THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Department of Computing
電子計算學系

LaBRADOR: Compact Proofs for R1CS from Module-SIS

GAO Shang

2025/07/10

Notion

- $\mathbb{Z}_q$: ring of integers mod q.
  - $\vec{a} \in \mathbb{Z}_q^m$, where the i-th element is $a_i \in \mathbb{Z}_q$.
- $\mathcal{R}_q$: polynomial ring $\mathbb{Z}_q[X]/(X^d + 1)$.
  - $\boldsymbol{a} = a_0 + a_1 \cdot X + \cdots + a_{d-1} \cdot X^{d-1} \in \mathcal{R}_q$, and $ct(\boldsymbol{a}) = a_0$ is the constant term of $\boldsymbol{a}$.
  - $\vec{\boldsymbol{a}} \in \mathcal{R}_q^m$, where the i-th element is $\boldsymbol{a}_i \in \mathcal{R}_q$.
  - $\sigma(\boldsymbol{a}) = a_0 + a_1 \cdot X^{-1} + \cdots + a_{d-1} \cdot X^{-(d-1)}$.
- $\langle *,* \rangle$: inner-product, works on $\mathbb{Z}_q$ $\langle \vec{a}, \vec{b} \rangle$ and $\mathcal{R}_q$ $\langle \vec{\boldsymbol{a}}, \vec{\boldsymbol{b}} \rangle$.
  - Let $\vec{a}, \vec{b} \in \mathbb{Z}_q^{md}$, we can also write $\vec{\boldsymbol{a}}, \vec{\boldsymbol{b}} \in \mathcal{R}_q^m$. Then $\langle \vec{a}, \vec{b} \rangle = ct\left(\langle \sigma(\vec{\boldsymbol{a}}), \vec{\boldsymbol{b}} \rangle\right)$.

LaBRADOR Relation

$$\mathfrak{R} = \left\{ (\mathcal{F}, \mathcal{F}', \beta); (\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) : \begin{array}{c} f(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) = \boldsymbol{0} \quad \forall f \in \mathcal{F} \\ ct\big(f'(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r)\big) = 0 \quad \forall f' \in \mathcal{F}' \\ \sum_{i=1}^{r} \|\vec{\boldsymbol{s}}_i\|_2^2 \leq \beta^2 \end{array} \right\},$$

- where $f(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r)$ is defined as:

$$f(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) = \sum_{i,j=1}^{r} \boldsymbol{a}_{i,j} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle + \sum_{i=1}^{r} \langle \vec{\boldsymbol{\phi}}_i, \vec{\boldsymbol{s}}_j \rangle - \boldsymbol{b},$$

- so does f'.
- f' ($\mathbb{Z}_q$-constraint form) can be extended to $\mathcal{R}_q$ ($\mathcal{R}_q$-constraint form).

LaBRADOR Overview

- Committing $\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r$.
- Proving $\sum_{i=1}^{r} \|\vec{\boldsymbol{s}}_i\|_2^2 \leq \beta^2$.
- Aggregating the results.
- Amortizing for better efficiency.
- Verifying.

Committing

- Committing $\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r$ is to build a binding relation ($\mathfrak{R}$ may not be binding).
- Naively: Ajtai for each $\vec{\boldsymbol{s}}_i$: $\vec{\boldsymbol{t}}_i = \boldsymbol{A}\vec{\boldsymbol{s}}_i \in \mathcal{R}_q^{\kappa}$.

- LaBRADOR: commitment of commitments.

Projecting

- Proving $\sum_{i=1}^{r} \|\vec{\boldsymbol{s}}_i\|_2^2 \leq \beta^2$ is the most challenging part in lattice-based proofs.

- Modular Johnson-Lindenstrauss Lemma: if $\|\Pi\vec{s}\|_2$ is small, then $\|\vec{s}\|_2$ is small.

- For $\Pr[\mathrm{C} = 0] = 1/2, \Pr[\mathrm{C} = 1] = \Pr[\mathrm{C} = -1] = 1/4$, if $\|\vec{s}\|_2 \geq b$, then

$$\Pr_{\Pi \leftarrow \mathrm{C}^{256 \times d}}\left[\|\Pi\vec{s}\|_2 < \sqrt{30}b\right] \leq 2^{-128}.$$

Projecting

- Let $\vec{p} = \sum_i \Pi_i \vec{s}_i \in \mathbb{Z}_q^{256}$ and $\vec{\pi}_i^{(j)}$ be the j-th row of Π_i. We have
$$\sum_i \left\langle \vec{\pi}_i^{(j)}, \vec{s}_i \right\rangle = p_j \quad \Longrightarrow \sum_i ct\left(\left\langle \sigma\left(\vec{\boldsymbol{\pi}}_i^{(j)}\right), \vec{\boldsymbol{s}}_i \right\rangle\right) - p_j = 0.$$
- They are in the $\mathbb{Z}_q$-constraint form.

Aggregating

- Aggregate $|\mathcal{F}'|$ functions $f'^{(\ell)} \in \mathcal{F}'$ and 256 derived projecting functions ($\mathbb{Z}_q$-constraints).

- Extent the $\mathbb{Z}_q$-constraints to $\mathcal{R}_q$-constraints.

- Aggregate $|\mathcal{F}|$ functions $f^{(k)} \in \mathcal{F}$ and extended functions.

Amortizing

- Now we only have one aggregated $\mathcal{R}_q$-constraint under the from

$$f(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) = \sum_{i,j=1}^{r} \boldsymbol{a}_{i,j} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle + \sum_{i=1}^{r} \langle \vec{\boldsymbol{\phi}}_i, \vec{\boldsymbol{s}}_j \rangle - \boldsymbol{b}.$$

Amortizing

- $\vec{\boldsymbol{z}} = \boldsymbol{c}_1 \vec{\boldsymbol{s}}_1 + \cdots + \boldsymbol{c}_r \vec{\boldsymbol{s}}_r$.
- For commitments $\vec{\boldsymbol{t}}_i = \boldsymbol{A}\vec{\boldsymbol{s}}_i$:

- For $\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle$, verifier computes $\langle \vec{\boldsymbol{z}}, \vec{\boldsymbol{z}} \rangle$:

- For$\langle \overrightarrow{\boldsymbol{\phi}}_i, \vec{\boldsymbol{s}}_j \rangle$, verifier computes $\langle \overrightarrow{\boldsymbol{\phi}}_i, \vec{\boldsymbol{z}} \rangle$:

Amortizing

- $\vec{\boldsymbol{g}}_{i,j}, \vec{\boldsymbol{h}}_{i,j}$ are short, but we can still decompose.
- But $\vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}$ are long, so the prover sends commitments of them.

Verifying

- Prover sends $\vec{\boldsymbol{t}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{z}}$.

- Verifier checks:
  - Commitment constraint:
  - $\vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}$ are correct:
  - Aggregated $\mathcal{R}_q$-constraint:
  - $\vec{\boldsymbol{t}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{z}}$ are short:

Recursion

- Prover does not send $\vec{\boldsymbol{t}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{z}}$ (regard them as a witness).
- We now have another $\mathfrak{R}$ relation, allowing us to conduct a recursion.
- Decomposing $\vec{\boldsymbol{z}}$ to avoid blowing up. No need to decompose $\vec{\boldsymbol{t}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}$.

Thanks!

Q&A