{\tilde{O}(b\sqrt{m_r n})}$), if Π_{crh} satisfies $f(\cdot)$-approximate collision resistance, then for $q(\ell) \le \frac{1}{3}f(\ell)$, Π^q_{rvc} is computationally binding.*

Proof. For any efficient adversary $\mathcal{A}$, we say $\mathcal{A}(\mathsf{pp})$ succeeds if it finds $(\mathsf{cm}, \mathbf{x}, \pi_o = \mathbf{r}, \mathbf{x}', \pi'_o = \mathbf{r}')$ such that

$$\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) = 1 \wedge \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}', \pi'_o, \ell_V) = 1 \wedge \|\mathbf{x} - \mathbf{x}'\| \ge 2^{-\ell_V}.$$

We have

$$\Pr_{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)}[\mathcal{A}(\mathsf{pp}) \text{ succeeds}] = \Pr_{\mathsf{pp}}[\mathcal{A}(\mathsf{pp}) \text{ succeeds} \wedge \mathbf{r} = \mathbf{r}'] + \Pr_{\mathsf{pp}}[\mathcal{A}(\mathsf{pp}) \text{ succeeds} \wedge \mathbf{r} \ne \mathbf{r}'].$$

We claim that both terms on the right hand side are negligible.

- If $\Pr_{\mathsf{pp}}[\mathcal{A}(\mathsf{pp}) \text{ succeeds} \wedge \mathbf{r} = \mathbf{r}']$ is non-negligible, then we use $\mathcal{A}$ to construct $\mathcal{A}'$ that breaks the approximate collision resistance of Π_{crh} as follows:

- On input $(1^\lambda, h, \ell_V)$, $\mathcal{A}'$ samples $\mathbf{A}_r \leftarrow \mathbb{T}^{n \times m_r}$ and sets $\mathsf{pp} = (h, \mathbf{A}_r)$.
- $\mathcal{A}'$ invokes $\mathcal{A}$ on input pp and obtains $(\mathsf{cm}, \mathbf{x}, \pi_o = \mathbf{r}, \mathbf{x}', \pi_o' = \mathbf{r}')$. If $\mathbf{r} = \mathbf{r}'$, $\mathcal{A}'$ outputs $(\mathbf{x}, \mathbf{x}')$. Otherwise, $\mathcal{A}'$ fails.

Since pp follows the same distribution as $\mathsf{Setup}(1^\lambda, 1^m)$, with non-negligible probability, $\mathcal{A}$ succeeds with $\mathbf{r} = \mathbf{r}'$. In this case,

$$\|\mathsf{cm} - \mathbf{c}_V\|_{\mathbb{T}} < q(\ell_V), \quad \|\mathsf{cm} - \mathbf{c}_V'\|_{\mathbb{T}} < q(\ell_V),$$

where

$$\mathbf{c}_V =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1, \quad \mathbf{c}_V' =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}') + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1.$$

This implies

$$2q(\ell_V) > \|\mathbf{c}_V - \mathbf{c}_V'\|_{\mathbb{T}} \geq \|(h_{\ell_V}(\mathbf{x}) - h_{\ell_V}(\mathbf{x}')) \bmod 1\|_{\mathbb{T}} - 2\epsilon_V.$$

Since $\|h_{\ell_V}(\mathbf{x}) - h_{\ell_V}(\mathbf{x}')\| < \frac{1}{50n} < \frac{1}{2}$, we have

$$\|h_{\ell_V}(\mathbf{x}) - h_{\ell_V}(\mathbf{x}')\| = \|(h_{\ell_V}(\mathbf{x}) - h_{\ell_V}(\mathbf{x}')) \bmod 1\|_{\mathbb{T}} < 2q(\ell_V) + 2\epsilon_V < f(\ell_V),$$

which means that $\mathcal{A}'$ finds $(\mathbf{x}, \mathbf{x}')$ such that $\|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell_V}$ and $\|h_{\ell_V}(\mathbf{x}) - h_{\ell_V}(\mathbf{x}')\| < f(\ell_V)$ with non-negligible probability.

- If $\Pr_{\mathsf{pp}}[\mathcal{A}(\mathsf{pp}) \text{ succeeds} \wedge \mathbf{r} \neq \mathbf{r}']$ is non-negligible, then we use $\mathcal{A}$ to construct $\mathcal{A}'$ that solves average-case $\mathrm{CSIS}_{m_r,\beta,\zeta}$, where $\beta = b\sqrt{m_r}, \zeta = \frac{1}{8n}$, as follows:
  - On input $(\mathbf{A}, \beta, \zeta)$, $\mathcal{A}'$ samples $h \leftarrow \mathsf{Gen}(1^\lambda, 1^m, 1^n)$ and sets $\mathsf{pp} = (h, \mathbf{A})$.
  - $\mathcal{A}'$ invokes $\mathcal{A}$ on input pp and obtains $(\mathsf{cm}, \mathbf{x}, \pi_o = \mathbf{r}, \mathbf{x}', \pi_o' = \mathbf{r}')$. If $\mathbf{r} \neq \mathbf{r}'$, $\mathcal{A}'$ outputs $\mathbf{z} = \mathbf{r} - \mathbf{r}'$. Otherwise, $\mathcal{A}'$ fails.

  When $\mathbf{A} \leftarrow \mathbb{T}^{n \times m_r}$, pp follows the same distribution as $\mathsf{Setup}(1^\lambda, 1^m)$, so with non-negligible probability, $\mathcal{A}$ succeeds with $\mathbf{r} \neq \mathbf{r}'$. In this case, $\mathbf{z} \neq 0$, $\|\mathbf{z}\|_2 \leq \beta$ and

$$\|\mathsf{cm} - \mathbf{c}_V\|_{\mathbb{T}} < q(\ell_V), \quad \|\mathsf{cm} - \mathbf{c}_V'\|_{\mathbb{T}} < q(\ell_V),$$

where

$$\mathbf{c}_V =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}) + \mathbf{A} \cdot \mathbf{r}) \bmod 1, \quad \mathbf{c}_V' =_{\epsilon_V} (h_{\ell_V}(\mathbf{x}') + \mathbf{A} \cdot \mathbf{r}') \bmod 1.$$

This implies

$$2q(\ell_V) > \|\mathbf{c}_V - \mathbf{c}_V'\|_{\mathbb{T}} \geq \|\mathbf{A}\mathbf{r} - \mathbf{A}\mathbf{r}' \bmod 1\|_{\mathbb{T}} - \|h_{\ell_V}(\mathbf{x})\| - \|h_{\ell_V}(\mathbf{x}')\| - 2\epsilon_V.$$

Since $\|h_{\ell_V}(\mathbf{x})\|, \|h_{\ell_V}(\mathbf{x}')\| < \frac{1}{100n}$, we have

$$\|\mathbf{A}\mathbf{z}\|_{\mathbb{T}} < 2q(\ell_V) + \frac{1}{50n} + 2\epsilon_V < \zeta,$$

which means that $\mathcal{A}'$ finds a solution $\mathbf{z}$ to the CSIS instance $(\mathbf{A}, \beta, \zeta)$. By Corollary 4.17, this contradicts the hardness of $\mathrm{SIVP}_{\tilde{O}(b\sqrt{m_r n})}$.

Together, we have $\Pr_{\mathsf{pp}}[\mathcal{A}(\mathsf{pp}) \text{ succeeds}]$ is negligible, which means that Π_{rvc}^q is computationally binding. □

Remark 6.9. *Combining Proposition 6.6 and Theorem 6.8, we require $g(\ell) < \frac{1}{12} f(\ell)$ for a valid verification threshold $q(\ell)$ to exist. For hash functions natively on real values, since g usually depends on the computation error bound ϵ_ℓ, this can be achieved by selecting a slightly stricter ϵ_ℓ.*

Statistically Hiding. We show that Π^q_{rvc} is statistically hiding under some restrictions on the implementation of the Commit algorithm.

We wish to show the statistical hiding property of Π^q_{rvc} by

$$\left\{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m), (\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P) : (\mathsf{pp}, \mathsf{cm})\right\} \approx_s \mathsf{Setup}(1^\lambda, 1^m) \times U(\mathbb{T}^n).$$

The first issue is that the Commit algorithm involves computation errors. If the error is unrestricted, the output of Commit may concentrate on a fine discrete grid in $\mathbb{T}^n$, making it impossible to be statistically close to $U(\mathbb{T}^n)$.

Even if we ignore the computation errors and aim to show

$$\left\{h \leftarrow \mathsf{Gen}, \mathbf{A}_r \leftarrow \mathbb{T}^{n\times m_r}, \mathbf{r} \leftarrow [b]^{m_r} : (h, \mathbf{A}_r, (h_{\ell_p}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1)\right\} \approx_s \mathsf{Gen} \times U(\mathbb{T}^{n\times m_r}) \times U(\mathbb{T}^n),$$

it is still impossible since for any fixed $h, \mathbf{A}_r$, the set $\{(h_{\ell_p}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 | \mathbf{r} \in [b]^{m_r}\}$ is discrete and hence zero-measure in $\mathbb{T}^n$.

Remark 6.10. *[BCHR22] encountered the same issue when constructing PKEs from hCLWE. To argue the security of their encryption schemes, ideally, they might want to show*

$$\{\mathbf{N} \leftarrow \mathcal{N}(0,1)^{n\times m}, \mathbf{t} \leftarrow \{1/\sqrt{m}, -1/\sqrt{m}\}^m : (\mathbf{N}, \mathbf{N}\mathbf{t})\} \approx_s \mathcal{N}(0,1)^{n\times m} \times \mathcal{N}(0,1)^n,$$

where $\mathcal{N}(0,1)$ denotes the standard normal distribution. This is impossible for the same reason. They resolved this issue by rounding the output to a discrete set, $\mathsf{round}_d : \mathbb{R} \to \{1, 2, \ldots, d\}$, such that $\mathsf{round}_d(\mathcal{N}(0,1)) = U(\{1, 2, \ldots, d\})$. Using an analog of LHL ([BCHR22], 4.5-4.8), they bounded the statistical distance between the two distributions:

$$\{\mathbf{N} \leftarrow \mathcal{N}(0,1)^{n\times m}, \mathbf{t} \leftarrow \{1/\sqrt{m}, -1/\sqrt{m}\}^m : (\mathbf{N}, \mathsf{round}_d(\mathbf{N}\mathbf{t}))\}, \quad \mathcal{N}(0,1)^{n\times m} \times U(\{1, 2, \ldots, d\}^n).$$

Unfortunately, for this rounding argument, the available upper bound $O\left(\sqrt{\frac{n \ln d}{\sqrt{m}}}\right)$ is not negligible when m is polynomially bounded.

Inspired by their approach, we can modify the Commit algorithm to include a rounding step after the computation. Even better, the uniform distribution and $(\bmod 1)$ operation in our setting enjoy some good number-theoretic properties, which allow us to get a negligible bound on the statistical distance:

Theorem 6.11. *Let $p \in \mathbb{N}^*$ satisfy $b < p < 1.2^{\frac{m_r}{n+1}}$, and define the rounding function by $\lfloor a \rceil_p = \lfloor pa \rceil / p$. Then for arbitrary $\mathbf{x}, \ell_P$, abbreviating $h_{\mathbf{x}} = h_{\ell_p}(\mathbf{x})$, we have*

$$\begin{aligned}\left\{h \leftarrow \mathsf{Gen}, \mathbf{A}_r \leftarrow \mathbb{T}^{n\times m_r}, \mathbf{r} \leftarrow [b]^{m_r} : \left(h, \mathbf{A}_r, \lfloor (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \rceil_p\right)\right\} \\ \approx_s \mathsf{Gen} \times U(\mathbb{T}^{n\times m_r}) \times U(\lfloor \mathbb{T}^n \rceil_p),\end{aligned}$$

where $\lfloor \mathbb{T}^n \rceil_p = \{\lfloor \mathbf{t} \rceil_p | \mathbf{t} \in \mathbb{T}^n\} = \mathbb{Z}^n_p / p$.

Proof. See Appendix A for the proof. □

Now we can modify the Commit algorithm to include the rounding step:

Scheme 6.12 (Modified Commit Algorithm from Scheme 6.3)**.** *The modified* Commit *algorithm is:*

- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$: *On input* $\mathsf{pp} = (h, \mathbf{A}_r)$, $\mathbf{x} \in \mathcal{M}^m$, ℓ_P, *choose a rounding parameter* p *with* $\max\left\{b, \frac{2}{\epsilon_P}\right\} < p < 1.2^{\frac{m_r}{n+1}}$ *where* $\epsilon_P = \frac{1}{4}q(\ell_P)$. *First compute* $h_{\ell_P}(\mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}, \ell_P)$. *Then uniformly sample a vector* $\mathbf{r} \leftarrow \{0, 1, \ldots, b-1\}^{m_r}$ *and compute*

$$\mathbf{c}_P =_{\epsilon'_P} (h_{\ell_P}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \in \mathbb{T}^n,$$

where the computation error is bounded by $\epsilon'_P = \frac{1}{p} \cdot 2^{-\omega(\log \lambda)}$. *Output* $(\mathsf{cm} = \lfloor \mathbf{c}_P \rfloor_p, \pi_o = \mathbf{r})$.

p is chosen to be a power of the base for the representation of real numbers, so that the rounding operation is equivalent to truncating the trailing digits in the representation and hence does not introduce additional computation error.

We point out that the modified Commit algorithm can be viewed as a special implementation of the original Commit algorithm, since

$$\|\mathsf{cm} - (h_{\ell_P}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r} \bmod 1)\|_{\mathbb{T}} \leq \|\mathsf{cm} - \mathbf{c}_P\|_{\mathbb{T}} + \epsilon'_P < \frac{1}{p} + \frac{1}{p} < \epsilon_P.$$

Thus, the correctness and computational binding properties of Π^q_{rvc} are preserved.

Now if we ignore ϵ'_P, the statistical hiding property of Π^q_{rvc} directly follows from Theorem 6.11. See Appendix B on how to handle ϵ'_P.

Corollary 6.13. *For* ℓ_P *such that* $m_r > 10n \log \frac{b}{q(\ell_P)}$, Π^q_{rvc} *with the modified* Commit *algorithm is statistically hiding.*

Proof. Under this condition, p with $\max\left\{b, \frac{2}{\epsilon_P}\right\} < p < 1.2^{\frac{m_r}{n+1}}$ can be found. The simulator $\mathcal{S}$ simply samples $\mathbf{u} \leftarrow U(\lfloor \mathbb{T}^n \rfloor_p)$ and outputs $\mathbf{u}$. □

7 Real Value Commitment from Continuous LWE

This section presents a commitment scheme for real-valued messages based on continuous LWE. It first describes the construction, then establishes computational hiding under the CLWE assumption, and finally proves statistical binding.

7.1 Commitment Scheme

Suppose the message space is $\mathcal{M} = [-M, M]$, where $M \geq 1$ is a polynomially bounded integer.

Scheme 7.1. *The real value commitment scheme based on CLWE is defined as:*

- $\mathsf{Setup}(1^\lambda, 1^m)$: *On input the security parameter* λ *and vector length* m, *set* $n = \lambda$ *and choose appropriate parameters* $k > n + m, \gamma \geq 2\sqrt{n}, 0 < \sigma \leq \beta < \frac{1}{100\sqrt{n}}$, *such that* $k\gamma/\sigma$ *is polynomially bounded. Sample* $\mathbf{A} \leftarrow D^{(n+m)\times k}_{\mathbb{R},1}$ *and output* $\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma)$.
- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$: *On input the public parameter* $\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma)$, *a message vector* $\mathbf{x} \in \mathcal{M}^m$, *and a precision length* ℓ_P, *sample a unit vector* $\mathbf{w} \leftarrow \mathcal{S}^{n-1}$ *and a noise vector* $\mathbf{e} \leftarrow \tilde{D}^k_{\mathbb{R},\sigma}$. *Compute*

$$\mathbf{c}_P =_{\epsilon_P} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1 \in \mathbb{T}^k,$$

where the computation error is bounded by $\epsilon_P = \frac{1}{4}\beta\sqrt{n}$. *Output the commitment* $\mathsf{cm} = \mathbf{c}_P$ *and the opening proof* $\pi_o = (\mathbf{w}, \mathbf{e})$.

- $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V)$: *On input the public parameter* $\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma)$, *the commitment* cm, *the message vector* $\mathbf{x}$, *the opening proof* $\pi_o = (\mathbf{w}, \mathbf{e})$, *and a precision length* ℓ_V, *compute*

$$\mathbf{c}_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1 \in \mathbb{T}^k,$$

where the computation error is bounded by $\epsilon_V = \frac{1}{4}\beta\sqrt{n}$.
If $\|\mathbf{x}\| \le M, \|\mathbf{w}\| \le M, \|\mathbf{e}\| < \beta\sqrt{n}, \|\mathbf{c}_V - \mathsf{cm}\|_{\mathbb{T}} \le \beta\sqrt{n}$, *output* 1; *otherwise, output* 0.

Since noticeable noise $\mathbf{e}$ is added to the commitment, there is no way to detect negligible deviation of the message, which means that the precision length is bounded by $O(\log n)$. We directly use fixed computation error bounds and verification threshold that handles all the supported precision lengths, so ℓ_P, ℓ_V do not appear in the algorithms.

Theorem 7.2. *Assume the hardness of* $\mathrm{CLWE}_{\sigma,\gamma}$, *when* $k \ge \Theta((n+m)\log n)$, *Scheme 7.1 is a statistically binding real value commitment scheme that works for* $\ell_V \le \ell_P < \log\left(\frac{\gamma}{100\beta\sqrt{n}}\right)$. *Moreover, with a post-processing rounding step in the commit algorithm described in Appendix C, it is also computationally hiding.*

Correctness of the scheme is ensured by the computation error bound. When both Commit and VerifyOpen are honestly executed, we have $\|\mathbf{c}_V - \mathbf{c}_P\|_{\mathbb{T}} \le \epsilon_P + \epsilon_V < \beta\sqrt{n}$, which implies that VerifyOpen will always output 1. Computationally hiding and statistically binding are shown in the next two subsections.

7.2 Computationally Hiding

In this subsection, we ignore the computation error in Commit for simplicity. See Appendix C on how to handle the computation error by slightly restricting the Commit algorithm.

Lemma 7.3. *Assume the hardness of* $\mathrm{CLWE}_{\sigma,\gamma}$ *where* σ, γ *satisfy the constraints in Scheme 7.1. Then for arbitrary* $\mathbf{x} \in \mathcal{M}^m$, *we have*

$$\left\{\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma) \leftarrow \mathsf{Setup}(1^\lambda, 1^m), \mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell) : (\mathbf{A}, \mathsf{cm})\right\} \approx_c D_{\mathbb{R},1}^{(n+m)\times k} \times U(\mathbb{T}^k).$$

Proof. Suppose $\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma)$, where $\mathbf{A} = \begin{pmatrix} \mathbf{A}_\mathbf{x} \\ \mathbf{A}_\mathbf{w} \end{pmatrix}, \mathbf{A}_\mathbf{x} \in \mathbb{R}^{m\times k}, \mathbf{A}_\mathbf{w} \in \mathbb{R}^{n\times k}$. We have

$$\begin{aligned}
&\left\{\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma) \leftarrow \mathsf{Setup}(1^\lambda, 1^m), \mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell) : (\mathbf{A}, \mathsf{cm})\right\} \\
=&\left\{\mathbf{A}_\mathbf{x} \leftarrow D_{\mathbb{R},1}^{m\times k}, \mathbf{A}_\mathbf{w} \leftarrow D_{\mathbb{R},1}^{n\times k}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow \tilde{D}_{\mathbb{R},\sigma}^k : \left(\begin{pmatrix} \mathbf{A}_\mathbf{x} \\ \mathbf{A}_\mathbf{w} \end{pmatrix}, \gamma\mathbf{A}_\mathbf{x}^T\mathbf{x} + \gamma\mathbf{A}_\mathbf{w}^T\mathbf{w} + \mathbf{e} \bmod 1\right)\right\} \\
\approx_s&\left\{\mathbf{A}_\mathbf{x} \leftarrow D_{\mathbb{R},1}^{m\times k}, \mathbf{A}_\mathbf{w} \leftarrow D_{\mathbb{R},1}^{n\times k}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow D_{\mathbb{R},\sigma}^k : \left(\begin{pmatrix} \mathbf{A}_\mathbf{x} \\ \mathbf{A}_\mathbf{w} \end{pmatrix}, \gamma\mathbf{A}_\mathbf{x}^T\mathbf{x} + \gamma\mathbf{A}_\mathbf{w}^T\mathbf{w} + \mathbf{e} \bmod 1\right)\right\} \\
\approx_c&\left\{\mathbf{A}_\mathbf{x} \leftarrow D_{\mathbb{R},1}^{m\times k}, \mathbf{A}_\mathbf{w} \leftarrow D_{\mathbb{R},1}^{n\times k}, \mathbf{u} \leftarrow U(\mathbb{T}^k) : \left(\begin{pmatrix} \mathbf{A}_\mathbf{x} \\ \mathbf{A}_\mathbf{w} \end{pmatrix}, \gamma\mathbf{A}_\mathbf{x}^T\mathbf{x} + \mathbf{u} \bmod 1\right)\right\} \\
=&D_{\mathbb{R},1}^{(n+m)\times k} \times U(\mathbb{T}^k),
\end{aligned}$$

where $\approx_c$ follows from the hardness of $\mathrm{CLWE}_{\sigma,\gamma}$, $\approx_s$ follows from the fact that $D_{\mathbb{R},\sigma}$ and its cut off distribution $\tilde{D}_{\mathbb{R},\sigma}$ are statistically close. □

Corollary 7.4. *Assume the hardness of $\mathrm{CLWE}_{\sigma,\gamma}$. Then Scheme 7.1 is computationally hiding.*

Proof. The simulator $\mathcal{S}$ simply samples $\mathbf{u} \in \mathbb{T}^k$ uniformly at random and outputs $\mathbf{u}$. The correctness of the simulator follows from Lemma 7.3. □

7.3 Statistically Binding

The goal of this section is to prove the following statement, which further implies the statistical binding property of the commitment scheme.

Proposition 7.5. *With β, γ defined as in Scheme 7.1, for $\Delta = \frac{100\sqrt{n}\beta}{\gamma}$, $k \geq \Theta((n+m)\log n)$, when $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$, with overwhelming probability, we have*

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{M}^m \text{ s.t. } \|\mathbf{x}-\mathbf{x}'\|_2 \geq \Delta; \quad \forall \mathbf{w}, \mathbf{w}' \text{ s.t. } \|\mathbf{w}\|, \|\mathbf{w}'\| \leq M; \quad \forall \mathbf{e}, \mathbf{e}' \text{ s.t. } \|\mathbf{e}\|, \|\mathbf{e}'\| < \beta\sqrt{n},$$

$$\mathbf{c}_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1, \quad \mathbf{c}'_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} + \mathbf{e}' \bmod 1 :$$

$$\|\mathbf{c}_V - \mathbf{c}'_V\|_{\mathbb{T}} > 2\beta\sqrt{n}.$$

We make use of two important lemmas to prove the proposition.

Lemma 7.6. *For arbitrary $\gamma > 0, 0 < \Delta < \frac{1}{\gamma}$ and $\mathbf{v} \in \mathbb{R}^{n+m}$ with $\|\mathbf{v}\|_2 \geq \Delta$, we have*

$$\Pr_{\mathbf{a} \leftarrow D_{\mathbb{R},1}^{n+m}} \left[\|\gamma \mathbf{a}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} > \frac{\gamma\Delta}{10} \right] > \frac{1}{2}.$$

Proof. Suppose $\theta = \gamma\Delta < 1$, $s = \gamma\|\mathbf{v}\|_2 \geq \theta$.
Notice that when $\mathbf{a}$ is sampled from $D_{\mathbb{R},1}^{n+m}$, $\gamma \mathbf{a}^T \mathbf{v}$ follows the distribution $D_{\mathbb{R},\gamma\|\mathbf{v}\|_2} = D_{\mathbb{R},s}$ with density function

$$D_{\mathbb{R},s}(x) = \frac{1}{s} e^{-\pi x^2/s^2}.$$

Thus we have

$$\begin{aligned} \Pr_{\mathbf{a} \leftarrow D_{\mathbb{R},1}^{n+m}} \left[\|\gamma \mathbf{a}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] &= \Pr_{x \leftarrow D_{\mathbb{R},s}} \left[\|x \bmod 1\|_{\mathbb{T}} \leq \frac{\theta}{10} \right] = \sum_{k \in \mathbb{Z}} \int_{k-\frac{\theta}{10}}^{k+\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx \\ &= \int_{-\frac{\theta}{10}}^{\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx + 2\sum_{k=1}^{+\infty} \int_{k-\frac{\theta}{10}}^{k+\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx. \end{aligned}$$

The first part is bounded by

$$\int_{-\frac{\theta}{10}}^{\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx \leq \frac{\theta}{5} \max_{x \in [-\frac{\theta}{10}, \frac{\theta}{10}]} \frac{1}{s} e^{-\pi x^2/s^2} = \frac{\theta}{5s} \leq \frac{1}{5}.$$

The second part is bounded by

$$2\sum_{k=1}^{+\infty} \int_{k-\frac{\theta}{10}}^{k+\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx \leq 2\sum_{k=1}^{+\infty} \frac{\theta}{5} \int_{k-1+\frac{\theta}{10}}^{k+\frac{\theta}{10}} D_{\mathbb{R},s}(x)dx = \frac{2\theta}{5} \int_{\frac{\theta}{10}}^{+\infty} D_{\mathbb{R},s}(x)dx \leq \frac{2\theta}{5} \cdot \frac{1}{2} \leq \frac{1}{5}.$$

Therefore,

$$\Pr_{\mathbf{a} \leftarrow D_{\mathbb{R},1}^{n+m}} \left[\|\gamma \mathbf{a}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] \leq \frac{2}{5} < \frac{1}{2}.$$

The lemma follows. □

Lemma 7.7. *For arbitrary $\gamma \geq 1$, $\frac{1}{\mathsf{poly}(n)} \leq \Delta < \frac{1}{\gamma}$, $m \leq \mathsf{poly}(n)$ and integer $M \leq \mathsf{poly}(n)$, for $k \geq \Theta((n+m)\log n)$, when $\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}$, with overwhelming probability, for all $\mathbf{v} \in [-2M, 2M]^{n+m}$ with $\|\mathbf{v}\|_2 \geq \Delta$, we have*

$$\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} > \frac{\gamma\Delta}{20}.$$

Proof. When $\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}$, with overwhelming probability, each entry of $\mathbf{A}$ is within $[-\sqrt{n}, \sqrt{n}]$. In the following, we assume this is the case.
Let $C = \left\lceil \frac{20(n+m)\sqrt{n}}{\Delta} \right\rceil$. Consider the set

$$S = \left\{ \mathbf{v} \in [-2M, 2M]^{n+m} : C\mathbf{v} \in \mathbb{Z}^{n+m}, \|\mathbf{v}\|_2 \geq \Delta \right\}.$$

Write $\mathbf{A} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_k \end{pmatrix}$. Then

$$\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} = \max_{1 \leq i \leq k} \|\gamma \mathbf{a}_i^T \mathbf{v} \bmod 1\|_{\mathbb{T}}.$$

$$\therefore \|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \iff \forall 1 \leq i \leq k, \|\gamma \mathbf{a}_i^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10}.$$

For each $\mathbf{v} \in S$, by Lemma 7.6,

$$\Pr_{\mathbf{a} \leftarrow D_{\mathbb{R},1}^{n+m}} \left[\|\gamma \mathbf{a}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] \leq \frac{1}{2}.$$

$$\therefore \Pr_{\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}} \left[\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] = \prod_{i=1}^{k} \Pr_{\mathbf{a}_i \leftarrow D_{\mathbb{R},1}^{n+m}} \left[\|\gamma \mathbf{a}_i^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] \leq 2^{-k}.$$

By union bound,

$$\Pr_{\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}} \left[\exists \mathbf{v} \in S, \|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \leq \frac{\gamma\Delta}{10} \right] \leq |S| \cdot 2^{-k}.$$

Notice that $|S| \leq (5MC)^{n+m} = 2^{O((n+m)\log n)}$. Choose $k \geq \log|S| + n = O((n+m)\log n)$. Then $|S| \cdot 2^{-k} \leq 2^{-n}$ is negligible. In other words, when $\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}$, with overwhelming probability, we have

$$\forall \mathbf{v} \in S, \|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} > \frac{\gamma\Delta}{10}.$$

We show that this implies the required statement. For arbitrary $\mathbf{v} \in [-2M, 2M]^{n+m}$ with $\|\mathbf{v}\|_2 \geq \Delta$, let $\overline{\mathbf{v}} = \lceil C\mathbf{v} \rceil_{|\cdot|} / C$. Here $\lceil x \rceil_{|\cdot|} = \mathrm{sgn}(x)\lceil |x| \rceil$. Then $\overline{\mathbf{v}} \in S$ and $\|\mathbf{v} - \overline{\mathbf{v}}\| \leq \frac{1}{C}$.
Since each entry of $\mathbf{A}$ is within $[-\sqrt{n}, \sqrt{n}]$, we have

$$\|\gamma \mathbf{A}^T (\mathbf{v} - \overline{\mathbf{v}})\| \leq \gamma(n+m)\sqrt{n} \cdot \frac{1}{C} \leq \frac{\gamma\Delta}{20}.$$

Therefore,

$$\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} \geq \|\gamma \mathbf{A}^T \overline{\mathbf{v}} \bmod 1\|_{\mathbb{T}} - \|\gamma A^T \mathbf{v} - \gamma A^T \overline{\mathbf{v}}\| > \frac{\gamma\Delta}{10} - \frac{\gamma\Delta}{20} = \frac{\gamma\Delta}{20}.$$

□

Now we prove Proposition 7.5.

Proof of Proposition 7.5. By Lemma 7.7, when $\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma) \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$, with overwhelming probability, for all $\mathbf{v} \in [-2M, 2M]^{n+m}$ with $\|\mathbf{v}\|_2 \geq \Delta$, we have

$$\|\gamma \mathbf{A}^T \mathbf{v} \bmod 1\|_{\mathbb{T}} > \frac{\gamma\Delta}{20} = 5\beta\sqrt{n}.$$

Now consider arbitrary $\mathbf{x}, \mathbf{x}' \in [-M, M]^m$ with $\|\mathbf{x} - \mathbf{x}'\|_2 \geq \Delta$,

$$\mathbf{c}_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1, \quad \mathbf{c}'_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} + \mathbf{e}' \bmod 1,$$

where $\mathbf{w}, \mathbf{w}' \in [-M, M]^n$ and $\|\mathbf{e}\|, \|\mathbf{e}'\| < \beta\sqrt{n}$.

Notice that

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} - \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} \in [-2M, 2M]^{n+m}, \left\| \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} - \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} \right\|_2 \geq \|\mathbf{x} - \mathbf{x}'\|_2 \geq \Delta.$$

Thus,

$$\left\| \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} - \gamma A^T \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} \bmod 1 \right\|_{\mathbb{T}} > 5\beta\sqrt{n}.$$

Since $\|\mathbf{e}\|, \|\mathbf{e}'\| < \beta\sqrt{n}$ by the accepting bounds, this implies

$$\|\mathbf{c}_V - \mathbf{c}'_V\|_{\mathbb{T}} > 5\beta\sqrt{n} - \|\mathbf{e}\| - \|\mathbf{e}'\| - 2\epsilon_V > 2\beta\sqrt{n}.$$

□

Corollary 7.8. *For $\ell_V < \log\left(\frac{\gamma}{100\beta\sqrt{n}}\right)$ and $k \geq \Theta((n+m)\log n)$, Scheme 7.1 is statistically binding.*

Proof. Suppose an unbounded adversary $\mathcal{A}(\mathsf{pp})$ outputs $(\mathsf{cm}, \mathbf{x}, \pi_o = (\mathbf{w}, \mathbf{e}), \mathbf{x}', \pi'_o = (\mathbf{w}', \mathbf{e}'))$ such that

$$\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell_V) = 1 \wedge \mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}', \pi'_o, \ell_V) = 1 \wedge \|\mathbf{x} - \mathbf{x}'\| \geq 2^{-\ell_V}.$$

This implies that $\mathbf{x}, \mathbf{x}' \in \mathcal{M}^m, \|\mathbf{x} - \mathbf{x}'\|_2 \geq \|\mathbf{x} - \mathbf{x}'\| > \frac{100\beta\sqrt{n}}{\gamma}, \|\mathbf{w}\|, \|\mathbf{w}'\| \leq M, \|\mathbf{e}\|, \|\mathbf{e}'\| < \beta\sqrt{n}$. The VerifyOpen algorithm computes

$$\mathbf{c}_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1, \quad \mathbf{c}'_V =_{\epsilon_V} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x}' \\ \mathbf{w}' \end{pmatrix} + \mathbf{e}' \bmod 1,$$

and ensures $\|\mathbf{c}_V - \mathsf{cm}\|_{\mathbb{T}} \leq \beta\sqrt{n}, \|\mathbf{c}'_V - \mathsf{cm}\|_{\mathbb{T}} \leq \beta\sqrt{n}$, which implies $\|\mathbf{c}_V - \mathbf{c}'_V\|_{\mathbb{T}} \leq 2\beta\sqrt{n}$. However, by Proposition 7.5, when $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$, this happens with negligible probability. Therefore, Scheme 7.1 is statistically binding. □

8 Sigma Protocol from CLWE Commitment

This section presents an application of the commitment scheme from Section 7, namely a Σ-protocol for real-valued messages. The protocol satisfies special soundness and provides honest-verifier zero knowledge (HVZK) for both accepting and **aborting** executions.

For simplicity, we ignore finite-precision computation errors and assume that all computations in the Σ-protocol are exact. A similar result holds when considering finite-precision errors, using techniques discussed in Appendix C.

Let us first give the definition of a Σ-protocol. Following [BKLP15, Definition 2.2], the definition allows a completeness error α and special soundness with respect to relaxed witness relations. Strong honest-verifier zero knowledge is defined following [DFPS23].

Definition 8.1 (Σ-Protocol with Abort)**.** *Let (P, V) be a two-party protocol between a prover P and a PPT verifier V. Let λ be the security parameter, and let $\mathcal{R} \subseteq \mathcal{R}'$ be binary relations. With challenge set $\mathcal{C}$, common public input x, and P's private input ω, the protocol is a Σ-protocol for $(\mathcal{R}, \mathcal{R}')$ if it consists of three messages: a first message t from P, a uniformly sampled challenge $c \leftarrow \mathcal{C}$ from V, and a proof ρ from P, and satisfies the following properties.*

1. Completeness. *There exists an α with $0 \le \alpha < 1 - 1/|\mathcal{C}|$ such that, for every $(x, \omega) \in \mathcal{R}$, the honest verifier V accepts an honest execution with probability at least $1 - \alpha$.*

2. Special soundness. *There is an efficient extractor* Ext *that, given x and two accepting transcripts (t, c_0, ρ_0) and (t, c_1, ρ_1) for x with distinct $c_0, c_1 \in \mathcal{C}$, returns ω^* such that $(x, \omega^*) \in \mathcal{R}'$ with overwhelming probability.*

3. Strong honest-verifier zero knowledge. *There is an efficient simulator* Sim *such that, for every witness ω and prescribed $c \in \mathcal{C}$, randomly sampled public parameters pp, and all statement $x = (y, \mathsf{pp})$ such that $(x, \omega) \in \mathcal{R}$,*

$$\mathsf{Sim}(1^\lambda, x, c) \approx_{\text{comp}} \mathsf{View}_{V,c}(1^\lambda, x, \omega),$$

where indistinguishability is over the joint randomness of Setup *and the real or simulated execution. The simulator receives no witness. Here $\mathsf{View}_{V,c}$ denotes V's complete view of an honest execution with prescribed challenge c: the public input, messages, and decision.*

Let us remark that, in the definition of HVZK above, the verifier's view includes aborts and is not conditioned on acceptance.

Parameters and relations. Let $\lambda = n \ge 2$ be the security parameter and let $m \ge 1$ be an integer. Sample the public commitment key $\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+1)\times m}$, and let

$$L(s, \mathbf{w}, \mathbf{e}) = \gamma \mathbf{A}^\mathsf{T} \begin{pmatrix} s \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \pmod 1.$$

Let $\mathbf{y} = L(s, \mathbf{w}, \mathbf{e}) \in \mathbb{T}^m$ be the public commitment. Let $B \ge 1$ be a common coordinate bound for (s, w). Let $d = n + m + 1$, and let $K \ge 4d$ be a public integer. All dimensions, B, and K are polynomially bounded. Choose $\gamma \ge 2\sqrt{n}$ and $0 < \beta < 1/(100\sqrt{n})$ such that $m\gamma/\beta$ is polynomially bounded, and set

$$\sigma = \frac{\beta}{2K}, \qquad b = \sigma\sqrt{n} = \frac{\beta\sqrt{n}}{2K}.$$

Variable	Meaning	Function of n	Default Setting
n	Security parameter	n	$\lambda = n \geq 2$
m	Number of columns of $\mathbf{A}$	$\mathrm{poly}(n)$	$m = O(n \log n)$
$\mathbf{A}$	Public commitment key	$(n+1) \times m$	$\mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+1)\times m}$
γ	Density of Layers	$\gamma \leq \mathrm{poly}(n)$	$\gamma \geq 2\sqrt{n}$
β	Verification error parameter	$\beta^{-1} \leq \mathrm{poly}(n)$	$0 < \beta < \frac{1}{100\sqrt{n}}$
σ	Gaussian error width	$\sigma^{-1} \leq \mathrm{poly}(n)$	$\sigma = \frac{\beta}{2K}$
B	Secret bound	$B \leq \mathrm{poly}(n)$	$B \geq 1$
b	Error bound	$b^{-1} \leq \mathrm{poly}(n)$	$b = \sigma\sqrt{n}$
d	Dimension of an opening	$\mathrm{poly}(n)$	$d = n + m + 1$
K	Mask expansion factor	$K \leq \mathrm{poly}(n)$	$K \in \mathbb{N}, K \geq 4d$
$\mathcal{C}$	Challenge set	Constant size	$\mathcal{C} = \{0, 1\}$
p	Acceptance probability	$\Theta(1)$	$p = ((K-1)/K)^d > 3/4$
α	Completeness error	$O(1)$	$\alpha = 1 - p < 1/4$

Table 2: Summary of global parameters and derived quantities in this section.

Then $2Kb = \beta\sqrt{n}$ and $m\gamma/\sigma$ is polynomially bounded. For each $\ell > 0$, define

$$\mathcal{D}_\ell = \{(s, \mathbf{w}, \mathbf{e}) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m : |s| \leq \ell B, \quad \|\mathbf{w}\|_\infty \leq \ell B, \quad \|\mathbf{e}\|_\infty < \ell b\}.$$

Since $\mathcal{D}_\ell$ is obtained by scaling each of the d coordinates of $\mathcal{D}_1$ by ℓ, it follows that

$$\mathrm{vol}(\mathcal{D}_\ell) = \ell^d \,\mathrm{vol}(\mathcal{D}_1).$$

For a vector $\mathbf{v} \in \mathbb{R}^d$, define the translated sets by

$$\mathcal{D}_\ell + \mathbf{v} = \{\mathbf{z} + \mathbf{v} : \mathbf{z} \in \mathcal{D}_\ell\},$$
$$\mathcal{D}_\ell - \mathbf{v} = \{\mathbf{z} - \mathbf{v} : \mathbf{z} \in \mathcal{D}_\ell\}.$$

Let the public parameters be

$$\mathsf{pp} = (n, m, \mathbf{A}, \gamma, \beta, \sigma, B, K).$$

Let the public input be

$$x = (\mathbf{y}, \mathsf{pp}).$$

The relations are defined as follows:

$$\mathcal{R} = \{(x, (s, \mathbf{w}, \mathbf{e})) : \mathbf{y} = L(s, \mathbf{w}, \mathbf{e}), \quad |s| \leq B, \quad \|\mathbf{w}\|_\infty \leq B, \quad \|\mathbf{e}\|_\infty < b\},$$
$$\mathcal{R}' = \{(x, (s, \mathbf{w}, \mathbf{e})) : \mathbf{y} = L(s, \mathbf{w}, \mathbf{e}), \quad |s| \leq 2KB, \quad \|\mathbf{w}\|_\infty \leq 2KB, \|\mathbf{e}\|_\infty < 2Kb\}.$$

Protocol. Let $\mathcal{C} = \{0, 1\}$ be the challenge space. The prover knows $\omega = (s, \mathbf{w}, \mathbf{e})$ with $(x, \omega) \in \mathcal{R}$.

1. **First message.** Sample $\mathbf{w}_0$ uniformly from the unit sphere in $\mathbb{R}^n$ and $\mathbf{e}_0 \leftarrow \tilde{D}_{\mathbb{R},\sigma}^m$, as in Section 7. Let $\omega_0 = (0, \mathbf{w}_0, \mathbf{e}_0)$ and independently sample $\mathbf{v} \leftarrow \mathsf{U}(\mathcal{D}_K)$. Set

$$(r, \mathbf{r}_w, \mathbf{r}_e) = \omega_0 + \mathbf{v}, \qquad \mathbf{t} = L(r, \mathbf{r}_w, \mathbf{r}_e).$$

Send $\mathbf{t}$.

2. **Challenge.** The verifier samples $c \leftarrow \{0,1\}$ uniformly and sends c.

3. **Response and verification.** Compute

$$z_s = r + cs, \qquad \mathbf{z}_w = \mathbf{r}_w + c\mathbf{w}, \qquad \mathbf{z}_e = \mathbf{r}_e + c\mathbf{e}.$$

Let $\mathbf{h} = \mathbf{v} + c\omega$. If $\mathbf{h} \notin \mathcal{D}_{K-1}$, send $\perp$. Otherwise send $\mathbf{z} = (z_s, \mathbf{z}_w, \mathbf{z}_e)$. The verifier rejects $\perp$ and malformed messages. For a well-formed response, it checks $c \in \{0,1\}$ and $\mathbf{z} \in \mathcal{D}_K$, and accepts if and only if

$$L(z_s, \mathbf{z}_w, \mathbf{z}_e) = \mathbf{t} + c\mathbf{y} \pmod 1.$$

Theorem 8.2 (CLWE-based Σ-protocol for real values)**.** *For the parameters specified in Table 2, assume that there is no quantum polynomial-time algorithm that can solve* SIVP_η *or* GapSVP_η *for some* $\eta = \widetilde{O}(n/\sigma)$*. Then the protocol described above is a* Σ*-protocol with abort for* $(\mathcal{R}, \mathcal{R}')$ *and challenge set* $\mathcal{C} = \{0,1\}$*. It satisfies completeness with*

$$\alpha = 1 - \left(\frac{K-1}{K}\right)^d < \frac{1}{2}.$$

It also satisfies special soundness with respect to $\mathcal{R}'$ *and computational strong honest-verifier zero knowledge, including aborting executions.*

We prove the theorem using the following three lemmas, which establish completeness, special soundness, and computational strong honest-verifier zero knowledge, respectively.

Lemma 8.3 (Completeness)**.** *For either challenge, an honest execution is accepted with probability*

$$p = \left(\frac{K-1}{K}\right)^d \geq \frac{3}{4}.$$

Hence completeness holds with $\alpha = 1 - p < 1/2 = 1 - 1/|\mathcal{C}|$.

Proof. First we show that, for either challenge, the prover does not abort with probability p. It does not abort exactly when $\mathbf{h} \in \mathcal{D}_{K-1}$. Since $\mathbf{v}$ is uniform on $\mathcal{D}_K$, $\mathbf{h} = \mathbf{v} + c\omega$ is uniform on $\mathcal{D}_K + c\omega$. Each coordinate of $(s, \mathbf{w})$ has magnitude at most B, and each coordinate of $\mathbf{e}$ has magnitude less than b. Therefore, $\mathcal{D}_{K-1} \subseteq \mathcal{D}_K + c\omega$.

$$\begin{aligned}
\Pr[\mathbf{h} \in \mathcal{D}_{K-1}] &= \frac{\mathrm{vol}(\mathcal{D}_{K-1})}{\mathrm{vol}(\mathcal{D}_K + c\omega)} \\
&= \frac{\mathrm{vol}(\mathcal{D}_{K-1})}{\mathrm{vol}(\mathcal{D}_K)} \\
&= \frac{(K-1)^d \,\mathrm{vol}\,\mathcal{D}_1}{K^d \,\mathrm{vol}\,\mathcal{D}_1} = p.
\end{aligned}$$

If the prover does not abort, $\mathbf{h} \in \mathcal{D}_{K-1}$. Since $\omega_0 \in \mathcal{D}_1$, the response $\mathbf{z} = \omega_0 + \mathbf{h} \in \mathcal{D}_K$ and

$$L(z_s, \mathbf{z}_w, \mathbf{z}_e) = \mathbf{t} + c\mathbf{y} \pmod 1.$$

The verifier therefore accepts. Thus the acceptance probability is also p. Since $d \geq 3$ and $K \geq 4d$, Bernoulli's inequality gives,

$$p = \left(\frac{K-1}{K}\right)^d \geq 1 - \frac{d}{K} \geq \frac{3}{4}.$$

In particular, $\alpha = 1 - p < 1/2$, as required for binary challenges. □

Lemma 8.4 (Special soundness)**.** *Given two accepting transcripts* $(\mathbf{t}, 0, \rho_0)$ *and* $(\mathbf{t}, 1, \rho_1)$ *for* x*, an efficient extractor outputs* ω^* *satisfying* $(x, \omega^*) \in \mathcal{R}'$ *with overwhelming probability.*

Proof. Assume the accepting transcripts are

$$(\mathbf{t}, 0, z_{0,s}, \mathbf{z}_{0,w}, \mathbf{z}_{0,e}), \qquad (\mathbf{t}, 1, z_{1,s}, \mathbf{z}_{1,w}, \mathbf{z}_{1,e}).$$

The extractor outputs $\omega^* = (s^*, \mathbf{w}^*, \mathbf{e}^*)$, where

$$s^* = z_{1,s} - z_{0,s}, \qquad \mathbf{w}^* = \mathbf{z}_{1,w} - \mathbf{z}_{0,w}, \qquad \mathbf{e}^* = \mathbf{z}_{1,e} - \mathbf{z}_{0,e}.$$

Subtracting the two verification equations gives

$$\begin{aligned} \mathbf{y} &= L(z_{1,s}, \mathbf{z}_{1,w}, \mathbf{z}_{1,e}) - L(z_{0,s}, \mathbf{z}_{0,w}, \mathbf{z}_{0,e}) \\ &= L(s^*, \mathbf{w}^*, \mathbf{e}^*) \pmod 1. \end{aligned}$$

Since both transcripts are accepted, $\mathbf{z}_0, \mathbf{z}_1 \in \mathcal{D}_K$, and hence

$$\begin{aligned} |s^*| &\le |z_{1,s}| + |z_{0,s}| \le 2KB, \\ \|\mathbf{w}^*\|_\infty &\le \|\mathbf{z}_{1,w}\|_\infty + \|\mathbf{z}_{0,w}\|_\infty \le 2KB, \\ \|\mathbf{e}^*\|_\infty &\le \|\mathbf{z}_{1,e}\|_\infty + \|\mathbf{z}_{0,e}\|_\infty < 2Kb. \end{aligned}$$

Therefore, $(x, \omega^*) \in \mathcal{R}'$. □

Lemma 8.5 (Strong honest-verifier zero knowledge)**.** *Assume that there is no quantum polynomial-time algorithm that can solve* SIVP_η *or* GapSVP_η *for some* $\eta = \widetilde{O}(n/\sigma)$*. There is an efficient simulator* Sim *such that, for every witness* $\omega = (s, \mathbf{w}, \mathbf{e}) \in \mathcal{D}_1$ *and prescribed challenge* $c \in \mathcal{C}$*, and randomly sampled* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *such that* $(x = (L(\omega), \mathsf{pp}), \omega) \in \mathcal{R}$*,*

$$\mathsf{Sim}(1^\lambda, x, c) \;\approx_{\mathrm{comp}}\; \mathsf{View}_{V,c}(1^\lambda, x, \omega).$$

Proof. Let c be a prescribed challenge and let $p = ((K-1)/K)^d$. Acceptance occurs if and only if $\mathbf{h} = \mathbf{v} + c\omega \in \mathcal{D}_{K-1}$. For fixed c and ω, this event occurs with probability p over the uniform choice of $\mathbf{v}$. The simulator can therefore first choose acceptance with probability p and abort with probability $1 - p$.

- *Acceptance.* The simulator samples ω_0' as in the protocol and independently samples $\mathbf{h}' \leftarrow \mathsf{U}(\mathcal{D}_{K-1})$. It computes

  $$\mathbf{z}' = \omega_0' + \mathbf{h}', \qquad \mathbf{t}' = L(\mathbf{z}') - c\mathbf{y} \pmod 1,$$

  and outputs $(1^\lambda, x, \mathbf{t}', c, \mathbf{z}', \mathsf{accept})$.

  Using $\mathbf{y} = L(\omega)$, the simulator's first message $\mathbf{t}'$ satisfies

  $$\begin{aligned} \mathbf{t}' &= L(\mathbf{z}') - c\mathbf{y} \\ &= L(\omega_0' + \mathbf{h}' - c\omega) \pmod 1. \end{aligned}$$

  The real and simulated views conditioned on acceptance are

  $$\begin{aligned} \text{Real:} \quad & (1^\lambda, x, L(\omega_0 + \mathbf{h} - c\omega), c, \omega_0 + \mathbf{h}, \mathsf{accept}), \\ \text{Simulated:} \quad & (1^\lambda, x, L(\omega_0' + \mathbf{h}' - c\omega), c, \omega_0' + \mathbf{h}', \mathsf{accept}). \end{aligned}$$

Conditioned on acceptance, $\mathbf{h}$ is uniformly distributed on $\mathcal{D}_{K-1}$ and thus has the same distribution as $\mathbf{h}'$. Moreover, ω_0 and ω_0' are selected from the same distribution and are independent of $\mathbf{h}$ and $\mathbf{h}'$, respectively. Consequently, $(\omega_0, \mathbf{h})$ and $(\omega_0', \mathbf{h}')$ are identically distributed. Therefore, the real and simulated views are statistically indistinguishable.

- *Abort.* The simulator samples ω_0' as in the protocol and independently samples $\mathbf{v}' \leftarrow \mathsf{U}(\mathcal{D}_K)$. It computes $\mathbf{t}' = L(\omega_0' + \mathbf{v}')$ and outputs $(1^\lambda, x, \mathbf{t}', c, \perp, \mathsf{reject})$.

  Let $\mathcal{E} = \{\mathbf{v} + c\omega \notin \mathcal{D}_{K-1}\}$ denote the abort event. The real and simulated views conditioned on abort are

$$\begin{aligned} \text{Real:} \quad & (1^\lambda, x, L(\omega_0 + \mathbf{v}), c, \perp, \mathsf{reject}) \mid \mathcal{E}, \\ \text{Simulated:} \quad & (1^\lambda, x, L(\omega_0' + \mathbf{v}'), c, \perp, \mathsf{reject}). \end{aligned}$$

  By Lemma 7.3,

$$\begin{aligned} (x, L(\omega_0)) &\approx_{\text{comp}} (x, \mathsf{U}(\mathbb{T}^m)), \\ (x, L(\omega_0')) &\approx_{\text{comp}} (x, \mathsf{U}(\mathbb{T}^m)). \end{aligned}$$

  Therefore,

$$\begin{aligned} (x, L(\omega_0 + \mathbf{v})) \mid \mathcal{E} &= (x, L(\omega_0) + L(\mathbf{v})) \mid \mathcal{E} \\ &\approx_{\text{comp}} (x, \mathsf{U}(\mathbb{T}^m) + L(\mathbf{v})) \mid \mathcal{E} \\ &= (x, \mathsf{U}(\mathbb{T}^m)) \\ &\approx_{\text{comp}} (x, L(\omega_0') + L(\mathbf{v}')) \\ &= (x, L(\omega_0' + \mathbf{v}')). \end{aligned}$$

  So real and simulated views are computationally indistinguishable.

The accepting views are statistically indistinguishable and the aborting views are computationally indistinguishable. Thus the full views, including aborts, are computationally indistinguishable. □

9 A Generic Method of Finding Collisions

In this section, we try to attack the binding property of a compressing real value commitment scheme $\Pi_{\mathsf{rvc}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyOpen})$ with an analytical approach. Assume $\ell_V = \ell_P = \ell$ for simplicity. We view the three algorithms as black boxes. We make several assumptions:

- **Perfect correctness:** For any precision length ℓ, message $\mathbf{x} \in \mathcal{M}^m$, $\mathsf{pp} \leftarrow \mathsf{Setup}$, $(\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell)$, we have $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}, \mathbf{x}, \pi_o, \ell) = 1$.
- **Open sets:** The message space $\mathcal{M}$ and the randomness space used by Commit are open sets.
- **Compressing commitment:** The message dimension is larger than the commitment dimension.
- **Stable verification:** For any $(\mathsf{pp}, \mathbf{x}, \ell)$ and $(\mathsf{cm}, \pi_o) \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell)$, when cm' is sufficiently close (up to $2^{-\mathsf{poly}(\lambda)}$) to cm, we have $\mathsf{VerifyOpen}(\mathsf{pp}, \mathsf{cm}', \mathbf{x}, \pi_o, \ell) = 1$.

9.1 Smooth Functions

We analyze the function $\psi(\mathbf{x}) = \mathsf{cm}, (\mathsf{cm}, \pi_o) = \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell)$ for some fixed randomness. Intuitively, since $m > n$, as long as ψ is smooth enough in the neighborhood of some regular point $\mathbf{x}_0$ (with full-rank Jacobian matrix), the set $\{\mathbf{x} : \psi(\mathbf{x}) = \psi(\mathbf{x}_0)\}$ should be a smooth manifold with dimension at least $m - n$. By going along the tangential direction of this manifold, we can find another point $\mathbf{x}'$ such that $\|\mathbf{x}' - \mathbf{x}_0\|$ is large but $\|\psi(\mathbf{x}') - \psi(\mathbf{x}_0)\|$ is small.

Lemma 9.1. *Assume that ψ is second-order continuously differentiable in the ball $\mathcal{B}_2(\mathbf{x}_0, r)$ and the second-order derivative of ψ is polynomially bounded by M in this ball. For any $\alpha < r$, there exists an efficient algorithm that, given oracle access to ψ, finds $\mathbf{x}_1$ such that $\|\mathbf{x}_1 - \mathbf{x}_0\|_2 = \alpha$ and $\|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_0)\|_2 \leq M\alpha^2$.*

Proof. In this case, given any point $\mathbf{x} \in \mathcal{B}_2(\mathbf{x}_0, r)$, we can approximate the Jacobian matrix $\mathbf{J}(\mathbf{x})$ of ψ at $\mathbf{x}$ up to an exponentially small error using the finite difference method. The Taylor expansion of ψ at $\mathbf{x}$ gives

$$\|\psi(\mathbf{x} + \Delta) - \psi(\mathbf{x}) - \mathbf{J}(\mathbf{x})\Delta\|_2 \leq M\|\Delta\|_2^2.$$

Since $\mathbf{J}(\mathbf{x}_0)$ is an $n \times m$ matrix with $m > n$, we can always find a unit vector Δ_1 such that $\mathbf{J}(\mathbf{x}_0)\Delta_1 = 0$. Let $\mathbf{x}_1 = \mathbf{x}_0 + \alpha\Delta_1$, then we have

$$\|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_0)\|_2 = \|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_0) - \mathbf{J}(\mathbf{x}_0)(\mathbf{x}_1 - \mathbf{x}_0)\|_2 \leq M\|\mathbf{x}_1 - \mathbf{x}_0\|_2^2 = M\alpha^2.$$

□

Lemma 9.2. *Further assume that the minimum singular value of $\mathbf{J}(\mathbf{x})$ is at least σ for any $\mathbf{x} \in \mathcal{B}_2(\mathbf{x}_0, r)$. Let $\alpha = \min\{\frac{\sigma}{4M}, \frac{2r}{3}\}$. There exists an efficient algorithm that, given oracle access to ψ, finds $\mathbf{x}, \mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\|_2 \geq \frac{\alpha}{2}$ and $\|\psi(\mathbf{x}) - \psi(\mathbf{x}')\|_2$ is exponentially small.*

Proof. First find $\mathbf{x}_1 \in \mathcal{B}_2(\mathbf{x}_0, r)$ such that $\|\mathbf{x}_1 - \mathbf{x}_0\|_2 = \alpha$ and $\|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_0)\|_2 \leq M\alpha^2$ using Lemma 9.1. We can iteratively find points closer and closer to the above manifold.

Specifically, for $i = 1, 2, \ldots$, suppose $\mathbf{x}_i \in \mathcal{B}_2(\mathbf{x}_0, r)$, find the minimum solution Δ_{i+1} to the linear system

$$\mathbf{J}(\mathbf{x}_i)\Delta_{i+1} = \psi(\mathbf{x}_0) - \psi(\mathbf{x}_i). \tag{4}$$

We have $\|\Delta_{i+1}\|_2 \leq \frac{1}{\sigma}\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_0)\|_2$. Let $\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta_{i+1}$, we have that

$$\|\psi(\mathbf{x}_{i+1}) - \psi(\mathbf{x}_0)\|_2 = \|\psi(\mathbf{x}_{i+1}) - \psi(\mathbf{x}_i) - \mathbf{J}(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i)\|_2 \leq M\|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2^2 \leq \frac{M}{\sigma^2}\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_0)\|_2^2,$$

where the first equation holds by Equation 4, the first inequality holds by Taylor expansion of ψ at $\mathbf{x}_i \in \mathcal{B}_2(\mathbf{x}_0, r)$, and the second inequality holds since $\|\Delta_{i+1}\|_2 \leq \frac{1}{\sigma}\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_0)\|_2$.

Consider the sequence $\{l_i = \|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_0)\|_2\}$. We have $l_{i+1} \leq \frac{M}{\sigma^2} \cdot l_i^2$ and $l_1 \leq M\alpha^2$. By induction, we can show that $l_i \leq M\alpha^2$ for any i. This further gives

$$l_{i+1} \leq \frac{M}{\sigma^2} \cdot l_i^2 \leq \frac{M^2\alpha^2}{\sigma^2} \cdot l_i \leq \frac{l_i}{16}.$$

At the same time, we show that $\mathbf{x}_{i+1}$ stays inside $\mathcal{B}_2(\mathbf{x}_0, r)$ (which is necessary for the induction hypothesis.) Notice that

$$\|\mathbf{x}_{i+1} - \mathbf{x}_1\|_2 = \left\|\sum_{j=1}^{i} \Delta_{j+1}\right\|_2 \leq \sum_{j=1}^{i} \|\Delta_{j+1}\|_2 \leq \frac{1}{\sigma}\sum_{j=1}^{i} l_j < \frac{2l_1}{\sigma} \leq \frac{\alpha}{2}.$$

Thus, by the choice $\alpha \le \frac{2r}{3}$,

$$\|\mathbf{x}_{i+1} - \mathbf{x}_0\|_2 \le \|\mathbf{x}_1 - \mathbf{x}_0\|_2 + \|\mathbf{x}_{i+1} - \mathbf{x}_1\|_2 < \alpha + \frac{\alpha}{2} = \frac{3}{2}\alpha \le r\,.$$

In sum, $\|\psi(\mathbf{x}_i) - \psi(\mathbf{x}_0)\|_2$ converges exponentially to zero, while $\|\mathbf{x}_i - \mathbf{x}_0\|_2 \ge \|\mathbf{x}_1 - \mathbf{x}_0\|_2 - \|\mathbf{x}_i - \mathbf{x}_1\|_2 \ge \frac{\alpha}{2}$. Thus, after polynomially many iterations, we get $\mathbf{x} = \mathbf{x}_0, \mathbf{x}' = \mathbf{x}_i$ with $\|\mathbf{x} - \mathbf{x}'\|_2 \ge \frac{\alpha}{2}$ while $\|\psi(\mathbf{x}) - \psi(\mathbf{x}')\|_2$ is exponentially small. □

Remark 9.3 (Handling computation errors)**.** *For simplicity, we assume that the evaluation of ψ is exact. Our attack extends when ψ is evaluable up to a small exponential error $2^{-\mathsf{poly}(\lambda)}$ so that we can preserve the geometric decay of l_i in Lemma 9.2.*

Corollary 9.4. *If Π_{rvc} satisfies all the assumptions above, and $\frac{1}{r}, M, \frac{1}{\sigma}$ are all polynomially bounded, then Π_{rvc} cannot be binding. In particular, let $\alpha = \min\{\frac{\sigma}{4M}, \frac{2r}{3}\}$, we can find a collision for all precision lengths $\ell = \ell(\lambda)$ such that $2^{-\ell} \le \frac{\alpha}{2\sqrt{m}}$.*

9.2 Additively Homomorphic Functions

Suppose $\varphi : \mathbb{R}^m \to \mathbb{R}^n$ satisfies the additively homomorphic property, i.e., $\varphi(\mathbf{x}+\mathbf{y}) = \varphi(\mathbf{x}) + \varphi(\mathbf{y})$ in a neighborhood of 0. Existing results about Cauchy's functional equation give that φ is linear in this neighborhood.

Proposition 9.5. *If $\varphi : \mathcal{B}_2(0, r) \to \mathbb{R}^n$ is additively homomorphic and continuous, then there exists a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ such that $\varphi(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ for any $\mathbf{x} \in \mathcal{B}_2(0, r)$.*

Proof sketch. Assume $r \in \mathbb{Q}$. Notice that $\varphi(\mathbf{x}+\mathbf{y}) = \varphi(\mathbf{x}) + \varphi(\mathbf{y})$ implies that for any $a, b \in \mathbb{Z}$, $\varphi(a\mathbf{x}) = a\varphi(\mathbf{x})$ and $\varphi(\frac{1}{b}\mathbf{x}) = \frac{1}{b}\varphi(\mathbf{x})$, which means that

$$\forall c \in \mathbb{Q}, \varphi(c\mathbf{x}) = c\varphi(\mathbf{x}).$$

Let $\{\mathbf{e}_1, \ldots, \mathbf{e}_m\}$ be the standard basis of $\mathbb{R}^m$. Let

$$\mathbf{A} = \begin{pmatrix} \frac{2}{r}\varphi(\frac{r}{2}\mathbf{e}_1) & \frac{2}{r}\varphi(\frac{r}{2}\mathbf{e}_2) & \cdots & \frac{2}{r}\varphi(\frac{r}{2}\mathbf{e}_m) \end{pmatrix},$$

then for any $\mathbf{x} \in \mathcal{B}_2(0, r) \cap \mathbb{Q}^m$, we have $\varphi(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$. Since φ is continuous and $\mathbb{Q}^m$ is dense in $\mathbb{R}^m$, we have $\varphi(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x}$ for any $\mathbf{x} \in \mathcal{B}_2(0, r)$. □

Corollary 9.6. *If $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell; \mathbf{r})$, where $\mathbf{r}$ is the randomness, is compressing, additively homomorphic and continuous in a neighborhood of 0, w.r.t. $\mathbf{x}$ or w.r.t. $(\mathbf{x}, \mathbf{r})$, then Π_{rvc} cannot be binding.*

Proof. By Proposition 9.5, we have

$$\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell; \mathbf{r}) = \mathbf{A}_{\mathsf{pp},\ell,\mathbf{r}} \cdot \mathbf{x} \text{ or } \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell; \mathbf{r}) = \mathbf{A}_{\mathsf{pp},\ell} \cdot (\mathbf{x}; \mathbf{r})$$

in a neighborhood of 0. In both cases, Commit is affine in $\mathbf{x}$.

Let $\psi(\mathbf{x}) = \mathsf{cm}$, $(\mathsf{cm}, \pi_o) = \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell)$ for some fixed randomness. Then ψ is second-order continuously differentiable and the second-order derivative of ψ is zero ($M = 0$). By Lemma 9.1, we can find $\mathbf{x}_0 = 0, \mathbf{x}_1$ such that $\|\mathbf{x}_1 - \mathbf{x}_0\|_2 = \alpha$ while $\|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_0)\|_2 \le M\alpha^2 = 0$. They yield a collision to the commitment scheme, contradicting the binding property. □

Remark 9.7. *Proposition 9.5 does not hold when the output of φ is a general Abelian group instead of a vector space, since there may exist multiple $\varphi(\frac{1}{b}\mathbf{x})$ satisfying $b\varphi(\frac{1}{b}\mathbf{x}) = \varphi(\mathbf{x})$.*

Nonetheless, if the output group is $\mathbb{T}^n$, we can consider a neighborhood of 0 such that $\|\varphi(\mathbf{x})\|_{\mathbb{T}} < \frac{1}{4}$. Under this condition, $2\varphi(\frac{1}{2}\mathbf{x}) = \varphi(\mathbf{x})$ gives a unique solution $\varphi(\frac{1}{2}\mathbf{x})$, so we can still derive the linearity of φ on $(\cup_{i=0}^{\infty}\mathbb{Z}/2^i)^m$. Since $(\cup_{i=0}^{\infty}\mathbb{Z}/2^i)^m$ is also dense in $\mathbb{R}^m$, we can extend the linearity to this neighborhood by continuity. This captures those commitment schemes with mod 1 *operation at the end.*

References

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

[APKP24] Kasra Abbaszadeh, Christodoulos Pappas, Jonathan Katz, and Dimitrios Papadopoulos. Zero-knowledge proofs of training for deep neural networks. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pages 4316–4330. ACM, 2024.

[BCHR22] Andrej Bogdanov, Miguel Cueto Noval, Charlotte Hoffmann, and Alon Rosen. Public-key encryption from homogeneous CLWE. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 565–592. Springer, 2022.

[BDG+26] Dor Bitan, Zachary DeStefano, Shafi Goldwasser, Yuval Ishai, Yael Tauman Kalai, and Justin Thaler. Sum-check protocol for approximate computations. In *Advances in Cryptology – EUROCRYPT 2026, Part VII*, volume 16547 of *Lecture Notes in Computer Science*, pages 209–238. Springer, 2026.

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *Computer Security – ESORICS 2015, Part I*, volume 9326 of *Lecture Notes in Computer Science*, pages 305–325. Springer, 2015. Definition 2.2 and Section 4.1.

[BRST21] Joan Bruna, Oded Regev, Min Jae Song, and Yi Tang. Continuous LWE. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 694–707. Association for Computing Machinery, 2021.

[BRV26] Andrej Bogdanov, Alon Rosen, and Neekon Vafa. Statistically undetectable backdoors in deep neural networks. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 8794–8823. PMLR, 2026.

[BRVV26] Andrej Bogdanov, Alon Rosen, Neekon Vafa, and Vinod Vaikuntanathan. Adaptive robustness of hypergrid johnson-lindenstrauss. In *STOC*, pages 210–221. ACM, 2026.

[BSCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society, 2014.

[BTFM+26] Zoë Ruha Bell, Anvith Thudi, Olive Franzese-McLaughlin, Nicolas Papernot, and Shafi Goldwasser. Efficient public verification of private ML via regularization. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, 2017.

[CWSK24] Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. ZKML: An optimizing system for ML inference in zero-knowledge proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pages 560–574. ACM, 2024.

[DCB25] Trisha Datta, Binyi Chen, and Dan Boneh. VerITAS: Verifying image transformations at scale. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 4606–4623. IEEE, 2025.

[DFPS23] Julien Devevey, Pouria Fallahpour, Alain Passelègue, and Damien Stehlé. A detailed analysis of fiat-shamir with aborts. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2023.

[FGZ+26] Alexander Frolov, Jianfeng Guo, Xinyi Zhao, Trisha Datta, Dan Boneh, and Ian Miers. zk-Cinema: Proving video provenance in zero knowledge. Cryptology ePrint Archive, Paper 2026/1598, 2026.

[GGJ+23] Sanjam Garg, Aarushi Goel, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Guru-Vamsi Policharla, and Mingyuan Wang. Experimenting with zero-knowledge proofs of training. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1880–1894. ACM, 2023.

[GJJZ22] Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Yinuo Zhang. Succinct zero knowledge for floating point computations. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1203–1216. ACM, 2022.

[GKVZ22] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 931–942. IEEE, 2022.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, NY, USA, 1987. ACM.

[GVV22] Aparna Gupte, Neekon Vafa, and Vinod Vaikuntanathan. Continuous LWE is as hard as LWE & applications to learning Gaussian mixtures. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2022.

[KHSS22] Daniel Kang, Tatsunori Hashimoto, Ion Stoica, and Yi Sun. Scaling up trustless DNN inference with zero-knowledge proofs. arXiv:2210.08674, 2022.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[MR18] Payman Mohassel and Peter Rindal. ABY3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 35–52. ACM, 2018.

[MZ17] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38. IEEE, 2017.

[PGL+26] Qi Pang, Radhika Garg, Ziling Liu, Hanshen Xiao, Virginia Smith, Wenting Zheng, and Xiao Wang. Noisette: Certifying differential privacy mechanisms efficiently. Cryptology ePrint Archive, Paper 2026/074, 2026.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology – CRYPTO '89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1990.

[TGM+25] Gefei Tan, Adrià Gascón, Sarah Meiklejohn, Mariana Raykova, Xiao Wang, and Ning Luo. Founding zero-knowledge proof of training on optimum vicinity. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, pages 1173–1187, New York, NY, USA, 2025. Association for Computing Machinery.

[VV25] Neekon Vafa and Vinod Vaikuntanathan. Symmetric perceptrons, number partitioning and lattices. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 2191–2202. ACM, 2025.

[YCBC24] Chhavi Yadav, Amrita Roy Chowdhury, Dan Boneh, and Kamalika Chaudhuri. FairProof: Confidential and certifiable fairness for neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 55682–55705. PMLR, 2024.

[ZDK+25] Tianyu Zhang, Shen Dong, O. Deniz Kose, Yanning Shen, and Yupeng Zhang. FairZK: A scalable system to prove machine learning fairness in zero-knowledge. In *Proceedings of the 2025 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2025.

A Proof of Theorem 6.11

We restate Theorem 6.11.

Theorem A.1. *Let $p \in \mathbb{N}^*$ satisfy $b < p < 1.2^{\frac{m_r}{n+1}}$, and define the rounding function by $\lfloor a \rceil_p = \lfloor pa \rceil / p$. Then for arbitrary $\mathbf{x}, \ell_P$, abbreviating $h_{\mathbf{x}} = h_{\ell_P}(\mathbf{x})$, we have*

$$\begin{aligned}\left\{h \leftarrow \mathsf{Gen}, \mathbf{A}_r \leftarrow \mathbb{T}^{n\times m_r}, \mathbf{r} \leftarrow [b]^{m_r} : \left(h, \mathbf{A}_r, \lfloor (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \rceil_p\right)\right\} \\ \approx_s \mathsf{Gen} \times U(\mathbb{T}^{n\times m_r}) \times U(\lfloor \mathbb{T}^n \rceil_p),\end{aligned}$$

where $\lfloor \mathbb{T}^n \rceil_p = \{\lfloor \mathbf{t} \rceil_p | \mathbf{t} \in \mathbb{T}^n\} = \mathbb{Z}_p^n / p$.

Proof. This is an analog of the proof of the Leftover Hash Lemma.
The TV distance between the above two distributions is

$$\begin{aligned}&\mathbb{E}_{h,\mathbf{A}_r}[d_{TV}(\{\mathbf{r} \leftarrow [b]^{m_r} : \lfloor (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \rceil_p\}, U(\mathbb{Z}_p^n / p))] \\ =&\frac{1}{2}\mathbb{E}_{h,\mathbf{A}_r}\left[\sum_{\mathbf{s}\in\mathbb{Z}_p^n}\left|\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r})\rfloor \bmod p = \mathbf{s}] - \frac{1}{p^n}\right|\right].\end{aligned}$$

By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\sum_{\mathbf{s}\in\mathbb{Z}_p^n}\left|\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r})\rfloor \bmod p = \mathbf{s}] - \frac{1}{p^n}\right| &\le \sqrt{p^n \sum_{\mathbf{s}}\left(\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r})\rfloor \bmod p = \mathbf{s}] - \frac{1}{p^n}\right)^2} \\ &= \sqrt{p^n \sum_{\mathbf{s}}\left(\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r})\rfloor \bmod p = \mathbf{s}]\right)^2 - 1}.\end{aligned}$$

Thus, the TV distance is bounded by

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}_{h,\mathbf{A}_r}\left[\sqrt{p^n\sum_{\mathbf{s}}\left(\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r})\rfloor \bmod p=\mathbf{s}]\right)^2-1}\right]\\
\le&\frac{1}{2}\sqrt{\mathbb{E}_{h,\mathbf{A}_r}\left[p^n\sum_{\mathbf{s}}\left(\Pr_{\mathbf{r}}[\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r})\rfloor \bmod p=\mathbf{s}]\right)^2-1\right]}\\
=&\frac{1}{2}\sqrt{p^n\Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r})\rfloor=\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r}')\rfloor \bmod p]-1},
\end{aligned}$$

where in the last line $\mathbf{r},\mathbf{r}'$ are independently sampled from $[b]^{m_r}$.

Below we bound the probability of the following event, called "collision":

$$\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r})\rfloor=\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r}')\rfloor \bmod p.$$

Divide the event into two cases:

$$\begin{aligned}
\Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\text{ collision }] &= \Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\text{ collision }\wedge\gcd(\mathbf{r}-\mathbf{r}')\neq 1]+\Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\text{ collision }\wedge\gcd(\mathbf{r}-\mathbf{r}')=1]\\
&\le \Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\gcd(\mathbf{r}-\mathbf{r}')\neq 1]+\Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\text{ collision }\mid\gcd(\mathbf{r}-\mathbf{r}')=1],
\end{aligned}$$

where $\gcd(\cdot)$ takes the greatest common divisor of all entries of a vector. The two parts can be bounded separately by the following two lemmas.

Lemma A.2. *For integers $b>1, m_r>0$, we have*

$$\Pr_{\mathbf{r},\mathbf{r}'\leftarrow[b]^{m_r}}[\gcd(\mathbf{r}-\mathbf{r}')\neq 1]<b\left(\frac{2}{3}\right)^{m_r}.$$

Proof. By union bound,

$$\begin{aligned}
\Pr_{\mathbf{r},\mathbf{r}'\leftarrow[b]^{m_r}}[\gcd(\mathbf{r}-\mathbf{r}')\neq 1] &\le \sum_{d=2}^{b}\Pr_{\mathbf{r},\mathbf{r}'\leftarrow[b]^{m_r}}[d\mid\gcd(\mathbf{r}-\mathbf{r}')]\\
&\le \sum_{d=2}^{b}\left(\frac{\lceil\frac{b}{d}\rceil}{b}\right)^{m_r}\le\sum_{d=2}^{b}\left(\frac{2}{3}\right)^{m_r}<b\left(\frac{2}{3}\right)^{m_r}.
\end{aligned}$$

□

Lemma A.3. *For any fixed $h,\mathbf{r},\mathbf{r}'$ with $\gcd(\mathbf{r}-\mathbf{r}')=1$, we have*

$$\Pr_{\mathbf{A}_r\leftarrow\mathbb{T}^{n\times m_r}}[\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r})\rfloor=\lfloor p(h_{\mathbf{x}}+\mathbf{A}_r\cdot\mathbf{r}')\rfloor \bmod p]=\frac{1}{p^n}.$$

Proof. Notice that

$$\begin{aligned}
&\lfloor ph_{\mathbf{x}}+p\mathbf{A}_r\cdot\mathbf{r}\rfloor=\lfloor ph_{\mathbf{x}}+p\mathbf{A}_r\cdot\mathbf{r}'\rfloor \bmod p && (5)\\
\iff&\lfloor ph_{\mathbf{x}}+(p\mathbf{A}_r \bmod 1)\mathbf{r}\rfloor+\lfloor p\mathbf{A}_r\rfloor\mathbf{r}=\lfloor ph_{\mathbf{x}}+(p\mathbf{A}_r \bmod 1)\mathbf{r}'\rfloor+\lfloor p\mathbf{A}_r\rfloor\mathbf{r}' \bmod p && (6)\\
\iff&\lfloor p\mathbf{A}_r\rfloor(\mathbf{r}-\mathbf{r}')=\lfloor ph_{\mathbf{x}}+(p\mathbf{A}_r \bmod 1)\mathbf{r}'\rfloor-\lfloor ph_{\mathbf{x}}+(p\mathbf{A}_r \bmod 1)\mathbf{r}\rfloor \bmod p. && (7)
\end{aligned}$$

When $\mathbf{A}_r$ is uniformly sampled from $\mathbb{T}^{n\times m_r}$, $(p\mathbf{A}_r \bmod 1, \lfloor p\mathbf{A}_r \rfloor)$ is uniformly distributed over $\mathbb{T}^{n\times m_r} \times \mathbb{Z}_p^{n\times m_r}$. Thus, conditioned on fixed $(p\mathbf{A}_r \bmod 1)$, the RHS of (7) is fixed, while on the LHS, $\lfloor p\mathbf{A}_r \rfloor$ is uniformly distributed over $\mathbb{Z}_p^{n\times m_r}$.

Since $\gcd(\mathbf{r}-\mathbf{r}') = 1$, there exists a vector $\mathbf{s} \in \mathbb{Z}^{m_r}$ such that $\gcd(\mathbf{s}) = 1, \langle \mathbf{s}, \mathbf{r}-\mathbf{r}' \rangle = 1$. Consider the bijection $\psi : \mathbb{Z}_p^{m_r} \to \mathbb{Z}_p^{m_r}$ defined by $\psi(\mathbf{z}) = \mathbf{z} + \mathbf{s}$. The graph of ψ consists of p^{m_r-1} cycles of length p, and for the p elements in each cycle, their inner products with $\mathbf{r}-\mathbf{r}'$ exactly give $\{0, 1, \ldots, p-1\}$. We conclude that the inner product between a uniformly random vector in $\mathbb{Z}_p^{m_r}$ and $\mathbf{r}-\mathbf{r}'$ is uniformly distributed over $\mathbb{Z}_p$.

Therefore, when $\lfloor p\mathbf{A}_r \rfloor$ is uniformly sampled from $\mathbb{Z}_p^{n\times m_r}$, with probability $\frac{1}{p^n}$, (7) holds. □

Together, the probability of collision is bounded by

$$\Pr_{h,\mathbf{A}_r,\mathbf{r},\mathbf{r}'}[\text{ collision }] \leq b\left(\frac{2}{3}\right)^{m_r} + \frac{1}{p^n}.$$

Thus, the TV distance is bounded by

$$\frac{1}{2}\sqrt{p^n\left(b\left(\frac{2}{3}\right)^{m_r} + \frac{1}{p^n}\right) - 1} = \frac{1}{2}\sqrt{p^n b\left(\frac{2}{3}\right)^{m_r}} < \frac{1}{2}\sqrt{\left(\frac{4}{5}\right)^{m_r}},$$

which is negligible. □

B Handling computation error when arguing the hiding property of the hash-based commitment scheme

Recall the modified Commit algorithm:

- $\mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P)$: On input $\mathsf{pp} = (h, \mathbf{A}_r)$, $\mathbf{x} \in \mathcal{M}^m$, ℓ_P, choose a rounding parameter $p > \frac{2}{\epsilon_P}$ where $\epsilon_P = \frac{1}{4}q(\ell_P)$. First compute $h_{\ell_P}(\mathbf{x}) = \mathsf{Hash}(h, \mathbf{x}, \ell_P)$. Then uniformly sample a vector $\mathbf{r} \leftarrow \{0, 1, \ldots, b-1\}^{m_r}$ and compute

  $$\mathbf{c}_P =_{\epsilon'_P} (h_{\ell_P}(\mathbf{x}) + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \in \mathbb{T}^n,$$

  where the computation error is bounded by $\epsilon'_P = \frac{1}{p} \cdot 2^{-\omega(\log\lambda)}$. Output $(\mathsf{cm} = \lfloor \mathbf{c}_P \rfloor_p, \pi_o = \mathbf{r})$.

Consider the set

$$E = \{\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{T}^n : \exists 1 \leq i \leq n, \|pv_i \bmod 1\|_{\mathbb{T}} \leq p\epsilon'_P\} \subset \mathbb{T}^n.$$

Notice that for $\mathbf{r} \in \mathbb{Z}^{m_r} \backslash \{0\}$, when $\mathbf{A}_r \leftarrow U(\mathbb{T}^{n\times m_r})$, $\mathbf{A}_r \cdot \mathbf{r}$ is uniformly distributed over $\mathbb{T}^n$. Thus,

$$\Pr_{h,\mathbf{A}_r,\mathbf{r}}[(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \in E] \leq \Pr_{h,\mathbf{A}_r,\mathbf{r}}[\mathbf{r} = 0] + \Pr_{h,\mathbf{A}_r,\mathbf{r}}[(h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \in E | \mathbf{r} \neq 0] = \frac{1}{b^{m_r}} + \frac{Vol(E)}{Vol(\mathbb{T}^n)},$$

which is negligible since

$$\frac{Vol(E)}{Vol(\mathbb{T}^n)} = 1 - (1 - 2p\epsilon'_P)^n \leq 2np\epsilon'_P = 2n \cdot 2^{-\omega(\log\lambda)}.$$

Observe that when $\mathbf{v} \notin E$, any $\mathbf{u}$ satisfying $\|\mathbf{u}-\mathbf{v}\|_{\mathbb{T}} \le \epsilon'_P$ must have $\lfloor \mathbf{u} \rfloor_p = \lfloor \mathbf{v} \rfloor_p$. Therefore,

$$\begin{aligned}
&\{\mathsf{pp} = (h, \mathbf{A}_r) \leftarrow \mathsf{Setup}(1^\lambda, 1^m), \mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell_P) : (\mathsf{pp}, \mathsf{cm})\} \\
=&\{h \leftarrow \mathsf{Gen}, \mathbf{A}_r \leftarrow \mathbb{T}^{n \times m_r}, \mathbf{r} \leftarrow [b]^{m_r}, \mathbf{c}_P =_{\epsilon'_P} (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 : \left(h, \mathbf{A}_r, \lfloor \mathbf{c}_P \rfloor_p\right)\} \\
\approx_s&\{h, \mathbf{A}_r, \mathbf{r}, \mathbf{c}_P =_{\epsilon'_P} (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 : \left(h, \mathbf{A}_r, \lfloor \mathbf{c}_P \rfloor_p\right) \mid (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \notin E\} \\
=&\{h, \mathbf{A}_r, \mathbf{r} : \left(h, \mathbf{A}_r, \lfloor (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \rfloor_p\right) \mid (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \notin E\} \\
\approx_s&\{h, \mathbf{A}_r, \mathbf{r} : \left(h, \mathbf{A}_r, \lfloor (h_{\mathbf{x}} + \mathbf{A}_r \cdot \mathbf{r}) \bmod 1 \rfloor_p\right)\} \\
\approx_s&\mathsf{Gen} \times U(\mathbb{T}^{n \times m_r}) \times U(\lfloor \mathbb{T}^n \rfloor_p).
\end{aligned}$$

C Handling computation error when arguing the hiding property of the CLWE-based commitment scheme

Originally, the Commit algorithm computes

$$\mathbf{c}_P =_{\epsilon_P} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1,$$

where the computation error is bounded by $\epsilon_P = \frac{1}{4}\beta\sqrt{n}$. If the error is unrestricted, the output of Commit may concentrate on a fine discrete grid in $\mathbb{T}^k$, making it potentially distinguishable from a uniform distribution over $\mathbb{T}^k$. To address this issue, we modify the Commit algorithm to include a rounding step after the computation.

Specifically, the modified Commit algorithm selects a rounding parameter $p \in \mathbb{N}^*$, computes

$$\mathbf{c}_P =_{\epsilon'_P} \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1,$$

and then outputs

$$\mathsf{cm} = \lfloor \mathbf{c}_P \rfloor_p = \lfloor p \cdot \mathbf{c}_P \rfloor / p.$$

p satisfies $\frac{1}{p} < \frac{\epsilon_P}{2}$, and ϵ'_P is set to be $\frac{1}{p} \cdot 2^{-\omega(\log n)}$. Also, p is chosen to be a power of the base for the representation of real numbers, so that the rounding operation is equivalent to truncating the trailing digits in the representation and hence does not introduce additional computation error.

Functionality. Since

$$\left\| \mathsf{cm} - \left(\gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1 \right) \right\|_{\mathbb{T}} \le \|\mathsf{cm} - \mathbf{c}_P\| + \epsilon'_P \le \frac{1}{p} + \frac{1}{p} < \epsilon_P,$$

the modified Commit algorithm can be viewed as a special implementation of the original Commit algorithm, and hence the correctness and statistical binding properties of the commitment scheme are preserved.

Computationally hiding. Now the simulator $\mathcal{S}$ samples $\mathbf{u} \in (\mathbb{Z}_p/p)^k$ uniformly at random and outputs $\mathbf{u}$ as the commitment. We show that under the same assumption as Lemma 7.3,

$$\left\{\mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma) \leftarrow \mathsf{Setup}(1^\lambda, 1^m), \mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell) : (\mathbf{A}, \mathsf{cm})\right\} \approx_c D_{\mathbb{R},1}^{(n+m)\times k} \times U((\mathbb{Z}_p/p)^k).$$

Proof. Fix an arbitrary $\mathbf{x}$. Denote $C(\mathbf{A}, \mathbf{w}, \mathbf{e}) = \gamma \mathbf{A}^T \begin{pmatrix} \mathbf{x} \\ \mathbf{w} \end{pmatrix} + \mathbf{e} \bmod 1$. Lemma 7.3 implies that

$$\left\{ \mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow \tilde{D}_{\mathbb{R},\sigma}^{k} : (\mathbf{A}, C(\mathbf{A}, \mathbf{w}, \mathbf{e})) \right\} \approx_c D_{\mathbb{R},1}^{(n+m)\times k} \times U(\mathbb{T}^k). \tag{8}$$

Consider the set

$$E = \{ \mathbf{v} = (v_1, \ldots, v_k) \in \mathbb{T}^k : \exists 1 \le i \le k, \ \| p v_i \bmod 1 \|_{\mathbb{T}} \le p\epsilon'_P \} \subset \mathbb{T}^k.$$

Since

$$\Pr_{\mathbf{u} \leftarrow U(\mathbb{T}^k)} [\mathbf{u} \in E] = \frac{Vol(E)}{Vol(\mathbb{T}^k)} = 1 - (1 - 2p\epsilon'_P)^k \le 2kp\epsilon'_P = 2k \cdot 2^{-\omega(\log k)}$$

is negligible, we have

$$\Pr_{\mathbf{A},\mathbf{w},\mathbf{e}} [C(\mathbf{A}, \mathbf{w}, \mathbf{e}) \in E]$$

is also negligible, otherwise the two distributions in Eq. (8) can be distinguished by checking whether the second component is in E.

Observe that when $\mathbf{v} \notin E$, any $\mathbf{u}$ satisfying $\|\mathbf{u} - \mathbf{v}\|_{\mathbb{T}} \le \epsilon'_P$ must have $\lfloor \mathbf{u} \rceil_p = \lfloor \mathbf{v} \rceil_p$. Therefore,

$$\begin{aligned}
&\left\{ \mathsf{pp} = (\mathbf{A}, \gamma, \beta, \sigma) \leftarrow \mathsf{Setup}(1^\lambda, 1^m), \mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, \mathbf{x}, \ell) : (\mathbf{A}, \mathsf{cm}) \right\} \\
=& \left\{ \mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}, \mathbf{w} \leftarrow U(\mathcal{S}^{n-1}), \mathbf{e} \leftarrow \tilde{D}_{\mathbb{R},\sigma}^{k}, \mathbf{c}_P =_{\epsilon'_P} C(\mathbf{A}, \mathbf{w}, \mathbf{e}) : \left(\mathbf{A}, \lfloor \mathbf{c}_P \rceil_p\right) \right\} \\
\approx_c& \left\{ \mathbf{A}, \mathbf{w}, \mathbf{e}, \ \mathbf{c}_P =_{\epsilon'_P} C(\mathbf{A}, \mathbf{w}, \mathbf{e}) : \left(\mathbf{A}, \lfloor \mathbf{c}_P \rceil_p\right) \mid C(\mathbf{A}, \mathbf{w}, \mathbf{e}) \notin E \right\} \\
=& \left\{ \mathbf{A}, \mathbf{w}, \mathbf{e} : \left(\mathbf{A}, \lfloor C(\mathbf{A}, \mathbf{w}, \mathbf{e}) \rceil_p\right) \mid C(\mathbf{A}, \mathbf{w}, \mathbf{e}) \notin E \right\} \\
\approx_c& \left\{ \mathbf{A}, \mathbf{w}, \mathbf{e} : \left(\mathbf{A}, \lfloor C(\mathbf{A}, \mathbf{w}, \mathbf{e}) \rceil_p\right) \right\} \\
\approx_c& \left\{ \mathbf{A} \leftarrow D_{\mathbb{R},1}^{(n+m)\times k}, \mathbf{u} \leftarrow U(\mathbb{T}^k) : \left(\mathbf{A}, \lfloor \mathbf{u} \rceil_p\right) \right\} = D_{\mathbb{R},1}^{(n+m)\times k} \times U((\mathbb{Z}_p/p)^k).
\end{aligned}$$

□