CSE 575: Advanced Cryptography
Fall 2024
Lecture 25 (last lecture!!!)

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

- Announcements
- “Prescribed” permutation check PIOP
- Compiling (P)IOPs to arguments
- Semester retrospective

Agenda for this lecture

- Announcements
- “Prescribed” permutation check PIOP
- Compiling (P)IOPs to arguments
- Semester retrospective

Announcements

- Final is out, due 12/14 11:59pm.
  - Take **at most** one week, but you choose when to start.
- Rest of class: no typeset lecture notes.
  - Can get extra credit for typesetting
- This is our last class!
  - *Please* do your course evals – I really appreciate feedback!

Agenda for this lecture

- Announcements
- “Prescribed” permutation check PIOP
- Compiling (P)IOPs to arguments
- Semester retrospective

"Prescribed" permutation check

Let $W: \Omega \to \Omega$ be poly that permutes Ω according to $\sigma: [k] \to [k]$

$$W(\omega^i) = \omega^{\sigma(i)} \quad \text{for } i \in k$$

Prove f, g are the same on Ω;

up to W: $f(a) = g(W(a))$ $\forall a$

Observe: if $(W(a), f(a))$'s a perm of $(a, g(a))$

then $f(a) = g(W(a)) \quad \forall a \in \Omega$

Proof:

$$U = \{(W(1), f(1)), (W(\omega), f(\omega)), \ldots, (W(\omega^{k-1}), f(\omega^{k-1}))\}$$

$$V = \{(1, g(1)), (\omega, g(\omega)), \ldots, (\omega^{k-1}, g(\omega^{k-1}))\}$$

"Prescribed" permutation check

Observe: if $(w(a), f(a))$ is a perm of $(a, g(a))$

then $f(a) = g(w(a)) \quad \forall a \in \Omega$

Proof:

$U = [(w(1), f(1)), (w(w), f(w)), \ldots, (w(w^{k-1}), f(w^{k-1}))]$

$V = [(1, g(1)), (w, g(w)), \ldots, (w^{k-1}, g(w^{k-1}))]$

Our claim implies that $\exists \sigma$ s.t. $U[\sigma(i)] = V[i] \; \forall i$

Let $\sigma(i) = j$

$U[j] = V[i]$

$(w(w^i), f(w^i)) = (w^i, g(w^i))$

$w(w^i) = w^i$

$f(w^i) = g(w^i) \Rightarrow f(w^i) = g(w(w^i))$

$\square$

"Prescribed" permutation check

$$\hat{f}(X,Y) = \prod_{a \in \Omega} (X - Y w(a) - f(a))$$

$$\hat{g}(X,Y) = \prod_{a \in \Omega} (X - Y a - g(a))$$

total degree k

Lemma: $\hat{f} \equiv \hat{g}$ iff $(w(a), f(a))$ is a permutation of $(a, g(a))$

Proof: ($\Rightarrow$) Use unique factorization of $\hat{f}, \hat{g}$

($\Leftarrow$) construct $\hat{f}, \hat{g}$, observe they must be same b/c factors are the same

"Prescribed" permutation check

$P(S, P, X, W, pp)$:

compute trace T,
interpolate it

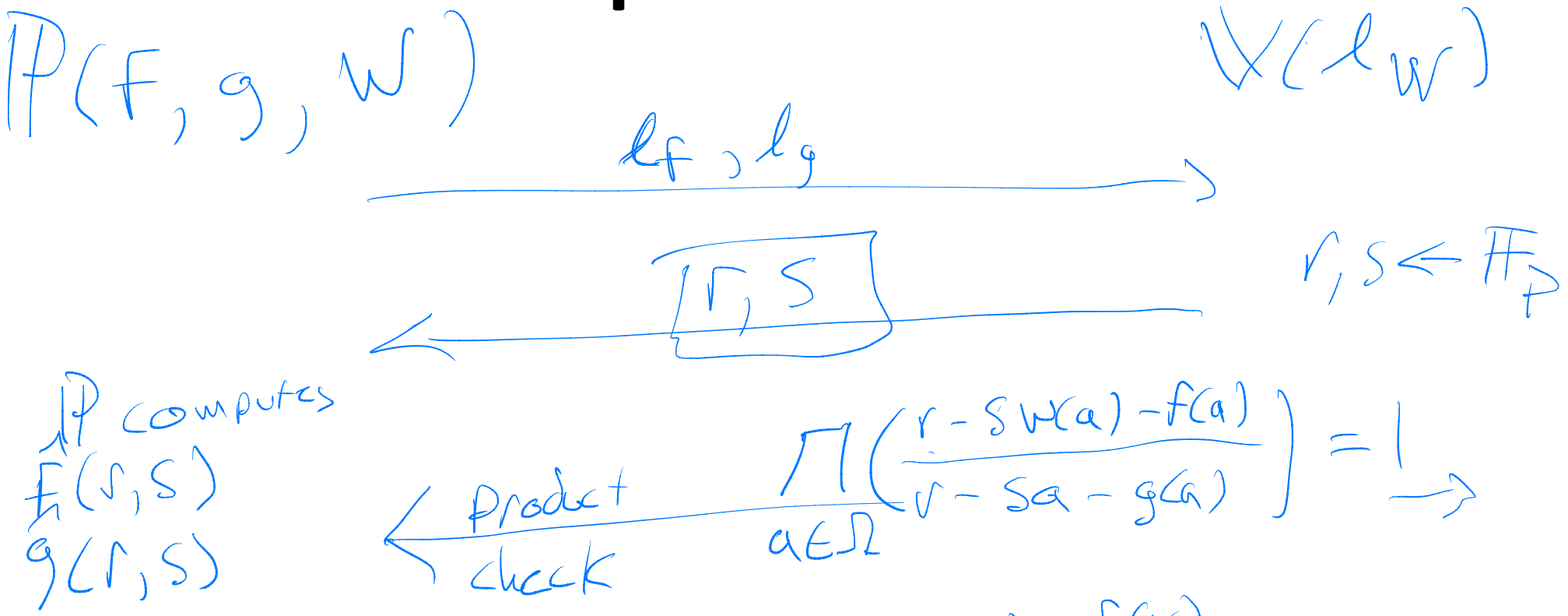
ℓ_T →

Plonk PIOP $V(\ell_S, \ell_P, X)$:

build ℓ_1 poly v

PIOPs of ℓ_T for checks:
1. pub input
2. gate
3. wiring
4. output

"Prescribed" permutation check



$$h(X) = \frac{r - sW(X) - f(X)}{r - sX - g(X)}$$

Completeness: follows from $\hat{f}/\hat{g}$

Soundness: $\leq \frac{2d}{|\mathbb{F}|}$ by soundness of product check

Agenda for this lecture

- Announcements
- “Prescribed” permutation check PIOP
- Compiling (P)IOPs to arguments
- Semester retrospective

Security of arguments

Compile (P)IOP to argument:

1. Everywhere you see label ℓ_f, replace with $C_f \leftarrow \mathsf{Com}(f)$
2. Ver randomness the same
3. When P sends eval of f at z, send $(\pi, v) \leftarrow \mathsf{PC.Eval}(pp, C_f, f, z, r)$
4. Ver's final check verifies all π using $\mathsf{PC.Ver}$

[BCS16]

Security of arguments

Thm: Let $\Pi = \langle P, V, (d_1, \ldots, d_k) \rangle$ be a PIOP for $\mathcal{R}$ and $PC = \langle Setup, Com, Eval, Ver \rangle$ be a poly commit. If Π is complete/sound and PC is eval binding*, then

$$A[\Pi, PC]$$

is complete / sound

Proof: non-trivial !

Compiling PIOPs to arguments

The rest of the way to zk-SNARKs

1. Fiat-Shamir:
   interactive arg $\Rightarrow$ non-interactive argument

2. ZK:
   define ZK-PIOP + hiding for PC,
   show transform gives ZK arg

3. Knowledge Soundness

Agenda for this lecture

- Announcements
- “Prescribed” permutation check PIOP
- Compiling (P)IOPs to arguments
- Semester retrospective

Looking back

Info-theoretic security

Comp. ind

Sym crypto

asymmetric

ZKPs

PIOPs