Computation-Trace iO (CiO) and its Applications

Yu-Chi Chen, Academia Sinica, Taiwan
Sherman Chow, Chinese Univ of Hong Kong
Kai-Min Chung, Academia Sinica, Taiwan
Russell Lai, Chinese Univ of Hong Kong
Wei-Kai Lin, Academia Sinica, Taiwan
Hong-Sheng Zhou, Virginia Commonwealth Univ
Example: Delegated Computation

- Client: $\tilde{O}(|A| + |x|)$ size encoding, $\tilde{O}(|y|)$ verification time
- Server: $\tilde{O}(\text{Time}(F, x))$ for comp., $\tilde{O}(\text{Space}(F, x))$ for storage
- Model of Computation:
  - Circuit: no random access / branching
  - Turing Machine (TM) / RAM: no parallelism
  - Parallel RAM: random access & parallelism
Example: Delegated Computation

- Client: $\tilde{O}(\|F\| + \|x\|)$ size encoding, $\tilde{O}(\|y\|)$ verification time
- Server: $\tilde{O}(\text{Time}(F, x))$ for comp., $\tilde{O}(\text{Space}(F, x))$ for storage
- Model of Computation:
  - Circuit: no random access / branching
  - Turing Machine (TM) / RAM: no parallelism
  - Parallel RAM: random access & parallelism
Example: Delegated Computation

- Client: $\tilde{O}(|A| + |x|)$ size encoding, $\tilde{O}(|y|)$ verification time
- Server: $\tilde{O}(\text{Time}(F, x))$ for comp., $\tilde{O}(\text{Space}(F, x))$ for storage
- Model of Computation:
  - Circuit: no random access / branching
  - Turing Machine (TM) / RAM: no parallelism
  - Parallel RAM: random access & parallelism
Example: Delegated Computation

Client

\[
\text{Encode}(F, x) \quad \rightarrow \quad y, \pi \\
\leftarrow \quad \text{Acc. / Rej.}
\]

Server

- Client: $\tilde{O}(|A| + |x|)$ size encoding, $\tilde{O}(|y|)$ verification time
- Server: $\tilde{O}(\text{Time}(F, x))$ for comp., $\tilde{O}(\text{Space}(F, x))$ for storage
- Model of Computation:
  - Circuit: no random access / branching
  - Turing Machine (TM) / RAM: no parallelism
  - Parallel RAM: random access & parallelism

6/9
Delegated Computation

- Client efficiency: Encoding: $\tilde{O}(|F| + |x|)$; Verify: $\tilde{O}(|y|)$
- Server efficiency: Preserve PRAM efficiency of $F(x)$
  - Parallel time, total time, space complexity
- Public verifiability
- Privacy: program, input, output
- Reusability, Persistent memory, etc.
Delegated Computation

**Ideal**

[CHJV15,BGLPT15]: Del. for RAM w/ space-dependent encoding

Client encoding: $\tilde{O}(|F| + |x| + \text{Space}(F, x))$


- Client efficiency: Encoding: $\tilde{O}(|F| + |x|);$ Verify: $\tilde{O}(|y|)$
- Server efficiency: Preserve PRAM efficiency of $F(x)$
  - *Parallel time*, total time, space complexity
- Public verifiability
- Privacy: program, input, output
- Reusability, Persistent memory, *etc.*
**Ideal Delegated Computation**

[CHJV15, BGLPT15]: Del. for RAM w/ space-dependent encoding

Client encoding: $\tilde{O}(|F| + |x| + \text{Space}(F, x))$

[Canetti-Holmgren-Jain-Vaikuntanathan; Bitansky-Garg-Lin-Pass-Telang]

[KLW15, PR14]: Ideal Delegation for TM

Server efficiency: TM complexity of $F(x)$

[Koppula-Lewko-Waters; Paneth-Rothblum]

- Client efficiency: Encoding: $\tilde{O}(|A| + |x|)$; Verify: $\tilde{O}(|y|)$
- Server efficiency: Preserve PRAM efficiency of $F(x)$
  - Parallel time, total time, space complexity
- Public verifiability
- Privacy: program, input, output
- Reusability, Persistent memory, etc.
Delegated Computation

**Ideal** Delegated Computation

[CHJV15,BGLPT15]: Del. for RAM w/ space-dependent encoding

Client encoding: $\tilde{O}(|F| + |x| + \text{Space}(F, x))$

[Canetti-Holmgren-Jain-Vaikuntanathan; Bitansky-Garg-Lin-Pass-Telang]

[KLW15,PR14]: *Ideal* Delegation for TM

Server efficiency: TM complexity of $F(x)$

[Koppula-Lewko-Waters; Paneth-Rothblum]

**Our Result:** *Ideal* Del. for PRAM based on iO for circuit

- Client efficiency: Encoding: $\tilde{O}(|F| + |x|)$; Verify: $\tilde{O}(|y|)$
- Server efficiency: Preserve PRAM efficiency of $F(x)$
  - *Parallel time*, total time, space complexity
- Public verifiability
- Privacy: program, input, output
- Reusability, Persistent memory, *etc.*
**Computation-Trace**

- Computation $\Pi = \text{Program } F + \text{Input } x$
- *Computation trace* = “all snapshots of memory and CPU states”
Computation-Trace

- Computation $\Pi = \text{Program } F + \text{Input } x$
- **Computation trace** = “all snapshots of memory and CPU states”

Initial Memory

Memory access $\text{access}^0 = (\text{loc, bit, R/W})$

CPU Step0

Initial State

$\Pi = (F, x)$
Computation-Trace

- Computation $\Pi = \text{Program } F + \text{Input } x$
- **Computation trace** = “all snapshots of memory and CPU states”

\[ \Pi = (F, x) \]
Computation-Trace

- Computation $\Pi = \text{Program } F + \text{Input } x$
- **Computation trace** = “all snapshots of memory and CPU states”

\[
\Pi = (F, x)
\]

Computations trace = $(\text{mem}^0, \text{state}^0, \text{access}^0, \ldots \text{mem}^T, \text{state}^T, \text{access}^T)$
ComputaGon trace $= (\text{mem}_0, \text{state}_0, \text{access}_0, \ldots, \text{mem}_T, \text{state}_T, \text{access}_T)$ (in RAM model)

**Efficiency:** Obfuscation time $= \text{poly}(|\Pi|)$ (indep. of evaluation of $\Pi$)

Evaluation complexity $= \text{PRAM}$ complexity of $\Pi$
**Computation-Trace iO (CiO)**

- If $\Pi$ and $\Pi'$ generate the same computation trace, then $\text{CiO}(\Pi) \approx \text{CiO}(\Pi')$

**Efficiency:** Obfuscation time = $\text{poly}(|\Pi|)$ (indep. of evaluation of $\Pi$)

Evaluation complexity = PRAM complexity of $\Pi$

---

**Diagram:**

- $\Pi = (F, x)$
  - Computation trace = $(\text{mem}^0, \text{state}^0, \text{access}^0, \ldots, \text{mem}^T, \text{state}^T, \text{access}^T)$

- $\Pi' = (F', x)$
  - Computation trace = $(\text{mem}'^0, \text{state}'^0, \text{access}'^0, \ldots, \text{mem}'^T, \text{state}'^T, \text{access}'^T)$

---

(in RAM model)
Compution-Trace iO (CiO)

• If $\Pi$ and $\Pi'$ generate the same computation trace,
Then $\text{CiO}(\Pi) \approx \text{CiO}(\Pi')$

Efficiency: Obfuscation time = $\text{poly}(|\Pi|)$ (indep. of evaluation of $\Pi$)

Evaluation complexity = PRAM complexity of $\Pi$
CiO for PRAM
CiO($\Pi_{Sig}$) → CiO for PRAM

Signature

Ideal Delegation for PRAM w/o Privacy
CiO(π_{Sig})

CiO for PRAM

Signature

Oblivious PRAM
Public-Key Enc.
Signature

Ideal Delegation
for PRAM
w/o Privacy

Ideal Delegation
for PRAM
with Privacy
CiO($\Pi_{Sig}$) for PRAM

Ideal Delegation for PRAM w/o Privacy

Signature

Oblivious PRAM Public-Key Enc. Signature

Ideal Delegation for PRAM with Privacy

New Primitive for Modular Design
CiO for PRAM
CiO for PRAM

OPRAM

PKE

Fully Succinct Rand. Encoding
for PRAM
CiO for PRAM

OPRAM
PKE

Fully Succinct
Rand. Encoding
for PRAM

Ideal Del.
for PRAM

NIZK
for PRAM

Functional Enc.
for PRAM

MPC
for PRAM

iO
for PRAM

24/9
CiO for PRAM

OPRAM

PKE

Fully Succinct

Rand. Encoding

for PRAM

[CHJV15, GHRW13, BGLPT15, KLW15]

Ideal Del.

for PRAM

NIZK

for PRAM

Functional Enc.

for PRAM

MPC

for PRAM

iO

for PRAM
CiO for PRAM

OPRAM

PKE

Fully Succinct Rand. Encoding for PRAM

Ideal Del. for PRAM

[NIZK, Functional Enc. for PRAM, iO for PRAM]

MPC for PRAM

Various Applications

[GHRW13, CHJV15, BGLPT15, KLW15]
Technical Challenges
CiO for TM/RAM

- generalization of [KLW]
CiO for TM/RAM

- generalization of [KLW]

CiO for PRAM

Technical Challenges
• CiO for TM/RAM
  – generalization of [KLW]
• CiO for PRAM
  – E.g. [KLW] \( \Rightarrow \) CiO for PRAM with size \( \Omega(#\text{CPU}) \)
    \( \therefore \) hybrids need to hardwire all CPU states
• CiO for TM/RAM
  – generalization of [KLW]
• CiO for PRAM
  – E.g. [KLW] \(\Rightarrow\) CiO for PRAM with size \(\Omega(#\text{CPU})\)
    ∴ hybrids need to hardwire all CPU states
CiO for **TM/RAM**

“Branch & Combine” PRAM emulation (hardwire $\log(#CPU)$)

• CiO for **TM/RAM**
  – generalization of [KLW]

• CiO for **PRAM**
  – E.g. [KLW] ⇒ CiO for PRAM with size $\Omega(#CPU)$
  ∴ hybrids need to hardwire all CPU states

CiO for **PRAM**

Fully Succinct Rand. Encoding for **PRAM**

Technical Challenges
CiO for **TM/RAM**

“Branch & Combine”
PRAM emulation
(hardware $\log(#CPU)$)

CiO for **PRAM**

• **CiO for TM/RAM**
  – generalization of [KLW]
• **CiO for PRAM**
  – E.g. [KLW] ⇒ CiO for PRAM
    with size $\Omega(#CPU)$
  ⊢ hybrids need to hardwire
    *all* CPU states
• Fully Succ. RE for PRAM
  – hiding by PKE & OPRAM
  – *yet, cannot* just use OPRAM
    ⊢ state, mem not hidden by iO
  – E.g. [CHJV]: “strong” ORAM via
    forward-in-time hybrids
  ⇒ space-dependent encoding

**Technical Challenges**
**CiO for TM/RAM**

“Branch & Combine”
PRAM emulation
(hardwire $\log(#CPU)$)

**CiO for PRAM**

- generalization of [KLW]
- E.g. [KLW] $\Rightarrow$ CiO for PRAM
  with size $\Omega(#CPU)$
    - hybrids need to hardwire
      all CPU states
- Fully Succ. RE for PRAM
  - hiding by PKE & OPRAM
  - *yet, cannot* just use OPRAM
    - state, mem not hidden by iO
  - E.g. [CHJV]: “strong” ORAM via forward-in-time hybrids
    $\Rightarrow$ space-dependent encoding

**Fully Succinct Rand. Encoding for PRAM**

- Puncturable oblivious PRAM
- back-in-time hybrids

---

**Technical Challenges**