

Notes on Spartan

arnaucube

April 2023

Abstract

Notes taken while reading about Spartan [1].

Usually while reading papers I take handwritten notes, this document contains some of them re-written to *LaTeX*.

The notes are not complete, don't include all the steps neither all the proofs.

Contents

1 R1CS into Sum-Check protocol	1
2 NIZKs with succinct proofs for R1CS	3
2.1 Full protocol	5

1 R1CS into Sum-Check protocol

Def 1.1. R1CS $\exists w \in \mathbb{F}^{m-|io|-1}$ such that $(A \cdot z) \circ (B \cdot z) = (C \cdot z)$, where $z = (io, 1, w)$.

Thm 4.1 \forall R1CS instance $x = (\mathbb{F}, A, B, C, io, m, n)$, \exists a degree-3 log m-variate polynomial G such that $\sum_{x \in \{0,1\}^{\log m}} G(x) = 0$ iff \exists a witness w such that $Sat_{R1CS}(x, w) = 1$.

We can view matrices $A, B, C \in \mathbb{F}^{m \times m}$ as functions $\{0,1\}^s \times \{0,1\}^s \rightarrow \mathbb{F}$ ($s = \lceil \log m \rceil$). For a given witness w to x , let $z = (io, 1, w)$. View z as a function $\{0,1\}^s \rightarrow \mathbb{F}$, so any entry in z can be accessed with a s -bit identifier.

$$F_{io}(x) = \left(\sum_{y \in \{0,1\}^s} A(x, y) \cdot Z(y) \right) \cdot \left(\sum_{y \in \{0,1\}^s} B(x, y) \cdot Z(y) \right) - \sum_{y \in \{0,1\}^s} C(x, y) \cdot Z(y)$$

Lemma 4.1. $\forall x \in \{0,1\}^s$, $F_{io}(x) = 0$ iff $Sat_{R1CS}(x, w) = 1$.

$F_{io}(\cdot)$ is a function, not a polynomial, so it can not be used in the Sum-check protocol.

$F_{io}(x)$ function is converted to a polynomial by using its polynomial extension $\tilde{F}_{io}(x) : \mathbb{F}^s \rightarrow \mathbb{F}$,

$$\tilde{F}_{io}(x) = \left(\sum_{y \in \{0,1\}^s} \tilde{A}(x,y) \cdot \tilde{Z}(y) \right) \cdot \left(\sum_{y \in \{0,1\}^s} \tilde{B}(x,y) \cdot \tilde{Z}(y) \right) - \sum_{y \in \{0,1\}^s} \tilde{C}(x,y) \cdot \tilde{Z}(y)$$

Lemma 4.2. $\forall x \in \{0,1\}^s$, $\tilde{F}_{io}(x) = 0$ iff $Sat_{R1CS}(x, w) = 1$.

(proof: $\forall x \in \{0,1\}^s$, $\tilde{F}_{io}(x) = F_{io}(x)$, so, result follows from Lemma 4.1.)

$\tilde{F}_{io}(\cdot)$: low-degree multivariate polynomial over \mathbb{F} in s variables. Verifier can check if $\sum_{x \in \{0,1\}^s} \tilde{F}_{io}(x) = 0$ using the Sum-check protocol.

But: $\sum_{x \in \{0,1\}^s} \tilde{F}_{io}(x) = 0 \not\iff F_{io}(x) = 0 \forall x \in \{0,1\}^s$. Bcs: the 2^s terms in the sum might cancel each other even when the individual terms are not zero.

Solution: combine $\tilde{F}_{io}(x)$ with $\tilde{e}q(t, x)$ to get $Q_{io}(t, x)$ as a zero-polynomial

$$Q_{io}(t) = \sum_{x \in \{0,1\}^s} \tilde{F}_{io}(x) \cdot \tilde{e}q(t, x)$$

where $\tilde{e}q(t, x) = \prod_{i=1}^s (t_i \cdot x_i + (1 - t_i) \cdot (1 - x_i))$, which is the MLE of e given x , e is the witness, $e = 1$ if $x = e$, 0 otherwise.

Basically $Q_{io}(\cdot)$ is a multivariate polynomial such that

$$Q_{io}(t) = \tilde{F}_{io}(t) \quad \forall t \in \{0,1\}^s$$

thus, $Q_{io}(\cdot)$ is a zero-polynomial iff $\tilde{F}_{io}(x) = 0 \forall x \in \{0,1\}^s$. \iff iff $\tilde{F}_{io}(\cdot)$ encodes a witness w such that $Sat_{R1CS}(x, w) = 1$.

To check that $Q_{io}(\cdot)$ is a zero-polynomial: check $Q_{io}(\tau) = 0$, $\tau \in^R \mathbb{F}^s$ (Schwartz-Zippel-DeMillo-Lipton lemma).

Recap

We have that $Sat_{R1CS}(x, w) = 1$ iff $F_{io}(x) = 0$.

To be able to use sum-check, we use its polynomial extension $\tilde{F}_{io}(x)$, using sum-check to prove that $\tilde{F}_{io}(x) = 0 \forall x \in \{0,1\}^s$, which means that $Sat_{R1CS}(x, w) = 1$.

To prevent potential canceling terms, we combine $\tilde{F}_{io}(x)$ with $\tilde{e}q(t, x)$, obtaining $G_{io,\tau}(x) = \tilde{F}_{io}(x) \cdot \tilde{e}q(t, x)$.

Thus $Q_{io}(t) = \sum_{x \in \{0,1\}^s} \tilde{F}_{io}(x) \cdot \tilde{e}q(t, x)$, and then we prove that $Q_{io}(\tau) = 0$, for $\tau \in^R \mathbb{F}^s$.

2 NIZKs with succinct proofs for R1CS

From Thm 4.1: to check R1CS instance $(\mathbb{F}, A, B, C, io, m, n)$ V can check if $\sum_{x \in \{0,1\}^s} G_{io,\tau}(x) = 0$, which through sum-check protocol can be reduced to $e_x = G_{io,\tau}(r_x)$, where $r_x \in \mathbb{F}^s$.

Recall: $G_{io,\tau}(x) = \tilde{F}_{io}(x) \cdot \tilde{e}q(\tau, x)$.

Evaluating $\tilde{e}q(\tau, r_x)$ takes $O(\log m)$, but to evaluate $\tilde{F}_{io}(r_x)$, V needs to evaluate

$$\tilde{A}(r_x, y), \tilde{B}(r_x, y), \tilde{C}(r_x, y), \tilde{Z}(y), \quad \forall y \in \{0,1\}^s$$

But: evaluations of $\tilde{Z}(y) \forall y \in \{0,1\}^s \iff (io, 1, w)$.

Solution: combination of 3 protocols:

- Sum-check protocol
- randomized mini protocol
- polynomial commitment scheme

Observation: let $\tilde{F}_{io}(r_x) = \overline{A}(r_x) \cdot \overline{B}(r_x) - \overline{C}(r_x)$, where

$$\overline{A}(r_x) = \sum_{y \in \{0,1\}^s} \tilde{A}(r_x, y) \cdot \tilde{Z}(y)$$

$$\overline{B}(r_x) = \sum_{y \in \{0,1\}^s} \tilde{B}(r_x, y) \cdot \tilde{Z}(y)$$

$$\overline{C}(r_x) = \sum_{y \in \{0,1\}^s} \tilde{C}(r_x, y) \cdot \tilde{Z}(y)$$

Prover makes 3 separate claims: $\overline{A}(r_x) = v_A$, $\overline{B}(r_x) = v_B$, $\overline{C}(r_x) = v_C$, then V evaluates:

$$G_{io,\tau}(r_x) = (v_A \cdot v_B - v_C) \cdot \tilde{e}q(r_x, \tau)$$

which equals to

$$= (\overline{A}(r_x) \cdot \overline{B}(r_x) - \overline{C}(r_x)) \cdot \tilde{e}q(r_x, \tau) = \left(\left(\sum_{y \in \{0,1\}^s} \tilde{A}(r_x, y) \cdot \tilde{Z}(y) \right) \cdot \left(\sum_{y \in \{0,1\}^s} \tilde{B}(r_x, y) \cdot \tilde{Z}(y) \right) - \sum_{y \in \{0,1\}^s} \tilde{C}(r_x, y) \cdot \tilde{Z}(y) \right) \cdot \tilde{e}q(r_x, \tau)$$

This would be 3 sum-check protocol instances (3 claims: $\overline{A}(r_x) = v_A$, $\overline{B}(r_x) = v_B$, $\overline{C}(r_x) = v_C$).

Instead, combine 3 claims into a single claim:

- V samples $r_A, r_B, r_C \in^R \mathbb{F}$, and computes $c = r_A v_A + r_B v_B + r_C v_C$.

- V, P use sum-check protocol to check:

$$r_A \cdot \bar{A}(r_x) + r_B \cdot \bar{B}(r_x) + r_C \cdot \bar{C}(r_x) == c$$

Let

$$\begin{aligned} L(r_x) &= r_A \cdot \bar{A}(r_x) + r_B \cdot \bar{B}(r_x) + r_C \cdot \bar{C}(r_x) \\ &= \sum_{y \in \{0,1\}^s} \left(r_A \cdot \tilde{A}(r_x, y) \cdot \tilde{Z}(y) + r_B \cdot \tilde{B}(r_x, y) \cdot \tilde{Z}(y) + r_C \cdot \tilde{C}(r_x, y) \cdot \tilde{Z}(y) \right) \\ &= \sum_{y \in \{0,1\}^s} M_{r_x}(y) \end{aligned}$$

$M_{r_x}(y)$ is a s -variate polynomial with $\deg \leq 2$ in each variable ($\iff \mu = s, l = 2, T = c$).

$$\begin{aligned} M_{r_x}(r_y) &= r_A \cdot \tilde{A}(r_x, r_y) \cdot \tilde{Z}(r_y) + r_B \cdot \tilde{B}(r_x, r_y) \cdot \tilde{Z}(r_y) + r_C \cdot \tilde{C}(r_x, r_y) \cdot \tilde{Z}(r_y) \\ &= (r_A \cdot \tilde{A}(r_x, r_y) + r_B \cdot \tilde{B}(r_x, r_y) + r_C \cdot \tilde{C}(r_x, r_y)) \cdot \tilde{Z}(r_y) \end{aligned}$$

only one term in $M_{r_x}(r_y)$ depends on prover's witness: $\tilde{Z}(r_y)$, the other terms can be computed locally by V in $O(n)$ time (Section 6 of the paper for sub-linear in n).

Instead of evaluating $\tilde{Z}(r_y)$ in $O(|w|)$ communications, P sends a commitment to $\tilde{w}(\cdot)$ (= MLE of the witness w) to V before the first instance of the sum-check protocol.

Recap

To check the R1CS instance, V can check $\sum_{x \in \{0,1\}^s} G_{io,\tau}(x) = 0$, which through the sum-check is reduced to $e_x = G_{io,\tau}(r_x)$, for $r_x \in \mathbb{F}^s$.

Evaluating $G_{io,\tau}(x)$ ($G_{io,\tau}(x) = \tilde{F}_{io}(x) \cdot \tilde{e}q(\tau, x)$) is not cheap. Evaluating $\tilde{e}q(\tau, r_x)$ takes $O(\log m)$, but to evaluate $\tilde{F}_{io}(r_x)$, V needs to evaluate $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{Z}, \forall y \in \{0,1\}^s$

P makes 3 separate claims: $\bar{A}(r_x) = v_A, \bar{B}(r_x) = v_B, \bar{C}(r_x) = v_C$, so V can evaluate $G_{io,\tau}(r_x) = (v_A \cdot v_B - v_C) \cdot \tilde{e}q(r_x, \tau)$

The previous claims are combined into a single claim (random linear combination) to use only a single sum-check protocol:

$$\text{P: } c = r_A v_A + r_B v_B + r_C v_C, \text{ for } r_A, r_B, r_C \in^R \mathbb{F}$$

$$\text{V, P: sum-check } r_A \cdot \bar{A}(r_x) + r_B \cdot \bar{B}(r_x) + r_C \cdot \bar{C}(r_x) == c$$

$c = L(r_x) = \sum_{y \in \{0,1\}^s} M_{r_x}(y)$, where $M_{r_x}(y)$ is a s -variate polynomial with $\deg \leq 2$ in each variable ($\iff \mu = s, l = 2, T = c$). Only $\tilde{Z}(r_y)$ depends on P's witness, the other terms can be computed locally by V.

Instead of evaluating $\tilde{Z}(r_y)$ in $O(|w|)$ communications, P uses a commitment to $\tilde{w}(\cdot)$ (= MLE of the witness w).

2.1 Full protocol

(Recall: Sum-Check params: μ : n vars, n rounds, l : degree in each variable upper bound, T : claimed result.)

- $pp \leftarrow \text{Setup}(1^\lambda)$: invoke $pp \leftarrow PC.Setup(1^\lambda, \log m)$; output pp
- $b \leftarrow \langle P(w), V(r) \rangle (\mathbb{F}, A, B, C, io, m, n)$:
 1. P: $(C, S) \leftarrow PC.Commit(pp, \tilde{w})$ and send C to V
 2. V: send $\tau \in^R \mathbb{F}^{\log m}$ to P
 3. let $T_1 = 0, \mu_1 = \log m, l_1 = 3$
 4. V: set $r_x \in^R \mathbb{F}^{\mu_1}$
 5. Sum-check 1. $e_x \leftarrow \langle P_{SC}(G_{io, \tau}), V_{SC}(r_x) \rangle (\mu_1, l_1, T_1)$
 6. P: compute $v_A = \bar{A}(r_x), v_B = \bar{B}(r_x), v_C = \bar{C}(r_x)$, send (v_A, v_B, v_C) to V
 7. V: abort with $b = 0$ if $e_x \neq (v_A \cdot v_B - v_C) \cdot \tilde{e}q(r_x, \tau)$
 8. V: send $r_A, r_B, r_C \in^R \mathbb{F}$ to P
 9. let $T_2 = r_A \cdot v_A + r_B \cdot v_B + r_C \cdot v_C, \mu_2 = \log m, l_2 = 2$
 10. V: set $r_y \in^R \mathbb{F}^{\mu_2}$
 11. Sum-check 2. $e_y \leftarrow \langle P_{SC}(M_{r_x}), V_{SC}(r_y) \rangle (\mu_2, l_2, T_2)$
 12. P: $v \leftarrow \tilde{w}(r_y[1..])$, send v to V
 13. $b_e \leftarrow \langle P_{PC.Eval}(\tilde{w}, S), V_{PC.Eval}(r) \rangle (pp, C, r_y, v, \mu_2)$
 14. V: abort with $b = 0$ if $b_e = 0$
 15. V: $v_z \leftarrow (1 - r_y[0]) \cdot \tilde{w}(r_y[1..]) + r_y[0] \cdot \widetilde{(io, 1)}(r_y[1..])$
 16. V: $v_1 \leftarrow \tilde{A}(r_x, r_y), v_2 \leftarrow \tilde{B}(r_x, r_y), v_3 \leftarrow \tilde{C}(r_x, r_y)$
 17. V: abort with $b = 0$ if $e_y \neq (r_A v_1 + r_B v_2 + r_C v_3) \cdot v_z$
 18. V: output $b = 1$

Section 6 of the paper, describes how in step 16, instead of evaluating $\tilde{A}, \tilde{B}, \tilde{C}$ at r_x, r_y with $O(n)$ costs, P commits to $\tilde{A}, \tilde{B}, \tilde{C}$ and later provides proofs of openings.

WIP: covered until sec.6

References

- [1] Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. Cryptology ePrint Archive, Paper 2019/550, 2019. <https://eprint.iacr.org/2019/550>.