Information-Theoretic Proofs + Cryptographic Commitments = Succinct Arguments

or "How to verifiably delegate computation?"

Russell W.F. Lai
Aalto University

Foundations Friday
2023-5-26

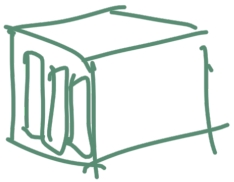
Verifiable Computation

Client

e.g. ML algorithms

e.g. large database

f, x



Server

f, x

y

$y \leftarrow f(x)$

Q: How can client know if indeed $y = f(x)$?

A: Server can "prove" it!

Goal

Design protocols such that

1) If $y = f(x)$, client will be convinced.

2) If $y \neq f(x)$, no efficient server can fool client.

More rigorously, want to show:

Efficient, convincing, cheating server $\Longrightarrow$ Efficient algorithm for solving computationally hard problems, e.g. factoring, discrete logarithm, LWE, ...

Argument / Proof Systems

- NP Languages $L_\lambda = \{ \text{stmt} : \exists \text{ wit s.t. } (\text{stmt}, \text{wit}) \in R_\lambda \}$

  $\lambda \in \mathbb{N}$ = security / size parameter.

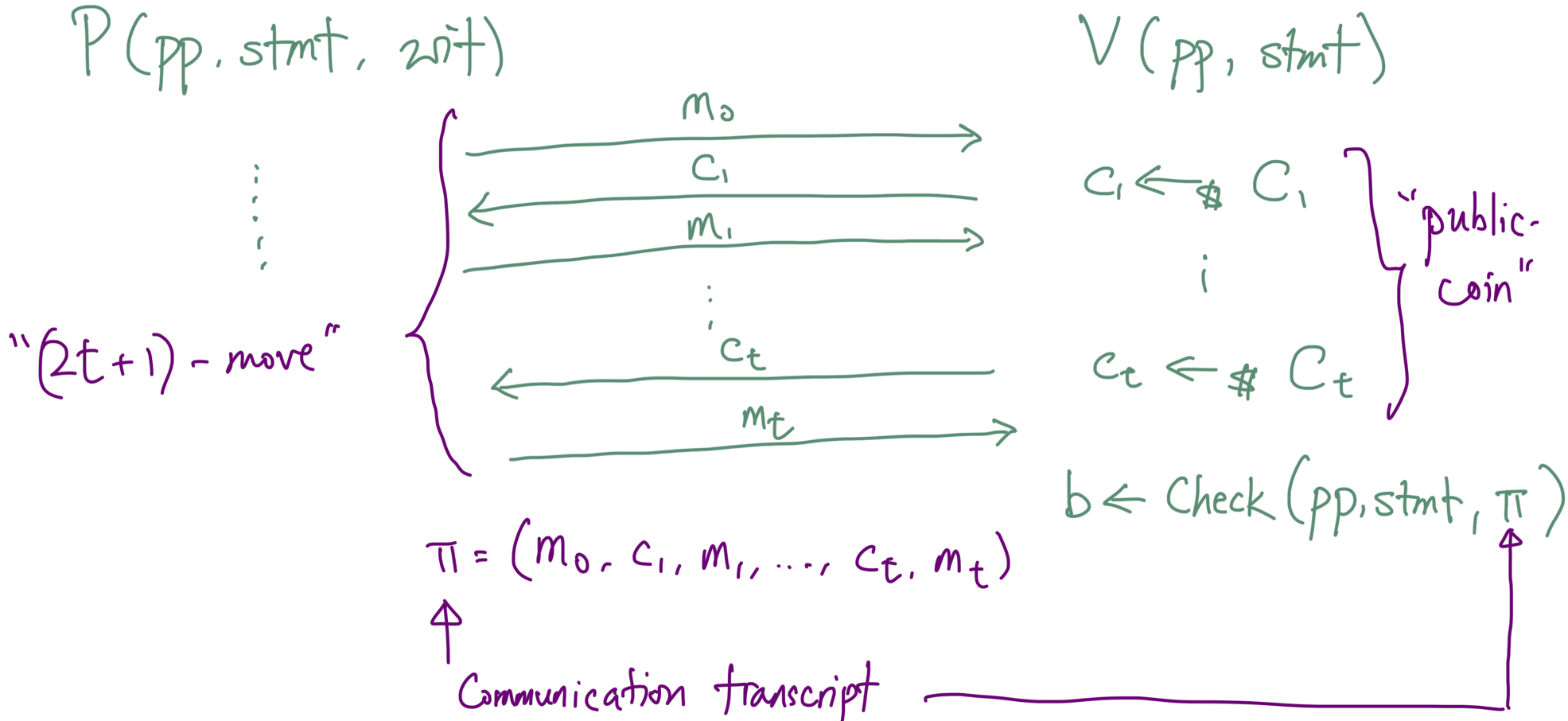
- Size of stmt = $|\text{stmt}| = \text{poly}(\lambda)$.

- Argument System $\Pi = (\text{Setup}, P, V)$ efficient algorithms

  (P: prover, V: verifier)

  - $pp \leftarrow \text{Setup}(1^\lambda)$
  - $b \leftarrow \langle P(pp, \text{stmt}, \text{wit}), V(pp, \text{stmt}) \rangle$

Typical Structure of $b \leftarrow \langle P(pp, stmt, wit), V(pp, stmt) \rangle$



Properties of Argument Systems I

- Completeness : If $(stmt, wit) \in R_\lambda$, then

$$\Pr\left[\langle P(pp, stmt, wit), V(pp, stmt)\rangle = 1\right] \approx 1$$

- Soundness : If $stmt \notin L_\lambda$, then $\forall\ P^*, wit^*$

$$\Pr\left[\langle P^*(pp, stmt, wit^*), V(pp, stmt)\rangle = 1\right] \approx 0$$

Argument : The above holds $\forall$ PPT P^*.

Proof : The above holds $\forall$ unbounded P^*

Properties of Argument Systems II

- Succinctness : $|\pi| \ll |stmt|$

  Stronger variant: $Time(V) \ll |stmt|$

  after pre-processing stmt.

- Non-Interactiveness : $t = 0$ or "1-move"

  In this case, write $\pi \leftarrow P(pp, stmt, wit)$

  $b \leftarrow V(pp, stmt, \pi)$

Fiat-Shamir Transformation

in the "random oracle model"

Theorem (informal):

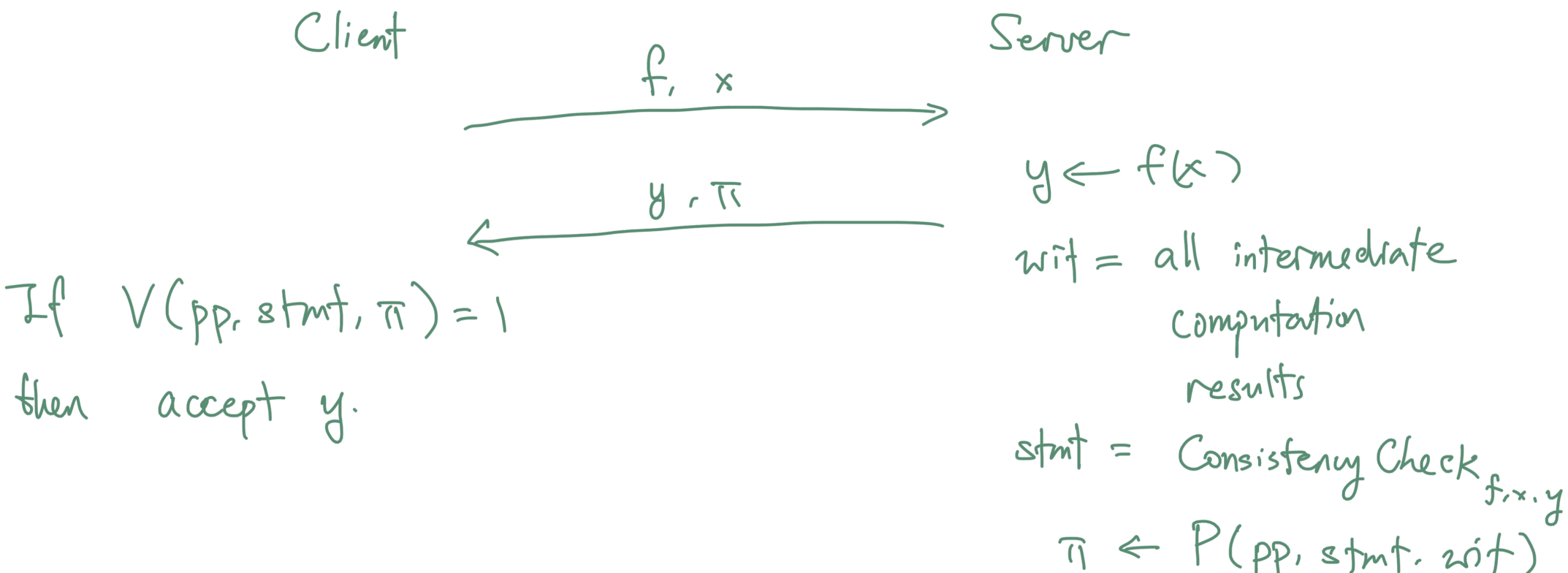
$\Pi \longmapsto FS(\Pi)$

Take-home: Just consider these

{ public-coin
  log(λ)-move
  complete
  "special-sound"
  succinct }

{ non-interative
  complete
  sound
  succinct } Succinct Non-interactive ARGument (SNARG)

Applications I Verifiable Computation



Applications II: Incrementally Verifiable Computation

$$x_0 \xrightarrow{f_1} x_1 \xrightarrow{f_2} x_2 \xrightarrow{f_3} \cdots \xrightarrow{f_t} x_t$$

- To verifiably compute f_i:
  - **Step(i, x_{i-1}, π_{i-1})**

    If π_{i-1} proves that $\exists (x_{i-2}, \pi_{i-2})$ s.t.

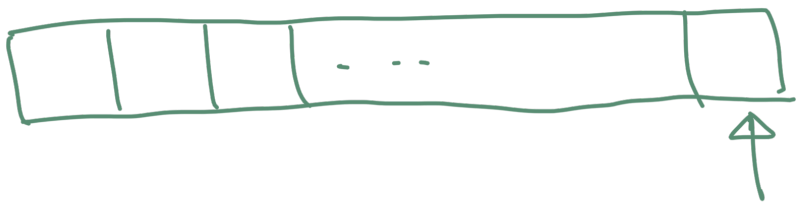
    $$x_{i-1} = \text{Step}(i-1, x_{i-2}, \pi_{i-2})$$

    then output $x_i = f_i(x_{i-1})$

  - Generate π_i proving that $\exists (x_{i-1}, \pi_{i-1})$ s.t.

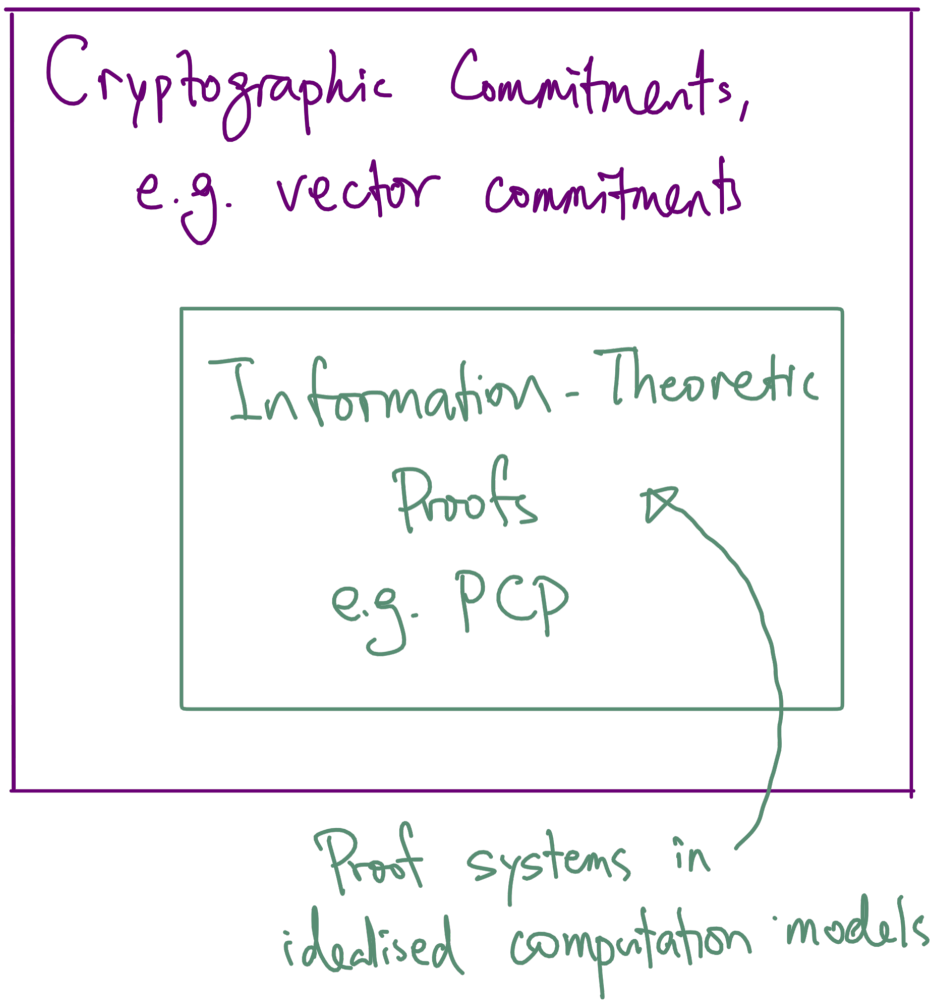
    $$x_i = \text{Step}(i, x_{i-1}, \pi_{i-1})$$

Applications III Aggregating Blockchain Transactions

Blockchain 

- Fee to insert $(tx_1, \ldots, tx_{100000})$
  $= 100000$ Aalto coin
- Fee to insert π : "$(tx_1, \ldots, tx_{100000})$ are valid"
  $= 100$ Aalto coin

SNARG Construction : Compilation Paradigm



succinct argument

= Π $\longmapsto$ $FS(\Pi)$

SNARG

Note: Direct constructions of SNARGs exist as well.

Plan

- Example of IT proofs:

  Probabilistically checkable proofs (PCP)

- Example of cryptographic commitments:

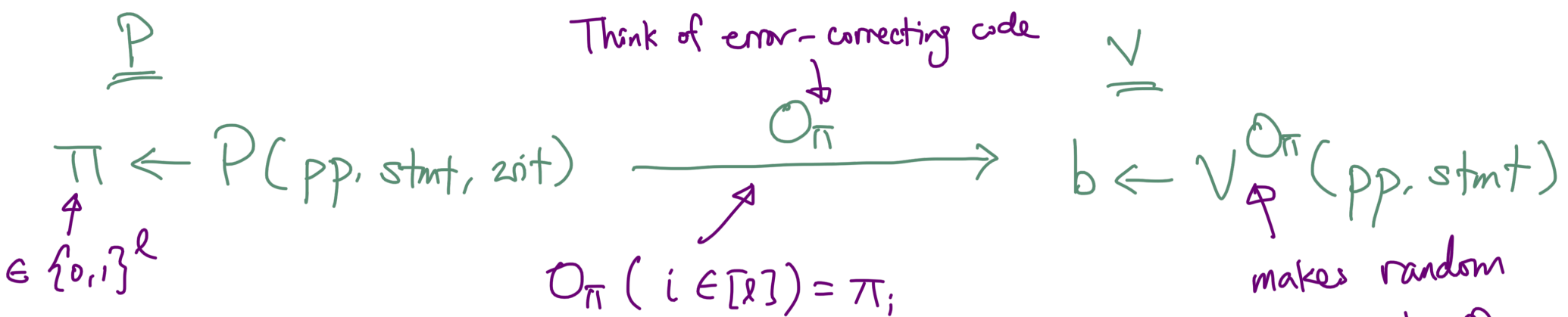
  Vector commitments (VC)

- Killian's succinct argument = PCP + VC

- More examples...

Probabilistically Checkable Proofs (PCP)

- Non-interactive proof systems in an idealised model



- PCP Theorem: $\forall$ NP language, $\exists$ PCP where
  - V uses $O(\lg \lambda)$ randomness
  - V makes $O(1)$ queries
  - Soundness error $= \frac{1}{2}$ $\xrightarrow{\lambda \text{ repetition}}$ $2^{-\lambda}$

Vector Commitment

Crypto

PPT algorithms $COM = (Setup, Com, Open, Verify)$

- $pp \leftarrow Setup(1^\lambda)$
- $(com, aux) \leftarrow Com(pp, x)$ — e.g. $x \in \{0,1\}^\ell$
- $w \leftarrow Open(pp, aux, i)$ — $i \in [\ell]$
- $b \leftarrow Verify(pp, com, (i, y), w)$

Correctness:

$Verify(pp, com, (i, x_i), w) = 1$

Succinctness:

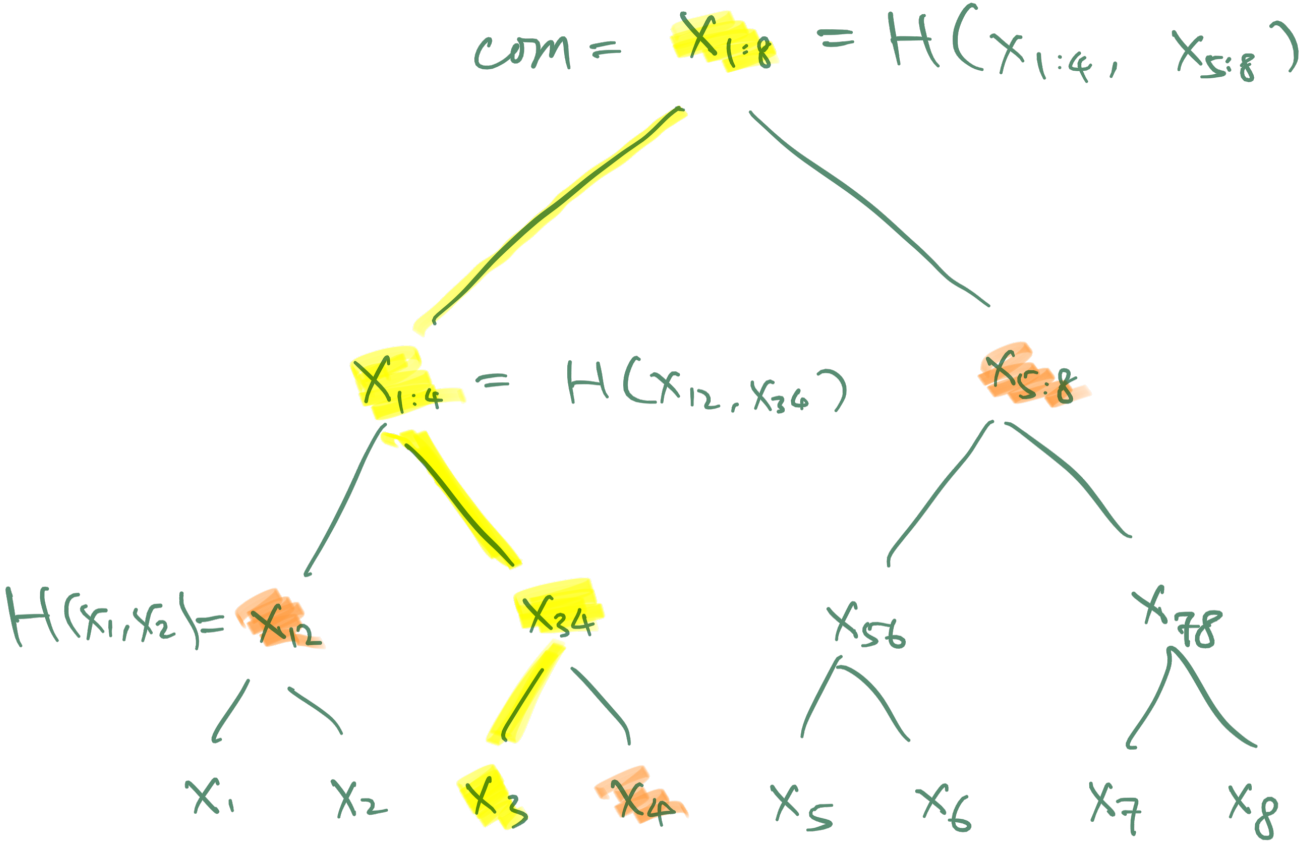
$|com| \ll \ell$

$|w| \ll \ell$

Position-Binding: No efficient adversary can find (com, i, w_0, w_1) s.t.

$$Verify(pp, com, (i,0), w_0) = Verify(pp, com, (i,1), w_1) = 1$$

Merkle-Tree Commitment



H collision-resistant ⇒ COM position-binding

H: collision-resistant hash function

Opening of x_3:

- root-to-leaf path
- siblings

Verify:

$com \stackrel{?}{=} x_{1:8}$

$x_{1:8} \stackrel{?}{=} H(x_{1:4}, x_{5:8})$

$x_{1:4} \stackrel{?}{=} H(x_{1:2}, x_{3:4})$

$x_{3:4} \stackrel{?}{=} H(x_3, x_4)$

Killian's Succinct Argument Π

$\Pi.P(pp, stmt, wit)$

$\pi \leftarrow PCP.P(pp, stmt, wit)$

$(com, aux) \leftarrow Com(pp, \pi)$

$|com| \ll |\pi|$

com →

$i_1, \ldots, i_\lambda \leftarrow [Q]$

← $i_1, \ldots, i_\lambda$

$\forall j \in [\lambda]$,

$w_j \leftarrow Open(pp, aux, i_j)$

$(\pi_{i_j}, w_j)_{j=1}^{\lambda}$ →

$\Pi.V(pp, stmt)$

$\forall j \in [\lambda]$,

$Verify(pp, com, (i_j, \pi_{i_j}), w_j)$

$\pi_{i_j} \in \{0,1\}$ $|w_j| \ll |\pi|$

Why sound? Intuition:

P commits to π before seeing $i_1, \ldots, i_\lambda$

Why are we not satisfied?

PCP:

- Very high prover time — large poly(|stmt|)
- Usually, PCPs support very "unexpressive" NP-complete languages
  $\Rightarrow$ High concrete overhead for NP reduction

Merkle-Tree Commitment:

- Opening size linear in # opened positions
  $\Rightarrow$ proof size $= O(\lambda^2 \lg \lambda)$.
  Can we do $O(\lambda \lg \lambda)$?

Subvector Commitment [L Malavolta 19]

"Strong root assumption" $\Rightarrow$ position-binding

Open to n positions at the cost of 1.

Setup(1^λ)

$N \leftarrow \text{RSAGen}(1^\lambda)$

$A \leftarrow \mathbb{Z}_N$

$e_1, \ldots, e_\ell$ primes

$$e = \prod_{i=1}^{\ell} e_i$$

$$S_i = A^{e/e_i} \mod N$$

$$pp = (N, A, (S_i, e_i)_{i=1}^{\ell})$$

Com$(pp, x \in \{0,1\}^\ell)$

$$com = C = \prod_{i=1}^{\ell} S_i^{x_i} \mod N$$

$$aux = x$$

Open$(com, aux, I \subseteq [\ell])$

$J = [\ell] \setminus I$, $e_J = \prod_{j \in J} e_j$

$$w_I = \prod_{j \in J} A^{x_j \frac{e_J}{e_j}}$$

Verify$(pp, com, (I, x_I), w_I)$

$$C \stackrel{?}{=} \prod_{i \in I} S_i^{x_i} \cdot w_I^{e_I}$$

Instantiations of Compilation Paradigm

Less powerful IT proofs →

IT Proofs	PCP	Linear PCP $\mathcal{O}_\pi : \mathbb{F}^\ell \to \mathbb{F}$ linear function w/ coeff π	Interactive Oracle Proofs (IOP) $\mathcal{O}_{\pi_1}, \mathcal{O}_{\pi_2}, \ldots, \mathcal{O}_{\pi_t}$ $\mathcal{O}_{\pi_i}(j) = \pi_{ij}$	Polynomial IOP $\mathcal{O}_{\pi_1}, \ldots, \mathcal{O}_{\pi_t}$ $\mathcal{O}_{\pi_i}(x) = f_{\pi_i}(x)$, $f_{\pi_i} \in \mathbb{F}[X]$
Commitments	(S)VC	Functional Commitment for linear functions Commit to $x \in \mathbb{F}^\ell$ Open to $L(x)$, L linear	(S)VC	Polynomial Commitment Commit to $f \in \mathbb{F}[X]$ Open to $f(x)$

More powerful commitments →

Rank-1 Constraint Satisfiability (R1CS)

Convenient NP-complete language for building arguments.

$$\text{stmt} = (\underbrace{A, B, C, D}_{\text{matrices}}, \underbrace{y}_{\text{vector}}) \quad \text{over } \mathbb{F}$$

$$\text{wit} = x \quad \text{s.t.}$$

$$Ax \circ Bx = Cx$$

$$Dx = y$$

$\circ$: Hadamard product

We will build a linear PCP for R1CS

Polynomial Interpolation

Let $\alpha_1, \dots, \alpha_\ell \in \mathbb{F}$ distinct evaluation points

$$f(X) = f_0 + f_1 X + \dots + f_{\ell-1} X^{\ell-1} \in \mathbb{F}[X]$$

Vandermonde matrix $V_{\alpha_1 \cdots \alpha_\ell} = \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{\ell-1} \\ 1 & \alpha_2 & \cdots & \alpha_2^{\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_\ell & \cdots & \alpha_\ell^{\ell-1} \end{bmatrix}$

$$V_{\alpha_1 \cdots \alpha_\ell} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{\ell-1} \end{bmatrix} = \begin{bmatrix} f(\alpha_1) \\ f(\alpha_2) \\ \vdots \\ f(\alpha_\ell) \end{bmatrix}$$

$V_{\alpha_1 \cdots \alpha_\ell}$ invertible

$\Rightarrow \forall \beta_1, \dots, \beta_\ell \in \mathbb{F}$, $\exists$ unique $g \in \mathbb{F}[X]$, $\deg(g) \le \ell - 1$, $g(\alpha_i) = \beta_i \quad \forall i \in [\ell]$

$$\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{\ell-1} \end{bmatrix} = V_{\alpha_1 \cdots \alpha_\ell}^{-1} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_\ell \end{bmatrix}$$

Linear PCP for R1CS

$Ax \circ Bx = Cx$
$Dx = y$

Let $f, g, h \in \mathbb{F}[X]$, $\alpha_1, \ldots, \alpha_{2\ell} \in \mathbb{F}$, s.t.

$\deg$ f, g: $\ell - 1$; h: $2\ell - 1$

$f(\alpha_i) = (Ax)_i \quad \forall i \in [\ell]$,
$g(\alpha_i) = (Bx)_i$

$$h(\alpha_i) = \begin{cases} (Cx)_i, & i \in [\ell] \\ f(\alpha_i) \cdot g(\alpha_i), & i \in [\ell+1, 2\ell] \end{cases}$$

$$\mathcal{O}_\pi : \pi = \begin{bmatrix} x \\ h(\alpha_{\ell+1}) \\ \vdots \\ h(\alpha_{2\ell}) \end{bmatrix}$$

Verifier queries:

- $q_A^T = [1\ r \cdots r^{\ell-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_\ell} (A \mid 0)$
- $q_B^T = [1\ r \cdots r^{\ell-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_\ell} (B \mid 0)$
- $q_C^T = [1\ r \cdots r^{\ell-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_{2\ell}} \begin{bmatrix} C & 0 \\ 0 & I_\ell \end{bmatrix}$
- $q_D^T = [1\ s \cdots s^{\ell-1}]\, [D \mid 0]$

where $r, s \leftarrow \mathbb{F}$ random

Linear PCP for RICS

$$\mathcal{O}_\pi : \pi = \begin{bmatrix} x \\ h(\alpha_{l+1}) \\ \vdots \\ h(\alpha_{2l}) \end{bmatrix}$$

Verifier queries:

- $q_A^T = [1\ r \cdots r^{l-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_l} (A \mid 0)$
- $q_B^T = [1\ r \cdots r^{l-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_l} (B \mid 0)$
- $q_C^T = [1\ r \cdots r^{l-1}]\, V^{-1}_{\alpha_1 \cdots \alpha_{2l}} \begin{bmatrix} C & 0 \\ 0 & I_l \end{bmatrix}$
- $q_D^T = [1\ s \cdots s^{l-1}]\, [D \mid 0]$

where $r, s \leftarrow \mathbb{F}$ random

Response	Expected
$q_A^T \pi = r_A$	$f(r)$
$q_B^T \pi = r_B$	$g(r)$
$q_C^T \pi = r_C$	$h(r)$
$q_D^T \pi = r_D$	$[1\ s \cdots s^{l-1}]\, y$

Verifier checks:

$$\begin{cases} r_A \cdot r_B \stackrel{?}{=} r_C \\ r_D \stackrel{?}{=} [1\ s \cdots s^{l-1}]\, y \end{cases}$$

Selected Questions I

- Compilation v.s. direct?

  State-of-the-art: Polynomial IOP + Polynomial commitment v.s. direct.

- (More) concretely efficient constructions?

  e.g. round complexity, proof size, prover time, verifier time, preprocessing, expressiveness of supported NP language

  Goal: Proving $y = f(x)$ as efficient as computing f

Selected Questions II

- Design "proof-friendly" algorithms

  e.g. use low-depth circuit, use low-degree polynomial, avoid or reduce control flow (eg. if-then-else), more arithmetic, less combinatoric, ...

- Discover mathematical structures for constructing arguments

  e.g. hinted and structured lattice problems

Summary

- Introduction to argument systems
- Applications
- Compilation paradigm for SNARGs
- PCP, VC, Killian's argument
- Different instantiations of compilation
- Linear PCP for R1CS
- Research directions

Recent related work:

https://ia.cr/2021/202

https://ia.cr/2022/941

Russell W.F. Lai

russell.lai@aalto.fi

russell-lai.hk

Aalto CS building B127