CS171: Cryptography

Lecture 22

Sanjam Garg

Plan for today

- Succinct Arguments.

Succinct Non-Interactive Argument System (SNARG)

structured reference string

π could be zero-knowledge or not

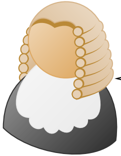
$\textcolor{red}{\mathsf{srs}}, x, w$

Prover

$C(x, w) = 1$

$$\xrightarrow[|\pi| \ll |C|]{\pi}$$

$\textcolor{red}{\mathsf{srs}}, x$



Verifier

Pre-processing depending only on C, $\textcolor{red}{\mathsf{srs}}$ and takes time $\mathrm{poly}(|C|, \textcolor{red}{\mathsf{srs}})$

Verification time is $\ll |C|$.

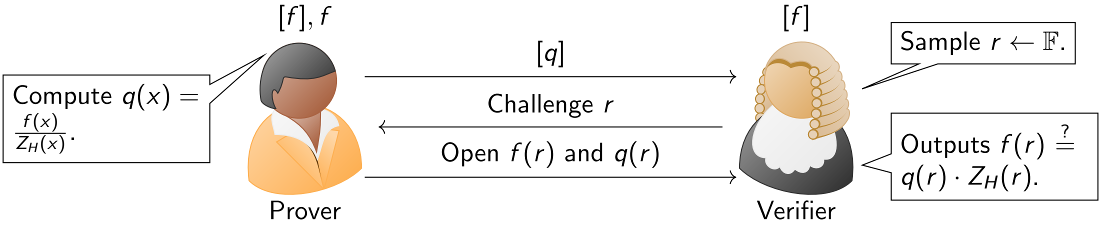
- Completeness: An honest prover should be able to convince an honest verifier with *overwhelming* probability.
- Soundness: A PPT cheating prover cannot generate an accepting proof for a false statement.
- Zero-Knowledge: The proof doesn't leak any information about the witness w.
  - Not all applications need zero knowledge, e.g. zk-rollups.

Polynomial Commitment

- P can commit to some polnomial f of some ia priori fixed maximum degree.
- We denote the commitment to f by $[f]$.
- The commiter can open $[f]$ at any input r and prove that the provided opening $f(r)$ is correct with respect to the previously provided commitment $[f]$.

Check the roots of a polynomial

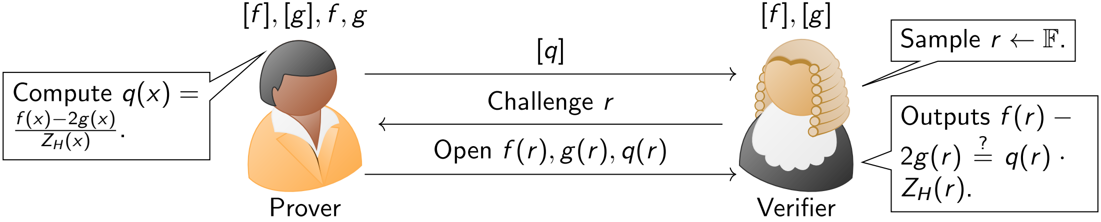
$$\forall x \in H, f(x) = 0$$



- $f(x) \neq 0$.
- $H = \{\omega, \omega^2 \dots \omega^n\}$.
- $Z_H(x) = \prod_{\alpha \in H}(x - \alpha) = x^n - 1$.
- $Z_H(r) = r^n - 1$ can be computed efficiently in $O(\log n)$ time using the repeated squaring algorithm.

Check relationship between two polynomials (Example)

$$\forall x \in H, f(x) = 2 \cdot g(x)$$



- $H = \{\omega, \omega^2 \dots \omega^n\}$.
- $Z_H(x) = \prod_{\alpha \in H}(x - \alpha) = x^n - 1$.
- $f(x)$ is a polynomial such that for each $\omega^i \in H$, $f(\omega^i) = 2 \cdot i$.
- $g(x)$ is a polynomial such that for each $\omega^i \in H$, $g(\omega^i) = i$.
- As polynomials $f(x) \neq 2g(x)$.
- $Z_H(r) = r^n - 1$ can be computed efficiently in $O(\log n)$ time using the repeated squaring algorithm.

Check two multisets A and B are the same

$$\{\forall x \in H, f(x)\} = \{\forall x \in H, g(x)\}$$

Set $z(x)$ such that $\forall i \in \{1 \dots n\}, z(\omega^i) = \prod_{j=1}^{i-1} \frac{a_j+\gamma}{b_j+\gamma}$.

$q(x) = \frac{z(x\omega)(g(x)+\gamma)-z(x)(f(x)+\gamma)}{Z_H(x)}$

$[f], [g], f, g$

Prover

Challenge γ ←

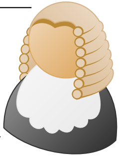
$[z], [q]$ →

Challenge r ←

Open $z(\omega), z(r), z(r\omega)$ →

Open $f(r), g(r), q(r)$

$[f], [g]$



Verifier

Sample $r \leftarrow \mathbb{F}$.

Outputs $z(\omega) \stackrel{?}{=} 1$ and $z(r\omega)(g(r)+\gamma) - z(r)(f(r)+\gamma) \stackrel{?}{=} q(r) \cdot Z_H(r)$.

- $A = \{a_1, \dots a_n\}$ and $B = \{b_1, \dots b_n\}$.
- $f(x)$ is a polynomial such that for each $\omega^i \in H$, $f(\omega^i) = a_i$.
- $g(x)$ is a polynomial such that for each $\omega^i \in H$, $g(\omega^i) = b_i$.
- As polynomials $f(x) \neq g(x)$.
- Note that $z(\omega) = z(\omega^{n+1}) = 1$. And we have that $\forall i \in 1, \dots n, z(\omega^{i+1}) = z(\omega^i) \cdot \frac{a_i+\gamma}{b_i+\gamma}$.
- Prove: (i) $z(\omega) = 1$, and (ii) $\forall x \in H$ we have that $z(x\omega)(g(x)+\gamma) - z(x)(f(x)+\gamma) = 0$.

Permutation Check: How to Prove — Warmup!

Permutation Check: How to check that $\sigma(a_1 ... a_n) = (b_1 ... b_n)$?

- $\sigma^{(1)} : \mathbb{F}^n \to \mathbb{F}^n$ as a function that permuates the input vector.
- $\sigma^{(2)} : \{1, \ldots n\} \to \{1, \ldots n\}$ as a function that maps input index to output index of the permuatation.
- $\sigma^{(3)} : H \to H$ as a polynomial that maps ω^i to $\omega^{\sigma^{(2)}(i)}$.
- For the identity permutation: $\sigma^{(3)}(x) = x$.
- Will refer to $\sigma^{(1)}$, $\sigma^{(2)}$, and $\sigma^{(3)}$ as just σ when clear from context.

Permutation Check: How to Prove — Warmup!

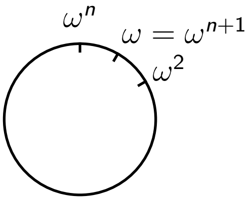
Permutation Check: How to check that $\sigma(a_1 ... a_n) = (b_1 ... b_n)$?
$f(x)$ and $g(x)$ are such that $\forall \omega^i \in H$, $f(\omega^i) = a_i$ and $g(\omega^i) = b_i$.
Pre-Processing Step: P and V generate $[\sigma]$ - commitment to σ in the pre-processing step.

How to test?

- For verifier chosen β, γ consider polynomial z (that Prover commits) such that for all $i \in \{1, ... n\}$

$$z(\omega^i) = \prod_{j=1}^{i-1} \frac{a_j + \beta \cdot \sigma(\omega^j) + \gamma}{b_j + \beta \cdot \omega^j + \gamma}$$

- Verifier checks that (i) $z(\omega) = 1$, and (ii) for all $x \in H$ we have that $\frac{z(x\omega)}{z(x)} = \frac{a_i + \beta \cdot \sigma(x) + \gamma}{b_i + \beta \cdot x + \gamma}$.
- **Note** that $z(\omega^{n+1}) = z(\omega) = 1$. (Only need to prove once!)
- Prove that $\forall x \in H$ we have that $z(x \cdot \omega) \cdot (g(x) + \beta \cdot x + \gamma)$ $- z(x) \cdot (f(x) + \beta \cdot \sigma(x) + \gamma) = 0$.



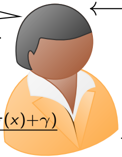
Prove that $\sigma(A) = B$

$$\sigma(f(\omega)\dots f(\omega^n)) = (g(\omega)\dots g(\omega^n))$$

Set $z(x)$ such that $\forall i \in \{1\dots n\}, z(\omega^i) = \prod_{j=1}^{i-1} \frac{a_j+\beta\sigma(\omega^j)+\gamma}{b_j+\beta\omega^j+\gamma}$.

$q(x) = \frac{z(x\omega)(g(x)+\beta x+\gamma)-z(x)(f(x)+\beta\sigma(x)+\gamma)}{Z_H(x)}$

$[f], [g], f, g, [\sigma]$



Prover

Challenge β, γ

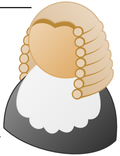
$[z], [q]$

Challenge r

Open $z(\omega), z(r), z(r\omega)$

Open $f(r), g(r), q(r), \sigma(r)$

$[f], [g], [\sigma]$



Verifier

Sample $r \leftarrow \mathbb{F}$.

Outputs $z(\omega) \stackrel{?}{=} 1$ and $z(r\omega)(g(r) + \beta r + \gamma) - z(r)(f(r) + \beta\sigma(r) + \gamma) \stackrel{?}{=} q(r)\cdot Z_H(r)$.

- Note that $z(\omega) = z(\omega^{n+1}) = 1$. And we have that $\forall i \in 1,\dots n, z(\omega^{i+1}) = z(\omega^i)\cdot \frac{a_i+\beta\sigma(\omega^i)+\gamma}{b_i+\beta\omega^i+\gamma}$.
- Prove: (i) $z(\omega) = 1$, and (ii) $\forall x \in H$ we have that $z(x\omega)(g(x) + \beta x + \gamma) - z(x)(f(x) + \beta\sigma(x) + \gamma) = 0$.

$\mathcal{P}$lon$\mathcal{K}$ Arithmetization

$\mathcal{C}(x, w) = 1$ be a circuit of n gates, where each gate is fan-in 2 and support $+$ and $\times$ gates.

Simplifying setting:

- Ignore zero-knowledge.
- Assume a polynomial commitment scheme (e.g. KZG) — use $[f]$ notation as commitment of f.
- Simplify construction — ignore optimizations.

For the i^{th} gate in the circuit we have a constraint on input wires values a_i, b_i and output wire value c_i specified by constants $q_{L,i}, q_{R,i}, q_{O,i}, q_{M,i}, q_{C,i}$:

$$q_{L,i} \cdot a_i + q_{R,i} \cdot b_i + q_{O,i} \cdot c_i + q_{M,i} \cdot a_i \cdot b_i + q_{C,i} = 0$$

$+$ gate then $q_{L,i}, q_{R,i} = 1, q_{O,i} = -1$ and $q_{M,i} = q_{C,i} = 0$.
$\times$ gate then $q_{L,i}, q_{R,i} = 0,\ q_{O,i} = -1$ and $q_{M,i} = 1,\ q_{C,i} = 0$.

$$q_L(x) = \sum_i q_{L,i} \cdot \delta_i(x),$$

where $\delta_i(x) = 1$ for $x = i$ and 0 for $x \in H \backslash \{i\}$.
Constraint Checks: Given $H \subset \mathbb{F}, |H| = n$, define degree-$n-1$ polynomial $q_L(x)$ such that for

Constraint Check: How to Prove!

Constraint Check: Prove that for all $x \in \{0, \ldots n-1\}$ we have:

$$q_L(x) \cdot a(x) + q_R(x) \cdot b(x) + q_O(x) \cdot c(x) + q_M(x) \cdot a(x) \cdot b(x) + (q_C(x) - PI(x)) = 0$$

How to test?

- The above is only true if there exists a quotient polynomial $T(x)$ such that
  $$q_L(x) \cdot a(x) + q_R(x) \cdot b(x) + q_O(x) \cdot c(x) + q_M(x) \cdot a(x) \cdot b(x) + (q_C(x) - PI(x)) = T(x) \cdot Z_H(x).$$
- It suffices for the verifier to check the following at a random point $\mathfrak{z} \in \mathbb{F}$:
  $$q_L(\mathfrak{z}) \cdot a(\mathfrak{z}) + q_R(\mathfrak{z}) \cdot b(\mathfrak{z}) + q_O(\mathfrak{z}) \cdot c(\mathfrak{z}) + q_M(\mathfrak{z}) \cdot a(\mathfrak{z}) \cdot b(\mathfrak{z}) + (q_C(\mathfrak{z}) - PI(\mathfrak{z})) = T(\mathfrak{z}) \cdot Z_H(\mathfrak{z}),$$
  where the verifier can compute $Z_H(\mathfrak{z})$ locally.
- Or, prover commits to $a(x), b(x), c(x)$ and $T(x)$ and opens $\overline{a} = a(\mathfrak{z}), \overline{b} = b(\mathfrak{z}), \overline{c} = c(\mathfrak{z})$ and $T(\mathfrak{z})$ for a verifier chosen $\mathfrak{z}$.
- Problem: But degree of $T(x)$ is large. So, commit to degree-$n-1$ T_{lo}, T_{mid}, T_{hi} such that $T_{lo}(x) + T_{mid}(x) \cdot x^n + T_{hi}(x) \cdot x^{2n} = T(x)$.
- It suffices to prove that the following polynomial is 0 for $x = \mathfrak{z}$:
  $$q_L(x) \cdot \overline{a} + q_R(x) \cdot \overline{b} + q_O(x) \cdot \overline{c} + q_M(x) \cdot \overline{a} \cdot \overline{b} + (q_C(x) - PI(\mathfrak{z})) - (T_{lo}(x) + T_{mid}(x) \cdot \mathfrak{z}^n + T_{hi}(x) \cdot \mathfrak{z}^{2n}) \cdot Z_H(\mathfrak{z}) = 0$$

Permutation Check: How to Prove!

- For a circuit dependent permutation $\sigma : [3n] \to [3n]$ we need to prove $\sigma(a_1, ...a_n, b_1, ...b_n, c_1, ...c_n) = (a_1, ...a_n, b_1, ...b_n, c_1, ...c_n)$.
- Define $\sigma^*(i) = \begin{cases} \omega^{\sigma(i)} & \text{if } \sigma(i) \in \{1...n\} \\ k_1 \cdot \omega^{\sigma(i)} & \text{if } \sigma(i) \in \{n+1...2n\} \\ k_2 \cdot \omega^{\sigma(i)} & \text{if } \sigma(i) \in \{2n+1...3n\} \end{cases}$
- For $c \in \{1, 2, 3\}$, $S_c, S_{\sigma c}$ be functions/polynomials $\{\omega^1, \ldots \omega^n\} \to H'$ where $H' = H \cup (k_1 \cdot H) \cup (k_2 \cdot H)$ and where $k_1, k_2 \in \mathbb{F}$ are such that $H, k_1 H, k_2 H$ give $3n$ distinct elements.

$$S_c(X) = \begin{cases} X & c = 1 \\ k_1 X & c = 2 \\ k_2 X & c = 3 \end{cases}, \quad S_{\sigma c}(X) = \begin{cases} \sigma^*(i) & \text{for } c = 1 \text{ on input } X = \omega^i \\ \sigma^*(i+n) & \text{for } c = 2 \text{ on input } X = \omega^i \\ \sigma^*(i+2n) & \text{for } c = 3 \text{ on input } X = \omega^i \end{cases}$$

- Same as before, we define $z(\omega) = 1$ and for all $i \in \{2, ...n\}$

$$z(\omega^i) = \prod_{j=1}^{i-1} \frac{a_j + \beta \cdot \omega^j + \gamma}{a_j + \beta \cdot \sigma^*(j) + \gamma} \cdot \frac{b_j + \beta \cdot k_1 \cdot \omega^j + \gamma}{b_j + \beta \cdot \sigma^*(j+n) + \gamma} \cdot \frac{c_j + \beta \cdot k_2 \cdot \omega^j + \gamma}{c_j + \beta \cdot \sigma^*(j+2n) + \gamma}$$

Permutation Check: How to Prove!

- Need to prove: $z(\omega) = 1$
- Need to prove that for $x \in \{\omega ... \omega^n\}$ we have that
$z(x\omega) \cdot ((a(x) + \beta S_{\sigma 1}(x) + \gamma)(b(x) + \beta S_{\sigma 2}(x) + \gamma)(c(x) + \beta S_{\sigma 3}(x) + \gamma))-$
$z(x) \cdot ((a(x) + \beta x + \gamma) \quad (b(x) + k_1 \beta x + \gamma) \quad (c(x) + k_2 \beta x + \gamma)) = 0.$
- Writing in the same format as other equations, we need to prove $\exists T'$ such
that$z(x\omega) \cdot ((a(x) + \beta S_{\sigma 1}(x) + \gamma)(b(x) + \beta S_{\sigma 2}(x) + \gamma)(c(x) + \beta S_{\sigma 3}(x) + \gamma))-$
$z(x) \cdot ((a(x) + \beta x + \gamma) \quad (b(x) + k_1 \beta x + \gamma) \quad (c(x) + k_2 \beta x + \gamma)) = T'(x) Z_H(x)$
- Prove by opening at a random point.
- Verifier Precomputes $[S_{\sigma 1}], [S_{\sigma 2}], [S_{\sigma 3}]$.

Making it non-interactive — Fiat-Shamir Heuristic

- Rather than the verifier specifying uniform values, obtain them by computing $H(\texttt{transcript})$ where `transcript` is the current value of all the messages so far.

Final Notes

- **Interoperatability among implementations:** Presentation is a simplified version of the construction from the $\mathcal{P}\mathfrak{lon}\mathcal{K}$ paper. https://eprint.iacr.org/2019/953.pdf
- **Security proofs are brittle:** Small changes in the scheme can affect security. Be careful when you depart from specifications.