latticefold plus

Folding framework

1. Committed CCS to Linear CCCS
2. Decomposition
3. Fold: LCCCS x LCCCS to LCCCS
    - norm bound
    - row check and linear check

Improvement
1. Multi-foldings
2. Double commitment
3. Range proof

1. Notions

1.1 Cyclotomic rings

- $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$
- $R = \mathbb{Z}[X]/(X^d + 1)$ is the ring of integers of the $2d$-cyclotomic field, where d is a power of two.
- $R_q = \mathbb{Z}_q[X]/(X^d + 1)$
- $\|x\|_\infty = \mathsf{max}_{i \in n}(x_i)$ is the ℓ_∞ norm of a vector $\vec{x} \in \mathbb{R}^n$.
- f, an element over R or R_q
- $\mathbf{b}$, bold, a vector over R or R_q
- $\mathbf{b}[i], \mathbf{b}_i$, the i-th element of a vector $\mathbf{b}$.
- $\langle \mathbf{a}, \mathbf{b} \rangle$, inner product of two vectors $\mathbf{a}, \mathbf{b}$.

1.2 Relations

- $\mathcal{R}$, an NP relation
    - $\mathcal{R}_{acc}$, accumulation
    - $\mathcal{R}_{comp}$, linear CCS, evaluation at a random point.
    - $\mathcal{R}_{lin,B}$, a general linear relation with norm bound B.
    - $\mathcal{R}_{open} = \{(\mathbb{x}, \mathbb{w}) = (\mathsf{cm}_\mathbf{f}, \mathbf{f})\}$, opening a commitment: $\mathbf{f}$ is a valid opening of the commitment $\mathsf{cm}_\mathbf{f}$.
    - $\mathcal{R}_{dopen}$, opening of a double commitment.
    - $\mathcal{R}_{m,in}$
    - $\mathcal{R}_{m,out}$
    - $\mathcal{R}_{rg}$, range check, norm check

1.3 Operations

- sgn, the sign of an integer, $\mathsf{sgn}(a) \in \{-1, 0, 1\}$.
- exp, the operation that takes an integer exponent and maps it to a monomial element, $\mathsf{exp}(a) = X^a$.
- EXP, the "border-aware" monomial operator,

$$\mathsf{EXP}(a) := \begin{cases} \{\mathsf{exp}(a)\} & \text{if } a \neq 0 \\ \{0, 1, X^{d/2}\} & \text{if } a = 0 \end{cases}$$

- pow, the vector form of exp: it maps an integer vector to a vector over R_q (or, depending on context, to a matrix).
- com, computes the Ajtai commitment of the input vector (refer to 4.2.1.2)
- dcom, computes the double commitment of the input vector
- flat, flattens a matrix into a long vector by concatenating its column vectors
- split, performs base-d' decomposition on a matrix

2. Multi-Folding

Latticefold+ consists of three reductions of knowledge: commit, fold, and decompose, in the following steps.

Step 1. Commit

Reduce the CCS relation $\mathcal{R}_{comp}$ to a general linear relation $\mathcal{R}_{lin,B}$.

In this step, the (reduction-of-knowledge) prover outputs the witness $\mathbb{w}$ and the statement $\mathbb{x}$ of the relation $\mathcal{R}_{lin,B}$.

Latticefold+ converts the CCS instancs into linear committed CCS, which is called a "general linear relation" $\mathcal{R}_{lin,B}$.

This step is the same as Hypernova and Latticefold, except the commitment is improved to a double commitment.

Step 2. Fold

From $L \geq 2$ instances in $\mathcal{R}_{lin,B} \times \ldots \times \mathcal{R}_{lin,B}$ to $\mathcal{R}_{lin,B^2}$.

In this step, it folds the given L instances of $\mathcal{R}_{lin,B}$ algebraically. In latticefold, there are only 2 CCS instances.

Step 3. Decompose

From $\mathcal{R}_{lin,B^2}$ back to 2 $\mathcal{R}_{lin,B}$ instance, using the B-nary decomposition technique (as in latticefold).

$$\left.\begin{array}{l} R_{lin,B}(x,w) \\ R_{lin,B}(x,w) \\ \vdots \\ R_{lin,B}(x,w) \end{array}\right\} L \xrightarrow{\text{fold}} R_{lin,B^2}(\Sigma x, \Sigma w) \xrightarrow{\text{decompose}} \left\{\begin{array}{l} R_{lin,B} \\ R_{lin,B} \end{array}\right.$$

3. Tools

3.1 MLE over rings

For a function $f(\mathbf{b}) : \{0,1\}^k \to R$, or a vector $\mathbf{f} = [f(0), \ldots, f(2^k - 1)]$, the MLE function $\tilde{f}(\mathbf{x})$ is

$$\tilde{f}(\mathbf{x}) := \sum_{\mathbf{b}\in\{0,1\}} f(\mathbf{b}) \cdot eq(\mathbf{b}, \mathbf{x}),$$

where $eq(\mathbf{b}, \mathbf{x}) := \Pi_{i\in[k]}[(1 - \mathbf{b}_i)(1 - \mathbf{x}_i) + \mathbf{b}_i\mathbf{x}_i]$.

The evaluation of $\tilde{f}$ at the point $\mathbf{r}$ is $\tilde{f}(\mathbf{r}) = \sum_{\mathbf{b}\in\{0,1\}} f(\mathbf{b}) \cdot eq(\mathbf{b}, \mathbf{r})$.

We can “eliminate” $\mathbf{b}$ and express $eq(\mathbf{b}, \mathbf{r})$ using only $\mathbf{r}$.
By the definition of $eq(\mathbf{b}, \mathbf{r})$, when $\mathbf{b} \in \{0,1\}^k$, all values $eq(\mathbf{b}, \mathbf{r})$ form a length-2^k vector as follows:

$$\begin{aligned}
&\mathbf{b} = (0, \ldots, 0), \quad eq(\mathbf{b}, \mathbf{r}) = \Pi_{i\in[k]}(1 - \mathbf{r}_i) \\
&\mathbf{b} = (0, \ldots, 1), \quad eq(\mathbf{b}, \mathbf{r}) = \mathbf{r}_k \cdot \Pi_{i\in[k-1]}(1 - \mathbf{r}_i) \\
&\ldots \\
&\mathbf{b} = (1, \ldots, 1), \quad eq(\mathbf{b}, \mathbf{r}) = \Pi_{i\in[k]}\mathbf{r}_i.
\end{aligned}$$

By the definition of the tensor product, the vector

$$(\Pi_{i\in[k]}(1 - \mathbf{r}_i), \mathbf{r}_k \cdot \Pi_{i\in[k-1]}(1 - \mathbf{r}_i), \ldots, \Pi_{i\in[k]}\mathbf{r}_i) = \otimes_{i\in[k]}(1 - \mathbf{r}_i, \mathbf{r}_i),$$

denoted by $\mathsf{tensor}(\mathbf{r})$.

Therefore, we have $\tilde{f}(\mathbf{r}) = \langle \mathbf{f}, \mathsf{tensor}(\mathbf{r}) \rangle$.

3.2 Double Commitment

3.2.1 Commit to any (large-norm) vectors

3.2.1.1 Gadget Matrix

For an integer $n \geq 1$, define a “gadget matrix” used to compose base-b digits into a value:

$$\mathbf{G}_{b,n} = \mathbf{I}_n \otimes [1 \; b \; b^2 \; \cdots \; b^{k-1}] \in \mathcal{R}_q^{n \times nk}.$$

Correspondingly, we use $\mathbf{G}_{b,k}^{-1} : \mathcal{R}_q^n \to \mathcal{R}_q^{kn}$ to denote base-b decomposition.

Then, for an arbitrary (large) vector $\mathbf{m} \in R^m$, $\mathbf{G}_{b,k}^{-1}(\mathbf{m})$ means: decompose all coefficients of $\mathbf{m}$ into base b, and then pack (pad) them into a vector (over $\mathbb{Z}^{nkd}$). Hence $\|\mathbf{G}_{b,k}^{-1}(\mathbf{m})\| < b$.

More generally, the $\mathbf{G}^{-1}$ decomposition can also be applied to a matrix. To make the next step convenient, we concatenate the decomposed matrix's column vectors into a single long vector. In Latticefold+, this process is denoted by $\mathsf{split}(\mathbf{M})$.

3.2.1.2 Ajtai commitment

For a message vector $\mathbf{m} \in R^m$ with $\|\mathbf{m}\|_\infty \le b$, the Ajtai commitment works as follows:

- KeyGen: given the security parameter λ, generate a commitment key $\mathbf{A} \in R_q^{n \times m}$.
- Com: given the commitment key $\mathbf{A} \in R_q^{n \times m}$ and message $\mathbf{m} \in R^m$, compute the commitment $\mathsf{cm}_\mathbf{m} = \mathbf{Am}$.

Similarly, when the message is a matrix $\mathbf{M}$, we can still commit to it using the operation defined by Com. In Latticefold+, this is denoted by $\mathsf{com}(\mathbf{M}) = \mathbf{AM}$.

If $\|\mathbf{m}'\|_\infty > b$, we can apply $\mathbf{G}^{-1}$ decomposition and then compute the commitment to $\mathbf{m}'$ as

$$\mathsf{cm}_\mathbf{m} = \mathbf{A}\mathbf{G}^{-1}(\mathbf{m}').$$

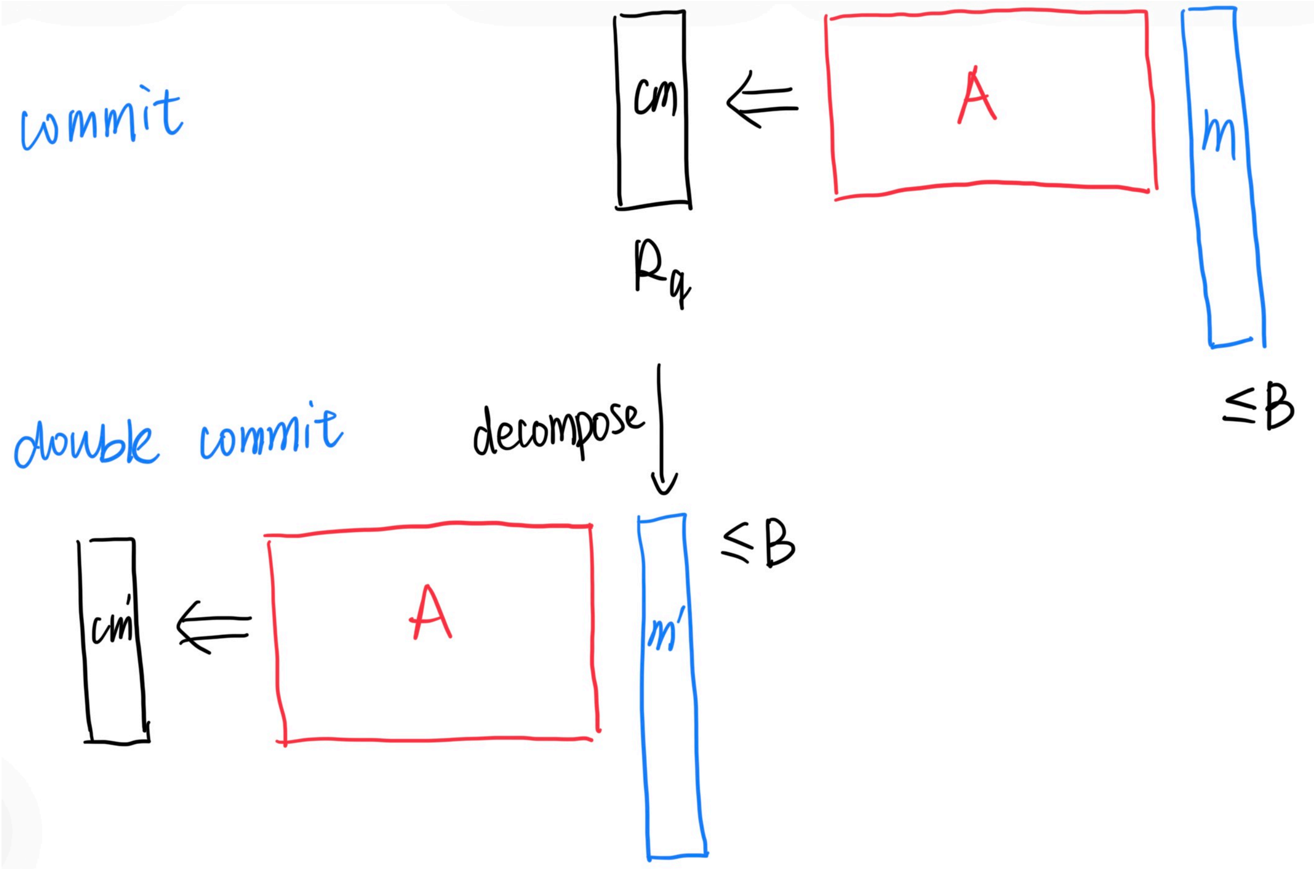
For a matrix $\mathbf{M}'$ with $\|\mathbf{M}'\|_\infty > b$, we can commit via $\mathsf{com}(\mathsf{split}(\mathbf{M}'))$.

3.2.2 Double Commitment

Like Greyhound and Labrador, Latticefold+ applies a double commitment to reduce communication costs.

In a word, a double commitment is a "commitment of a commitment." We use dcom (double commit) to denote this process. For a message matrix $\mathbf{M}$,

$$\mathsf{dcom} := \mathsf{com}(\mathsf{split}(\mathsf{com}(\mathbf{M}))).$$



This double commitment can reduce proof size but it also eliminate the additive homomorphism.

4.2.3 Prove the consistency of a double commitment

- Consistency: dcom and com are computed as in the above figure.

Folding

Formally, $\mathcal{R}_{lin,B}$ is defined below.

$$\mathcal{R}_{lin,B} = \left\{ (\mathbb{i}, \mathbb{x}, \mathbb{w}) \middle| \begin{array}{l} \mathbb{x} = (\mathsf{cm}_{\mathbf{f}}, \mathbf{r}, \mathbf{v}_1, \ldots, \mathbf{v}_{n_{lin}}), \\ \mathbb{w} = \mathbf{f}, \\ \mathsf{cm}_{\mathbf{f}} = \mathsf{Com}(\mathbf{f}), \\ \|\mathbf{f}\|_\infty < B, \\ \forall i \in [n_{lin}],\ \langle \mathbf{M}^{(i)}\mathbf{f}, \mathsf{tensor}(\mathbf{r}) \rangle = \mathbf{v}_i \end{array} \right\}, \tag{2}$$

where $\langle \mathbf{M}^{(i)}\mathbf{f}, \mathsf{tensor}(\mathbf{r}) \rangle = \mathbf{v}_i$ means that $\mathbf{M}^{(i)}\mathbf{f}$ evaluates to $\mathbf{v}_i$ at the point $\mathbf{r}$, and $\mathbb{i}$ is an index that includes the public parameters of the relation.