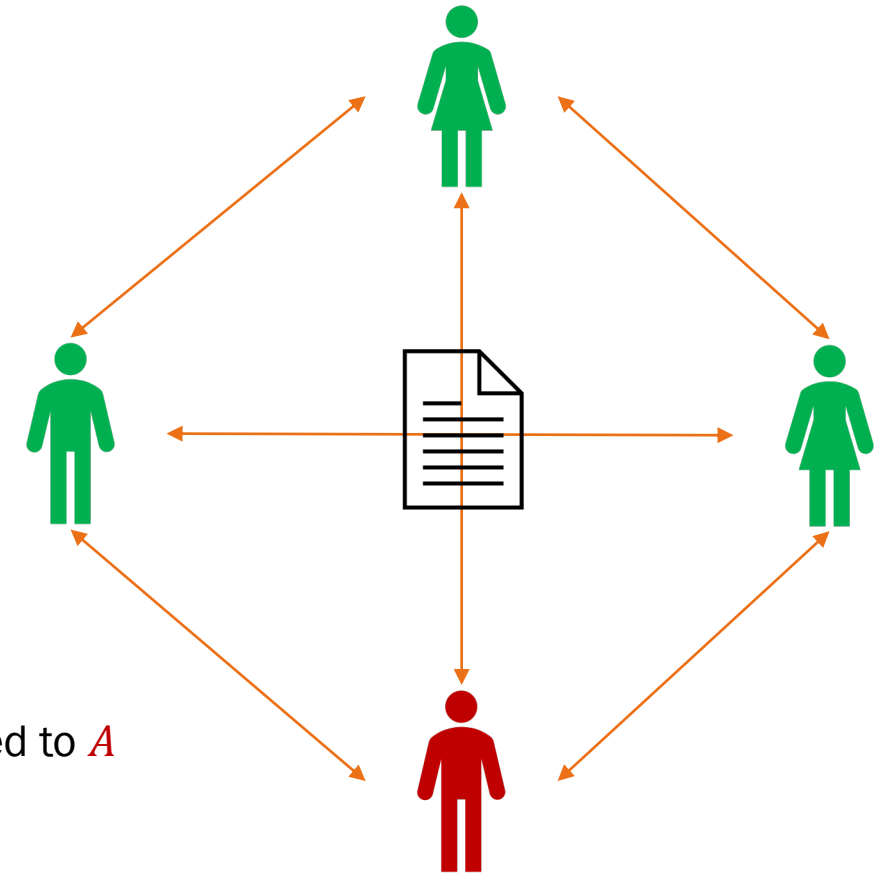
Attaining GOD Beyond Honest Majority With Friends and Foes

Aditya Hegde, Nishat Koti, Varsha Bhat Kukkala, **Shravani Patil**, Arpita Patra, Protik Paul

Secure Multiparty Computation

- n mutually distrusting parties $P_1, P_2, \ldots, P_n$
- P_i has private input x_i
- n-input function f
- t corrupted by a (centralized) adversary A

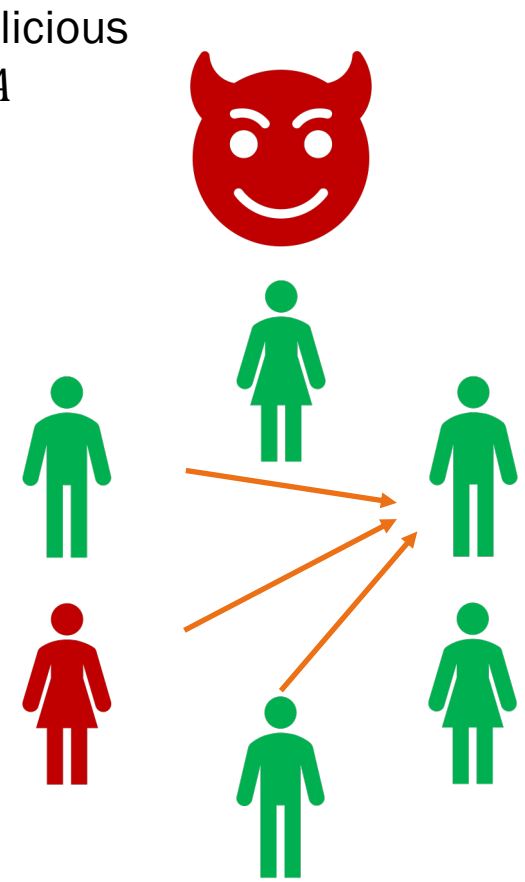


Goals:

- Correctness: Compute $y = f(x_1, x_2, \ldots, x_n)$
- Privacy: Nothing beyond the function output is revealed to A

View Leakage

- Loophole in the classical security definition:
  - Adversary can leak information to honest parties
  - Protocol might require leaking info to honest parties → GOD via TTP!



Alon, Bar, Eran Omri, and Anat Paskin-Cherniavsky. "MPC with friends and foes." *Annual International Cryptology Conference*, 2020.

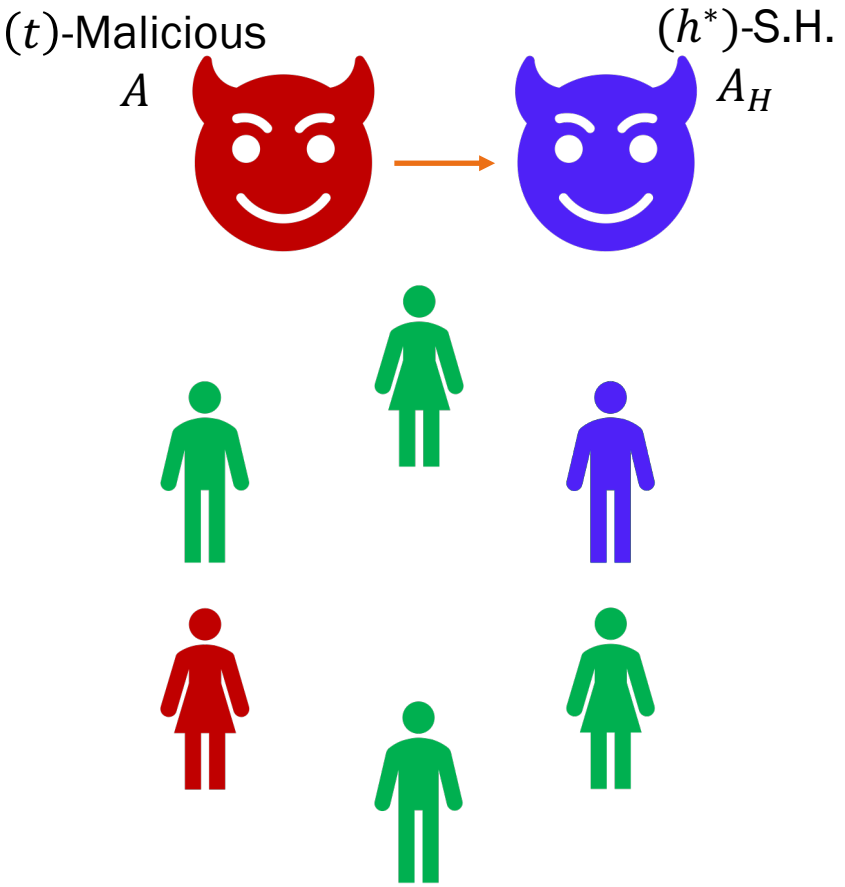
Friends and Foes (FaF)

(t, h^*)-FaF security

- t maliciously corrupted parties (Foes)
- h^* semi-honest parties (Friends)

$$\text{View}_{😈}^{\text{Ideal}} \approx \text{View}_{😈}^{\text{Real}}$$

$$\text{View}_{😈}^{\text{Ideal}} \approx \text{View}_{😈}^{\text{Real}}$$



Alon, Bar, Eran Omri, and Anat Paskin-Cherniavsky. "MPC with friends and foes." *Annual International Cryptology Conference*, 2020.

Our Results

- Theoretical
  - Necessity of Oblivious Transfer (OT) for generic (t, h^*)-FaF secure protocol with $n \leq 2t + 2h^*$

- Practical
  - QuadSquad: Robust/Fair $(1,1)$-FaF secure 4PC
    - Optimal corruption threshold

Any n-party functionality f can be computed with robust/fair computational (t, h^*)-FaF security, iff $n > 2t + h^*$.

Our Results

- Theoretical
  - Necessity of Oblivious Transfer (OT) for generic (t, h^*)-FaF secure protocol with $n \leq 2t + 2h^*$

- Practical
  - QuadSquad: Robust/Fair $(1,1)$-FaF secure 4PC
    - Optimal corruption threshold
    - Preprocessing paradigm
    - Operates over rings
  - Application in Privacy-Preserving Machine Learning (PPML)

Preprocessing Phase
- Input independent
- Computationally intensive tasks

Online Phase
- Input dependent

Necessity of OT

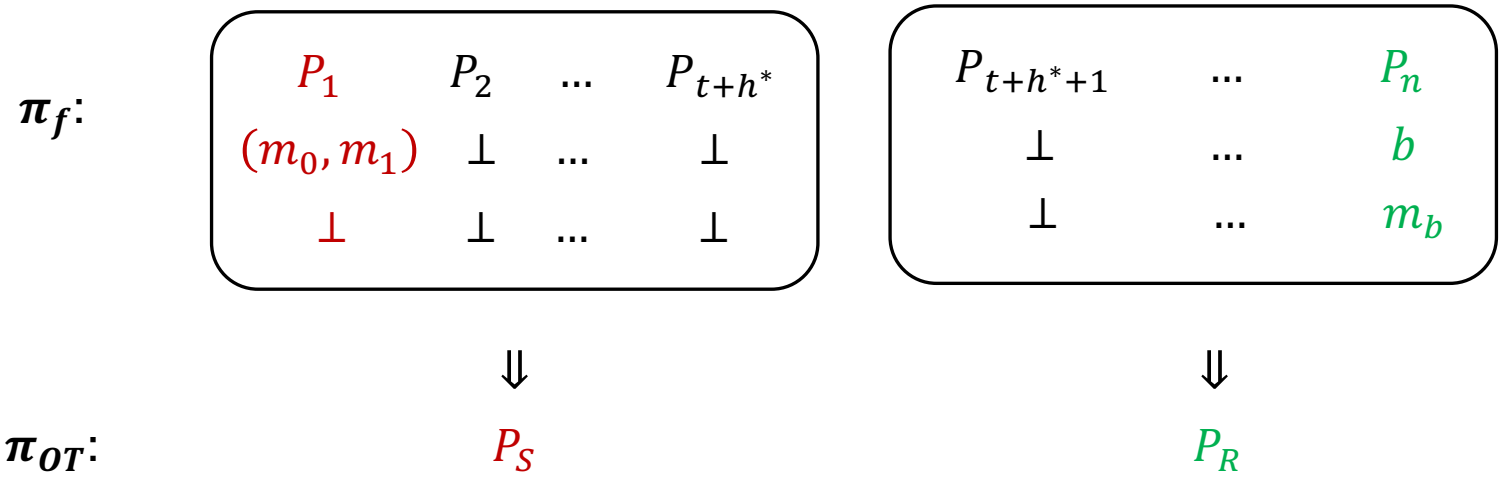
$\boldsymbol{\pi_f}$: (t, h^*)-FaF Protocol

$f(x_1, x_2, \dots, x_n) = (y_1, y_2, \dots, y_n)$

$\Rightarrow$

$\boldsymbol{\pi_{OT}}$: 2-party OT Protocol

$f\big((m_0, m_1), b\big) = (\perp, m_b)$



Our Results

- Theoretical
  - Necessity of Oblivious Transfer (OT) for generic (t, h^*)-FaF secure protocol with $n \leq 2t + 2h^*$

- Practical
  - QuadSquad: Robust/Fair $(1,1)$-FaF secure 4PC
    - Optimal corruption threshold
    - Preprocessing paradigm
    - Operates over rings
  - Application in Privacy-Preserving Machine Learning (PPML)

Comparison with existing work

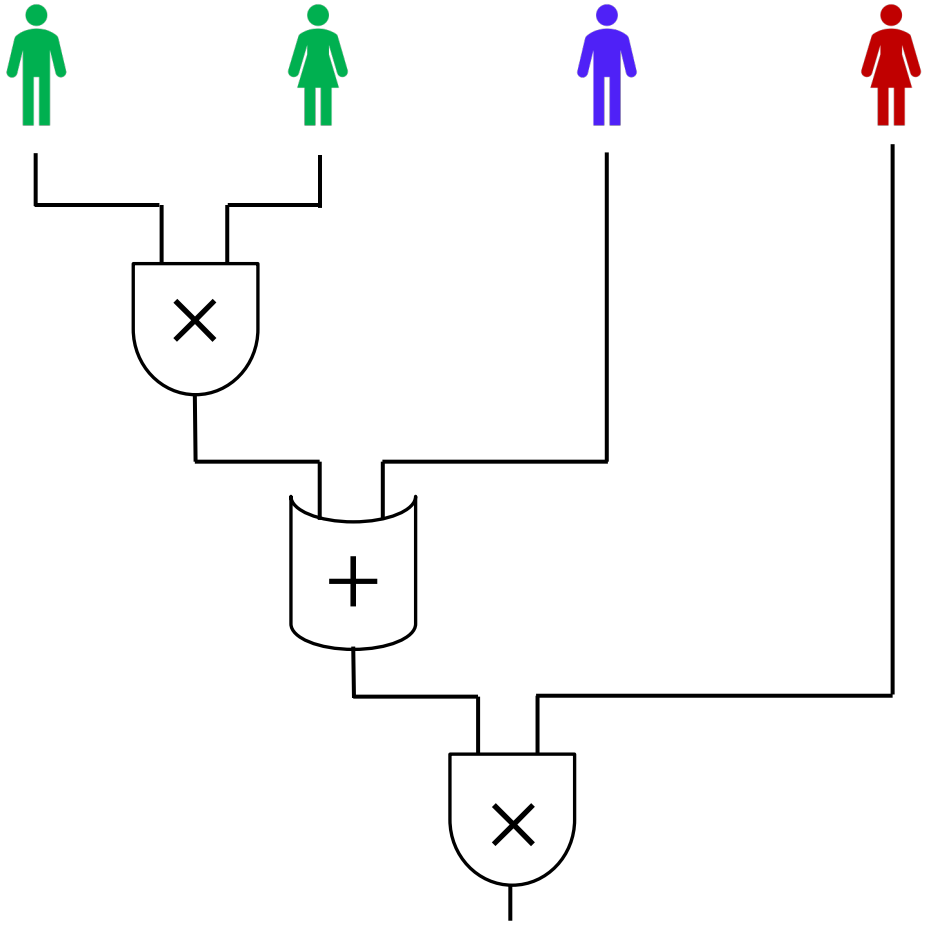
Ref.	Preprocessing Comm.	Online		Model	Security
		Rounds	Comm.		
Tetrad (R)	2	1	3	Honest Majority ($t = 1$)	GOD
Fantastic Four (R)	NA	1	6	Honest Majority ($t = 1$)	GOD
MASCOT (F)	7713	2	12	Dishonest Majority ($t = 3$)	Abort
QuadSquad (R)	**1558**	**3**	**7**	**FaF** ($t = 1, h^* = 1$)	**Fair**
QuadSqaud (R)	**3110**	**3**	**7**	**FaF** ($t = 1, h^* = 1$)	**GOD**

^ The communication cost is reported in terms of field/ring elements.

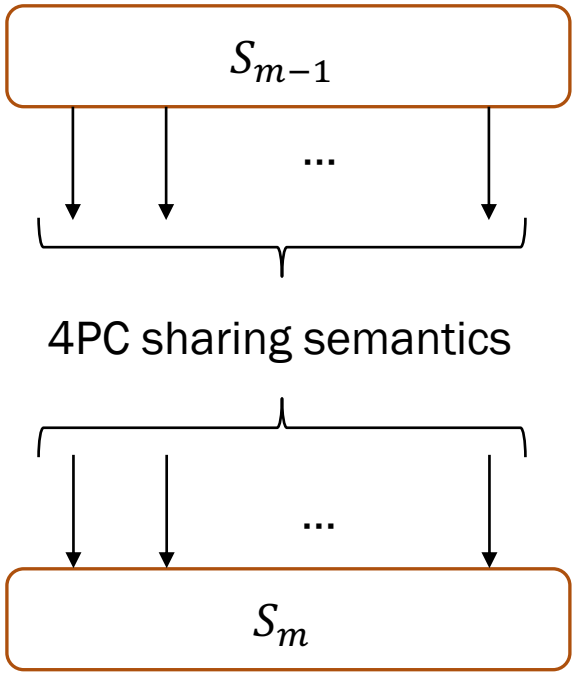
* Koti, Nishat, et al. "Tetrad: actively secure 4pc for secure training and inference." *arXiv preprint arXiv:2106.02850*(2021).
* Dalskov, Anders, Daniel Escudero, and Marcel Keller. "Fantastic Four: Honest-Majority Four-Party Secure Computation With Malicious Security." *USENIX Security Symposium,* 2021.
* Keller, Marcel, Emmanuela Orsini, and Peter Scholl. "MASCOT: faster malicious arithmetic secure computation with oblivious transfer." *ACM CCS,* 2016.

Approach

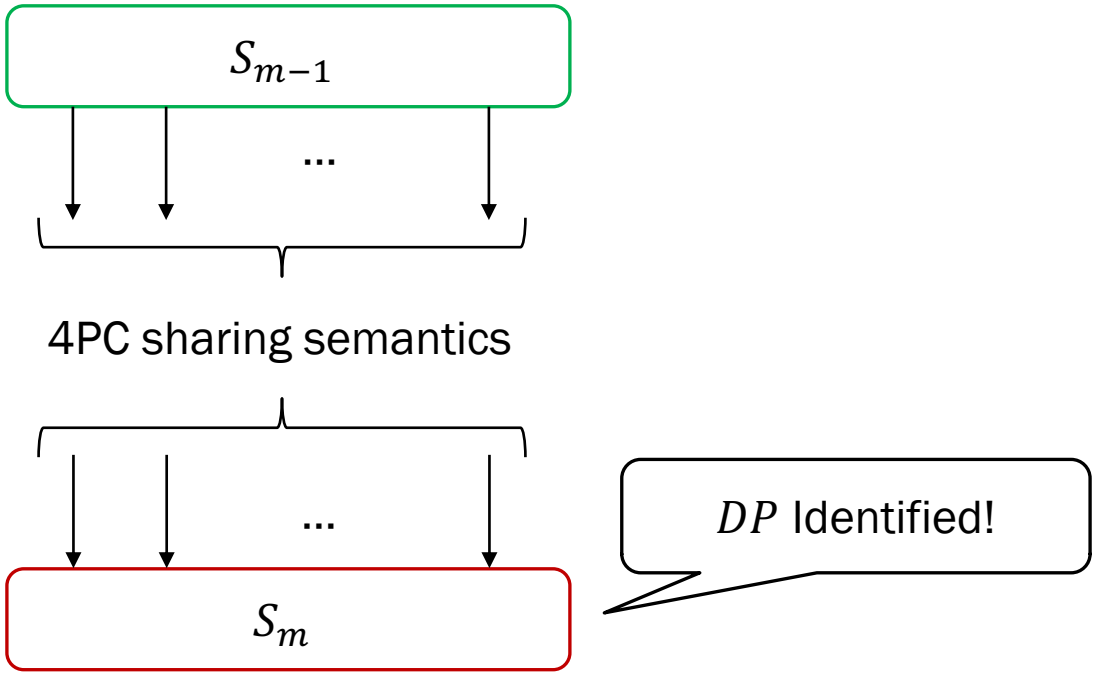
- Function expressed as a circuit
- Evaluated in a topological order
  - Input sharing
  - Evaluation
  - Output reconstruction
- High-level idea for sub-protocols
  - Completes successfully, OR
  - Identifies a dispute pair DP (includes corrupt party)
  - Run semi-honest 2PC (for GOD)
- Segmented evaluation to get GOD for free



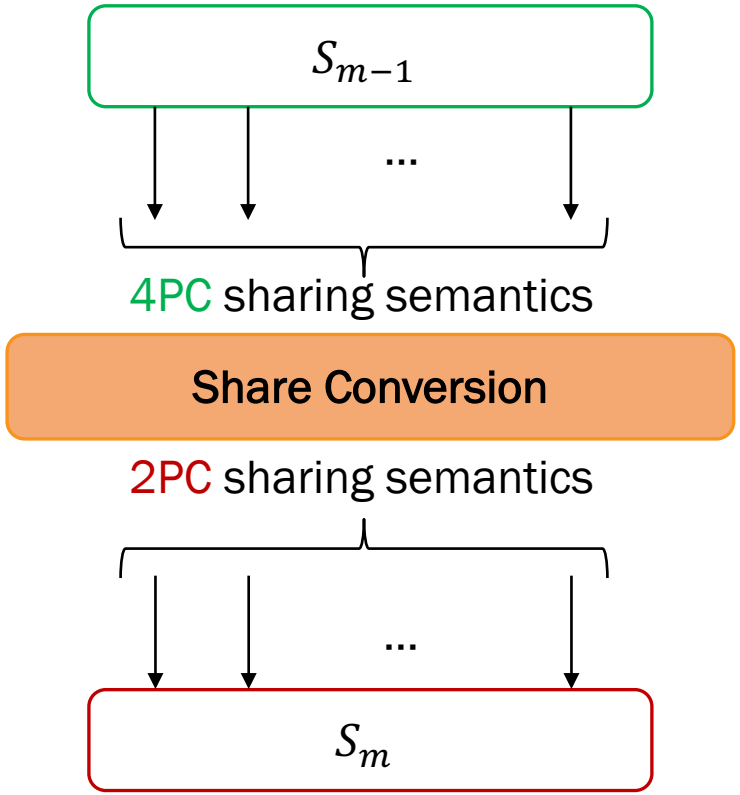
Fair to GOD: Segmentation



Fair to GOD: Segmentation



Fair to GOD: Segmentation



Sharing Semantics

(n, t) Replicated Secret Sharing (RSS)
- Total number of shares = $\binom{n}{t}$
- Shares per party = $\binom{n-1}{t}$

- QuadSquad
  - $(4,2)$ RSS: Accounts for $t = 1, h^* = 1$
  - $s = \langle s \rangle_{12} + \langle s \rangle_{13} + \langle s \rangle_{14} + \langle s \rangle_{23} + \langle s \rangle_{24} + \langle s \rangle_{34}$

$(\langle s \rangle_{12}, \langle s \rangle_{13}, \langle s \rangle_{14})$

$(\langle s \rangle_{12}, \langle s \rangle_{23}, \langle s \rangle_{24})$

- Contrast with H.M./D.M. Setting
  - Honest Majority: $(4,1)$-RSS $\Longrightarrow$ 4 shares per secret, High Redundancy
  - Dishonest Majority: $(4,3)$-RSS $\Longrightarrow$ 4 shares per secret, Low redundancy

Sharing Semantics

$(\beta_x, \langle\alpha_x\rangle_{12}, \langle\alpha_x\rangle_{23}, \langle\alpha_x\rangle_{24})$

$(\beta_x, \langle\alpha_x\rangle_{12}, \langle\alpha_x\rangle_{13}, \langle\alpha_x\rangle_{14})$

$(\beta_x, \langle\alpha_x\rangle_{13}, \langle\alpha_x\rangle_{23}, \langle\alpha_x\rangle_{34})$

$(\beta_x, \langle\alpha_x\rangle_{14}, \langle\alpha_x\rangle_{24}, \langle\alpha_x\rangle_{34})$

$[\![\cdot]\!]$-sharing of a value x:

- $\langle\cdot\rangle$-sharing of random α_x
- Masked value $\beta_x = x + \alpha_x$

Can be viewed as $\langle\cdot\rangle$-sharing of x:

$$x = \beta_x - \alpha_x$$

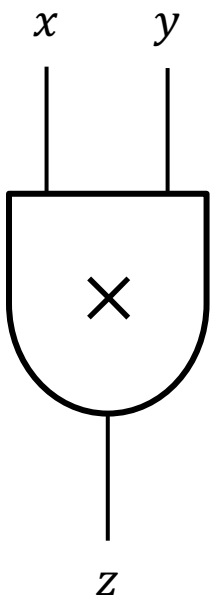
$$\langle x\rangle = \beta_x - \langle\alpha_x\rangle$$

Multiplication

- Given $[\![x]\!] = (\beta_x, \langle \alpha_x \rangle)$ and $[\![y]\!] = (\beta_y, \langle \alpha_y \rangle)$
- Compute $[\![z]\!] = (\beta_z, \langle \alpha_z \rangle)$

$$\begin{aligned}\beta_z &= z + \alpha_z \\ &= xy + \alpha_z \\ &= (\beta_x - \alpha_x)(\beta_y - \alpha_y) + \alpha_z \\ &= \beta_x\beta_y - \beta_x\alpha_y - \beta_y\alpha_x + \alpha_x\alpha_y + \alpha_z\end{aligned}$$

- Can be reduced to 1 $\langle \cdot \rangle$-reconstruction (online)

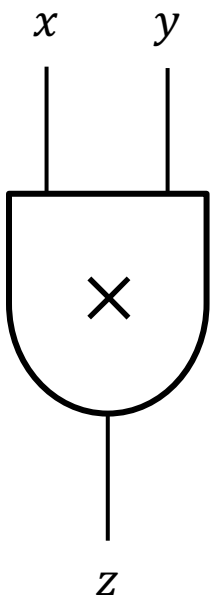


Multiplication

- Given $[\![x]\!] = (\beta_x, \langle \alpha_x \rangle)$ and $[\![y]\!] = (\beta_y, \langle \alpha_y \rangle)$
- Compute $[\![z]\!] = (\beta_z, \langle \alpha_z \rangle)$

$$\begin{aligned}\langle \beta_z \rangle &= z + \alpha_z \\ &= xy + \alpha_z \\ &= (\beta_x - \alpha_x)(\beta_y - \alpha_y) + \alpha_z \\ &= \langle \beta_x \beta_y \rangle - \langle \beta_x \alpha_y \rangle - \langle \beta_y \alpha_x \rangle + \langle \alpha_x \alpha_y \rangle + \langle \alpha_z \rangle\end{aligned}$$

- Can be reduced to 1 $\langle \cdot \rangle$-reconstruction (online)

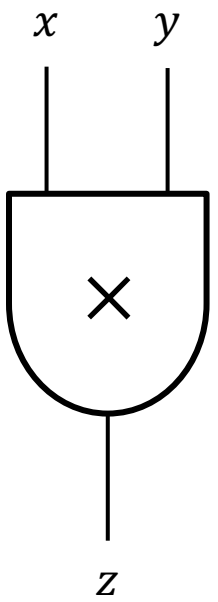


Multiplication

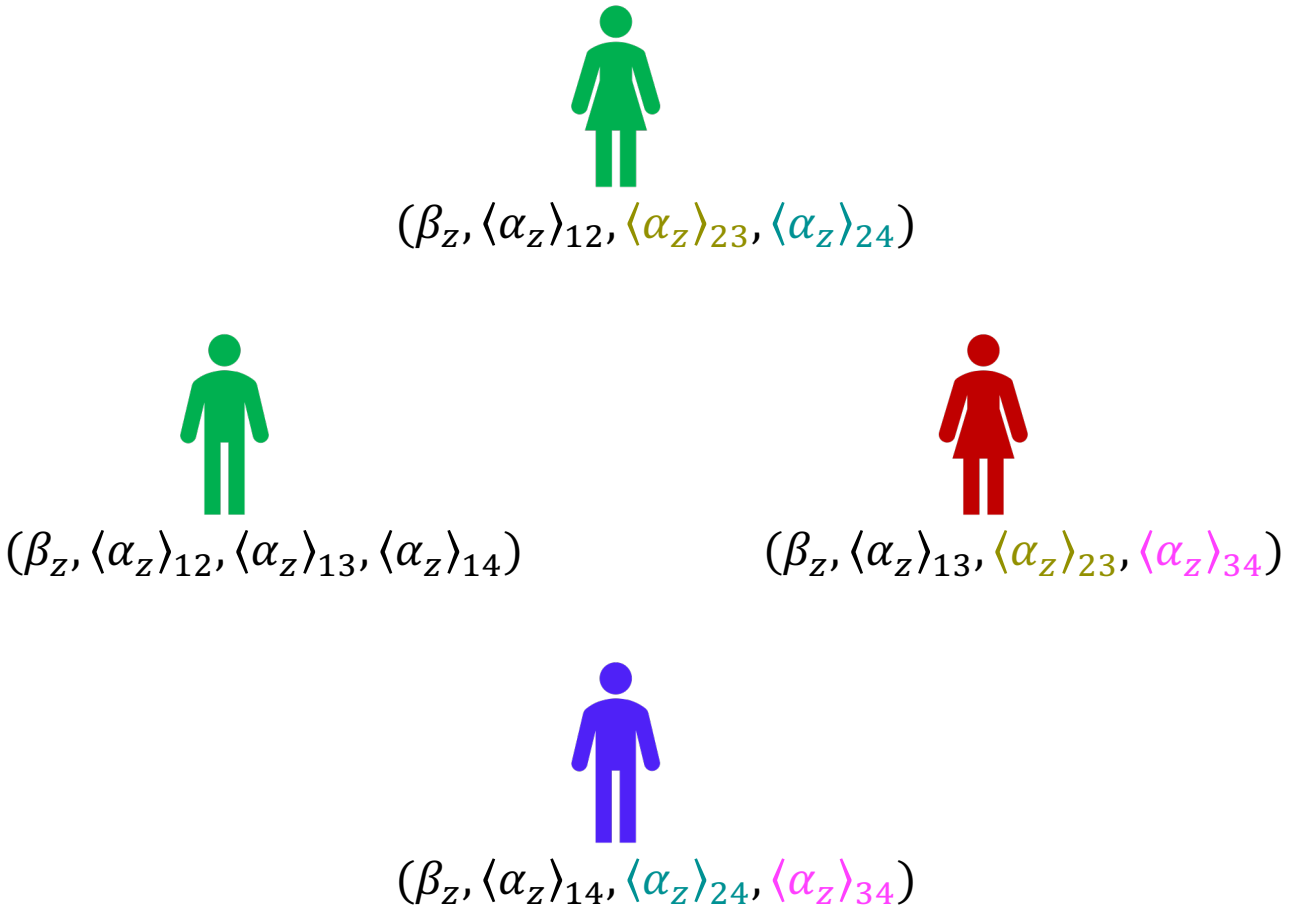
- Given $[\![x]\!] = (\beta_x, \langle \alpha_x \rangle)$ and $[\![y]\!] = (\beta_y, \langle \alpha_y \rangle)$
- Compute $[\![z]\!] = (\beta_z, \langle \alpha_z \rangle)$

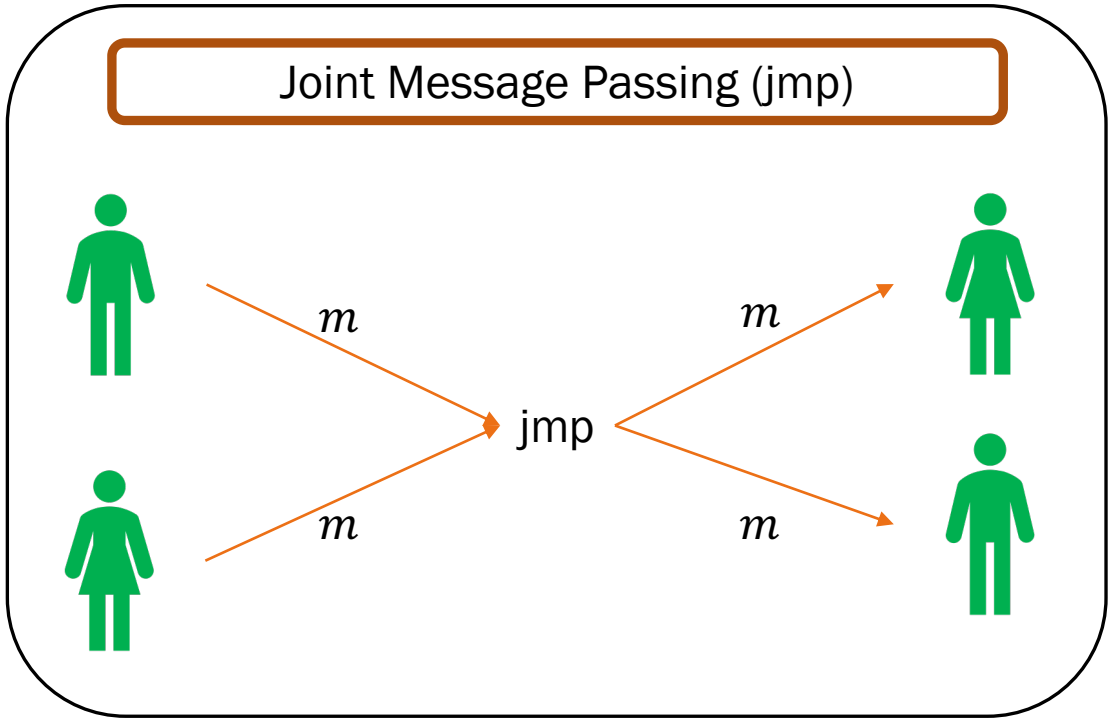
$$\begin{aligned}\langle \beta_z \rangle &= z + \alpha_z \\ &= xy + \alpha_z \\ &= (\beta_x - \alpha_x)(\beta_y - \alpha_y) + \alpha_z \\ &= \langle \beta_x \beta_y \rangle - \langle \beta_x \alpha_y \rangle - \langle \beta_y \alpha_x \rangle + \langle \alpha_x \alpha_y \rangle + \langle \alpha_z \rangle\end{aligned}$$

- Can be reduced to 1 $\langle \cdot \rangle$-reconstruction (online)
- Reduced to obtaining $\langle \alpha_x \alpha_y \rangle$ from $\langle \alpha_x \rangle$ and $\langle \alpha_y \rangle$ (preprocessing)



Reconstruction Phase





Koti, N., Pancholi, M., Patra, A., & Suresh, A., "SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning," *30th USENIX Security Symposium,* 2021.

Reconstruction Phase

$(\beta_z, \langle\alpha_z\rangle_{12}, \langle\alpha_z\rangle_{23}, \langle\alpha_z\rangle_{24})$

$(\beta_z, \langle\alpha_z\rangle_{12}, \langle\alpha_z\rangle_{13}, \langle\alpha_z\rangle_{14})$

$(\beta_z, \langle\alpha_z\rangle_{13}, \langle\alpha_z\rangle_{23}, \langle\alpha_z\rangle_{34})$

$(\beta_z, \langle\alpha_z\rangle_{14}, \langle\alpha_z\rangle_{24}, \langle\alpha_z\rangle_{34})$

Online:

- P_2, P_3 jmp-send $\langle\alpha_z\rangle_{23}$
- P_2, P_4 jmp-send $\langle\alpha_z\rangle_{24}$
- P_3, P_4 jmp-send $\langle\alpha_z\rangle_{34}$

Communication: 3 elements

Reconstruction towards all: 12 elements

Optimized reconstruction: 7 elements

Koti, N., Pancholi, M., Patra, A., & Suresh, A., "SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning," *30th USENIX Security Symposium,* 2021.

Triple Generation

$$\alpha_x\alpha_y = \sum_{1\le i<j\le 4}\langle\alpha_x\rangle_{ij}\cdot\sum_{1\le k<m\le 4}\langle\alpha_y\rangle_{km} = \sum\langle\alpha_x\rangle_{ij}\langle\alpha_y\rangle_{ij} + \sum\langle\alpha_x\rangle_{ij}\langle\alpha_y\rangle_{ik} + \sum\langle\alpha_x\rangle_{ij}\langle\alpha_y\rangle_{km}$$

	$\langle\alpha_x\rangle_{12}$	$\langle\alpha_x\rangle_{13}$	$\langle\alpha_x\rangle_{14}$	$\langle\alpha_x\rangle_{23}$	$\langle\alpha_x\rangle_{24}$	$\langle\alpha_x\rangle_{34}$
$\langle\alpha_y\rangle_{12}$	S_2	S_1	S_1	S_1	S_1	S_0
$\langle\alpha_y\rangle_{13}$	S_1	S_2	S_1	S_1	S_0	S_1
$\langle\alpha_y\rangle_{14}$	S_1	S_1	S_2	S_0	S_1	S_1
$\langle\alpha_y\rangle_{23}$	S_1	S_1	S_0	S_2	S_1	S_1
$\langle\alpha_y\rangle_{24}$	S_1	S_0	S_1	S_1	S_2	S_1
$\langle\alpha_y\rangle_{34}$	S_0	S_1	S_1	S_1	S_1	S_2

Our Results

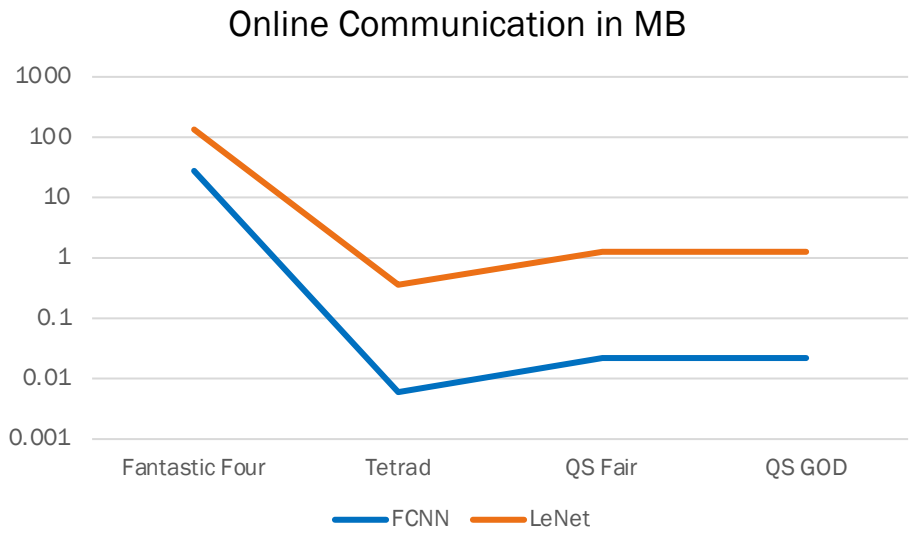
Ref.	Preprocessing Comm.	Online		Model	Security
		Rounds	Comm.		
Tetrad (R)	2	1	3	Honest Majority ($t = 1$)	GOD
Fantastic Four (R)	NA	1	6	Honest Majority ($t = 1$)	GOD
MASCOT (F)	7713	2	12	Dishonest Majority ($t = 3$)	Abort
QuadSquad (R)	**1558**	**3**	**7**	**FaF** ($t = 1, h^* = 1$)	**Fair**
QuadSqaud (R)	**3110**	**3**	**7**	**FaF** ($t = 1, h^* = 1$)	**GOD**

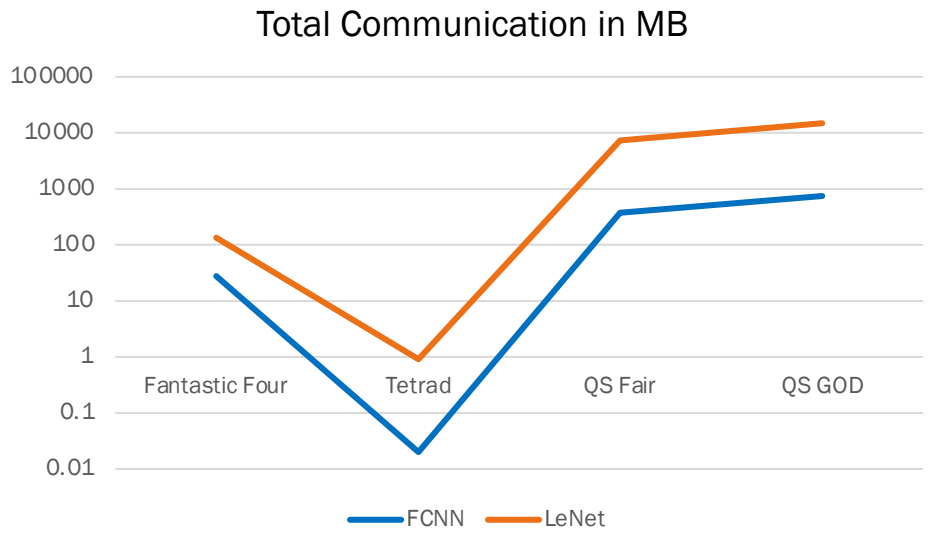
^ The communication cost is reported in terms of field/ring elements.

* Koti, Nishat, et al. "Tetrad: actively secure 4pc for secure training and inference." *arXiv preprint arXiv:2106.02850*(2021).
* Dalskov, Anders, Daniel Escudero, and Marcel Keller. "Fantastic Four: Honest-Majority Four-Party Secure Computation With Malicious Security." *USENIX Security Symposium,* 2021.
* Keller, Marcel, Emmanuela Orsini, and Peter Scholl. "MASCOT: faster malicious arithmetic secure computation with oblivious transfer." *ACM CCS,* 2016.

PPML Benchmarks

- Neural Network inference over MNIST dataset
- Benchmarking over WAN instantiated using Google Cloud
- Code available at: https://github.com/cris-iisc/quadsquad





Thank You!