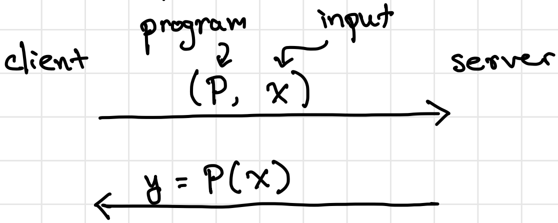
So far, we have focused on proving properties in a privacy-preserving manner. Next, we will look at achieving <u>short</u> proofs that are efficient to verify.

<u>Application</u>: verifiable computation



How do we know that the server computed the correct value?
↳ Can provide a proof $y = P(x)$. To be useful, checking the proof should be much faster than computing P.

<u>Main primitive</u>: aggregation scheme for proofs – batch argument for NP

<u>Setting</u>: given T statements $x_1, \ldots, x_T$ and circuit C, show that all of the statements are true (i.e., $\exists w_i : C(x_i, w_i) = 1$ for all $i \in [T]$).

<u>Naively</u>: Give out T proofs, one for each statement x_i.

<u>Goal</u>: Can we do better. Namely, can we batch prove T statements with a proof of size $o(T)$?

For now, we will not worry about zero-knowledge. Turns out that succinctness can be used to achieve zero-knowledge.

Let $C : \{0,1\}^n \times \{0,1\}^h \to \{0,1\}$ be circuit that computes an NP relation. Take $x_1, \ldots, x_T \in \{0,1\}^n$. We want to prove that there exist $w_1, \ldots, w_h \in \{0,1\}^h$ where $C(x_i, w_i) = 1$ for all $i \in [T]$.

Waters-Wu construction: follows commit-and-prove paradigm

Relies on a GOS-like structure: commit to all of the wires and give a proof that commitments associated with each gate are valid (i.e., consistent with the gate operation)

In GOS, commitments were BGN <u>encryptions</u> (as long as the input)

↳ Will not give succinct proofs

Starting point: <u>succinct</u> commitment scheme

- CRS: $g, h_1, \ldots, h_T$ where $h_i = g^{\alpha_i}$ (and $\alpha_i \xleftarrow{R} \mathbb{Z}_N$)
- Commit to a vector $v = (v_1, \ldots, v_T)$ as follows:

$$C = \prod_{i \in [T]} h_i^{v_i} \quad \leftarrow \text{commitment is now a single group element (independent of } T)$$

Construction overview:

- As in GOS, we index the wires in C in topological order
- Prover starts by computing $C(x_i, w_i)$ for each $i \in [T]$. Let $t_1^{(i)}, \ldots, t_s^{(i)}$ be the wire assignments associated with $C(x_i, w_i)$.
- For each wire $j \in [s]$, prover commits to vector $(t_j^{(1)}, \ldots, t_j^{(T)})$ — namely the values associated with wire j across all T instances. Let c_j be the commitment.

Similar to GOS, need to establish two properties

1. For all $i \in [T]$ and $j \in [s]$, $t_j^{(i)} \in \{0,1\}$
2. For each NAND gate (j_1, j_2, j_3), $t_{j_3}^{(i)} = \text{NAND}(t_{j_1}^{(i)}, t_{j_2}^{(i)})$ for all $i \in [T]$

We start with the first property.

Suppose $c = \prod_{i \in [T]} h_i^{v_i}$. <u>Goal</u>: prove that $v_i \in \{0,1\}$ for all $i \in [T]$.

As before: $v_i \in \{0,1\}$ if and only if $v_i(v_i - 1) = 0$ or equivalently, $v_i^2 = v_i$

$$
\begin{aligned}
e(c,c) &= e\left(\prod_{i \in [t]} h_i^{v_i}, \prod_{i \in [t]} h_i^{v_i}\right) \\
&= e\left(g^{\sum_i \alpha_i v_i}, g^{\sum \alpha_i v_i}\right) \\
&= e(g,g)^{[\sum_i \alpha_i v_i]^2}
\end{aligned}
$$

$$\left[\sum_{i \in [T]} \alpha_i v_i\right]^2 = \sum_{i \in [T]} \alpha_i^2 v_i^2 + \sum_{i \in [T]} \sum_{j \neq i} \alpha_i \alpha_j v_i v_j$$

if $v_i^2 = v_i$, then $\sum_{i \in [T]} \alpha_i^2 v_i^2 = \underline{\sum_{i \in [T]} \alpha_i^2 v_i}$

↳ Can we compute this in the exponent?

$$e\left(c, \prod_{i \in [T]} h_i\right) = e\left(g^{\sum \alpha_i v_i}, g^{\sum \alpha_i}\right) = e(g,g)^{(\sum_i \alpha_i v_i)(\sum \alpha_i)}$$

$$\left[\sum_{i \in [T]} \alpha_i v_i\right]\left[\sum_{i \in [T]} \alpha_i\right] = \sum_{i \in [T]} \alpha_i^2 v_i + \sum_{i \in [T]} \sum_{j \neq i} \alpha_i \alpha_j v_i$$

Main observation: if $v_i^2 = v_i$ for all $i \in [T]$, then

$$e(c,c) = \underbrace{e\left(c, \prod_{i \in [t]} h_i\right)}_{\text{Can be computed by the verifier}} \underbrace{e(g,g)^{\sum_{i \in [T]} \sum_{j \neq i} \alpha_i \alpha_j (v_i v_j - v_i)}}_{\text{"cross terms" [depend on } \alpha_i \alpha_j]}$$

Can be computed by the verifier

"cross terms" [depend on $\alpha_i \alpha_j$]

verifier cannot compute since it does not know values of v_i, v_j

<u>Solution</u>: give out $u_{ij} = g^{\alpha_i \alpha_j}$ for all $i \neq j$

<u>Construction</u>: $crs = \left(g, \{h_i\}_{i\in[T]}, \{u_{ij}\}_{i\neq j}, A = \prod_{i\in[T]} h_i\right)$

Commitment to $v = (v_1, \dots, v_T)$: $c = \prod_{i\in[T]} h_i^{v_i}$

To prove $v_i \in \{0,1\}$, compute $\pi = \prod_{i\in[T]} \prod_{j\neq i} u_{ij}^{v_i v_j - v_i}$

To check the proof, check that

$$e(c, c) \stackrel{?}{=} e(c, A)\, e(g, u)$$

This corresponds to the following relation in the exponent:

$$\sum_{i\in[T]} \sum_{j\in[T]} \alpha_i \alpha_j v_i v_j \stackrel{?}{=} \sum_{i\in[T]} \sum_{j\in[T]} \alpha_i \alpha_j v_i + \sum_{i\in[T]} \sum_{j\neq i} \alpha_i \alpha_j (v_i v_j - v_i)$$

Equality holds if $v_i^2 = v_i$ for all $i \in [T]$. (Completeness)

Soundness is more delicate - will defer to later.

Gate consistency can be implemented like in GOS

[by checking $v_{1,i} + v_{2,i} - 2v_{3,i} + 2 \in \{0,1\}$ for all i]

$$c_1 = \prod_{i\in[T]} h_i^{v_{1,i}}, \quad c_2 = \prod_{i\in[T]} h_i^{v_{2,i}}, \quad c_3 = \prod_{i\in[T]} h_i^{v_{3,i}}$$

Compute $c_1 c_2 c_3^{-2} \prod_{i\in[T]} h_i^2$:

$$c^* = c_1 c_2 c_3^{-2} \prod_{i\in[T]} h_i^2 = \left(\prod_{i\in[T]} h_i^{v_{1,i}}\right)\left(\prod_{i\in[T]} h_i^{v_{2,i}}\right)\left(\prod_{i\in[T]} h_i^{-2v_{3,i}}\right)\left(\prod_{i\in[T]} h_i^{2}\right)$$

$$= \prod_{i\in[T]} h_i^{v_{1,i} + v_{2,i} - 2v_{3,i} + 2}$$

c^* is a commitment to $V_{1,i} + V_{2,i} - 2V_{3,i} + 2$ for all $i \in [T]$

Can use previous approach to check that component of committed vector is a $\{0,1\}$ value.

How do we argue soundness?

We will consider non-adaptive soundness where the adversary has to choose the false statement before seeing the public parameters (crs).

Approach is to program a secret index i^* into the CRS. Given a valid proof on a statement $(x_1, \ldots, x_T)$, it will be possible to extract a witness w_{i^*} such that $C(x_{i^*}, w_{i^*}) = 1$.

[Cannot extract witness for all indices $i \in [T]$ from the same proof – why?]

As long as crs hides the index i^*, this suffices to show non-adaptive soundness:

- Fix any statement $(x_1, \ldots, x_T)$. If this is false, there exists $i^* \in [T]$ where x_{i^*} is false.
- Suppose we set the CRS to be extracting at i^*
- Adversary should still produce an accepting proof (otherwise, it breaks index hiding)
- If adversary produces valid proof in this case, then we extract a witness for x_{i^*}. But this is not possible (since no such witness exists!)

With complexity leveraging for index hiding, this suffices to show adaptive soundness.

Programming the CRS to extract on index i^*:

- Normal CRS: $h_1 = g_p^{\alpha_1}, \ldots, h_T = g_p^{\alpha_T}$, $u_{ij} = g_p^{\alpha_i \alpha_j} = h_i^{\alpha_j} = h_j^{\alpha_i}$ } as described, all elements are in the mod-p subgroup
- Binding CRS at index i^*: "lift" element h_i to the full group.

  [set $h_{i^*} = g^{\alpha_{i^*}}$ where g generates the full group
  cross terms involving i^* also lifted to the full group as a result]

- Binding CRS indistinguishable from real CRS by subgroup decision
- In binding mode, commitment to $v = (v_1, \ldots, v_T)$ is now

$$c = \prod_{i \in [T]} h_i^{v_i} = \underbrace{h_{i^*}^{v_{i^*}}}_{\substack{\text{in full} \\ \text{group}}} \underbrace{\prod_{i \neq i^*} h_i^{v_i}}_{\substack{\text{in order-}p \\ \text{subgroup}}}$$

- Extraction trapdoor is the factor p, which can be used to project away the modulo-p component:

$$c^p = h_{i^*}^{v_{i^*} p} \prod_{i \neq i^*} h_i^{v_i p}$$

$$= \left(g^{\alpha_{i^*} p}\right)^{v_{i^*}} \prod_{i \neq i^*} \left(g_p^{\alpha_i v_i p}\right)$$

$$= \left(g^{\alpha_{i^*} p}\right)^{v_{i^*}}$$

↳ have isolated component i^* and can see if $v_{i^*} = 0$ or $v_{i^*} = 1$

[if verification relations pass, these are the <u>only</u> two possibilities]

Essentially, when the CRS binds at index i^*, the proof system is <u>statistically sound</u> at index i^* (since we can extract the witness at i^*).

↳ Also called "somewhere statistical soundness"