Reducing Communication Channels in MPC

Marcel Keller [1,2] Dragos Rotaru [1,3]
Nigel Smart [1,3] **T**im Wood [1,3]

[1]University of Bristol

[2]Data61

[3]KU Leuven/COSIC ESAT

Outline

Outline

Goal

Generalising

MPC Tools

Performing MPC

What is MPC?

What is MPC?

P_7

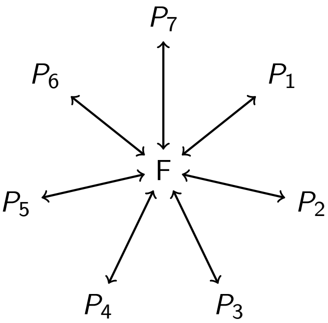
P_6 P_1

F

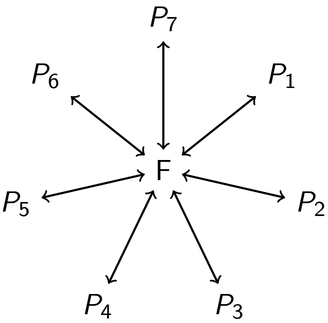
P_5 P_2

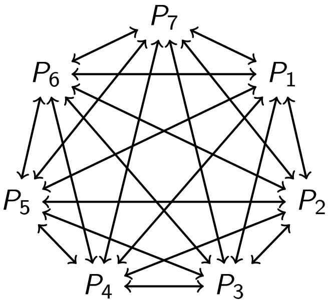
P_4 P_3

What is MPC?

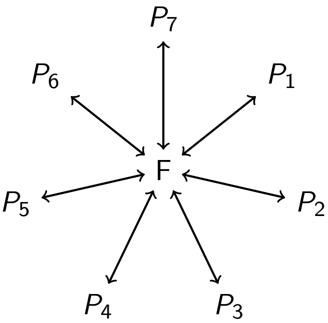


What is MPC?

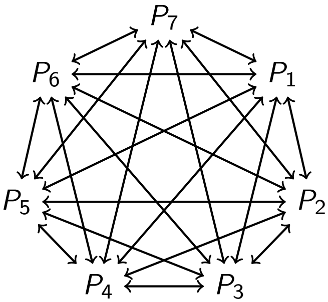




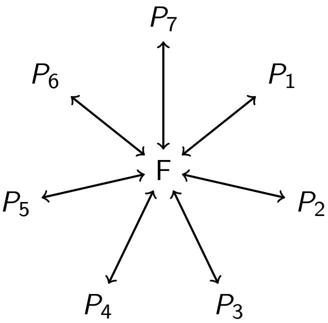
What is MPC?



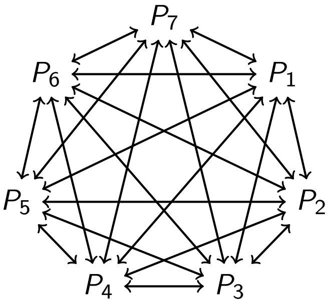
$\approx$



What is MPC?



$\approx$



Various guarantees:

Privacy/Secrecy

Correctness

Fairness

etc.

What is MPC?

Types:

- Garbled circuits
- Secret-sharing

What is MPC?

Types:

- Garbled circuits
- Secret-sharing

Examples:

- General MPC (e.g. SPDZ, MASCOT, Yao, etc.)
- PSI
- Auctions

What is MPC?

Types:

- Garbled circuits
- Secret-sharing

Examples:

- General MPC (e.g. SPDZ, MASCOT, Yao, etc.)
- PSI
- Auctions

Corruption Models:

- Active/Passive
- Static/Adaptive
- etc.

Goal

This work:

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

as part of overarching goal:

Efficient[1] MPC protocols for any access structure.

[1] communication/computation cost

Related Work

Previous best-known protocol was due to Maurer [Mau06]: passively-secure for $\mathcal{Q}_2$ structures, actively-secure for $\mathcal{Q}_3$.

[Mau06] Secure Multi-party Computation Made Simple, Journal of Discrete Applied Mathematics, 2006

Related Work

Previous best-known protocol was due to Maurer [Mau06]: passively-secure for $\mathcal{Q}_2$ structures, actively-secure for $\mathcal{Q}_3$.

Araki et al. [AFLNO16] give active security in the $(3, 1)$-threshold case with efficient "hash-check" authentication.

[Mau06] Secure Multi-party Computation Made Simple, Journal of Discrete Applied Mathematics, 2006

[AFLNO16] High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority, CCS 2016

Related Work

Previous best-known protocol was due to Maurer [Mau06]: passively-secure for $\mathcal{Q}_2$ structures, actively-secure for $\mathcal{Q}_3$.

Araki et al. [AFLNO16] give active security in the $(3, 1)$-threshold case with efficient "hash-check" authentication.

Our contribution:

Generalise to any $\mathcal{Q}_2$ access structure for any number of parties...

...and optimise the communication[2].

[Mau06] Secure Multi-party Computation Made Simple, Journal of Discrete Applied Mathematics, 2006

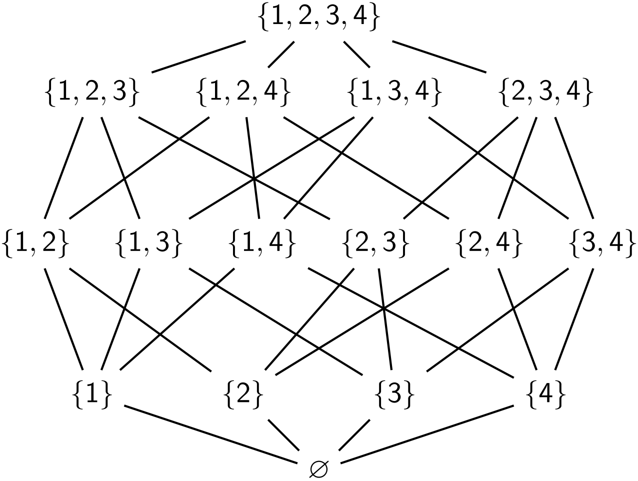
[AFLNO16] High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority, CCS 2016

[2] Asymptotics are hard to give because it depends on the access structure

Outline

Access Structures

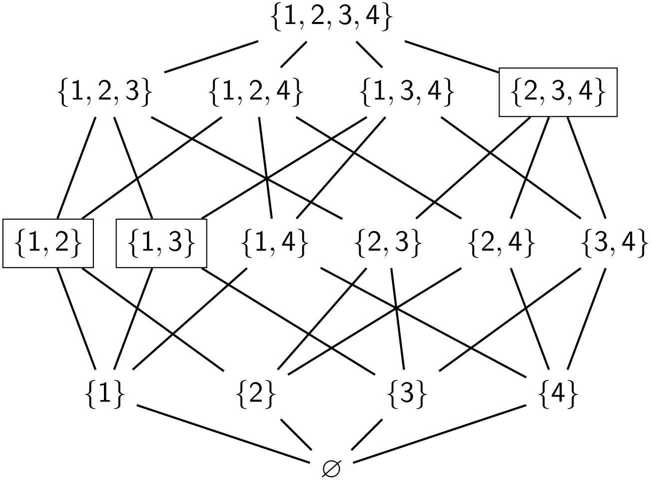
Definition by example



$\mathcal{Q}_2$: union of no two unqualified sets is $\{1,2,3,4\}$

Access Structures

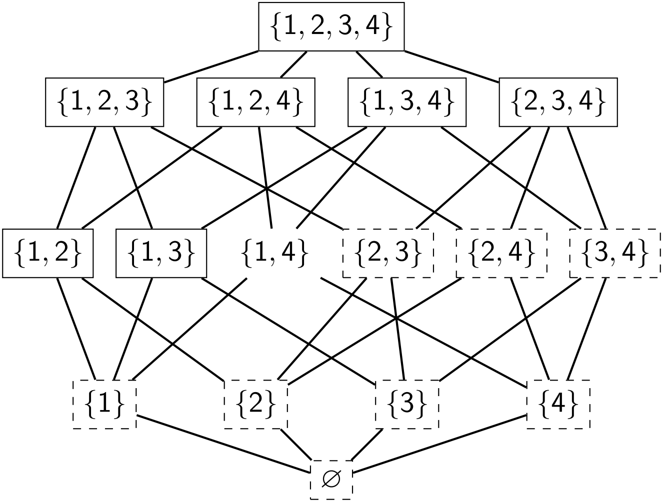
Specify minimally qualified sets



$\mathcal{Q}_2$: union of no two unqualified sets is $\{1,2,3,4\}$

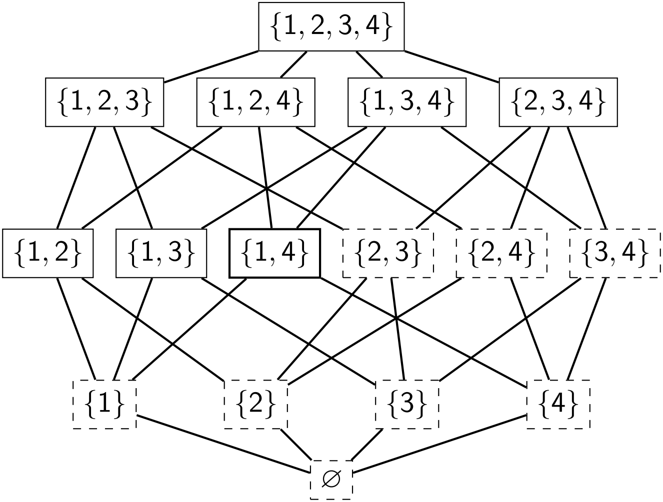
Access Structures

Check monotonicity



$\mathcal{Q}_2$: union of no two unqualified sets is $\{1,2,3,4\}$

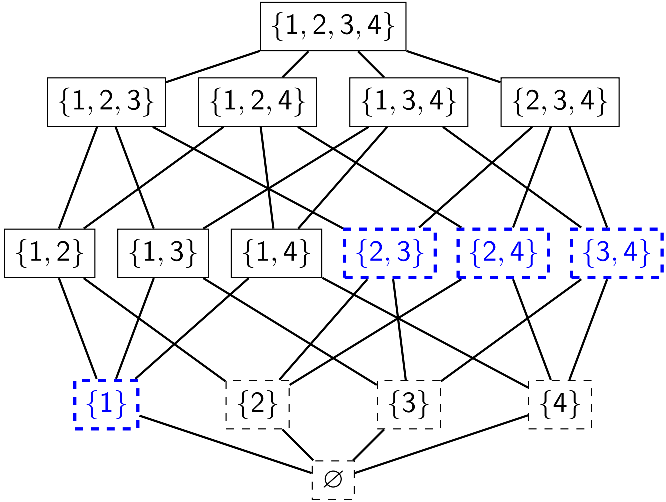
Decide on remaining sets



$\mathcal{Q}_2$: union of no two unqualified sets is $\{1, 2, 3, 4\}$

Access Structures

Determine maximally-unqualified sets



$\mathcal{Q}_2$: union of no two unqualified sets is $\{1,2,3,4\}$

Replicated Secret-sharing

Starting with the access structure

$$\Delta^+ = \{\{1\}, \{2,3\}, \{2,4\}, \{3,4\}\}$$

we obtain replicated secret sharing by taking the complements

$$\mathcal{B} = \{\{2,3,4\}, \{1,4\}, \{1,3\}, \{1,2\}\}$$

and sharing a secret s by letting

$$s = s_{\{2,3,4\}} + s_{\{1,4\}} + s_{\{1,3\}} + s_{\{1,2\}}$$

where $\{s_B\}_{B \in \mathcal{B}} \xleftarrow{\$} \mathbb{F}$ subject to $s = \sum_{B \in \mathcal{B}} s_B$.
Then s_B is sent to all parties whose party index is in B.

Denote by $[\![s]\!]$

Replicated Secret-sharing

$$s = s_{\{2,3,4\}} + s_{\{1,4\}} + s_{\{1,3\}} + s_{\{1,2\}}$$

Thus the parties have shares as follows:

$$\begin{array}{lllll}
P_1: & & s_{\{1,2\}} & s_{\{1,3\}} & s_{\{1,4\}} \\
P_2: & s_{\{2,3,4\}} & s_{\{1,2\}} & & \\
P_3: & s_{\{2,3,4\}} & & s_{\{1,3\}} & \\
P_4: & s_{\{2,3,4\}} & & & s_{\{1,4\}}
\end{array}$$

Linear operations for free

$$[\![s]\!] + [\![t]\!] :$$

	P_1			P_2		P_3		P_4	
$[\![s]\!]$	$s_{\{1,2\}}$	$s_{\{1,3\}}$	$s_{\{1,4\}}$	$s_{\{1,2\}}$	$s_{\{2,3,4\}}$	$s_{\{1,3\}}$	$s_{\{2,3,4\}}$	$s_{\{1,4\}}$	$s_{\{2,3,4\}}$
$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
$[\![t]\!]$	$t_{\{1,2\}}$	$t_{\{1,3\}}$	$t_{\{1,4\}}$	$t_{\{1,2\}}$	$t_{\{2,3,4\}}$	$t_{\{1,3\}}$	$t_{\{2,3,4\}}$	$t_{\{1,4\}}$	$t_{\{2,3,4\}}$
$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$	$\|\|$
$[\![u]\!]$	$u_{\{1,2\}}$	$u_{\{1,3\}}$	$u_{\{1,4\}}$	$u_{\{1,2\}}$	$u_{\{2,3,4\}}$	$u_{\{1,3\}}$	$u_{\{2,3,4\}}$	$u_{\{1,4\}}$	$u_{\{2,3,4\}}$

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- Additions
- Multiplications

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

✓ Additions: for free

– Multiplications: we will require

Tool 1: Passive multiplication

Tool 2: Efficient opening procedure

Outline

Tool 1: Passive Multiplication

Theorem [1]

If $\mathcal{Q}_2$, each cross term is computable by *at least one* party.

P_1, P_2, P_3, P_4 can compute an additive sharing of the product:

$$\begin{aligned} st = \; & s_{\{2,3,4\}} \cdot t_{\{2,3,4\}} + s_{\{2,3,4\}} \cdot t_{\{1,4\}} + s_{\{2,3,4\}} \cdot t_{\{1,3\}} + s_{\{2,3,4\}} \cdot t_{\{1,2\}} \\ & s_{\{1,4\}} \cdot t_{\{2,3,4\}} + s_{\{1,4\}} \cdot t_{\{1,4\}} + s_{\{1,4\}} \cdot t_{\{1,3\}} + s_{\{1,4\}} \cdot t_{\{1,2\}} \\ & s_{\{1,3\}} \cdot t_{\{2,3,4\}} + s_{\{1,3\}} \cdot t_{\{1,4\}} + s_{\{1,3\}} \cdot t_{\{1,3\}} + s_{\{1,3\}} \cdot t_{\{1,2\}} \\ & s_{\{1,2\}} \cdot t_{\{2,3,4\}} + s_{\{1,2\}} \cdot t_{\{1,4\}} + s_{\{1,2\}} \cdot t_{\{1,3\}} + s_{\{1,2\}} \cdot t_{\{1,2\}} \end{aligned}$$

$$M_1 \cup M_2 \subsetneq \mathcal{P} \quad \forall M_1, M_2 \in \Delta^+$$

$$\Longleftrightarrow$$

$$B_1 \cap B_2 \neq \varnothing \quad \forall B_1, B_2 \in \mathcal{B}$$

Tool 1: Passive Multiplication

Theorem [1]

If $\mathcal{Q}_2$, each cross term is computable by *at least one* party.

P_1, P_2, P_3, P_4 can compute an additive sharing of the product:

$$\begin{aligned} st = \; & s_{\{2,3,4\}} \cdot t_{\{2,3,4\}} + s_{\{2,3,4\}} \cdot t_{\{1,4\}} + s_{\{2,3,4\}} \cdot t_{\{1,3\}} + s_{\{2,3,4\}} \cdot t_{\{1,2\}} \\ & s_{\{1,4\}} \cdot t_{\{2,3,4\}} + s_{\{1,4\}} \cdot t_{\{1,4\}} + s_{\{1,4\}} \cdot t_{\{1,3\}} + s_{\{1,4\}} \cdot t_{\{1,2\}} \\ & s_{\{1,3\}} \cdot t_{\{2,3,4\}} + s_{\{1,3\}} \cdot t_{\{1,4\}} + s_{\{1,3\}} \cdot t_{\{1,3\}} + s_{\{1,3\}} \cdot t_{\{1,2\}} \\ & s_{\{1,2\}} \cdot t_{\{2,3,4\}} + s_{\{1,2\}} \cdot t_{\{1,4\}} + s_{\{1,2\}} \cdot t_{\{1,3\}} + s_{\{1,2\}} \cdot t_{\{1,2\}} \end{aligned}$$

E.g. P_2 computes

$$u^{(2)} := s_{\{2,3,4\}} \cdot t_{\{1,2\}} + s_{\{1,2\}} \cdot t_{\{2,3,4\}} + s_{\{1,2\}} \cdot t_{\{1,2\}}$$

Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

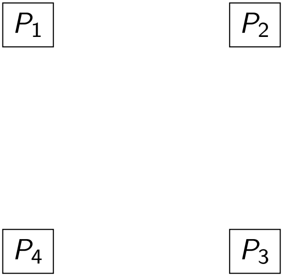
Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

E.g. P_1 additively splits $u^{(1)}$ as

$$u^{(1)} = u^{(1)}_{\{1,2\}} + u^{(1)}_{\{1,3\}} + u^{(1)}_{\{1,4\}} + u^{(1)}_{\{2,3,4\}}$$

and sends shares



Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

E.g. P_1 additively splits $u^{(1)}$ as

$$u^{(1)} = u^{(1)}_{\{1,2\}} + u^{(1)}_{\{1,3\}} + u^{(1)}_{\{1,4\}} + u^{(1)}_{\{2,3,4\}}$$

and sends shares

$$\begin{tikzcd} \boxed{P_1} \arrow[r, "u^{(1)}_{\{1,2\}}"] & \boxed{P_2} \\ \boxed{P_4} & \boxed{P_3} \end{tikzcd}$$

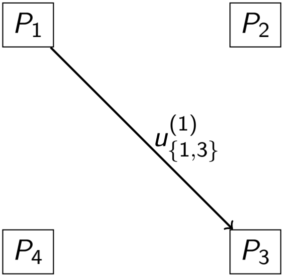
Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

E.g. P_1 additively splits $u^{(1)}$ as

$$u^{(1)} = u^{(1)}_{\{1,2\}} + u^{(1)}_{\{1,3\}} + u^{(1)}_{\{1,4\}} + u^{(1)}_{\{2,3,4\}}$$

and sends shares



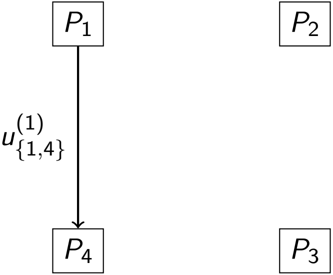
Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

E.g. P_1 additively splits $u^{(1)}$ as

$$u^{(1)} = u^{(1)}_{\{1,2\}} + u^{(1)}_{\{1,3\}} + u^{(1)}_{\{1,4\}} + u^{(1)}_{\{2,3,4\}}$$

and sends shares



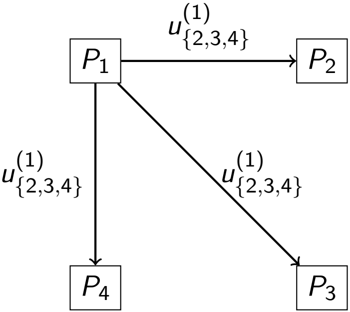
Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

E.g. P_1 additively splits $u^{(1)}$ as

$$u^{(1)} = u^{(1)}_{\{1,2\}} + u^{(1)}_{\{1,3\}} + u^{(1)}_{\{1,4\}} + u^{(1)}_{\{2,3,4\}}$$

and sends shares



Tool 1: Passive Multiplication – Maurer-style

Reshare each summand to get $[\![u^{(1)}]\!]$, $[\![u^{(2)}]\!]$, $[\![u^{(3)}]\!]$ and $[\![u^{(4)}]\!]$.

After all parties have reshared, sum shares locally:

$$[\![v]\!] := [\![u^{(1)}]\!] + [\![u^{(2)}]\!] + [\![u^{(3)}]\!] + [\![u^{(4)}]\!]$$

Tool 1: Passive Multiplication – Araki-style

Look for some assignment of sets in $\mathcal{B}$ to parties[3]:

$$\mathcal{B}_1 := \{\{1,4\}\}$$
$$\mathcal{B}_2 := \{\{1,2\}\}$$
$$\mathcal{B}_3 := \{\{1,3\}\}$$
$$\mathcal{B}_4 := \{\{2,3,4\}\}$$

such that

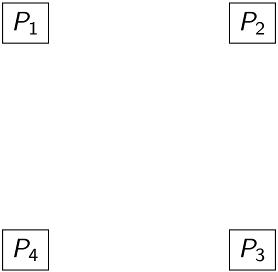
- every set assigned to P_i contains i
- every set is assigned to some party
- as many parties as possible are assigned at least one set

[3] Usually more sets than parties

Tool 1: Passive Multiplication – Araki-style

Recall a PRZS: $z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} = 0$, use it to mask the summands, and treat resulting shares as shares of the output.

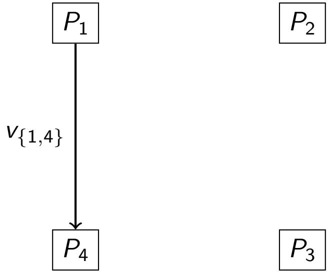
$$P_1 \text{ sets } v_{\{1,4\}} := u^{(1)} + z^{(1)} \text{ and sends to } P_4$$
$$P_2 \text{ sets } v_{\{1,2\}} := u^{(2)} + z^{(2)} \text{ and sends to } P_1$$
$$P_3 \text{ sets } v_{\{1,3\}} := u^{(3)} + z^{(3)} \text{ and sends to } P_1$$
$$P_4 \text{ sets } v_{\{2,3,4\}} := u^{(4)} + z^{(4)} \text{ and sends to } P_2 \text{ and } P_3$$



Tool 1: Passive Multiplication – Araki-style

Recall a PRZS: $z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} = 0$, use it to mask the summands, and treat resulting shares as shares of the output.

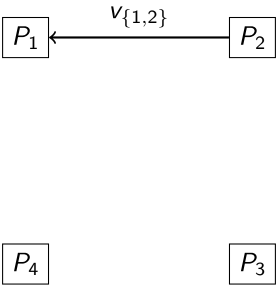
$$P_1 \text{ sets } v_{\{1,4\}} := u^{(1)} + z^{(1)} \text{ and sends to } P_4$$
$$P_2 \text{ sets } v_{\{1,2\}} := u^{(2)} + z^{(2)} \text{ and sends to } P_1$$
$$P_3 \text{ sets } v_{\{1,3\}} := u^{(3)} + z^{(3)} \text{ and sends to } P_1$$
$$P_4 \text{ sets } v_{\{2,3,4\}} := u^{(4)} + z^{(4)} \text{ and sends to } P_2 \text{ and } P_3$$



Tool 1: Passive Multiplication – Araki-style

Recall a PRZS: $z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} = 0$, use it to mask the summands, and treat resulting shares as shares of the output.

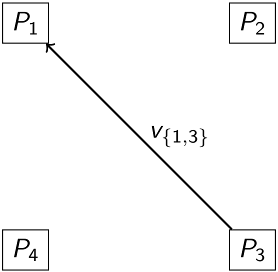
$$P_1 \text{ sets } v_{\{1,4\}} := u^{(1)} + z^{(1)} \text{ and sends to } P_4$$
$$P_2 \text{ sets } v_{\{1,2\}} := u^{(2)} + z^{(2)} \text{ and sends to } P_1$$
$$P_3 \text{ sets } v_{\{1,3\}} := u^{(3)} + z^{(3)} \text{ and sends to } P_1$$
$$P_4 \text{ sets } v_{\{2,3,4\}} := u^{(4)} + z^{(4)} \text{ and sends to } P_2 \text{ and } P_3$$



Tool 1: Passive Multiplication – Araki-style

Recall a PRZS: $z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} = 0$, use it to mask the summands, and treat resulting shares as shares of the output.

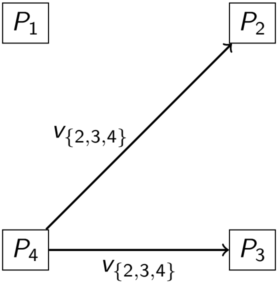
$$P_1 \text{ sets } v_{\{1,4\}} := u^{(1)} + z^{(1)} \text{ and sends to } P_4$$
$$P_2 \text{ sets } v_{\{1,2\}} := u^{(2)} + z^{(2)} \text{ and sends to } P_1$$
$$P_3 \text{ sets } v_{\{1,3\}} := u^{(3)} + z^{(3)} \text{ and sends to } P_1$$
$$P_4 \text{ sets } v_{\{2,3,4\}} := u^{(4)} + z^{(4)} \text{ and sends to } P_2 \text{ and } P_3$$



Tool 1: Passive Multiplication – Araki-style

Recall a PRZS: $z^{(1)} + z^{(2)} + z^{(3)} + z^{(4)} = 0$, use it to mask the summands, and treat resulting shares as shares of the output.

$$\begin{aligned}
P_1 \text{ sets } v_{\{1,4\}} &:= u^{(1)} + z^{(1)} \text{ and sends to } P_4 \\
P_2 \text{ sets } v_{\{1,2\}} &:= u^{(2)} + z^{(2)} \text{ and sends to } P_1 \\
P_3 \text{ sets } v_{\{1,3\}} &:= u^{(3)} + z^{(3)} \text{ and sends to } P_1 \\
P_4 \text{ sets } v_{\{2,3,4\}} &:= u^{(4)} + z^{(4)} \text{ and sends to } P_2 \text{ and } P_3
\end{aligned}$$



Tool 1: Passive Multiplication – Araki-style

No further local computation (addition) needed: parties hold $[\![v]\!]$.

Notice

- Not all parties communicate with each other;
- Total number of field elements sent is less than Maurer.

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications: we will require

  Tool 1: Passive multiplication

  Tool 2: Efficient opening procedure

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

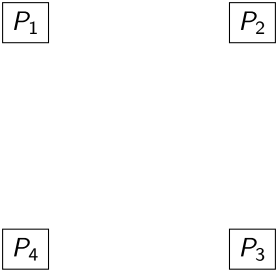
Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications: we will require
  - ✓ Tool 1: Passive multiplication – Araki-style
  - Tool 2: Efficient opening procedure

Tool 2: Opening – Maurer-style

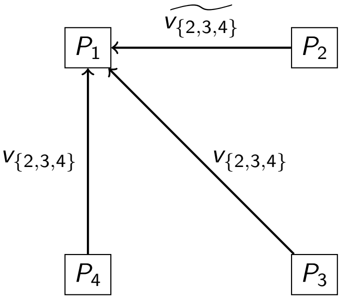
Every party broadcasts all of their shares.

Active security: every share is held by at least one honest party.



Every party broadcasts all of their shares.

Active security: every share is held by at least one honest party.

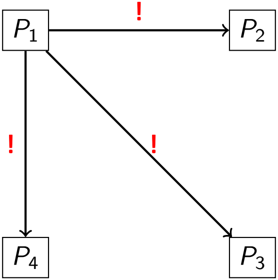


Tool 2: Opening – Maurer-style

Every party broadcasts all of their shares.

Active security: every share is held by at least one honest party.

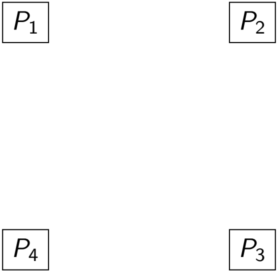
$\widetilde{v_{\{2,3,4\}}} \neq v_{\{2,3,4\}}$



Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:

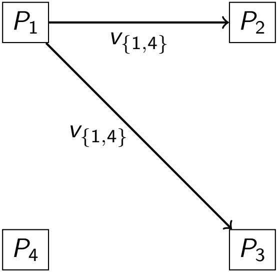


Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:

$\mathcal{B}_1 = \{\{1,4\}\}$

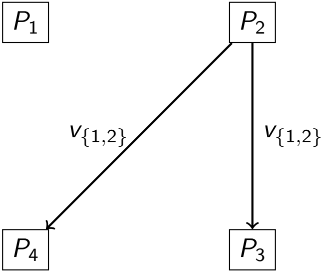


Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:

$\mathcal{B}_2 = \{\{1,2\}\}$

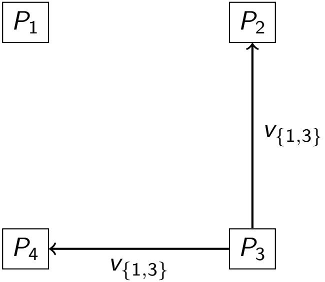


Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:

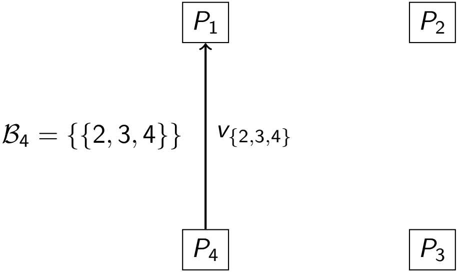
$\mathcal{B}_3 = \{\{1, 3\}\}$



Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:



Tool 2: Opening – Araki-style

Use the assignment of sets to parties:

Party in charge of a share sends to all who do not hold it:

Active security: Update hash function locally – all parties' hashes should agree:

$$P_1 \text{ computes } h_1 := H(..., v_{\{1,2\}}, v_{\{1,3\}}, v_{\{1,4\}}, \widetilde{v_{\{2,3,4\}}}, ...)$$
$$P_2 \text{ computes } h_2 := H(..., v_{\{1,2\}}, \widetilde{v_{\{1,3\}}}, \widetilde{v_{\{1,4\}}}, v_{\{2,3,4\}}, ...)$$
$$P_3 \text{ computes } h_3 := H(..., \widetilde{v_{\{1,2\}}}, v_{\{1,3\}}, \widetilde{v_{\{1,4\}}}, v_{\{2,3,4\}}, ...)$$
$$P_4 \text{ computes } h_4 := H(..., \widetilde{v_{\{1,2\}}}, \widetilde{v_{\{1,3\}}}, v_{\{1,4\}}, v_{\{2,3,4\}}, ...)$$

Batch-check to save on communication cost.

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

✓ Additions: for free

– Multiplications: we will require

✓ Tool 1: Passive multiplication – Araki-style

Tool 2: Efficient opening procedure

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications: we will require
  - ✓ Tool 1: Passive multiplication – Araki-style
  - ✓ Tool 2: Efficient opening procedure – using hashing

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications: we will require
  - ✓ Tool 1: Passive multiplication – Araki-style
  - ✓ Tool 2: Efficient opening procedure – using hashing

Now to do the actual multiplication...

Outline

Pre-processing Model

Offline/Online paradigm using Beaver's circuit randomisation:

Multiply $[\![x]\!]$ and $[\![y]\!]$ *online* given a "triple" $([\![a]\!], [\![b]\!], [\![ab]\!])$ from *offline*

$$[\![xy]\!] = (x+a)[\![y]\!] + (y+b)[\![x]\!] + [\![ab]\!] - (x+a)(y+b)[\![1]\!]$$

where

- $(x+a)$ and $(y+b)$ are opened secrets (i.e. use **Tool 2**: Opening on $[\![x]\!] + [\![a]\!]$ and $[\![y]\!] + [\![b]\!]$)
- $[\![1]\!]$ is any valid sharing of the value 1

$\rightsquigarrow$ **Offline phase: generate lots of random triples**

Generating Triples: 1. Generate random values

One-time key agreement: parties in each $B \in \mathcal{B}$ agree on a key.

Then for each $B \in \mathcal{B}$, compute $a_B := F_{k_B}(\text{count})$ to obtain $[\![a]\!]$.

$$\begin{aligned} a_{\{1,2\}} &:= F_{k_{\{1,2\}}}(\text{count}) \\ a_{\{1,3\}} &:= F_{k_{\{1,3\}}}(\text{count}) \\ a_{\{1,4\}} &:= F_{k_{\{1,4\}}}(\text{count}) \\ a_{\{2,3,4\}} &:= F_{k_{\{2,3,4\}}}(\text{count}) \end{aligned}$$

All parties increment count and then compute the shares as before: $b_B := F_{k_B}(\text{count})$; the parties obtain $[\![b]\!]$.

Generating Triples: 2. Passively multiply

Tool 1: Passive Multiplication

$$[\![ab]\!] := [\![a]\!] \cdot [\![b]\!]$$

Generating Triples: 3. Sacrifice for active security

Generate two triples,

$$([\![a]\!], [\![b]\!], [\![ab]\!]) \text{ and } ([\![a']\!], [\![b']\!], [\![a'b']\!])$$

Now use $([\![a']\!], [\![b']\!], [\![a'b']\!])$ to check that

$$[\![a]\!] \cdot [\![b]\!] - [\![ab]\!] = 0$$

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- – Multiplications: we will require
  - ✓ Tool 1: Passive multiplication – Araki-style
  - ✓ Tool 2: Efficient opening procedure – using hashing

Goal

Goal

Communication-efficient actively-secure MPC arithmetic circuit evaluation for any $\mathcal{Q}_2$ access structure.

Arithmetic circuits:

- ✓ Additions: for free
- ✓ Multiplications: we will require
  - ✓ Tool 1: Passive multiplication – Araki-style
  - ✓ Tool 2: Efficient opening procedure – using hashing

Costs

Comparison for a threshold access structure:

Tool 1: Passive Multiplication

	Maurer-style	Ours
# Channels[4]	$n \cdot (n-1)$	$n \cdot (n-t-1)$
# Field elements	$n \cdot \binom{n}{t}$	$n \cdot (n-t-1)$

Tool 2: Opening

	Maurer-style	Ours
# Channels[4]	$n \cdot (n-1)$	$\frac{1}{2} \cdot n \cdot (n-1)$
# Field elements	$n \cdot \binom{n}{t}$	$t \cdot \binom{n}{t}$

[4] Uni-directional

Implementation

https://github.com/KULeuven-COSIC/SCALE-MAMBA

Thanks!

Questions?

$|\Delta^+| > n$?

If the number of replicated shares exceeds the number of parties: e.g. $(5, 2)$-threshold:

$$\begin{aligned}\Delta^+ := \{&\{1,2\},\{1,3\},\{1,4\},\{1,5\},\{2,3\},\\ &\{2,4\},\{2,5\},\{3,4\},\{3,5\},\{4,5\}\}\end{aligned}$$

gives

$$\begin{aligned}\mathcal{B} = \{&\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\},\{1,3,5\},\\ &\{1,4,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\},\{3,4,5\}\}\end{aligned}$$

Assignment as before: e.g.

$$\begin{aligned}\mathcal{B}_1 &:= \{\{1,2,3\},\{1,2,4\}\} & \mathcal{B}_4 &:= \{\{1,4,5\},\{2,4,5\}\}\\ \mathcal{B}_2 &:= \{\{2,3,4\},\{2,3,5\}\} & \mathcal{B}_5 &:= \{\{1,2,5\},\{1,3,5\}\}\\ \mathcal{B}_3 &:= \{\{3,4,5\},\{1,3,4\}\} & &\end{aligned}$$

Optimisation using pre-shared keys

Share index

	123	124	125	134	135	145	234	235	245	345
P_1	$u^{(1)} - r_1$	r_1		r_3	r_5					
P_2		r_1					$u^{(2)} - r_2$	r_2	r_4	
P_3				r_3	r_5			r_2		$u^{(3)} - r_3$
P_4		r_1		r_3		$u^{(4)} - r_4$			r_4	
P_5			$u^{(5)} - r_5$		r_5			r_2	r_4	

$r_1 := F_{k_{\{1,2,4\}}}(\text{count})$

$r_2 := F_{k_{\{2,3,5\}}}(\text{count})$

$r_3 := F_{k_{\{1,3,4\}}}(\text{count})$

$r_4 := F_{k_{\{2,4,5\}}}(\text{count})$

$r_5 := F_{k_{\{1,3,5\}}}(\text{count})$