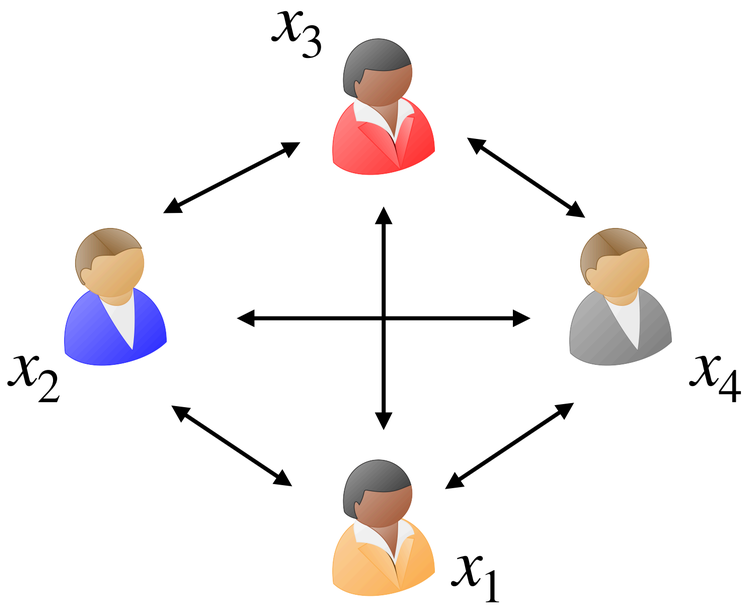
Scalable Multiparty Garbling

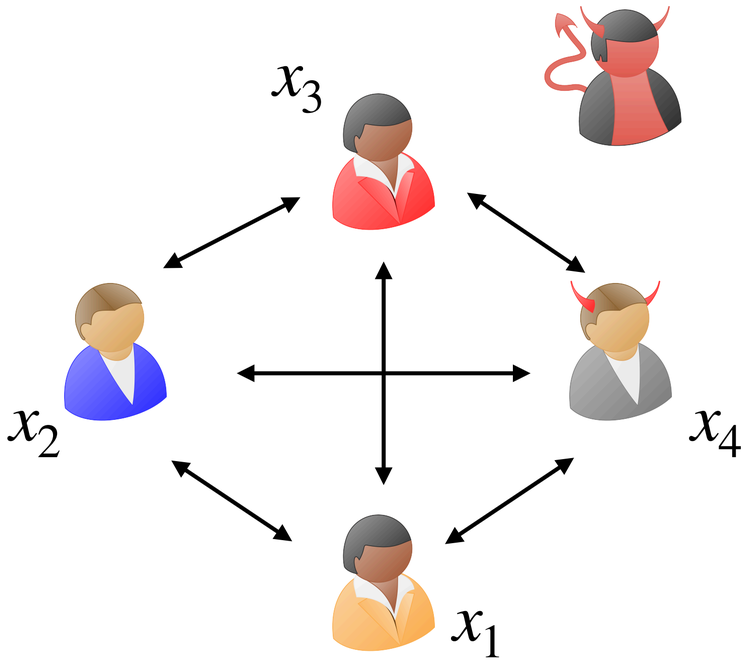
Gabrielle Beck Aarushi Goel **Aditya Hegde** Abhishek Jain Zhengzhong Jin Gabriel Kaptchuk

Secure Multiparty Computation



$$y = f(x_1, x_2, x_3, x_4)$$

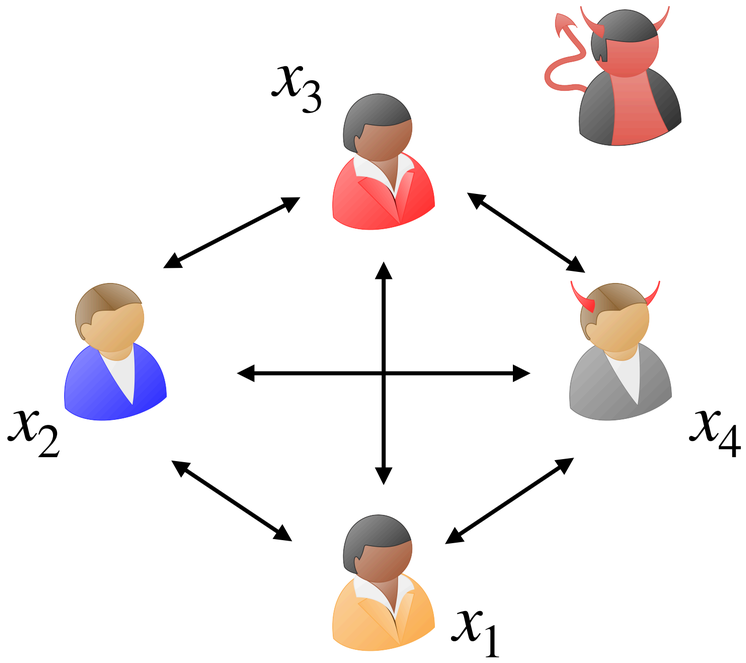
Secure Multiparty Computation



$$y = f(x_1, x_2, x_3, x_4)$$

Secure Multiparty Computation

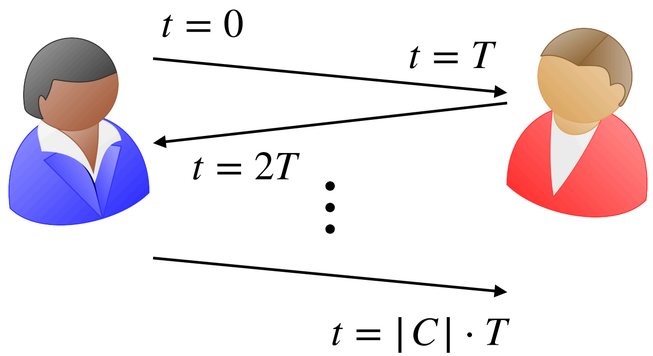
- Applications
  - Machine learning on distributed datasets.
  - Data as a service: secure access to shared and private data.
  - Securing digital assets and key management.



$$y = f(x_1, x_2, x_3, x_4)$$

MPC Over The Internet

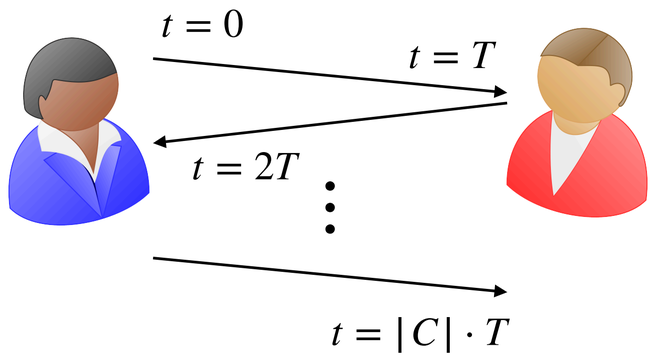
- Over the internet, network latency limits protocol runtime [WRK17b].



MPC Over The Internet

- Over the internet, network latency limits protocol runtime [WRK17b].

- Constant round MPC: Parties interact constant number of times.
  - Supports evaluating large-depth circuits over the internet.
  - 2 Approaches: FHE and Multiparty Garbling.



Constant Round MPC

- Best known constant round MPC has a total communication cost of

$$O(n^2 \cdot |C|)$$

Number of parties

Size of circuit

[HSS17,WRK17b,BCOOSS21].

- In the honest majority setting, non-constant round MPC with a total communication cost of

$$O(|C|)$$

[BGJK21,GSY21,GPS21,GPS22].

Problem: Bringing The Same Efficiency To Multiparty Garbling

Does there exist a multiparty garbling protocol with $O(|C|)$ communication in the honest majority setting?

MPC Protocols With $O(|C|)$ Communication

 Per party communication decreases as the number of parties increases.

MPC Protocols With $O(|C|)$ Communication

Per party communication decreases as the number of parties increases.

MPC that scales to hundreds and thousands of parties.

MPC Protocols With $O(|C|)$ Communication

 Per party communication decreases as the number of parties increases.

 MPC that scales to hundreds and thousands of parties.

 Honest majority is a more plausible assumption.

Our Contributions

- Scalable multiparty garbling
  - $O(|C|)$ communication complexity
  - Malicious security with abort
  - $t < n\left(\frac{1}{2} - \epsilon\right)$
  - Based on the LPN assumption over large fields
  - Practical: ~700 MB for garbling AES-128 with 250 parties.

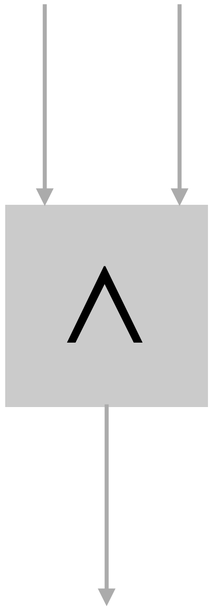
Agenda

- Structure Of Garbled Circuit
- Multiparty Garbling Protocol
- Efficiency Analysis

Agenda

- Structure Of Garbled Circuit
- Multiparty Garbling Protocol
- Efficiency Analysis

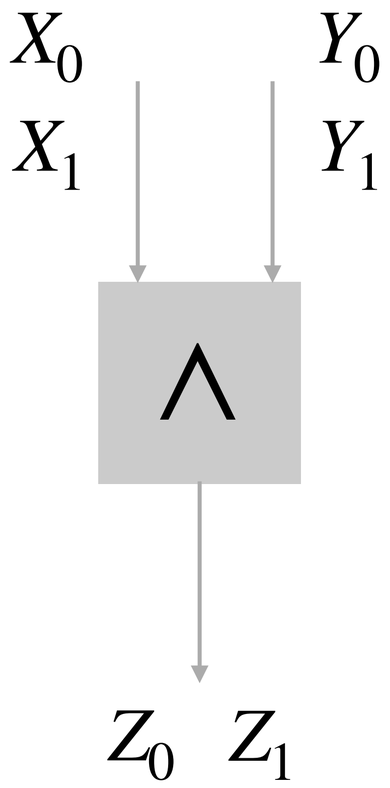
Two Party Garbling: Garbler



Truth Table

0	0	0
0	1	0
1	0	0
1	1	1

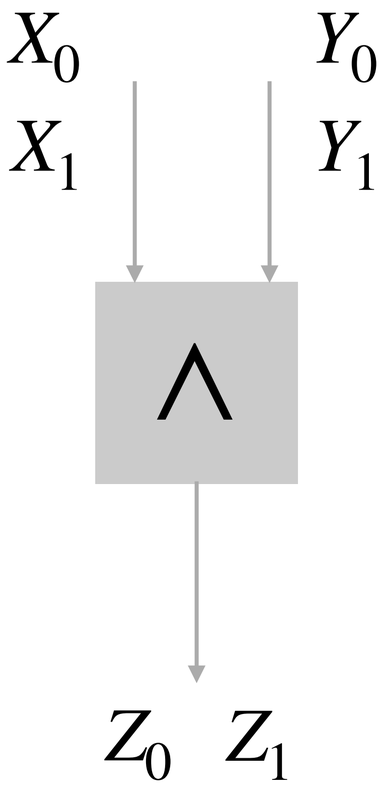
Two Party Garbling: Garbler



Truth Table

0	0	0
0	1	0
1	0	0
1	1	1

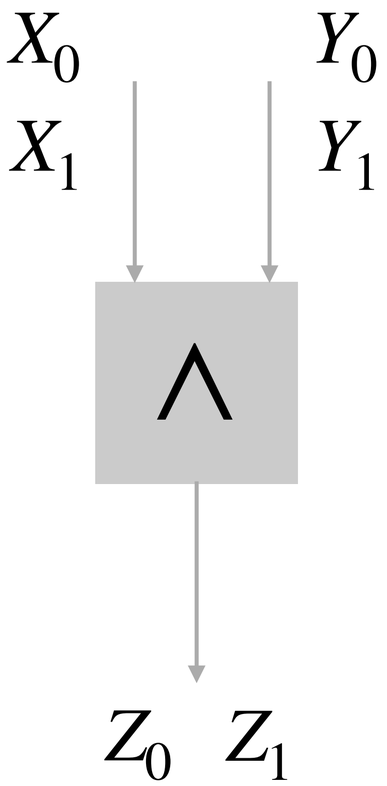
Two Party Garbling: Garbler



Encrypting The Truth Table

X_0	Y_0	Z_0
X_0	Y_1	Z_0
X_1	Y_0	Z_0
X_1	Y_1	Z_1

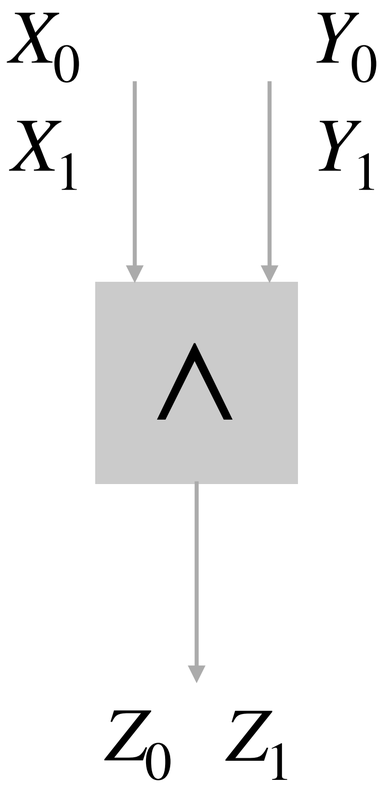
Two Party Garbling: Garbler



Encrypting The Truth Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	
X_1	Y_0	Z_0	
X_1	Y_1	Z_1	

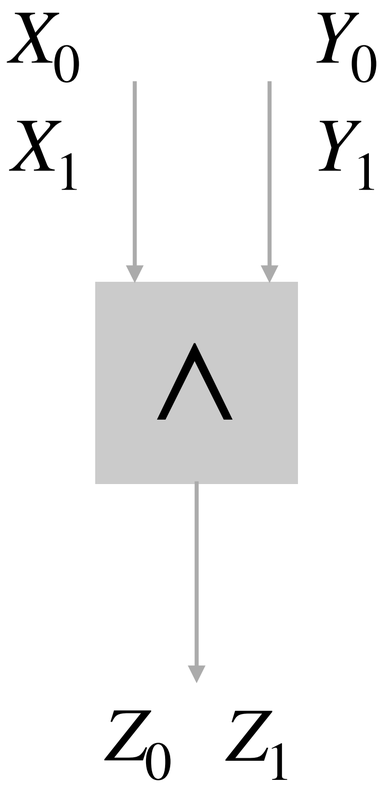
Two Party Garbling: Garbler



Encrypting The Truth Table

X_0	Y_0	Z_0	$\mathsf{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\mathsf{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\mathsf{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\mathsf{Enc}_{X_1,Y_1}(Z_1)$

Two Party Garbling: Garbler



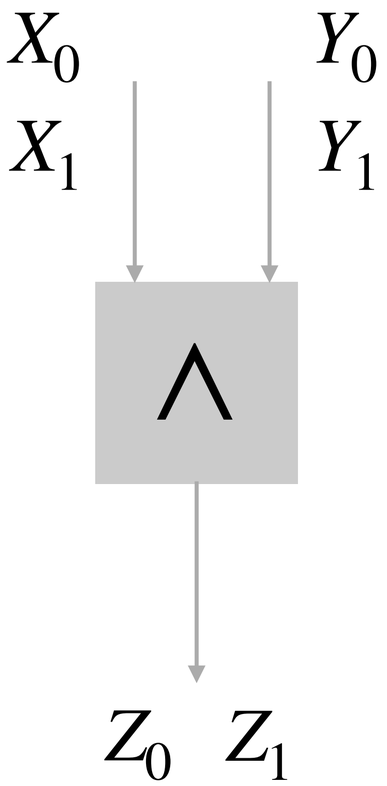
Encrypting The Truth Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Garbler



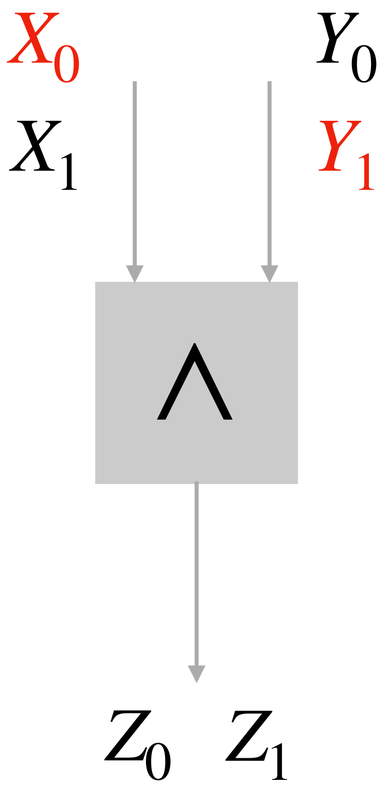
Encrypting The Truth Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

Computed and sent to evaluator. $\mathcal{G}$

Two Party Garbling: Evaluator



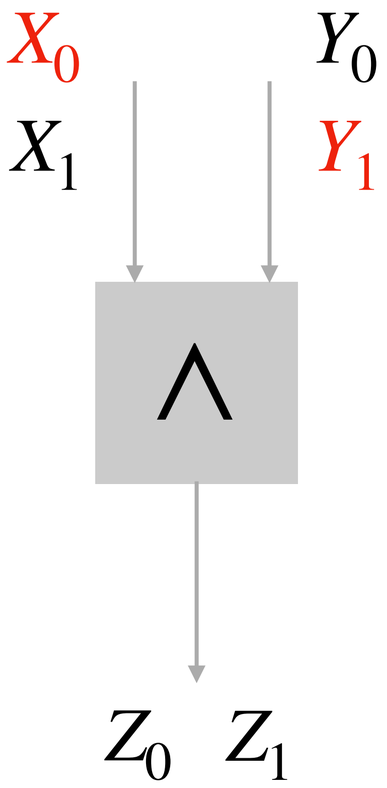
Garbled Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Evaluator



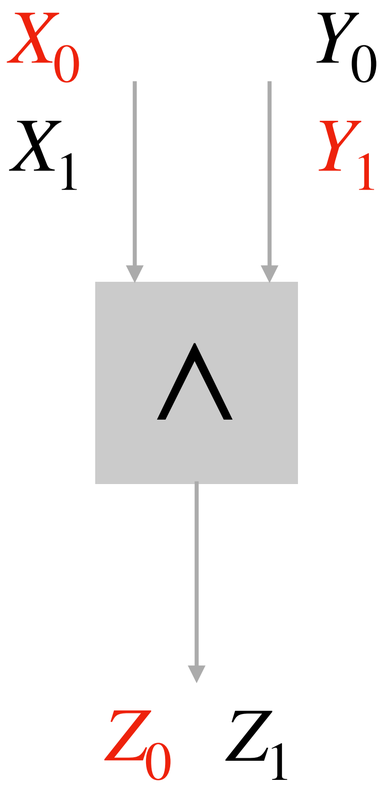
Garbled Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Evaluator



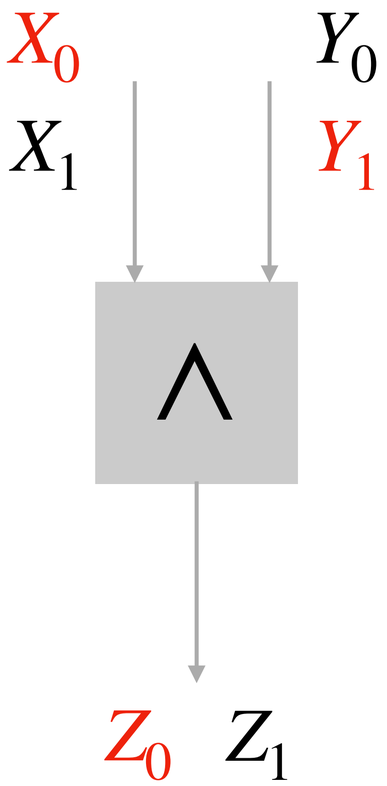
Garbled Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathcal{G}$

Two Party Garbling: Evaluator



Garbled Table

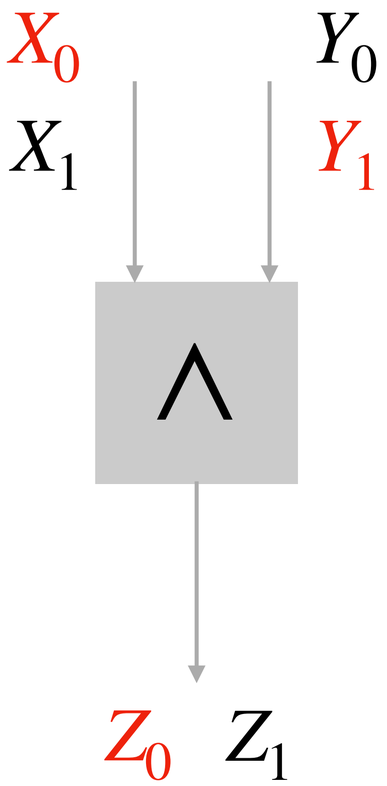
X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Evaluator should learn **exactly one** wire label.

Two Party Garbling: Evaluator



Garbled Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

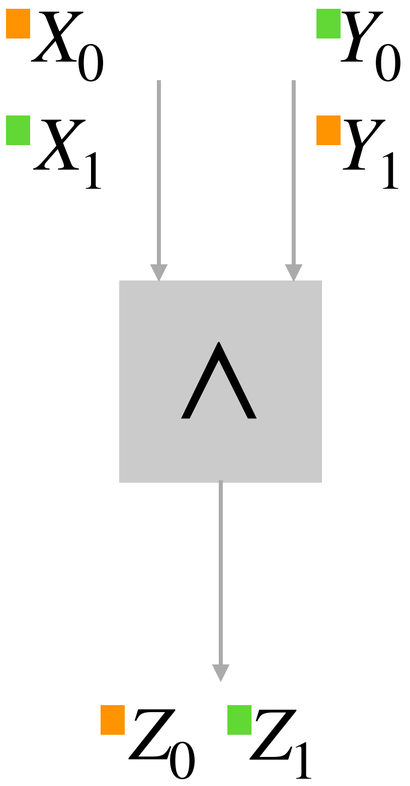
$\mathscr{G}$

Evaluator should learn **exactly one** wire label.

Which ciphertext to decrypt?

Two Party Garbling: Point And Permute

Color wire labels to indicate which ciphertext to decrypt.



Encrypting The Truth Table

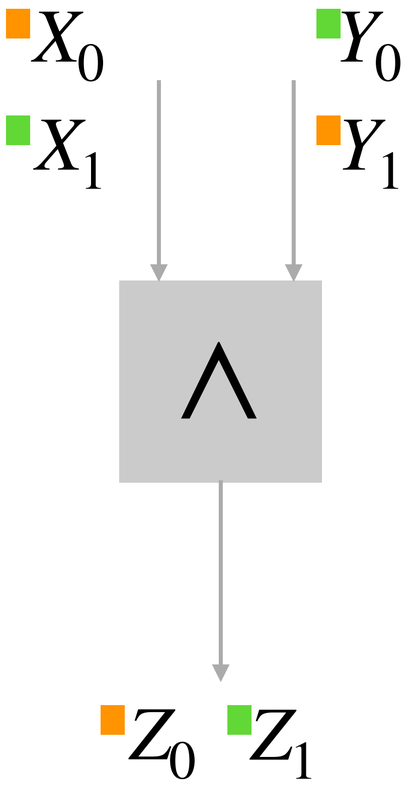
X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Point And Permute

Color wire labels to indicate which ciphertext to decrypt.



Encrypting The Truth Table

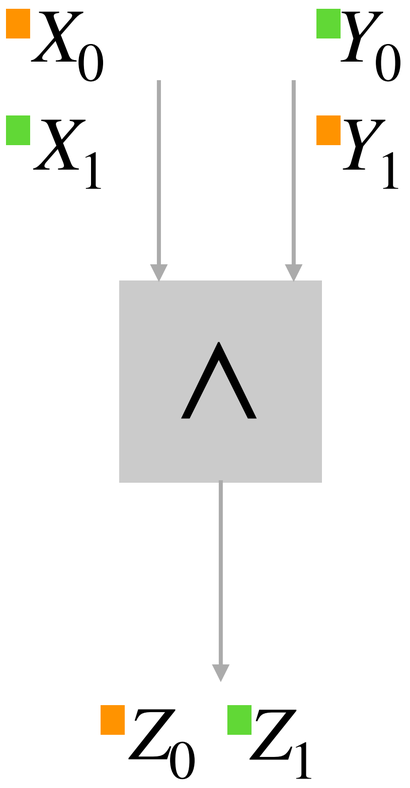
X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Point And Permute

Color wire labels to indicate which ciphertext to decrypt.



$\rightarrow 0$

$\rightarrow 1$

Encrypting The Truth Table

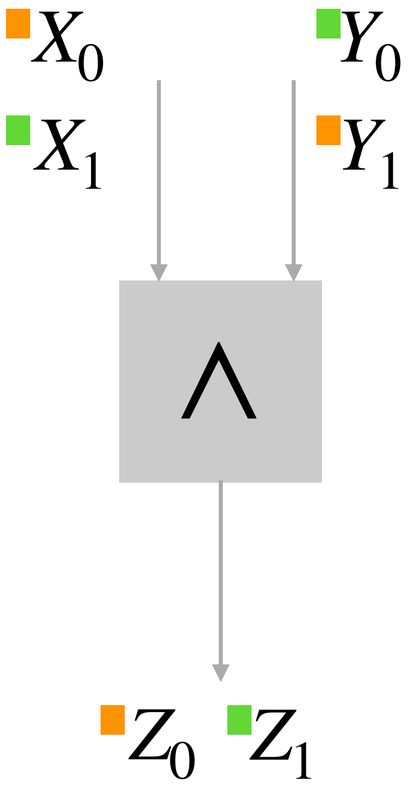
X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathscr{G}$

Two Party Garbling: Point And Permute

Color wire labels to indicate which ciphertext to decrypt.



$\blacksquare \rightarrow 0$

$\blacksquare \rightarrow 1$

Encrypting The Truth Table

X_0	Y_0	Z_0	$\text{Enc}_{X_0,Y_0}(Z_0)$
X_0	Y_1	Z_0	$\text{Enc}_{X_0,Y_1}(Z_0)$
X_1	Y_0	Z_0	$\text{Enc}_{X_1,Y_0}(Z_0)$
X_1	Y_1	Z_1	$\text{Enc}_{X_1,Y_1}(Z_1)$

Shuffled

$\mathcal{G}$

Coloring is independent of wire label assignment.

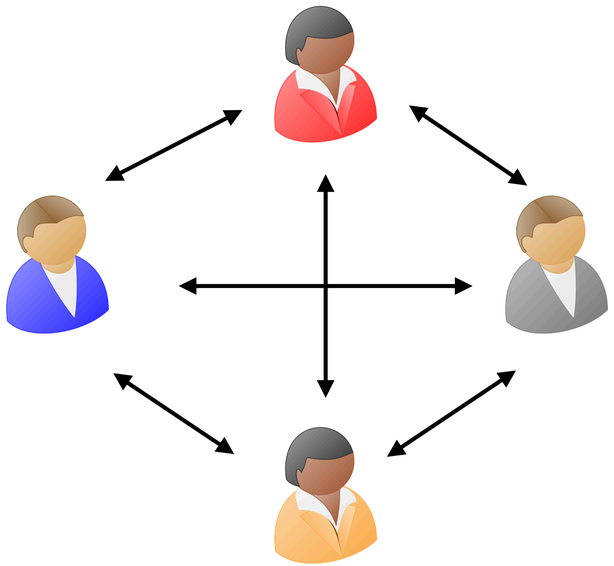
A Template For Multiparty Garbling [BMR90]

Non-constant round MPC → Constant round MPC

A Template For Multiparty Garbling [BMR90]

Non-constant round MPC → Constant round MPC

Garbling Phase

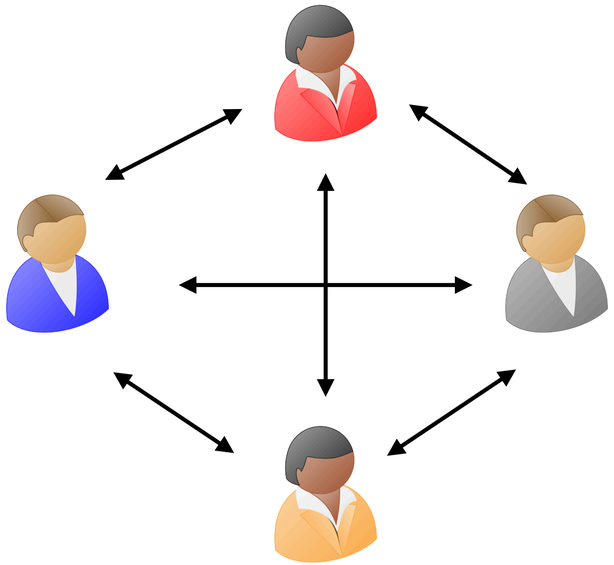


$\mathscr{G} \quad X_0, Y_1$

A Template For Multiparty Garbling [BMR90]

Non-constant round MPC → Constant round MPC

Garbling Phase



$\mathscr{G} \quad X_0, Y_1$

Evaluation Phase

$\mathsf{Eval}(\mathscr{G}, X_0, Y_1)$

$\mathsf{Eval}(\mathscr{G}, X_0, Y_1)$

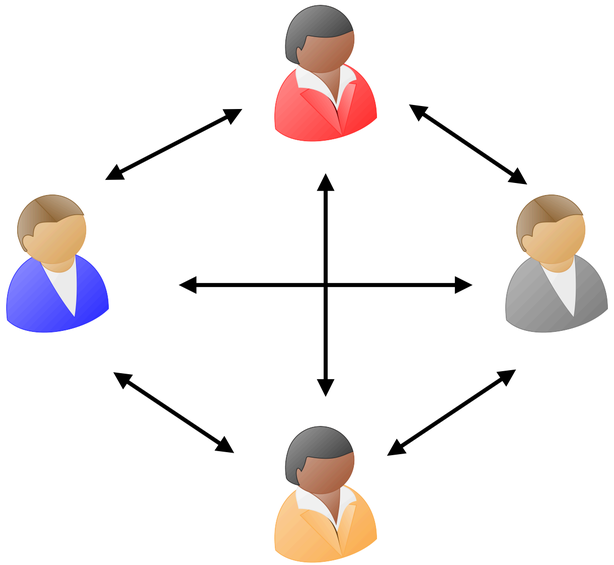
$\mathsf{Eval}(\mathscr{G}, X_0, Y_1)$

$\mathsf{Eval}(\mathscr{G}, X_0, Y_1)$

A Template For Multiparty Garbling [BMR90]

Non-constant round MPC → Constant round MPC

Garbling Phase



$\mathscr{G} \quad X_0, Y_1$

Evaluation Phase

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

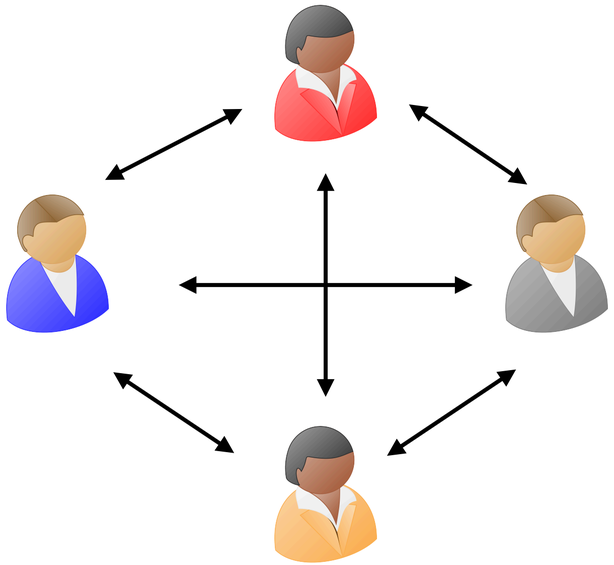
$\text{Eval}(\mathscr{G}, X_0, Y_1)$

- MPC ensures wire labels are private.
- All gates can be garbled in parallel $\Longrightarrow$ constant round MPC.

A Template For Multiparty Garbling [BMR90]

Non-constant round MPC → Constant round MPC

Garbling Phase



$\mathscr{G} \quad X_0, Y_1$

Evaluation Phase

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

$\text{Eval}(\mathscr{G}, X_0, Y_1)$

- MPC ensures wire labels are private.
- All gates can be garbled in parallel $\Longrightarrow$ constant round MPC.

$O(|C|)$ MPC for garbling $\Longrightarrow$ $O(|C|)$ constant round MPC?

Non-black box use of encryption.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05]

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Wire labels can't be known in the clear by any single party.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Wire labels can't be known in the clear by any single party.

X_1 Y_1 X_2 Y_2 X_3 Y_3

$X = X_1 \| X_2 \| X_3$ $Y = Y_1 \| Y_2 \| Y_3$

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Wire labels can't be known in the clear by any single party.

X_1 Y_1 X_2 Y_2 X_3 Y_3

$X = X_1 \| X_2 \| X_3 \qquad Y = Y_1 \| Y_2 \| Y_3$

$\text{PRF}_X(g) = PRF_{X_1}(g) \oplus PRF_{X_2}(g) \oplus PRF_{X_3}(g)$

Each party can locally evaluate PRF.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Wire labels can't be known in the clear by any single party.

X_1 Y_1 X_2 Y_2 X_3 Y_3

$X = X_1 \| X_2 \| X_3$ $Y = Y_1 \| Y_2 \| Y_3$

$\text{PRF}_X(g) = PRF_{X_1}(g) \oplus PRF_{X_2}(g) \oplus PRF_{X_3}(g)$

Each party can locally evaluate PRF.

Wire labels and hence encryptions are linear in n.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Use key-homomorphic PRFs [BLO17] $\rightarrow$ $|\mathcal{G}|$ independent of n.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Use key-homomorphic PRFs [BLO17] $\rightarrow$ $|\mathcal{G}|$ independent of n.

X_1 Y_1 X_2 Y_2 X_3 Y_3

$X = X_1 + X_2 + X_3$ $Y = Y_1 + Y_2 + Y_3$

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Use key-homomorphic PRFs [BLO17] $\rightarrow$ $|\mathcal{G}|$ independent of n.



X_1 Y_1 X_2 Y_2 X_3 Y_3

Homomorphism operator

$X = X_1 + X_2 + X_3$ $Y = Y_1 + Y_2 + Y_3$

$\text{PRF}_{X_1}(g) \tilde{+} \text{PRF}_{X_2}(g) \tilde{+} \text{PRF}_{X_3}(g) = \text{PRF}_X(g)$

Each party can locally evaluate PRF on its share.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Use key-homomorphic PRFs [BLO17] $\rightarrow$ $|\mathcal{G}|$ independent of n.



X_1 Y_1 X_2 Y_2 X_3 Y_3

$X = X_1 + X_2 + X_3$ $Y = Y_1 + Y_2 + Y_3$

Wire labels independent of n.

Homomorphism operator

$\text{PRF}_{X_1}(g) \tilde{+} \text{PRF}_{X_2}(g) \tilde{+} \text{PRF}_{X_3}(g) = \text{PRF}_X(g)$

Each party can locally evaluate PRF on its share.

Moving Away From Non-black Box Constructions

Garbling g^{th} gate: $\text{Enc}_{X,Y}(Z) = \text{PRF}_X(g) \oplus \text{PRF}_Y(g) \oplus Z$

Evaluate the PRF outside the MPC [DI05] $\rightarrow$ $|\mathcal{G}|$ is linear in n.

Use key-homomorphic PRFs [BLO17] $\rightarrow$ $|\mathcal{G}|$ independent of n.



X_1 Y_1 X_2 Y_2 X_3 Y_3

Homomorphism operator

$X = X_1 + X_2 + X_3$ $Y = Y_1 + Y_2 + Y_3$

$\text{PRF}_{X_1}(g) \tilde{+} \text{PRF}_{X_2}(g) \tilde{+} \text{PRF}_{X_3}(g) = \text{PRF}_X(g)$

Each party can locally evaluate PRF on its share.

Wire labels independent of n.

Does not extend to threshold setting.

Garbling Using Doubly Linear Homomorphic PRFs

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathsf{PRF}_{[X]}(g) = [PRF_X(g)]$$

Garbling Using Doubly Linear Homomorphic PRFs

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

$[X][Y][Z]$

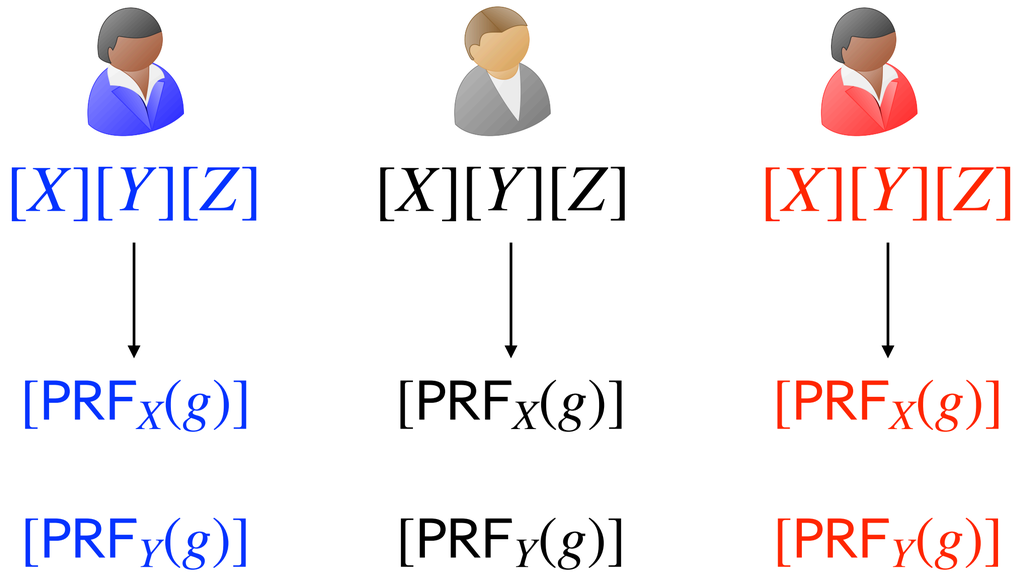
$[X][Y][Z]$

$[X][Y][Z]$

Garbling Using Doubly Linear Homomorphic PRFs

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

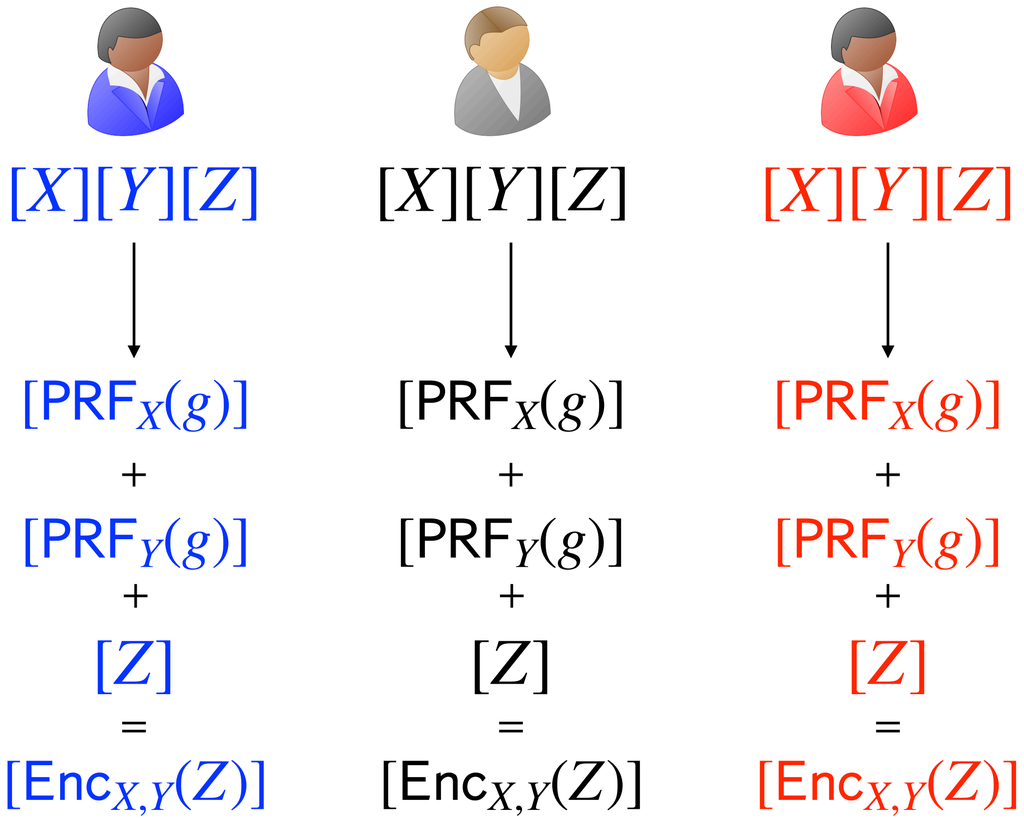
$$\mathsf{PRF}_{[X]}(g) = [PRF_X(g)]$$



Garbling Using Doubly Linear Homomorphic PRFs

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

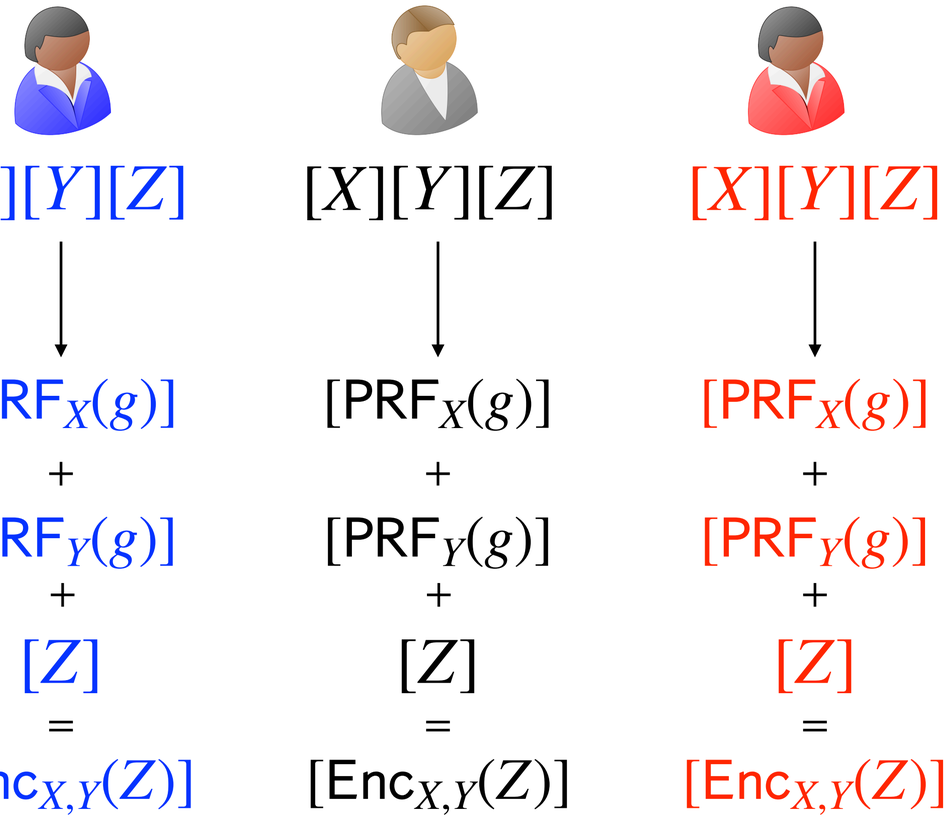
$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$



Garbling Using Doubly Linear Homomorphic PRFs

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$



No known constructions of DLH-PRFs.

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$ $\quad [k] \cdot A + [\epsilon] = [R_k]$

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$ $\quad$ $[k] \cdot A + [\epsilon] = [R_k]$

$[X][Y][Z]$

$[X][Y][Z]$

$[X][Y][Z]$

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$ $\quad$ $[k] \cdot A + [\epsilon] = [R_k]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$ $\quad$ $[k] \cdot A + [\epsilon] = [R_k]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$

$[k] \cdot A + [\epsilon] = [R_k]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

Avoid non-black box usage of encryption.

$|\mathcal{G}|$ is independent of number of parties.

Garbling Using LPN

DLH-PRF: Secret sharing of the key → Secret sharing of PRF expansion.

$$\mathrm{PRF}_{[X]}(g) = [PRF_X(g)]$$

Use pseudo-randomness from LPN instead of PRF evaluations.

Expand labels using LPN: $R_k = k \cdot A + \epsilon$ $\quad [k] \cdot A + [\epsilon] = [R_k]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[R_X] + [R_Y] + [Z]$
$= [\mathrm{Enc}_{X,Y}(Z)]$

Avoid non-black box usage of encryption.

$|\mathcal{G}|$ is independent of number of parties.

[BCOOSS21] used LPN for garbling in the dishonest majority setting.

Agenda

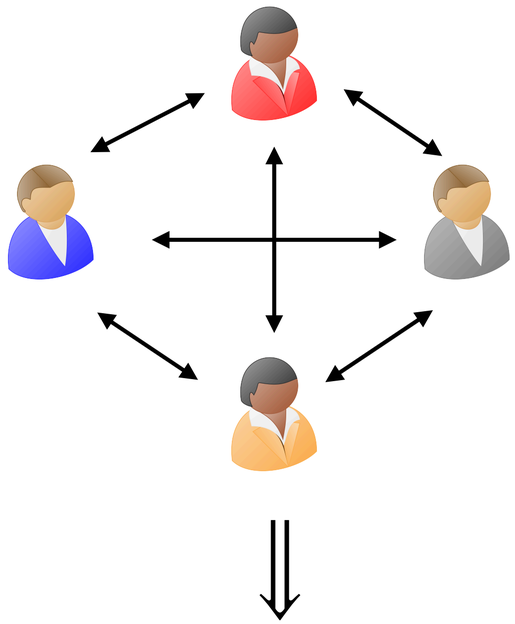
- Structure Of Garbled Circuit
  - Secret sharing of LPN keys and errors → secret sharing of ciphertext.
- Multiparty Garbling Protocol
- Efficiency analysis

Agenda

- Structure Of Garbled Circuit
- Multiparty Garbling Protocol
- Efficiency Analysis

Computing The Garbled Circuit

Pre-processing Phase



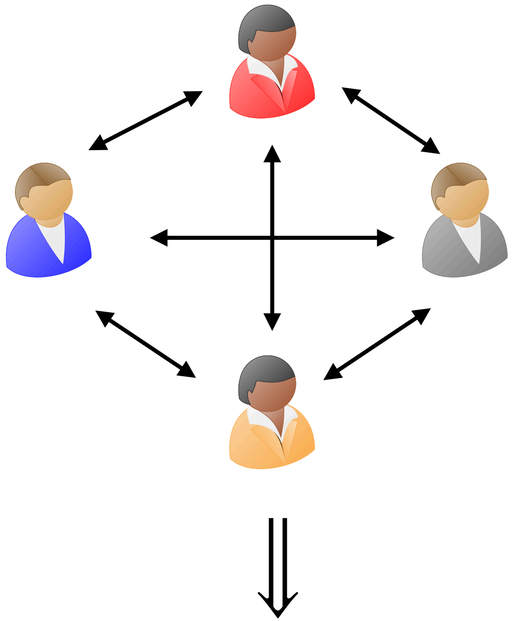
$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Computing The Garbled Circuit

Pre-processing Phase

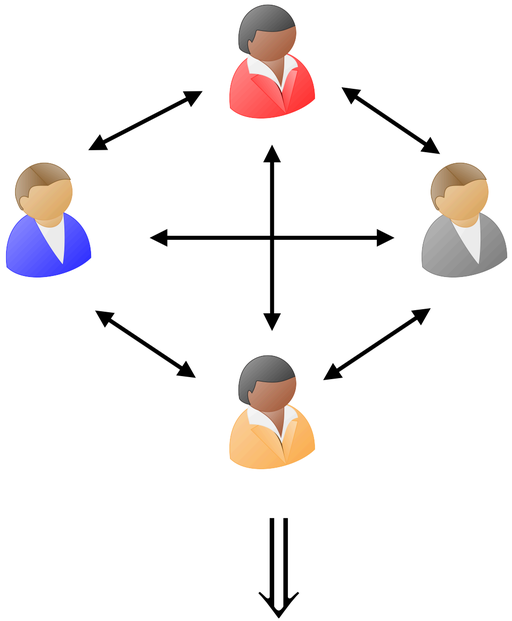


$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Garbling Phase

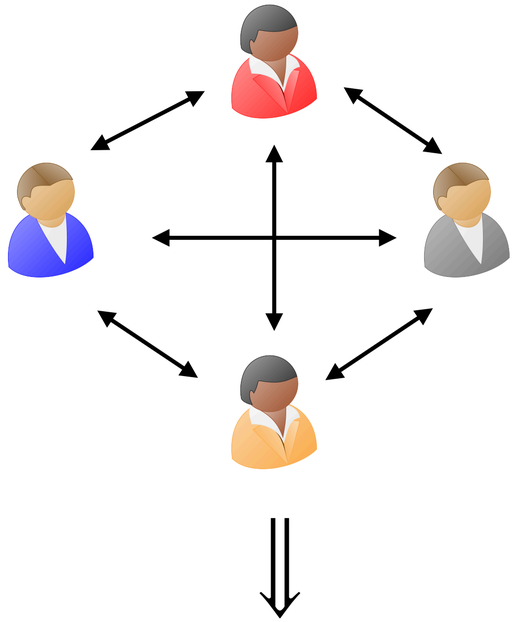


$\mathcal{G}$ - Garbled circuit

$\{X_{\text{inp}}\}$ - Wire labels for inputs

Computing The Garbled Circuit

Pre-processing Phase

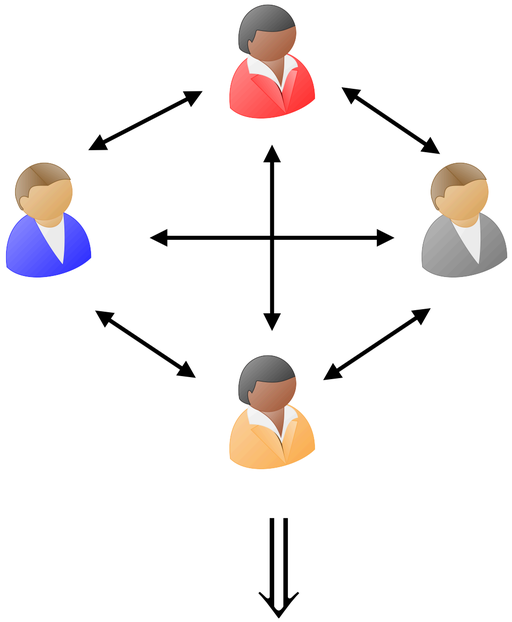


$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Garbling Phase



$\mathcal{G}$ - Garbled circuit

$\{X_{\text{inp}}\}$ - Wire labels for inputs

Evaluation Phase

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

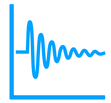
Packed Secret Sharing [FY92]

Regular Secret Sharing

v

Packed Secret Sharing [FY92]

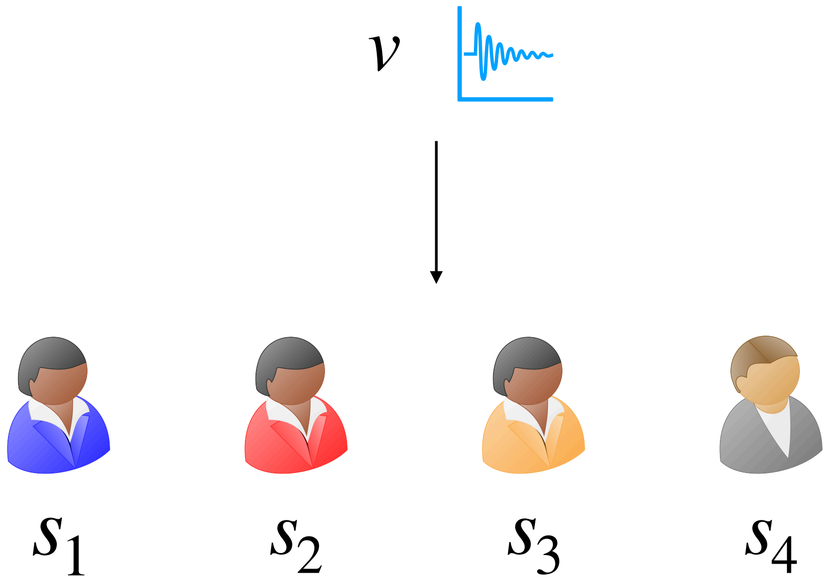
Regular Secret Sharing

v 



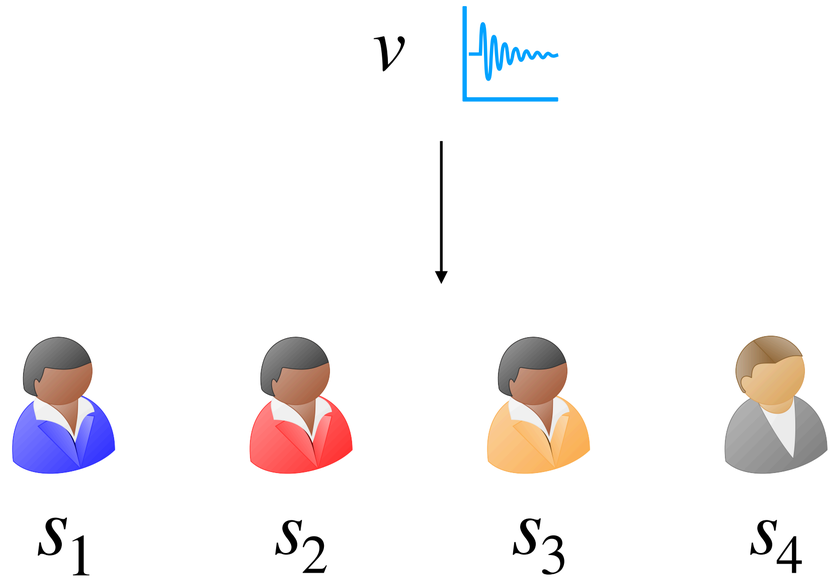

Packed Secret Sharing [FY92]

Regular Secret Sharing

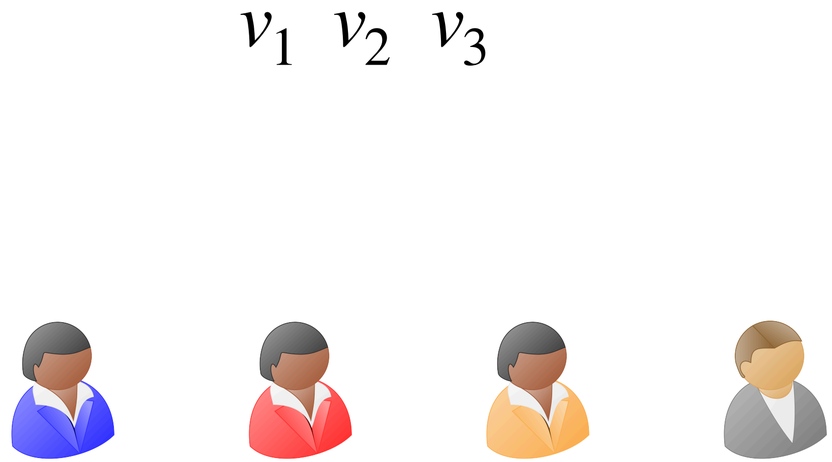


Packed Secret Sharing [FY92]

Regular Secret Sharing

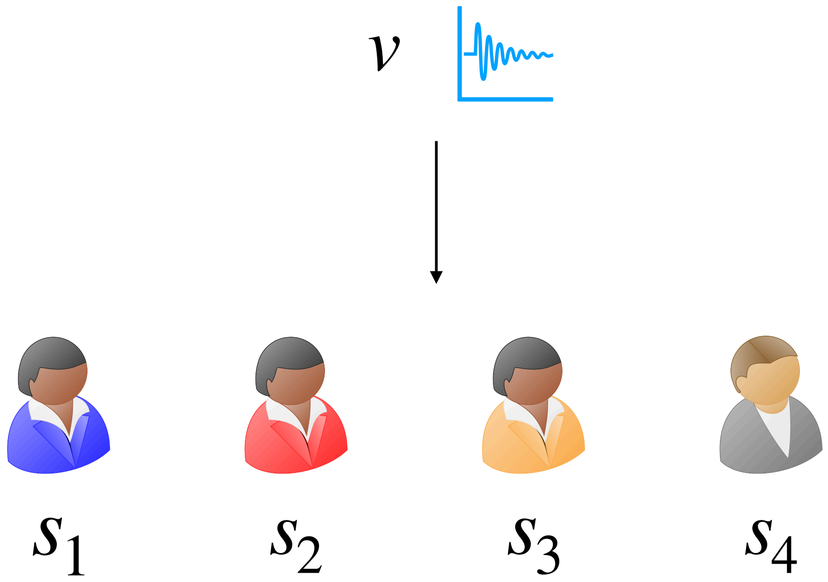


Packed Secret Sharing

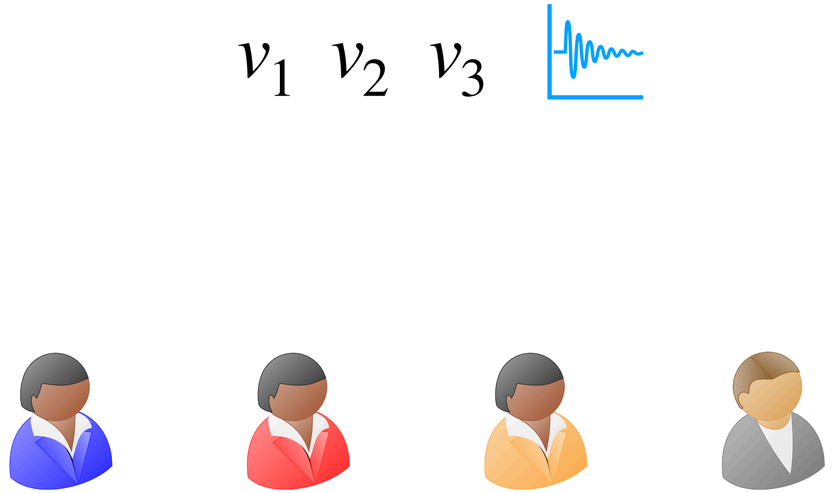


Packed Secret Sharing [FY92]

Regular Secret Sharing

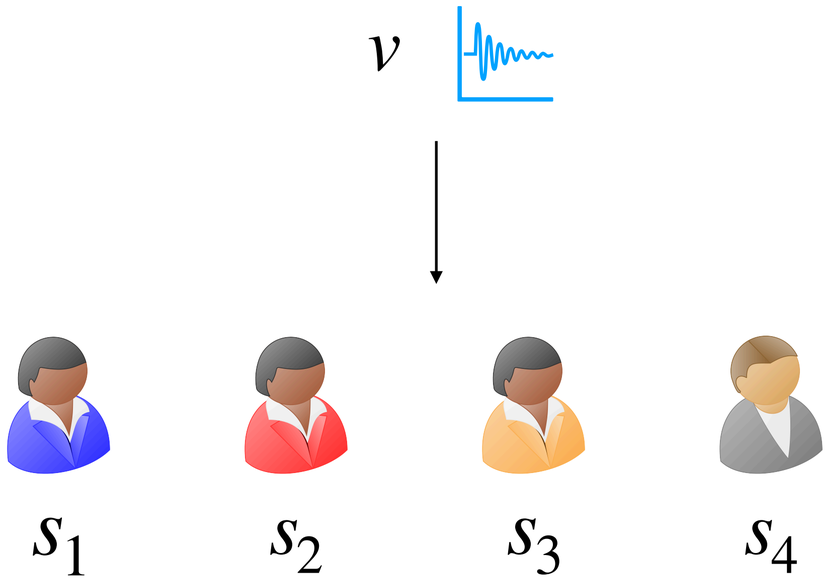


Packed Secret Sharing

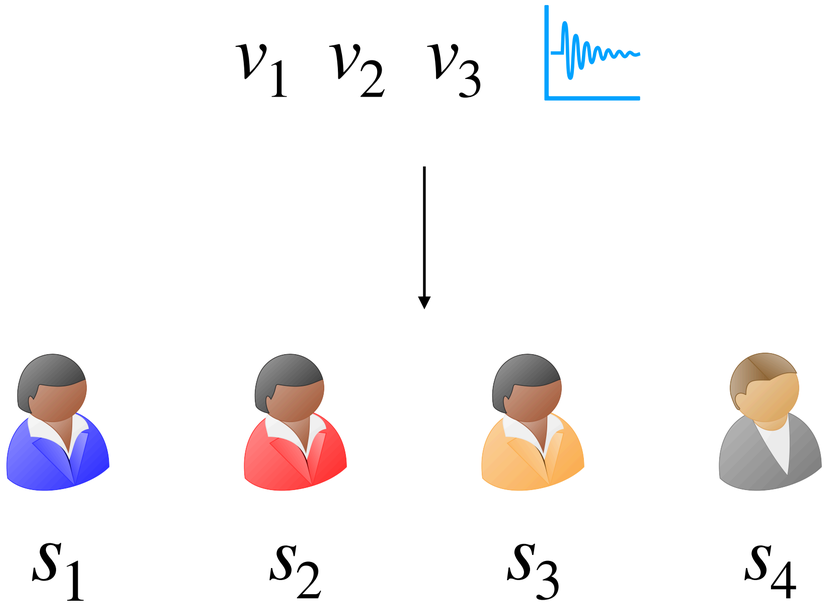


Packed Secret Sharing [FY92]

Regular Secret Sharing

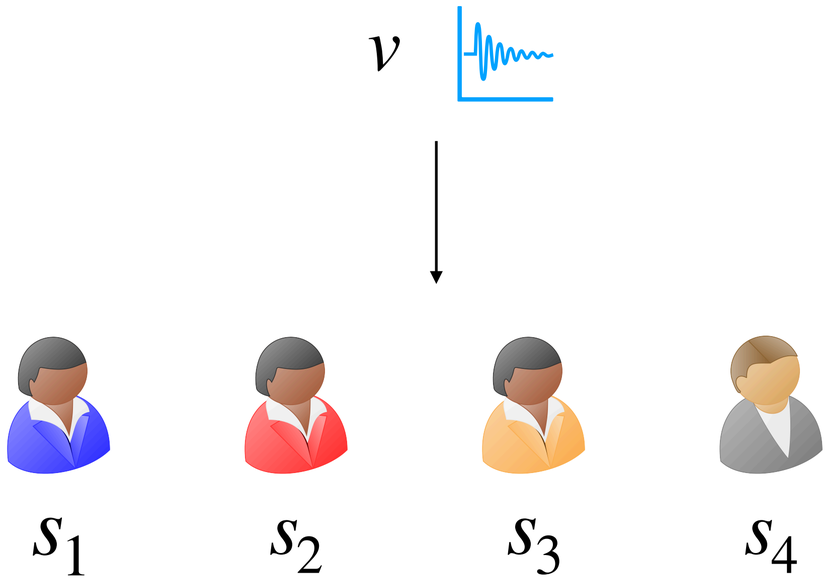


Packed Secret Sharing



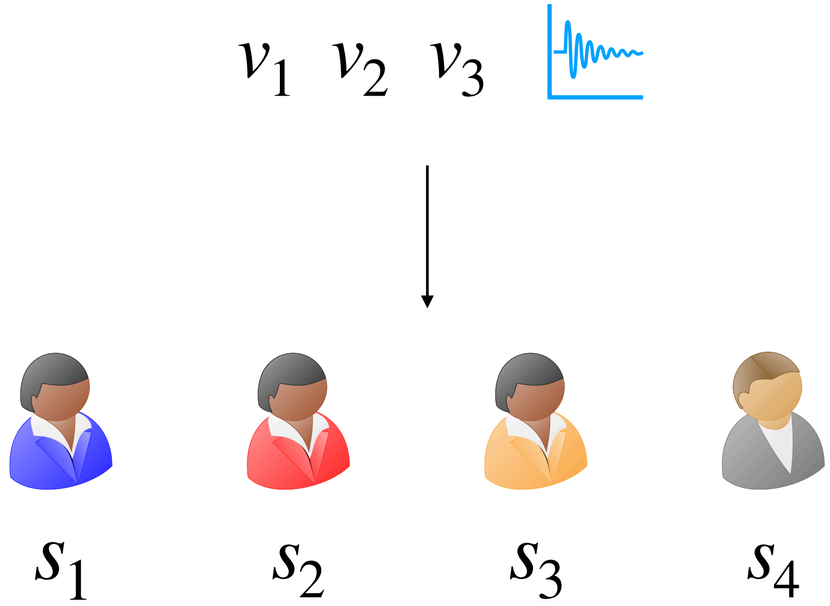
Packed Secret Sharing [FY92]

Regular Secret Sharing



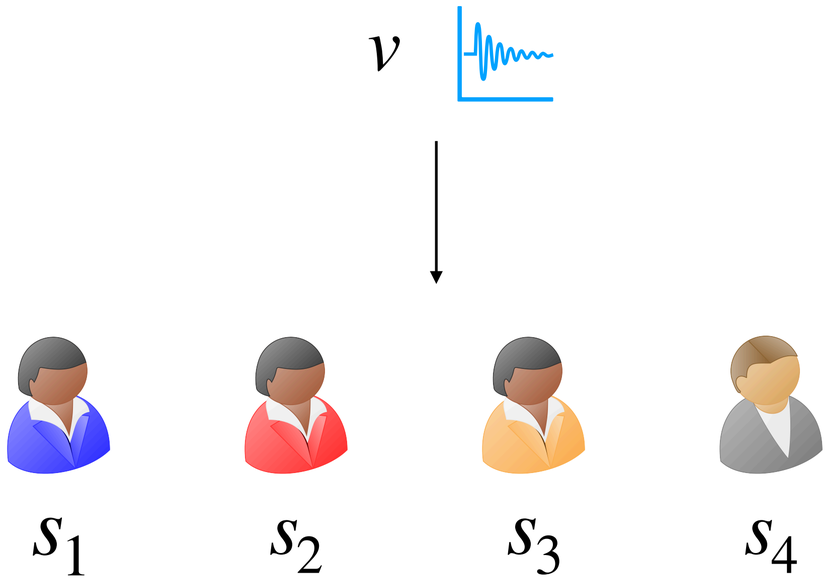
n shares → 1 secret

Packed Secret Sharing



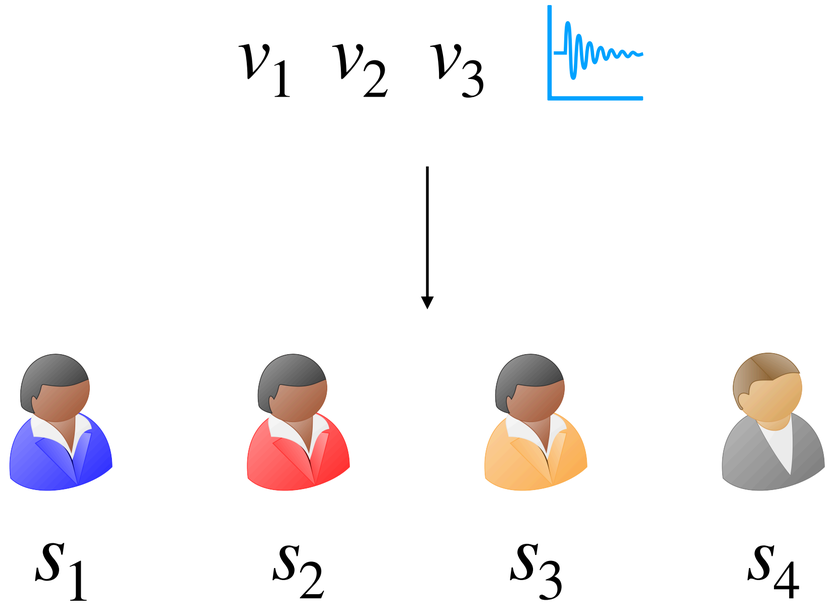
Packed Secret Sharing [FY92]

Regular Secret Sharing



n shares → 1 secret

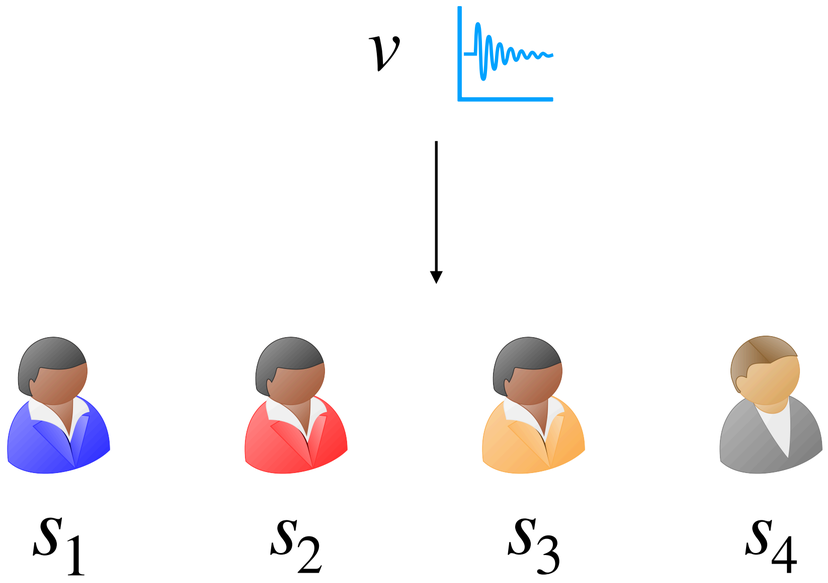
Packed Secret Sharing



n shares → ℓ secrets

Packed Secret Sharing [FY92]

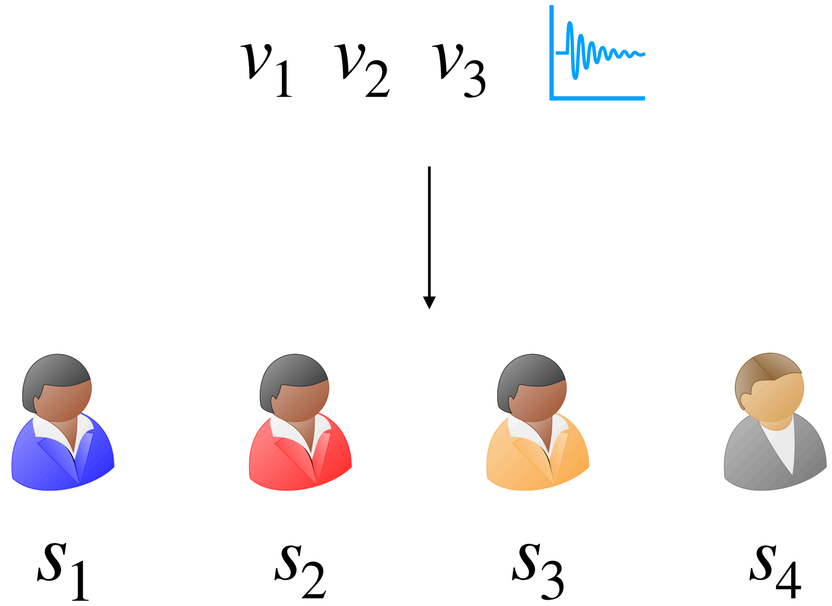
Regular Secret Sharing



n shares $\rightarrow$ 1 secret

Cannot reconstruct $\leq t$

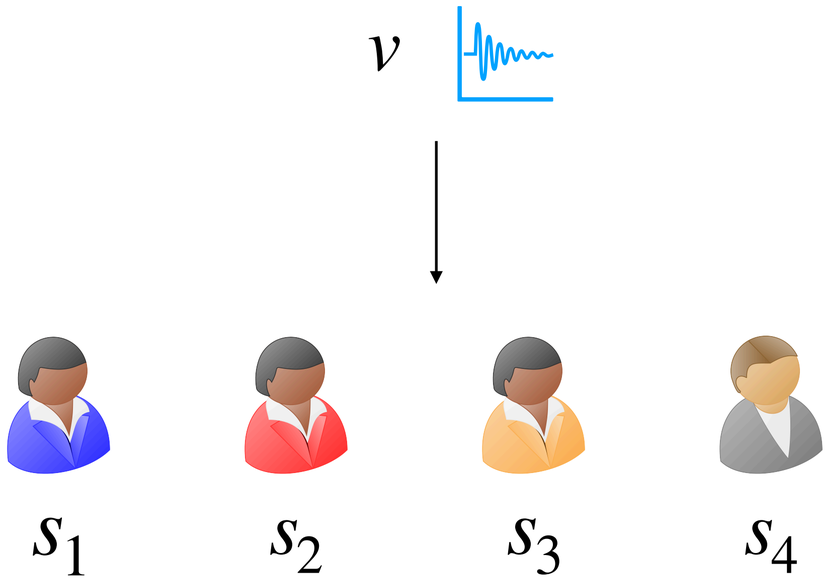
Packed Secret Sharing



n shares $\rightarrow$ ℓ secrets

Packed Secret Sharing [FY92]

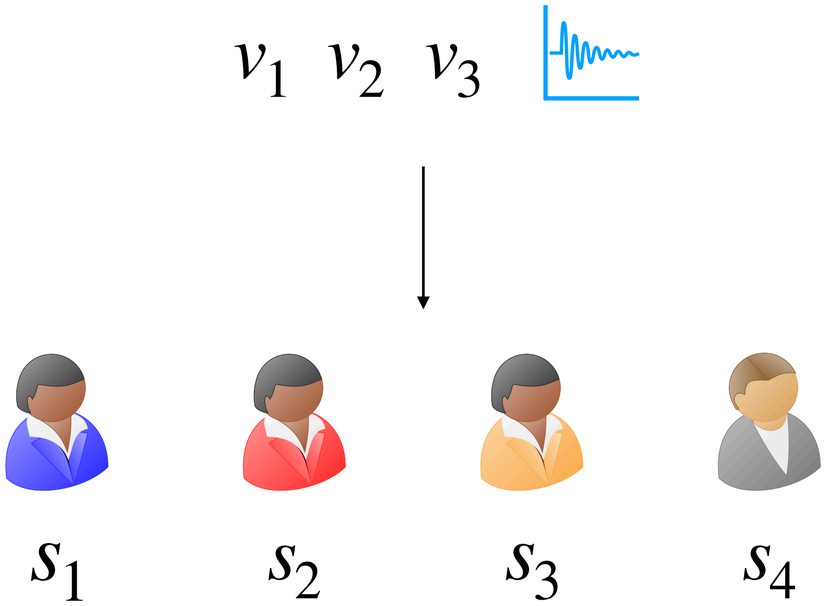
Regular Secret Sharing



n shares $\rightarrow$ 1 secret

Cannot reconstruct $\leq t <$ Can reconstruct

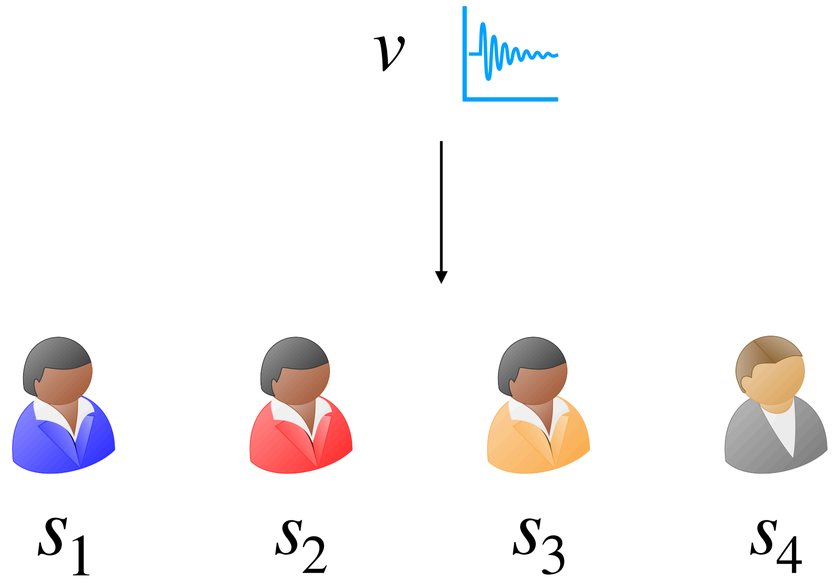
Packed Secret Sharing



n shares $\rightarrow$ ℓ secrets

Packed Secret Sharing [FY92]

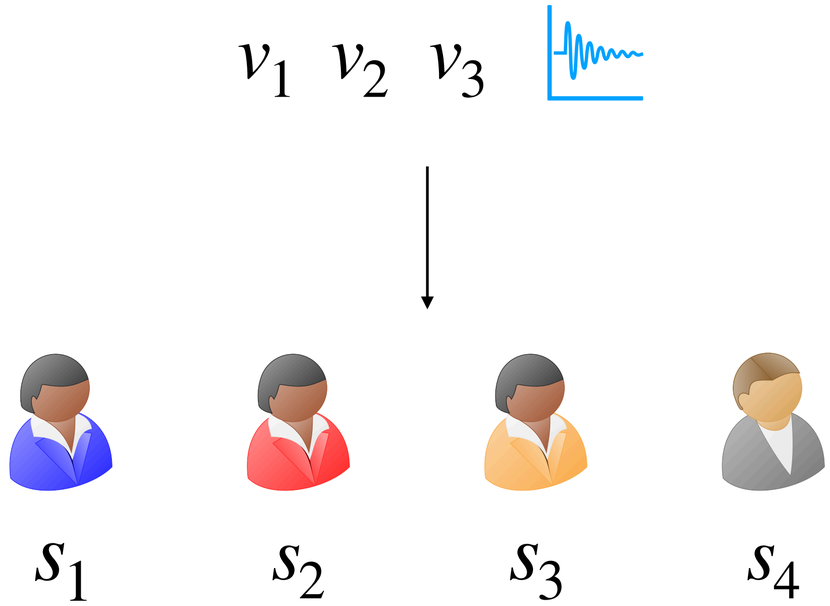
Regular Secret Sharing



n shares $\rightarrow$ 1 secret

Cannot reconstruct $\leq t <$ Can reconstruct

Packed Secret Sharing

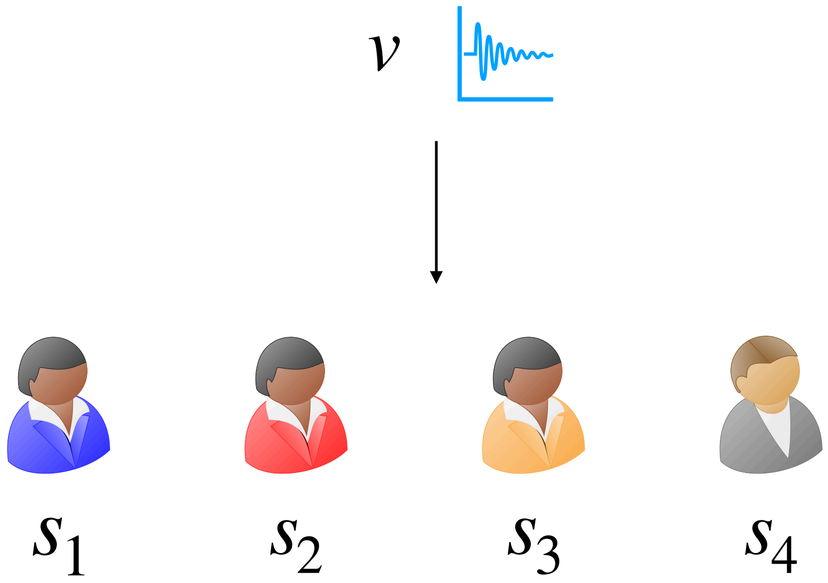


n shares $\rightarrow$ ℓ secrets

Cannot reconstruct $\leq t$

Packed Secret Sharing [FY92]

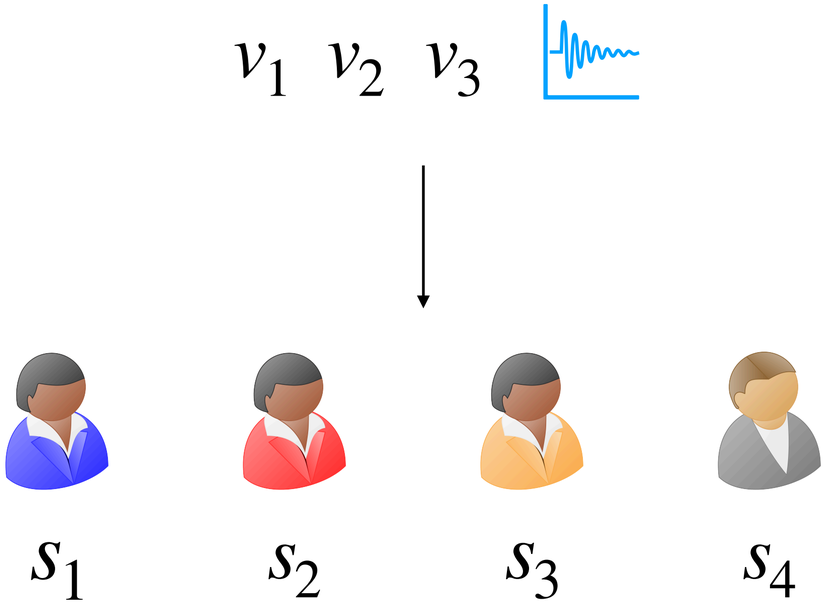
Regular Secret Sharing



n shares $\rightarrow$ 1 secret

Cannot reconstruct $\leq t <$ Can reconstruct

Packed Secret Sharing

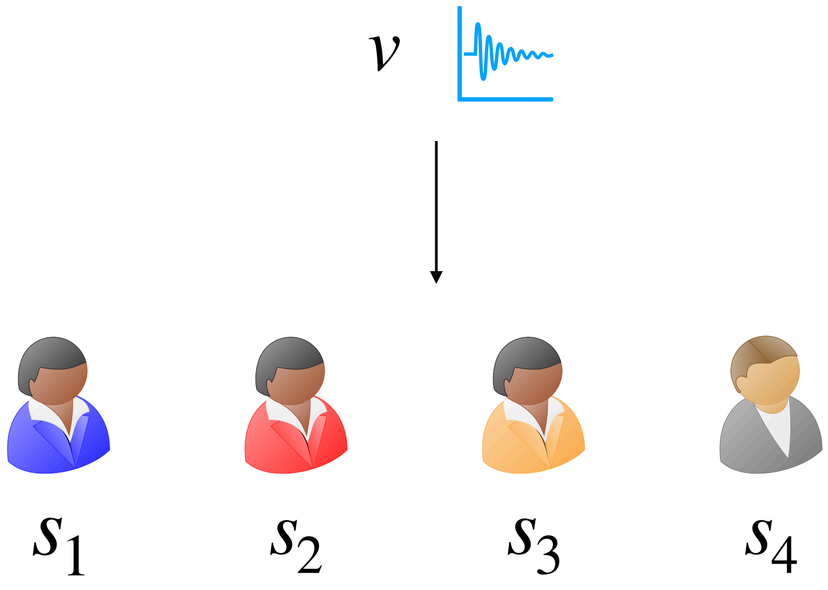


n shares $\rightarrow$ ℓ secrets

Cannot reconstruct $\leq t$ $\quad t + \ell \leq$ Can reconstruct

Packed Secret Sharing [FY92]

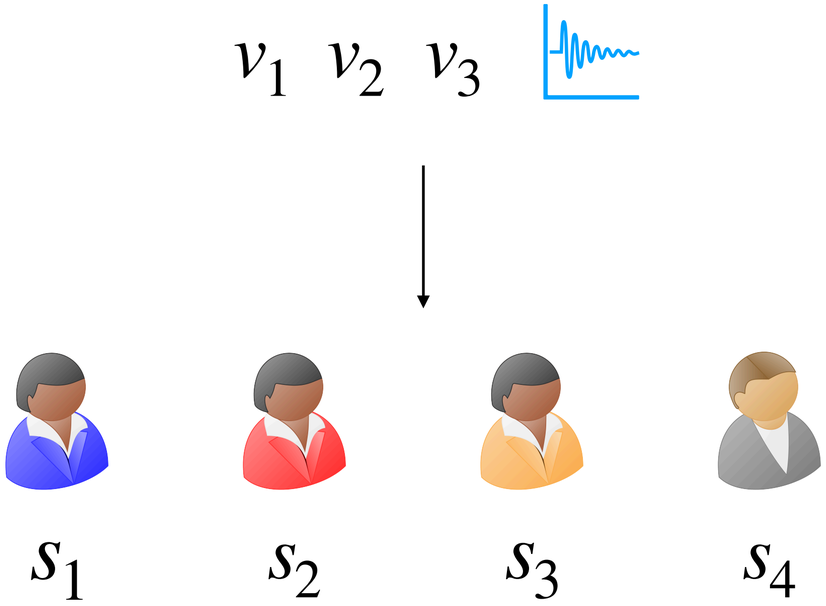
Regular Secret Sharing



n shares $\rightarrow$ 1 secret

Cannot reconstruct $\leq t <$ Can reconstruct

Packed Secret Sharing



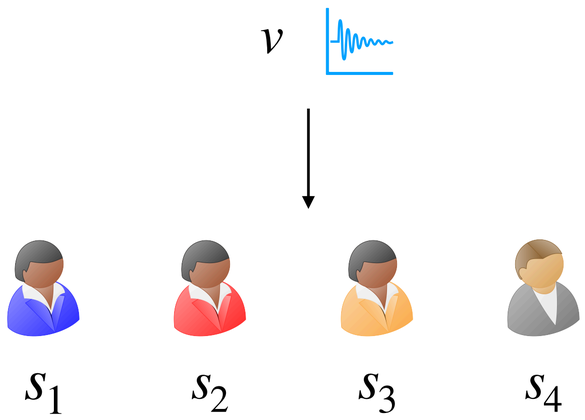
n shares $\rightarrow$ ℓ secrets

Cannot reconstruct $\leq t$ $\quad t + \ell \leq$ Can reconstruct

$$|\mathbb{F}| > n + \ell$$

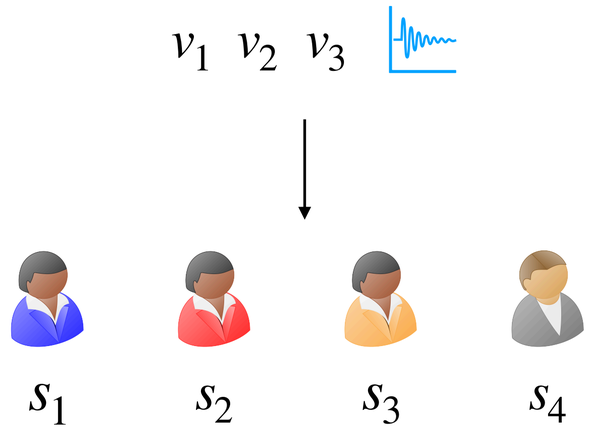
MPC Using Packed Secret Sharing

Regular Sharing



$O(n\,|\,C\,|\,)$ communication MPC

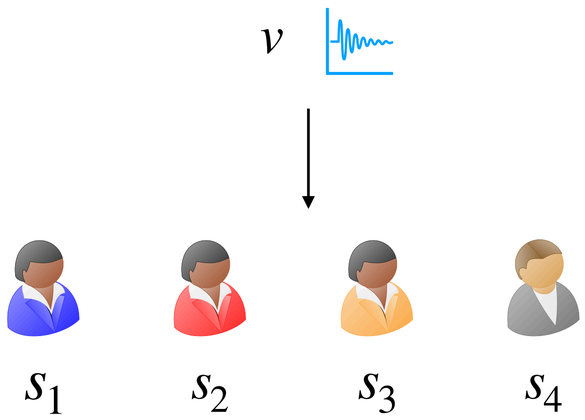
Packed Sharing



$\ell = O(n) \Longrightarrow O(\,|\,C\,|\,)$ communication MPC

MPC Using Packed Secret Sharing

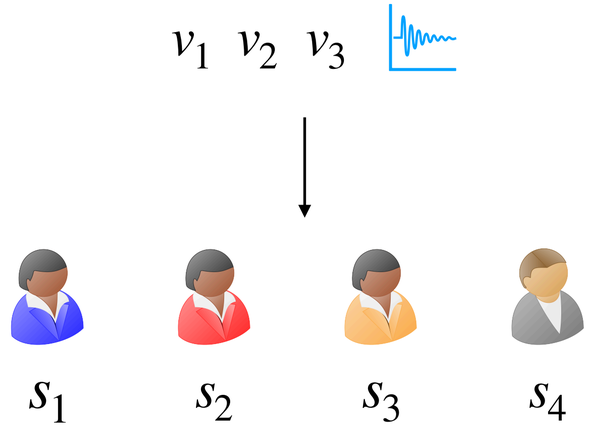
Regular Sharing



$O(n\,|\,C\,|\,)$ communication MPC

$$t = \frac{n-1}{2}$$

Packed Sharing

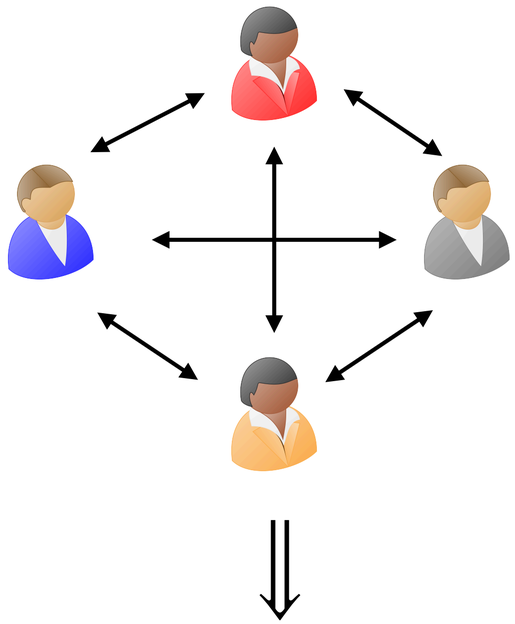


$\ell = O(n) \Longrightarrow O(\,|\,C\,|\,)$ communication MPC

$$t = \left(\frac{1}{2} - \epsilon\right) \cdot n$$

Computing The Garbled Circuit

Pre-processing Phase

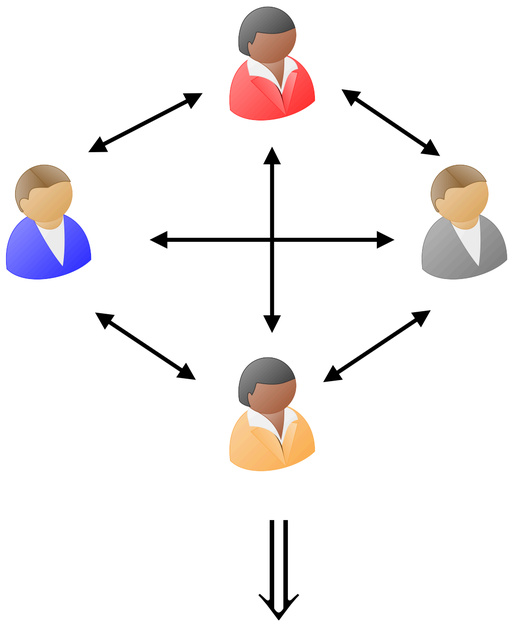


$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Garbling Phase



$\mathscr{G}$ - Garbled circuit

$\{X_{\text{inp}}\}$ - Wire labels for inputs

Evaluation Phase

$\text{Eval}(\mathscr{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathscr{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathscr{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathscr{G}, \{X_{\text{inp}}\})$

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

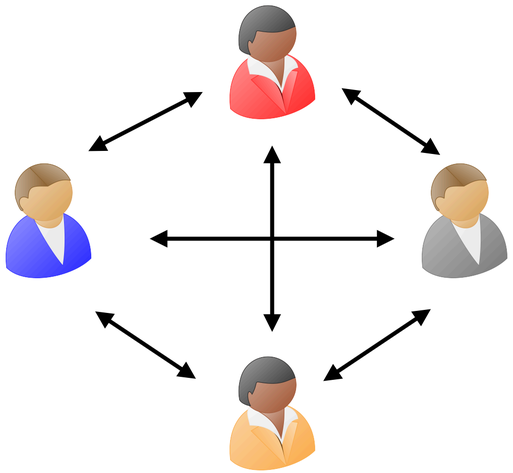
Generating shares of random elements in $\mathbb{F}^{\ell}$ [HN06].

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random elements in $\mathbb{F}^{\ell}$ [HN06].

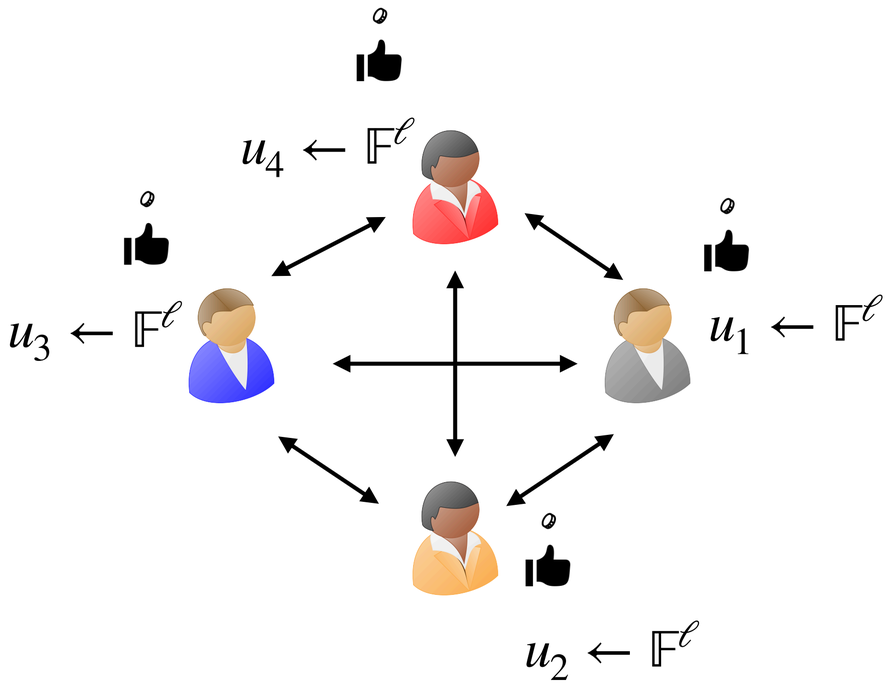


Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random elements in $\mathbb{F}^\ell$ [HN06].

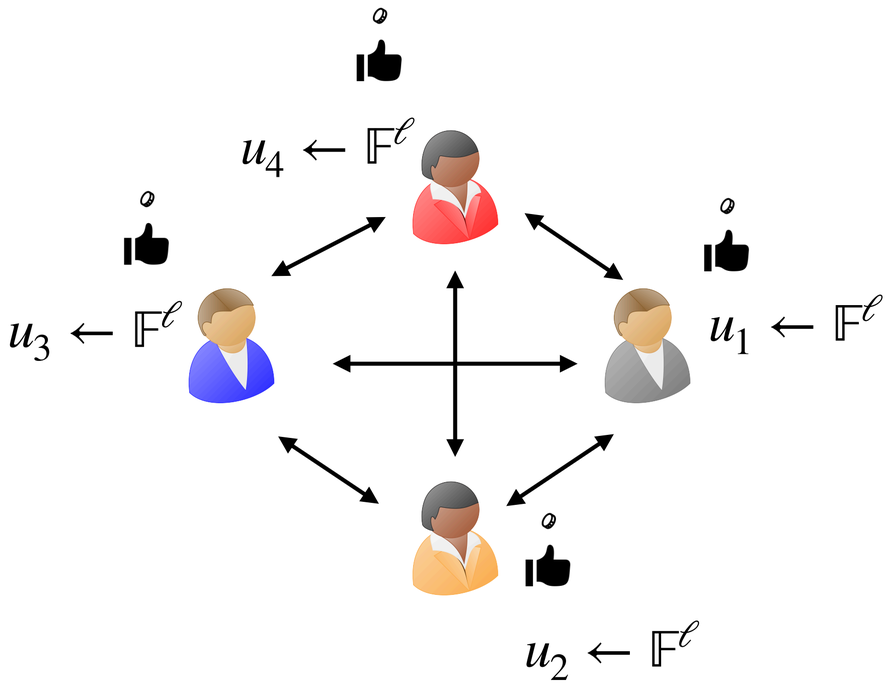


Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random elements in $\mathbb{F}^\ell$ [HN06].



$[u_1]$

$[u_2]$

$[u_3]$

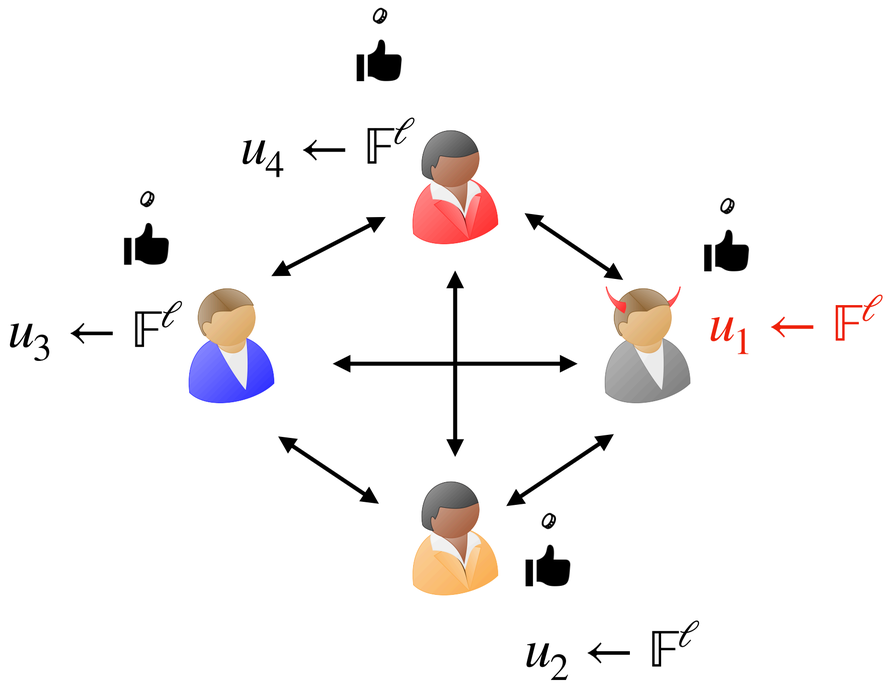
$[u_4]$

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random vectors in $\mathbb{F}^\ell$ [HN06].



$[u_1]$

$[u_2]$

$[u_3]$

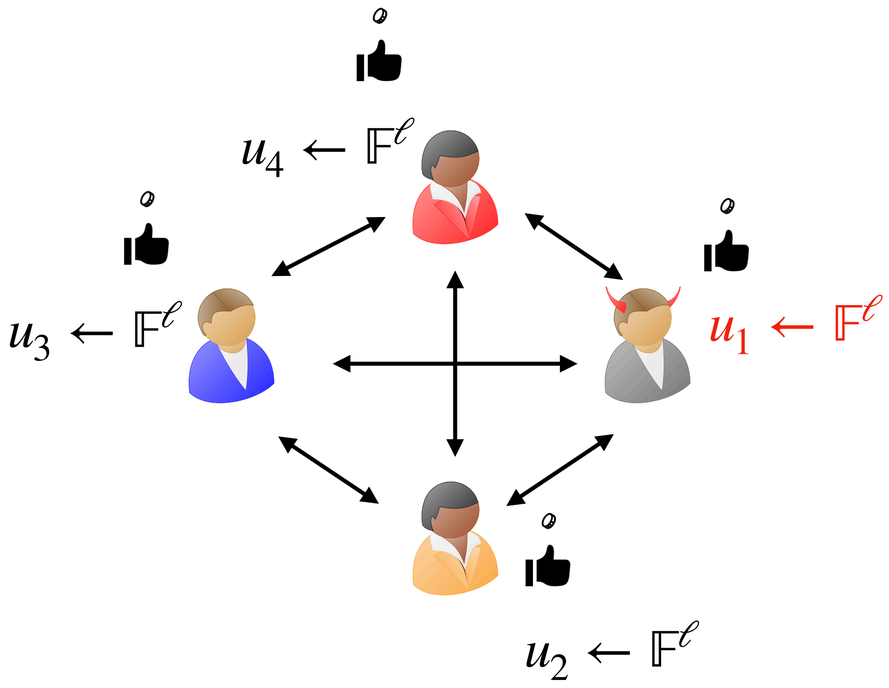
$[u_4]$

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random vectors in $\mathbb{F}^\ell$ [HN06].



Super-invertible matrix over $\mathbb{F}$

$$\begin{bmatrix} [u_1] \\ [u_2] \\ [u_3] \\ [u_4] \end{bmatrix} = \begin{bmatrix} [r_1] \\ [r_2] \\ [r_3] \end{bmatrix}$$

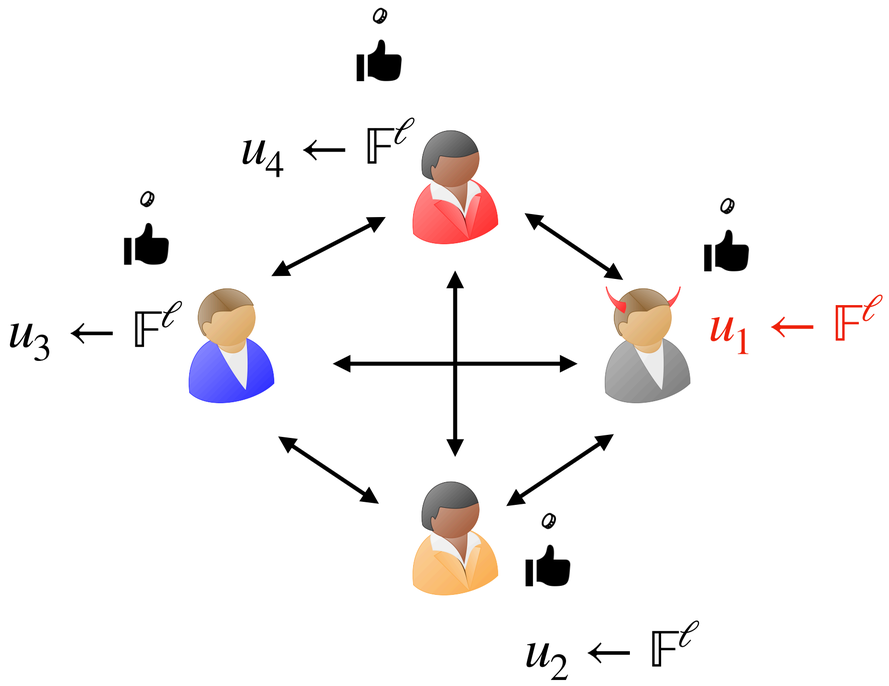
$n - t$ outputs

Computing Shares of Wire Labels

- Wire labels are LPN keys.
- LPN over larger fields since $|\mathbb{F}| > n + \ell$ due to packed secret sharing.

Number of secrets per sharing

Generating shares of random vectors in $\mathbb{F}^\ell$ [HN06].



Super-invertible matrix over $\mathbb{F}$

$$\begin{bmatrix} [u_1] \\ [u_2] \\ [u_3] \\ [u_4] \end{bmatrix} = \begin{matrix} n - t \text{ outputs} \\ [r_1] \\ [r_2] \\ [r_3] \end{matrix}$$

Total communication is $O(n^2)$.
- $\ell = O(n)$
- $O(n)$ outputs

Amortized communication is $O(1)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Computing Shares Of Random Bits

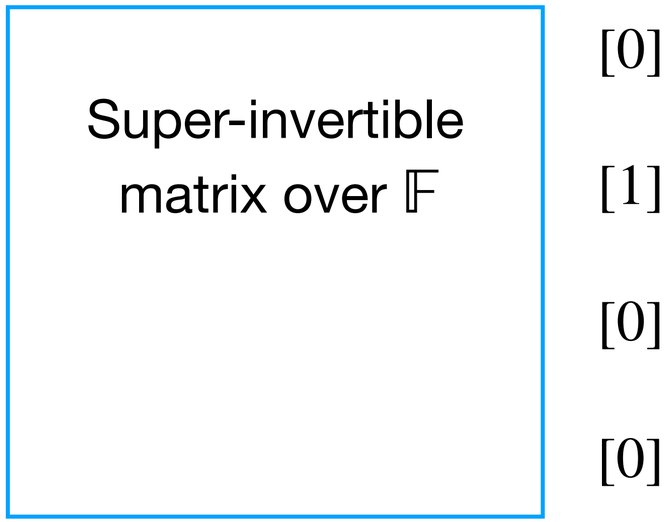
- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

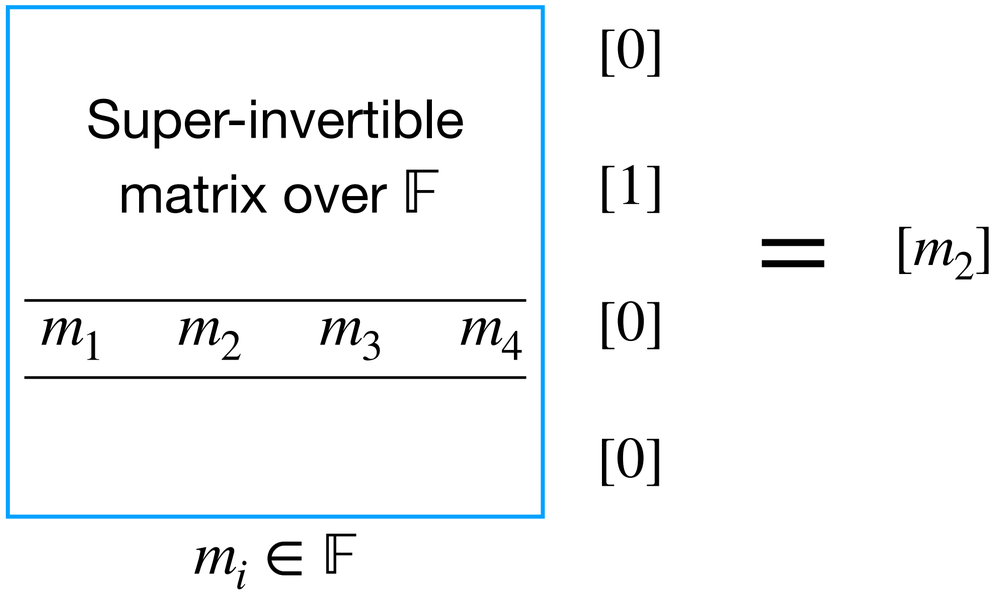
Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

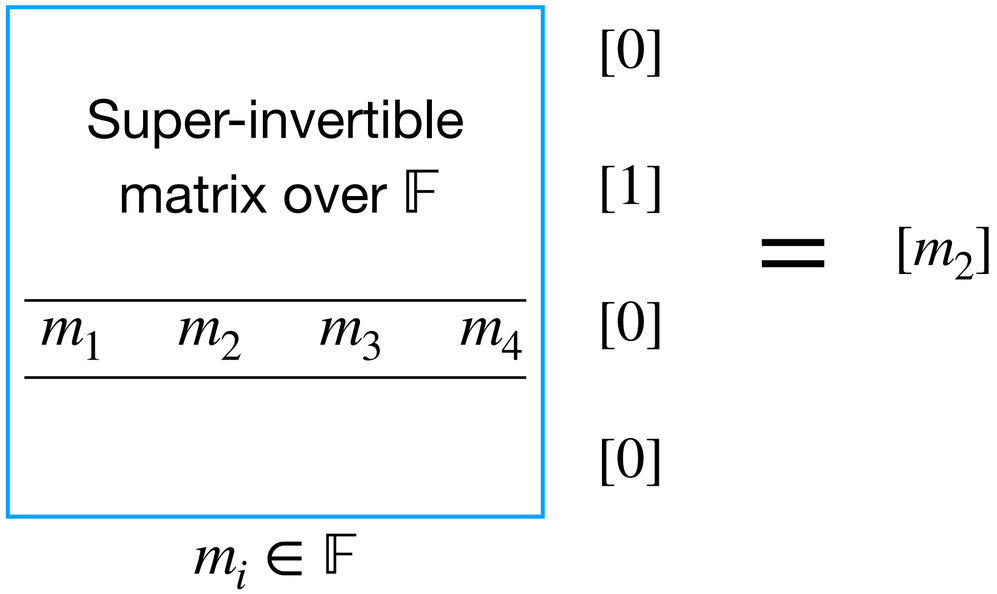
Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.

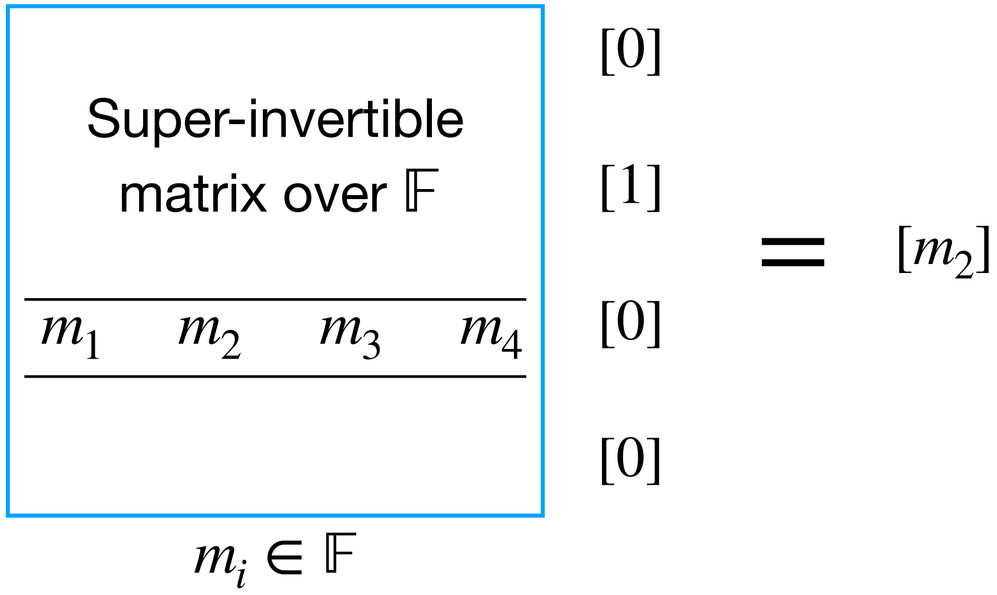


Arbitrary super-invertible matrices over $\mathbb{F}$ will not work.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



Arbitrary super-invertible matrices over $\mathbb{F}$ will not work.

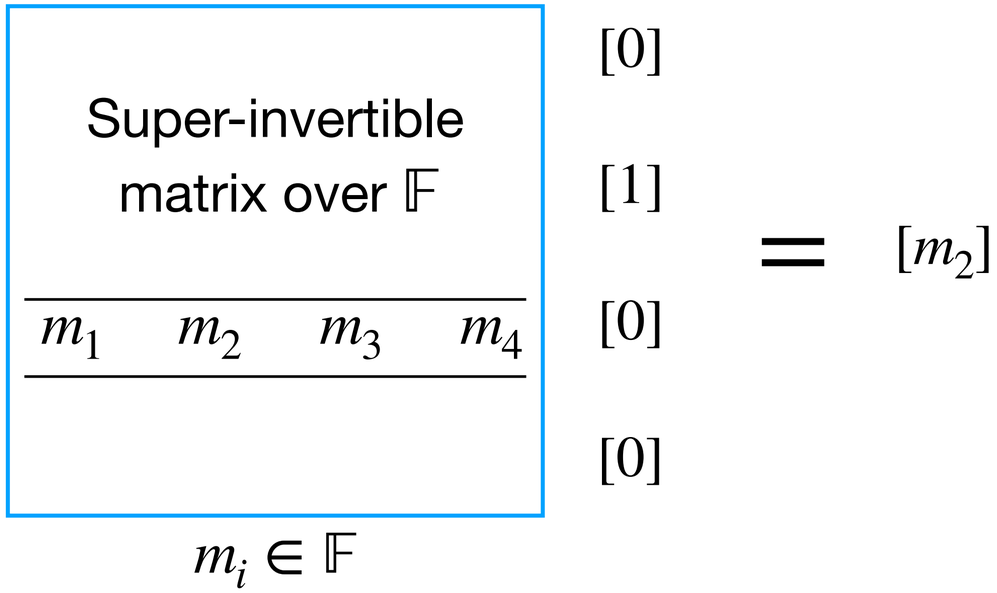
The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



Arbitrary super-invertible matrices over $\mathbb{F}$ will not work.

The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

MPC is done over $\mathbb{F} = \mathrm{GF}(2^k)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.

The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.

$$[u_1]$$

$$[u_2]$$

$$[u_3]$$

$$[u_4]$$

$$u_i \in \{0,1\}^{\ell}$$

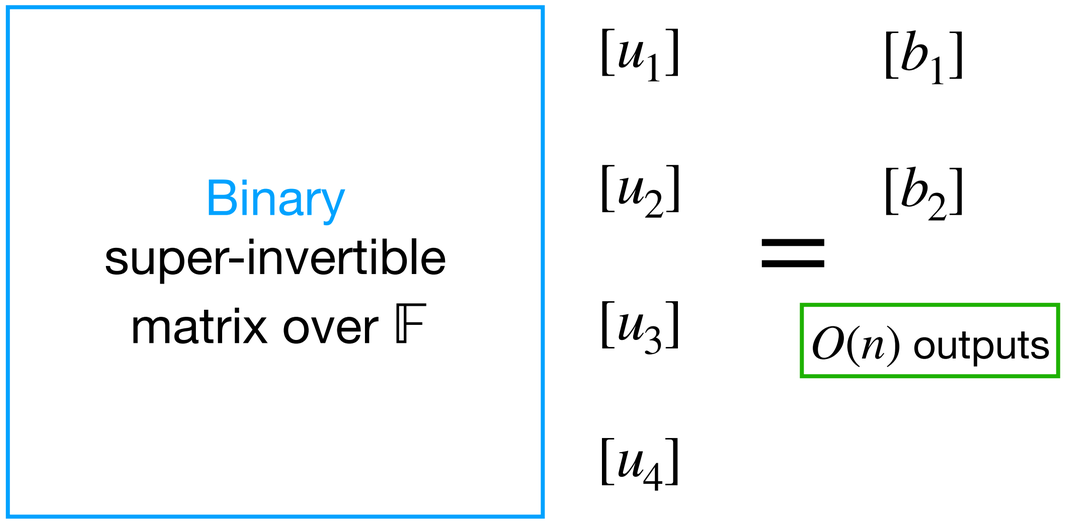
The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



$u_i \in \{0,1\}^\ell$

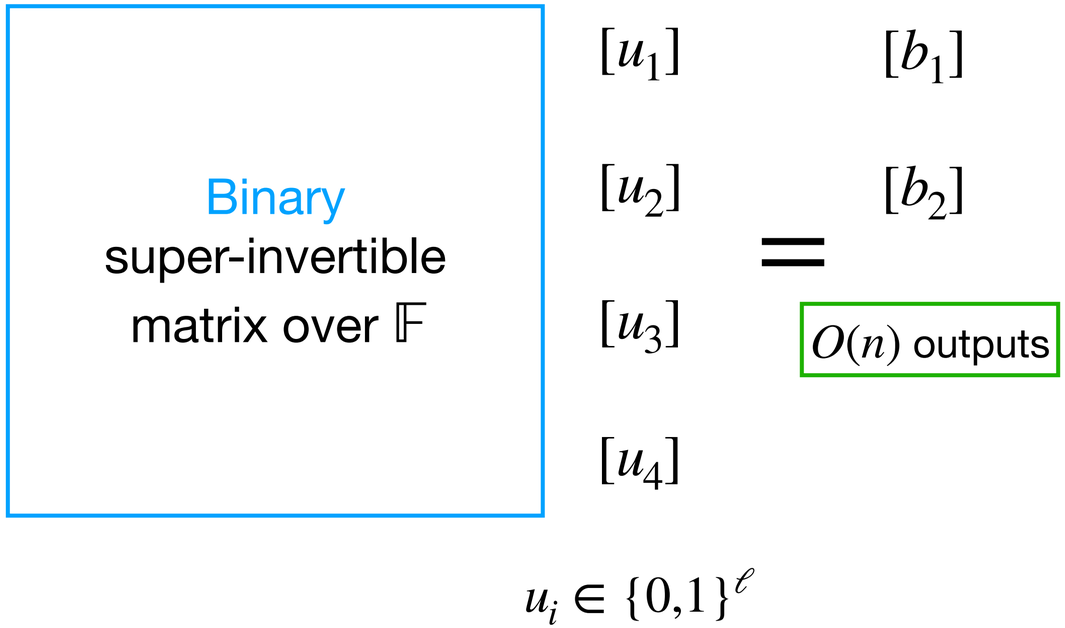
The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

Computing Shares Of Random Bits

- Secret sharings of random bits
  - Colors for wire labels.
  - Compute shares of LPN errors.

Problem: Secret sharing of a uniformly random bit over a larger field $\mathbb{F}$.



The generator matrix of any binary linear error correcting code is super-invertible over a field of characteristic 2.

Construct binary super-invertible matrix with order $n \times O(n)$.

Total communication is $O(n^2)$.

- $\ell = O(n)$
- $O(n)$ outputs

Amortized communication is $O(1)$.

Computing Shares Of LPN Errors

- Computing shares of secrets sampled from a Bernoulli distribution over $\mathbb{F}$.

$$P\left[\epsilon \leftarrow \mathbb{F}^{\ell}\right] = 2^{-p}$$

$$P\left[\epsilon = 0^{\ell}\right] = 1 - 2^{-p}$$

p is a constant

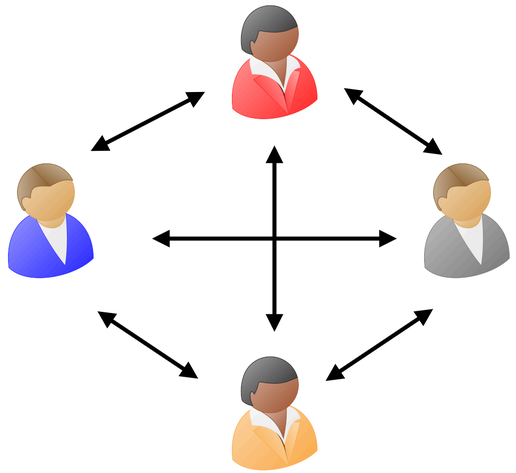
Computing Shares Of LPN Errors

- Computing shares of secrets sampled from a Bernoulli distribution over $\mathbb{F}$.

$$P\left[\epsilon \leftarrow \mathbb{F}^{\ell}\right] = 2^{-p}$$

$$P\left[\epsilon = 0^{\ell}\right] = 1 - 2^{-p}$$

p is a constant



$$[b_1] \quad [b_2] \quad \ldots \quad [b_p]$$

Packed sharing of random bits

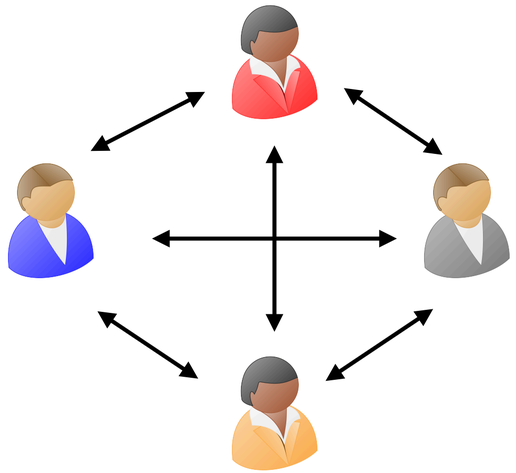
Computing Shares Of LPN Errors

- Computing shares of secrets sampled from a Bernoulli distribution over $\mathbb{F}$.

$$P\left[\epsilon \leftarrow \mathbb{F}^{\ell}\right] = 2^{-p}$$

$$P\left[\epsilon = 0^{\ell}\right] = 1 - 2^{-p}$$

p is a constant



$$[b_1] \times [b_2] \times \ldots \times [b_p]$$

Packed sharing of random bits

$$[\beta]$$

$$P[\beta = 1] = 2^{-p}$$

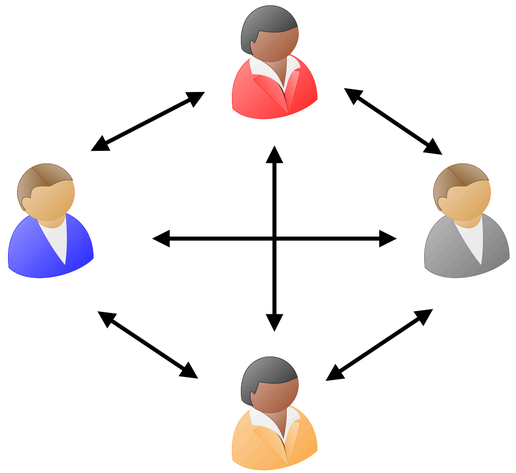
Computing Shares Of LPN Errors

- Computing shares of secrets sampled from a Bernoulli distribution over $\mathbb{F}$.

$$P\left[\epsilon \leftarrow \mathbb{F}^{\ell}\right] = 2^{-p}$$

$$P\left[\epsilon = 0^{\ell}\right] = 1 - 2^{-p}$$

p is a constant



$$[b_1] \times [b_2] \times \ldots \times [b_p]$$

Packed sharing of random bits

$$[\beta] \times [r]$$

$$P[\beta = 1] = 2^{-p}$$

$$\downarrow$$

$$[\epsilon]$$

Packed sharing of random elements

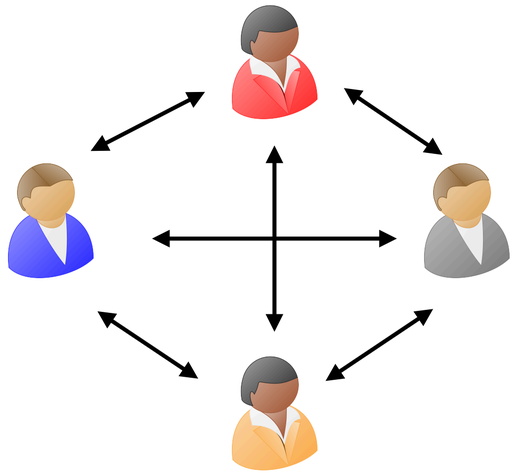
Computing Shares Of LPN Errors

- Computing shares of secrets sampled from a Bernoulli distribution over $\mathbb{F}$.

$$P\left[\epsilon \leftarrow \mathbb{F}^{\ell}\right] = 2^{-p}$$

$$P\left[\epsilon = 0^{\ell}\right] = 1 - 2^{-p}$$

p is a constant



$$[b_1] \times [b_2] \times \ldots \times [b_p]$$

Packed sharing of random bits

$$[\beta] \times [r]$$

$$P[\beta = 1] = 2^{-p}$$

$$\downarrow$$

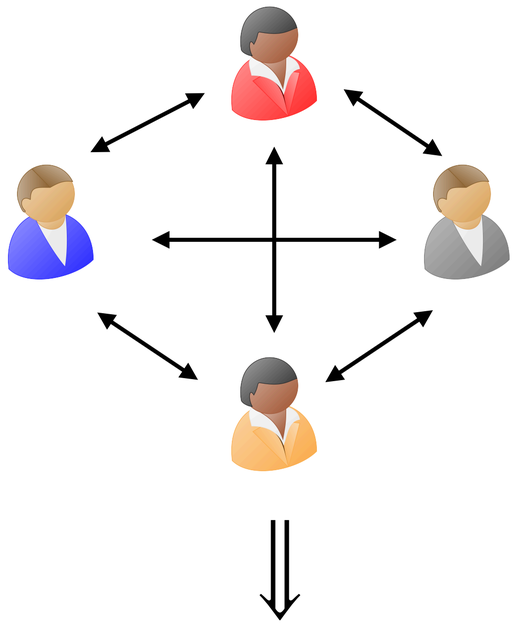
$$[\epsilon]$$

Packed sharing of random elements

$O(1)$ communication

Computing The Garbled Circuit

Pre-processing Phase

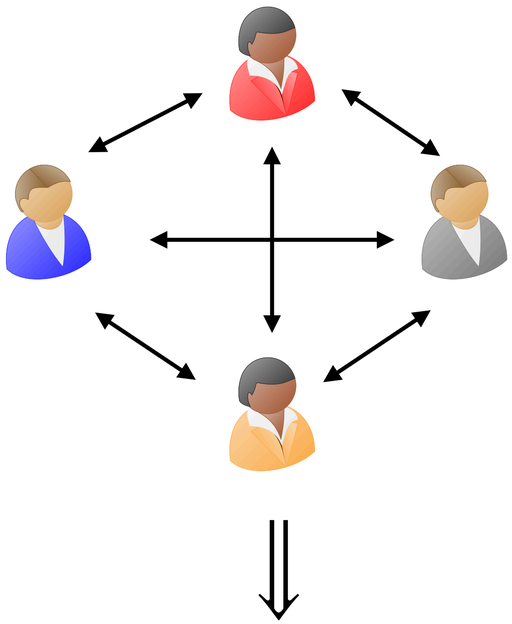


$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Garbling Phase



$\mathcal{G}$ - Garbled circuit

$\{X_{\text{inp}}\}$ - Wire labels for inputs

Evaluation Phase

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

Garbling Phase

Pre-processing phase

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

Garbling Phase

Pre-processing phase

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

Garbling phase

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

Garbling Phase

Pre-processing phase

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

Garbling phase

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

Shuffle ciphertexts to compute $\mathscr{G}$

Garbling Phase

Pre-processing phase

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

$[X][Y][Z]$
$[\epsilon_X][\epsilon_Y]$

Garbling phase

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

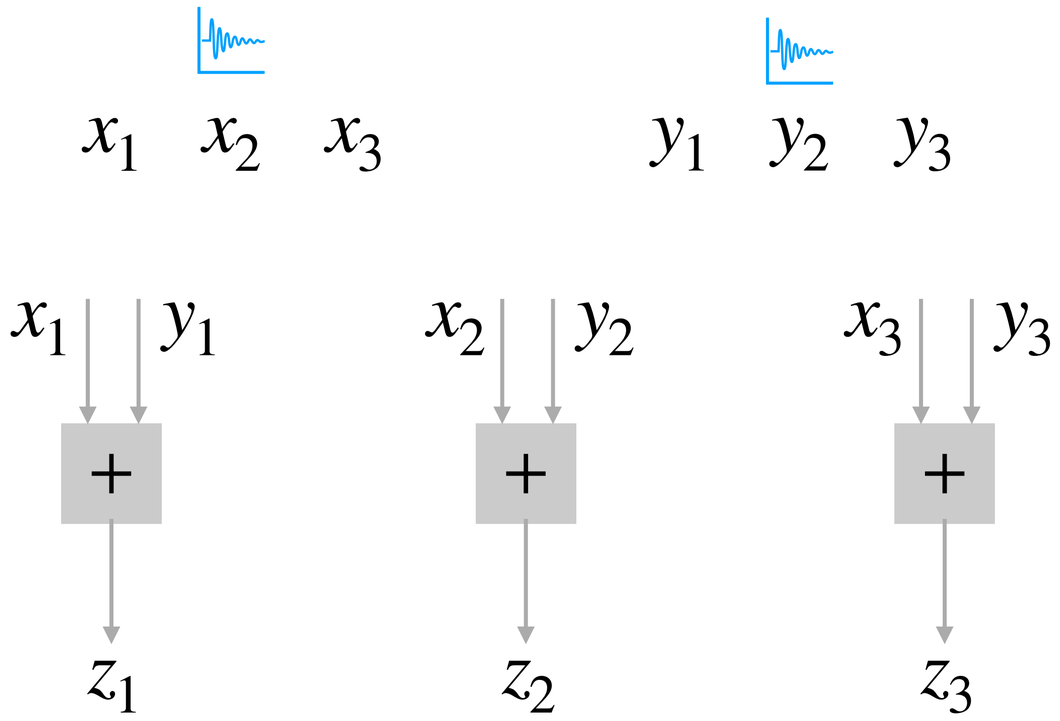
$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

$[R_X] + [R_Y] + [Z]$
$= [\mathsf{Enc}_{X,Y}(Z)]$

Shuffle ciphertexts to compute $\mathscr{G}$

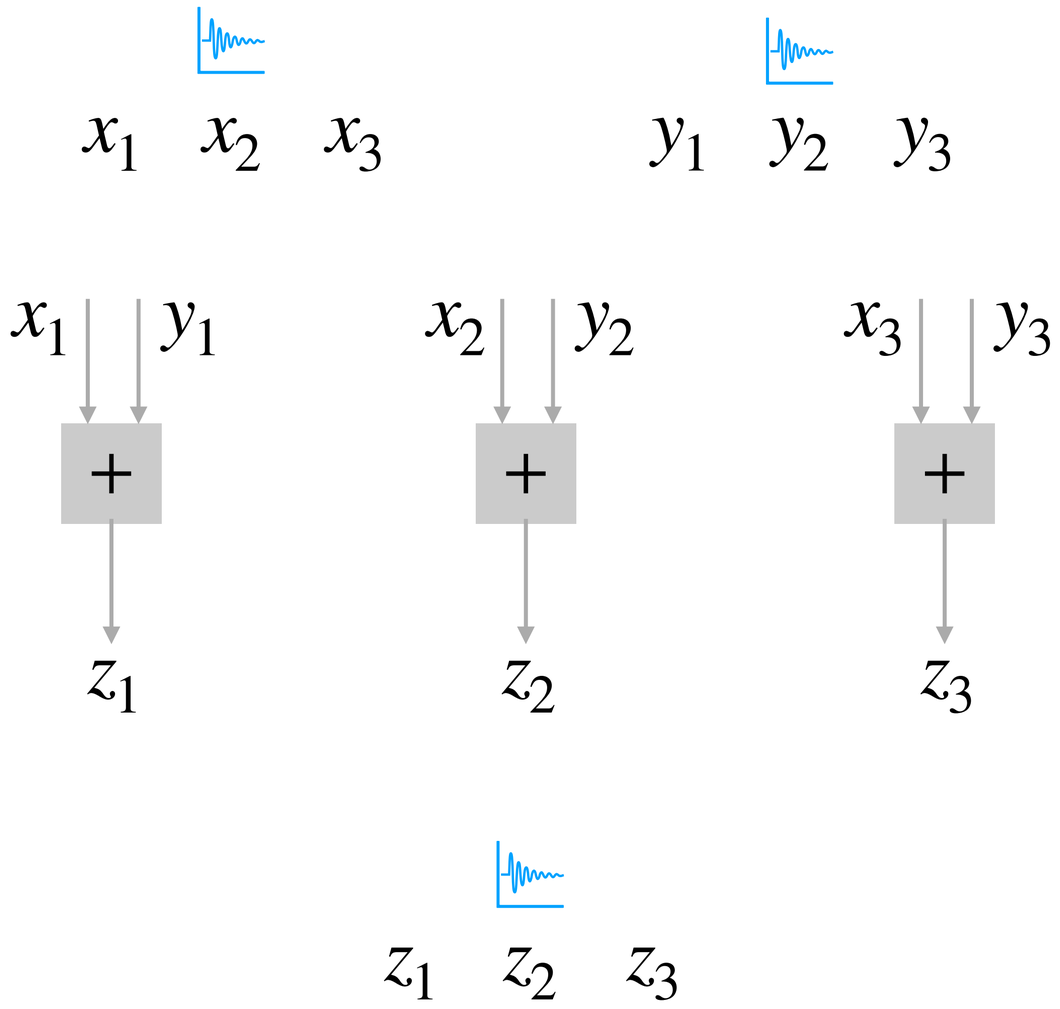
Alignment of secrets in packed sharing.

Alignment Of Secrets In Packed Sharing



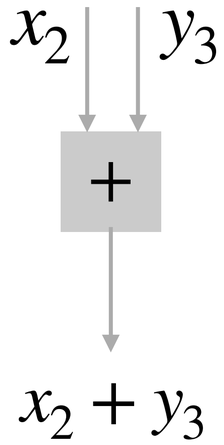
Alignment Of Secrets In Packed Sharing

Computation is straightforward when secrets are aligned.



Alignment Of Secrets In Packed Sharing

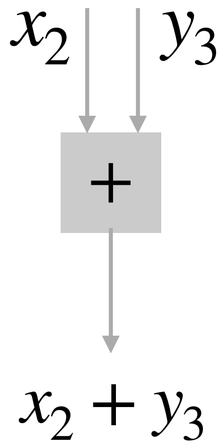
Computation is straightforward when secrets are aligned.



Alignment Of Secrets In Packed Sharing

Computation is straightforward when secrets are aligned.

Packed sharing requires secrets to be re-aligned.



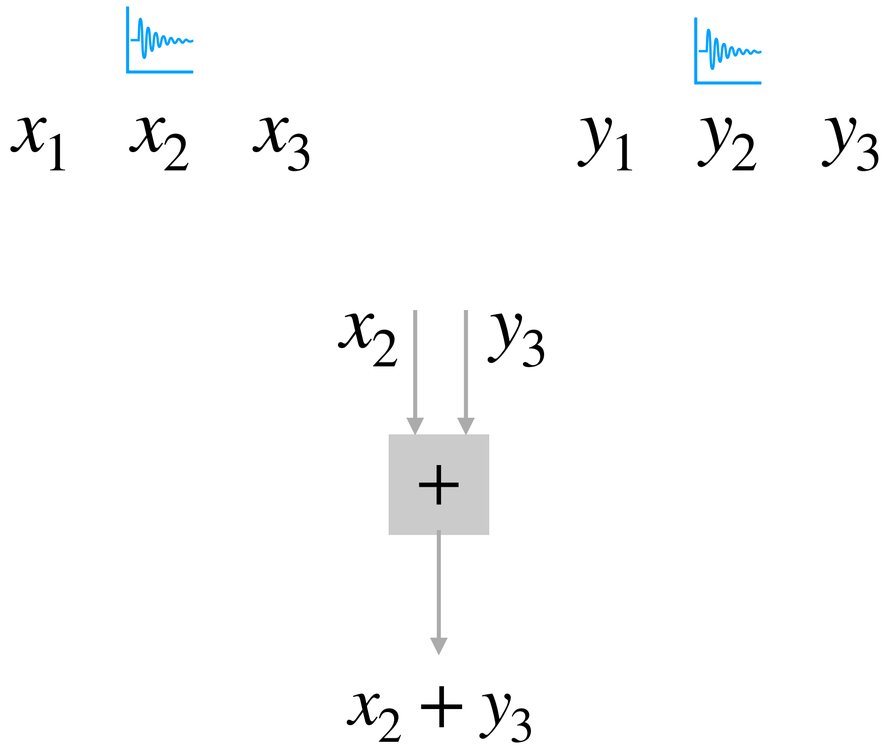
Alignment Of Secrets In Packed Sharing

Computation is straightforward when secrets are aligned.

Packed sharing requires secrets to be re-aligned.

Bottleneck is number of distinct re-alignments:

- Bound number of distinct re-alignments [DIK10, GIP15, GPS21].
- Restrict circuit type [BGJK21].
- Efficiently perform any re-alignment [GPS22].



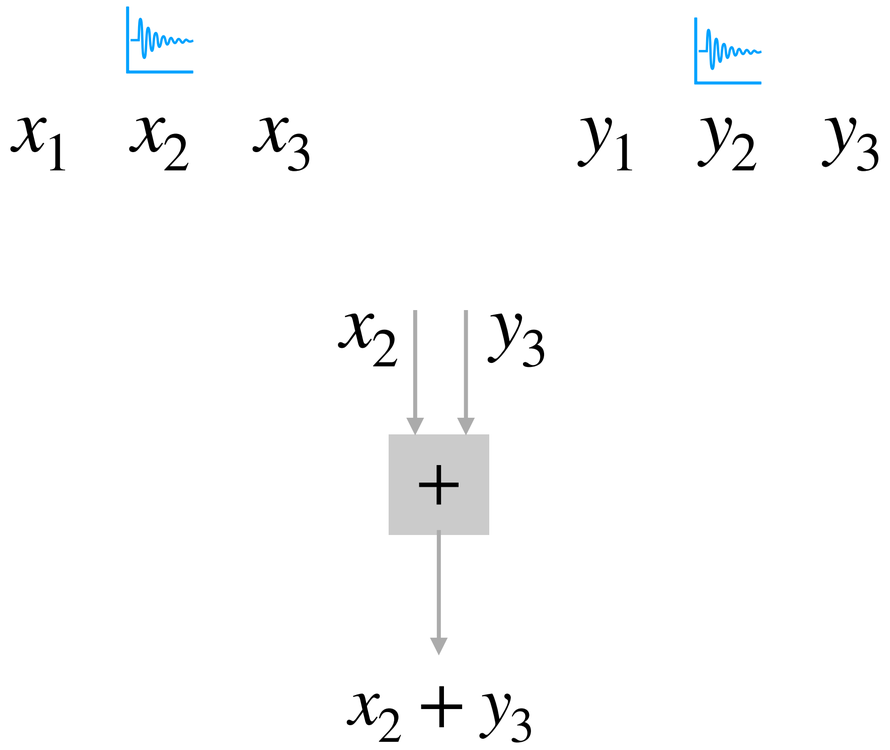
Alignment Of Secrets In Packed Sharing

Computation is straightforward when secrets are aligned.

Packed sharing requires secrets to be re-aligned.

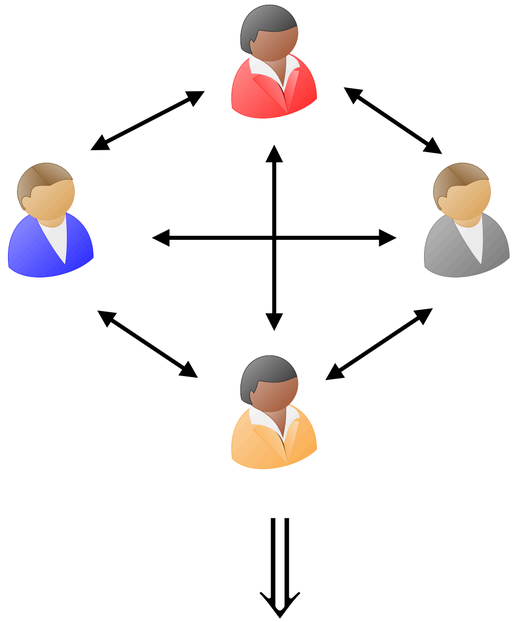
Bottleneck is number of distinct re-alignments:

- Bound number of distinct re-alignments [DIK10, GIP15, GPS21].
- Restrict circuit type [BGJK21].
- Efficiently perform any re-alignment [GPS22].



Computing The Garbled Circuit

Pre-processing Phase

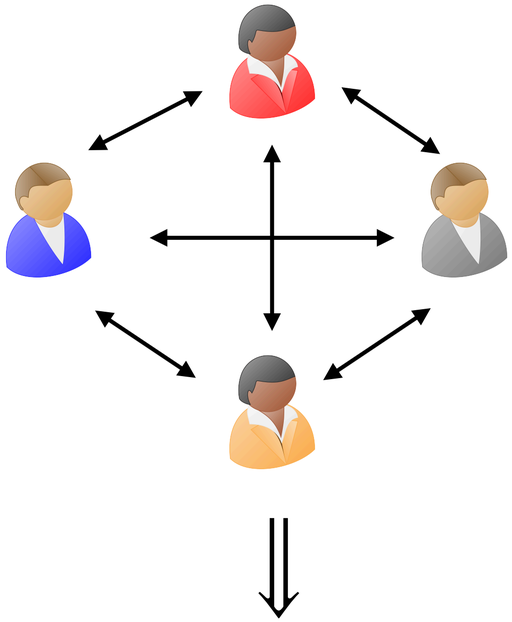


$[k]$ - Wire labels (LPN Keys)

$[\epsilon]$ - LPN Errors

$[b]$ - Wire label colors

Garbling Phase



$\mathcal{G}$ - Garbled circuit

$\{X_{\text{inp}}\}$ - Wire labels for inputs

Evaluation Phase

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

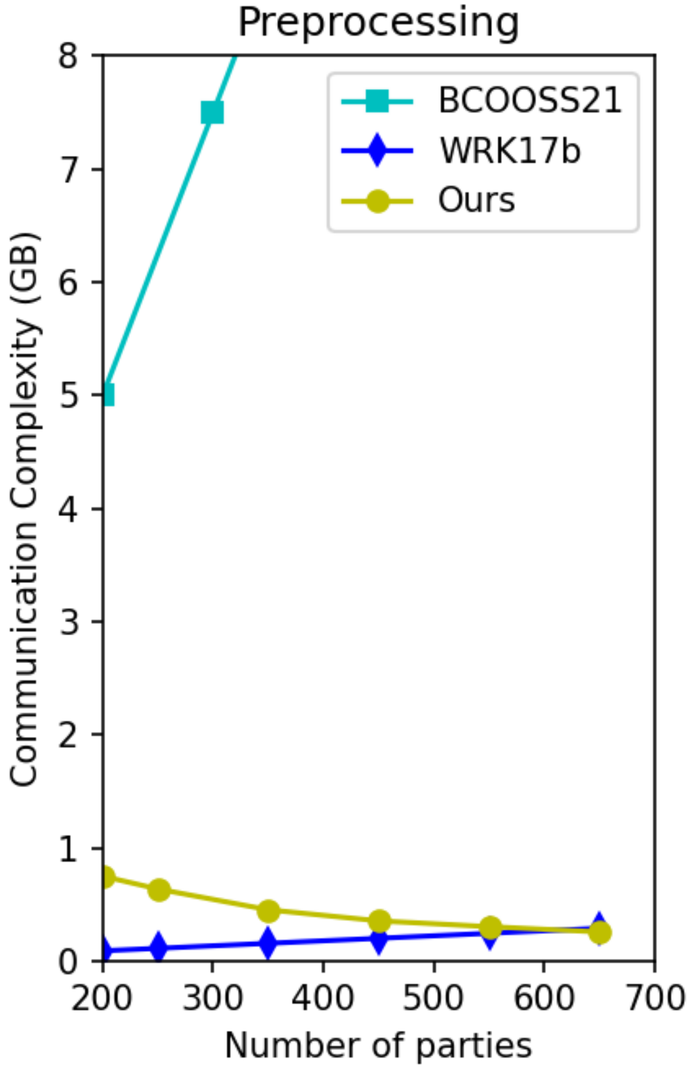
$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

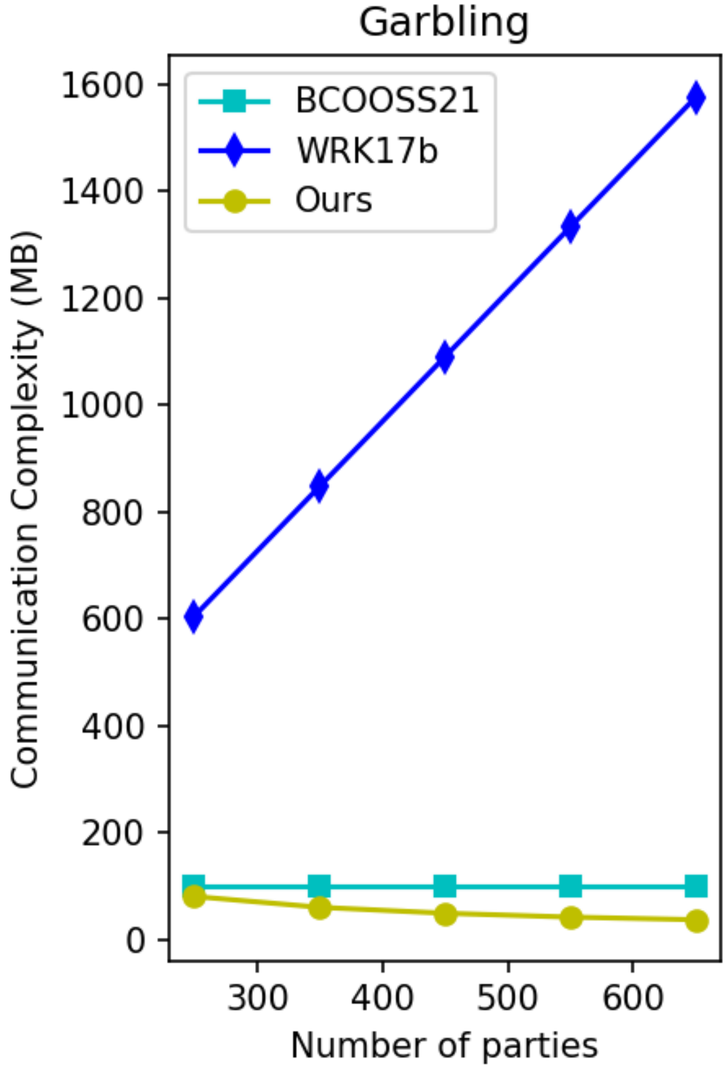
$\text{Eval}(\mathcal{G}, \{X_{\text{inp}}\})$

Agenda

- Structure Of Garbled Circuit
- Multiparty Garbling Protocol
- Efficiency Analysis

Communication Cost Estimates For Evaluating AES-128





Maliciously secure protocol

Per party costs

$$t \approx \frac{n}{4}$$

$$\ell \approx \frac{n}{4}$$

Bottlenecks To Concrete Efficiency

Semi-honest Protocol

- AES-128
- 8 threads per party
- $n = 250$
- $t = 50$
- $\ell = 50$

	Computation	Communication
Pre-processing Phase		
Total	1.94 min	240 MB

Bottlenecks To Concrete Efficiency

Semi-honest Protocol

- AES-128
- 8 threads per party
- $n = 250$
- $t = 50$
- $\ell = 50$

	Computation	**Communication**
Pre-processing Phase		
Total	1.94 min	240 MB
Error Generation	1.73 min	196 MB

Bottlenecks To Concrete Efficiency

Semi-honest Protocol

- AES-128
- 8 threads per party
- $n = 250$
- $t = 50$
- $\ell = 50$

	Computation	Communication
Pre-processing Phase		
Total	1.94 min	240 MB
Error Generation	1.73 min	196 MB

Low rate of binary error correcting codes.

Bottlenecks To Concrete Efficiency

Semi-honest Protocol

- AES-128
- 8 threads per party
- $n = 250$
- $t = 50$
- $\ell = 50$

	Computation	Communication
Pre-processing Phase		
Total	1.94 min	240 MB
Error Generation	1.73 min	196 MB
Garbling Phase		
Total	1.64 min	43 MB

Bottlenecks To Concrete Efficiency

Semi-honest Protocol

- AES-128
- 8 threads per party
- $n = 250$
- $t = 50$
- $\ell = 50$

	Computation	Communication
Pre-processing Phase		
Total	1.94 min	240 MB
Error Generation	1.73 min	196 MB
Garbling Phase		
Total	1.64 min	43 MB
Re-aligning Secrets	27.66 sec	28 MB

Future Directions

- Improving concrete efficiency
  - Concretely efficient protocol for error generation.
  - Incorporating existing GC optimizations.
- Constructing $O(|C|)$ communication multiparty garbling from weaker assumptions.

Thank you

ia.cr/2023/099