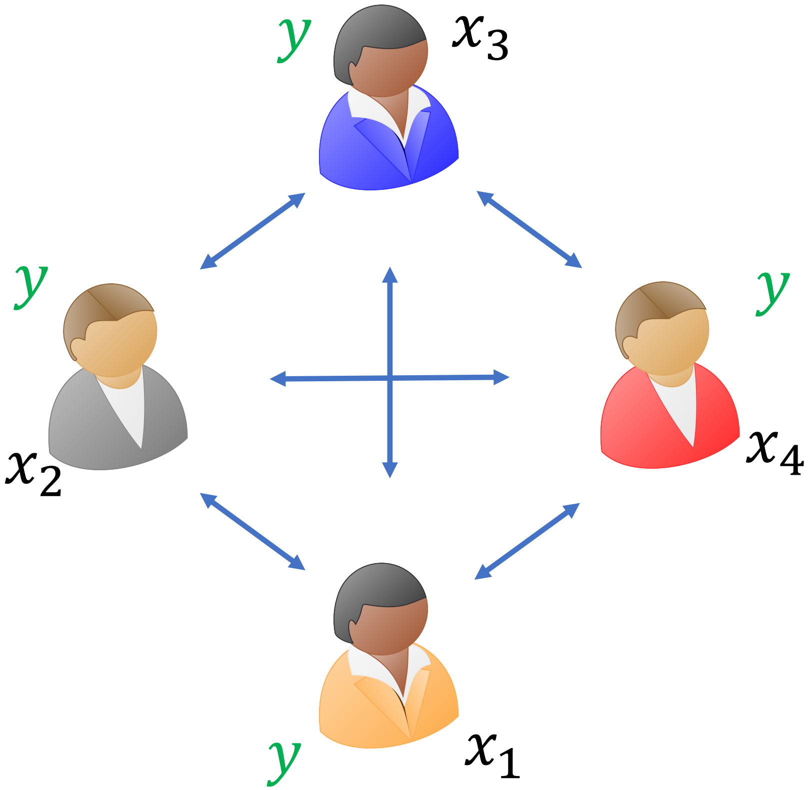
Scalable Multiparty Garbling

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

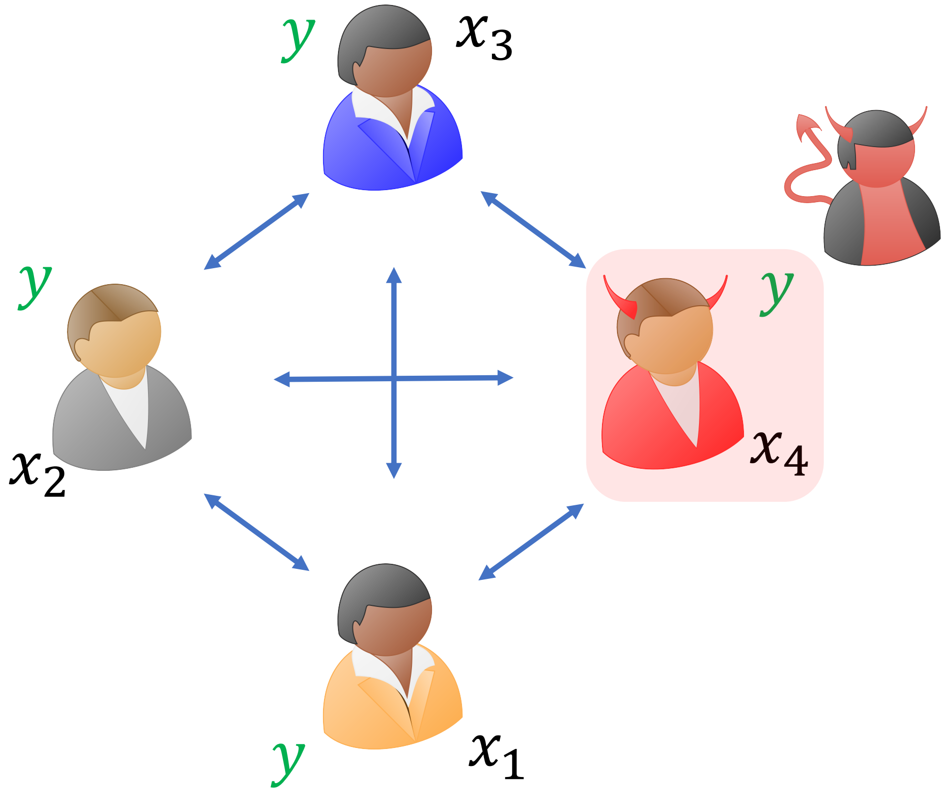
Johns Hopkins University   NTT Research   Massachusetts Institute of Technology   Boston University

Secure Multiparty Computation (MPC)



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

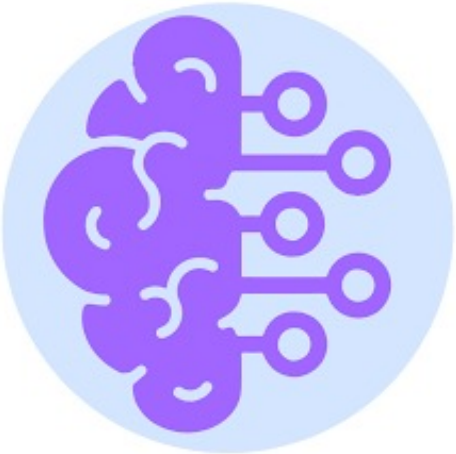
Secure Multiparty Computation (MPC)



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

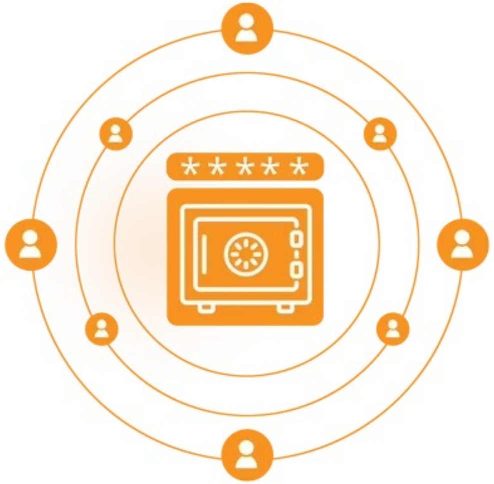
Adversary learns nothing beyond input of corrupt parties and function output

Applications Of MPC



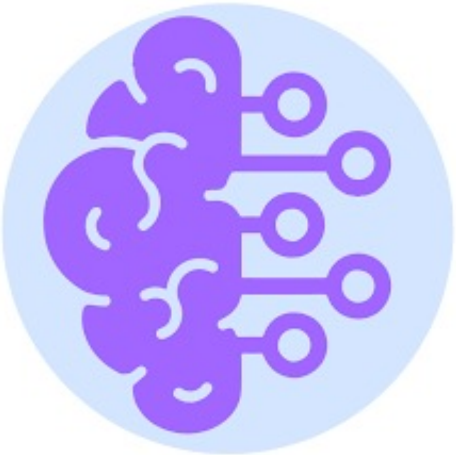
Machine learning on distributed datasets

Data as a Service (DaaS)



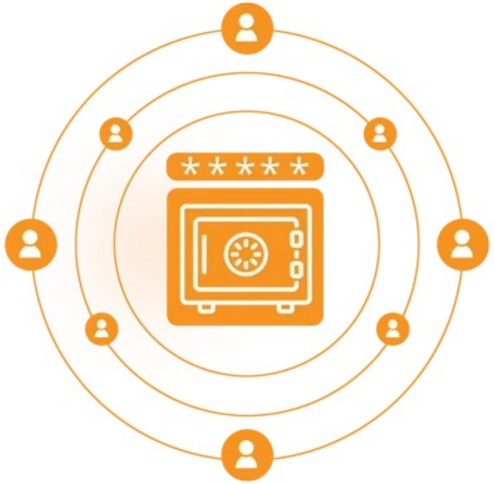
Securing digital assets and key management

Applications Of MPC



Machine learning on distributed datasets

Data as a Service (DaaS)

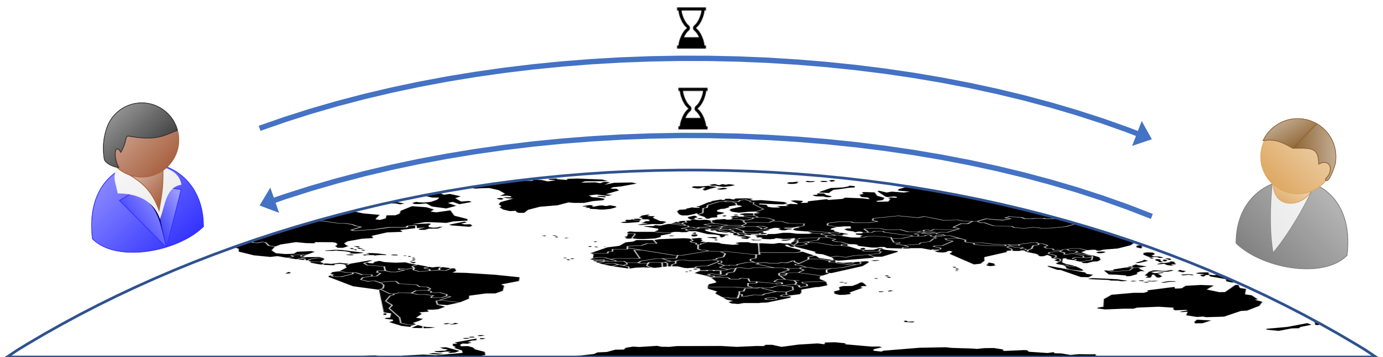


Securing digital assets and key management

Large computations over the internet

MPC Over The Internet

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

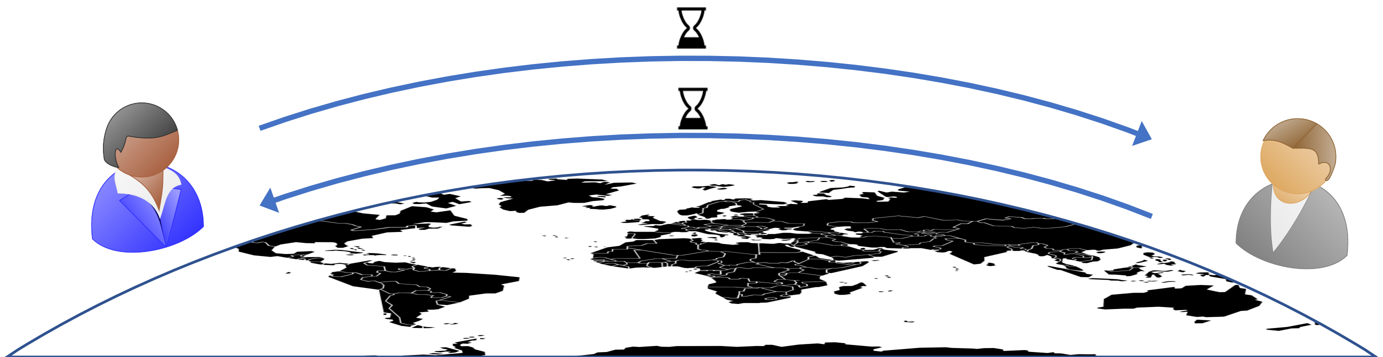


MPC Over The Internet

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

Constant Round MPC: Parties interact constant number of times

- Two approaches: FHE-based and Multiparty Garbling



Problem: Getting Best Of Both Worlds

n → Number of parties

$|C|$ → Size of circuit

Dishonest majority, multiparty garbling protocols with total communication cost of

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

[HSS17, WRK17b, BCOOSS21]

Problem: Getting Best Of Both Worlds

n → Number of parties

$|C|$ → Size of circuit

Dishonest majority, multiparty garbling protocols with total communication cost of

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

[HSS17, WRK17b, BCOOSS21]

Honest majority, non-constant round MPC with total communication cost of

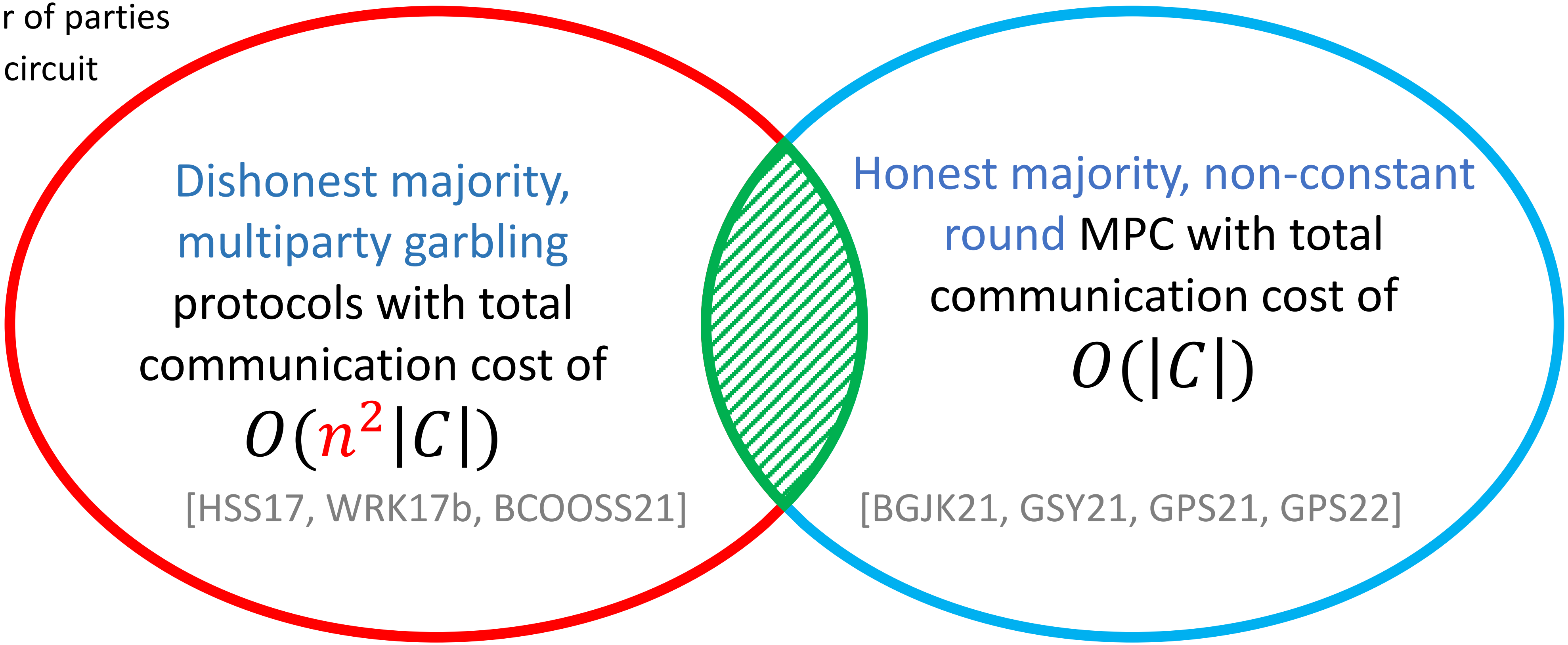
$$O(|C|)$$

[BGJK21, GSY21, GPS21, GPS22]

Problem: Getting Best Of Both Worlds

n → Number of parties

$|C|$ → Size of circuit



Is there 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

- Scales to hundreds of parties and can be used with large volunteer networks like Tor and Bitcoin

- Honest majority is a more plausible assumption

Our Contributions

Multiparty garbling protocol with $O(|C|)$ communication complexity

- Semi-honest and maliciously secure
- $t < n\left(\frac{1}{2} - \varepsilon\right)$ where $0 < \varepsilon < \frac{1}{2}$
- Based on Learning Parity with Noise (LPN) over large fields assumption
- Benchmarks and evaluation show that our protocol is practical

Outline

Template for multiparty garbling

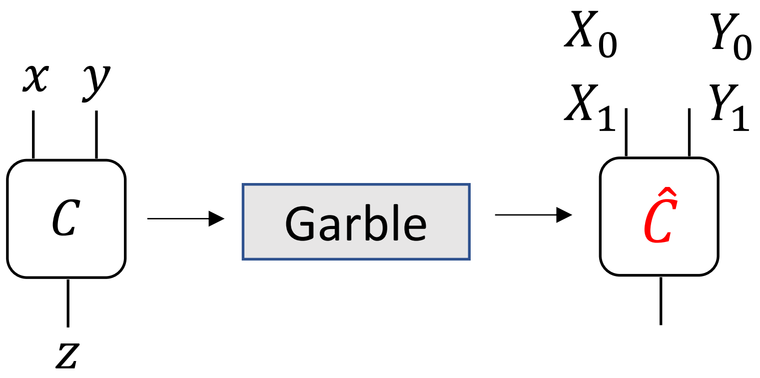
Overview of prior works

Key techniques in our protocol

Benchmarks and evaluation

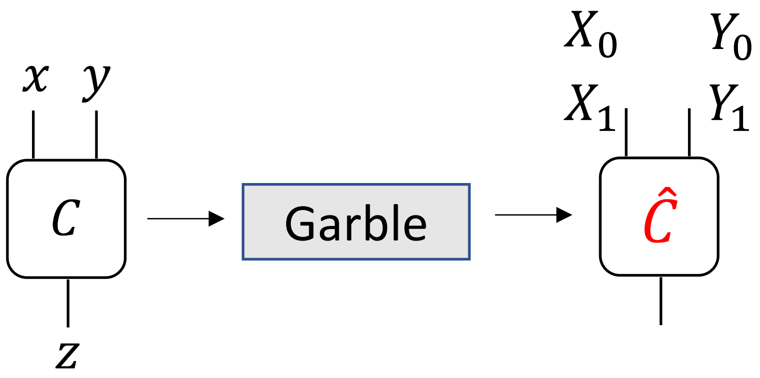
Review Of Garbled Circuits [Yao86]

Garbling

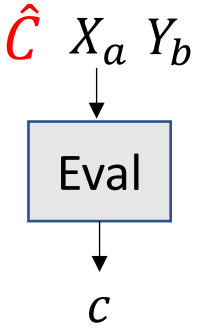


Review Of Garbled Circuits [Yao86]

Garbling



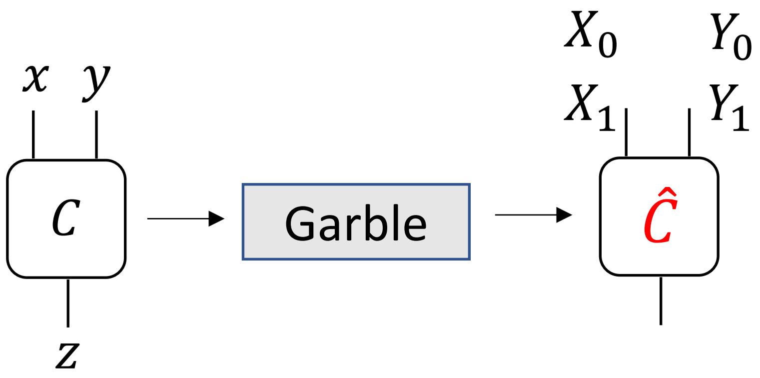
Evaluation



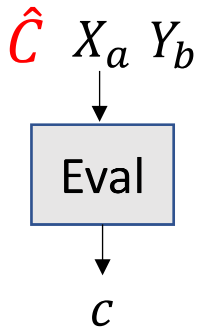
$c = C(a, b)$

Review Of Garbled Circuits [Yao86]

Garbling



Evaluation

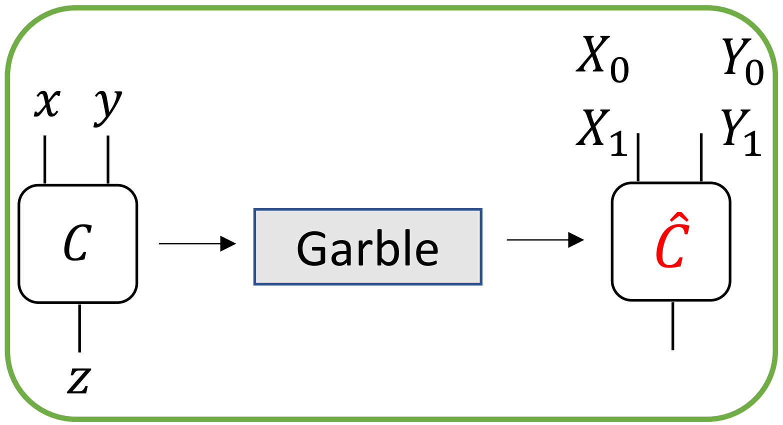


$$\boxed{c = C(a, b)}$$

Adversary having only $\hat{C}$ and one label per wire does not learn anything beyond the output

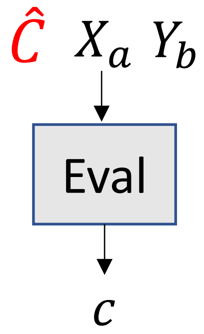
Review Of Garbled Circuits [Yao86]

Garbling



Garble the circuit using MPC!

Evaluation

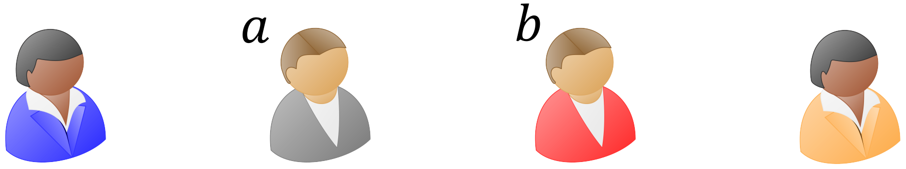


$$\boxed{c = C(a, b)}$$

Adversary having only $\hat{C}$ and one label per wire does not learn anything beyond the output

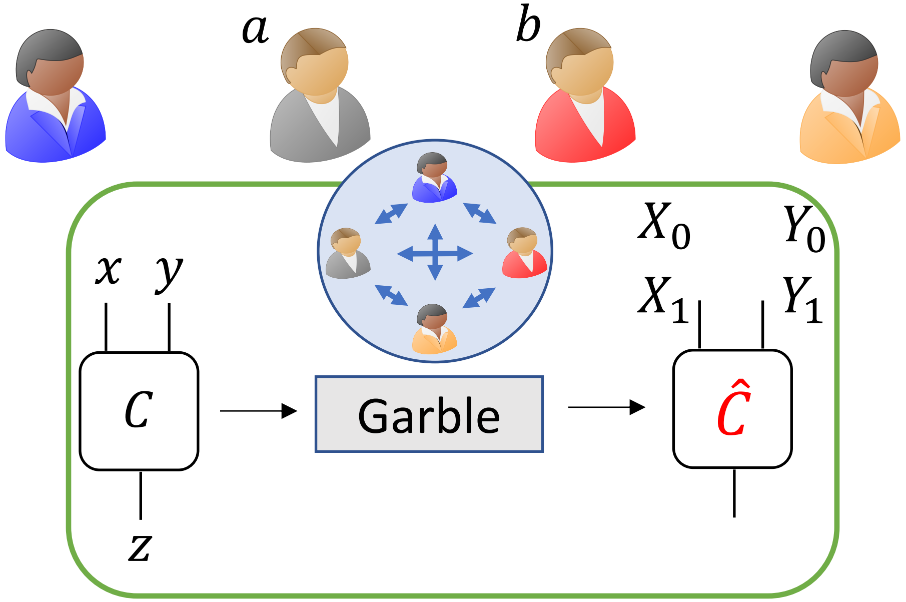
Template For Multiparty Garbling[BMR90]

Garbling Phase



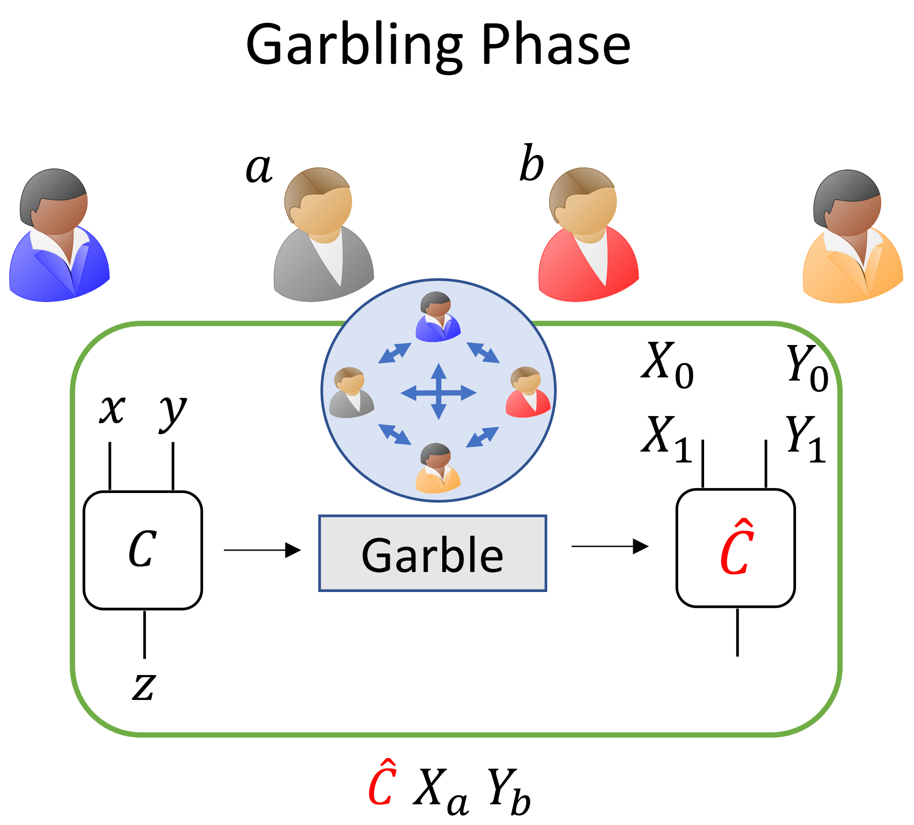
Template For Multiparty Garbling[BMR90]

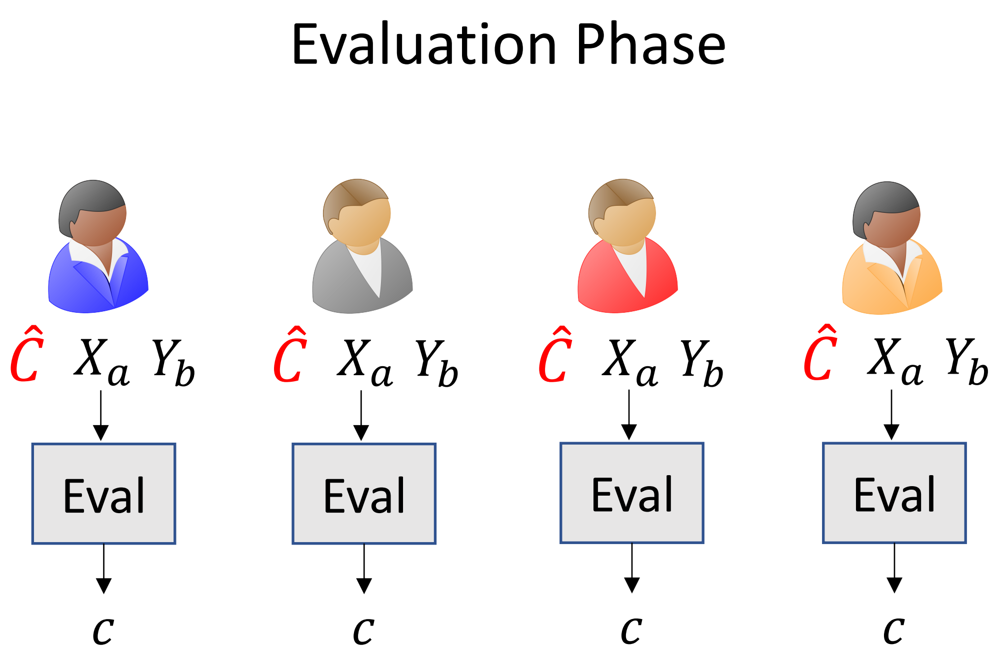
Garbling Phase



$\hat{C}$ X_a Y_b

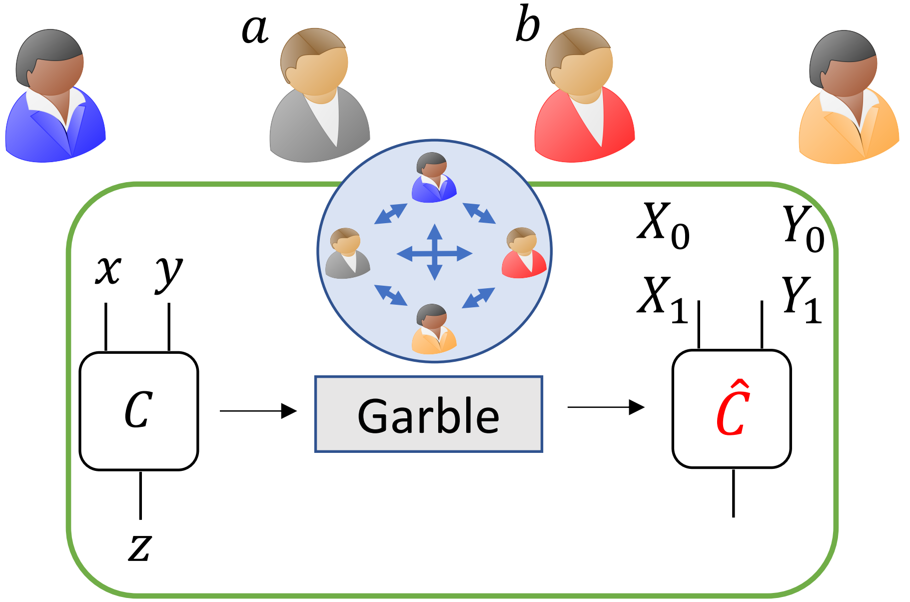
Template For Multiparty Garbling[BMR90]





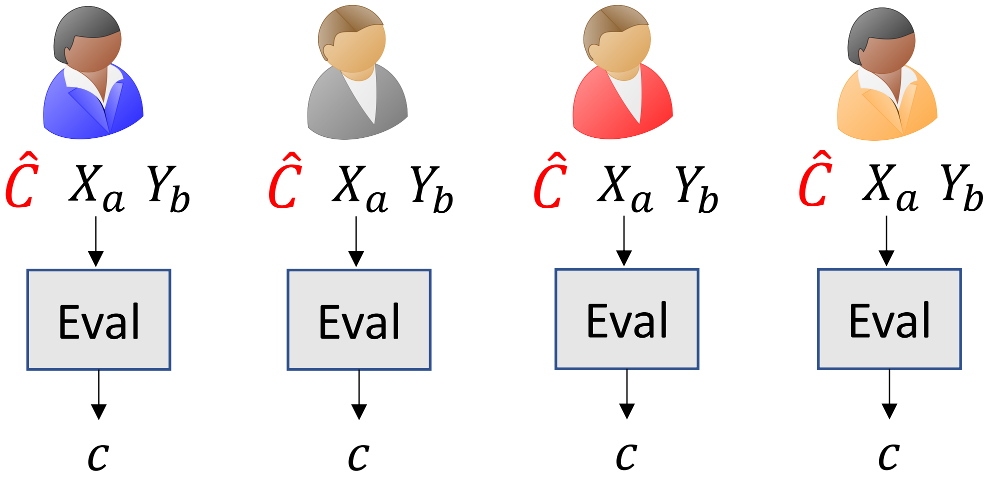
Template For Multiparty Garbling[BMR90]

Garbling Phase



$\hat{C}$ X_a Y_b

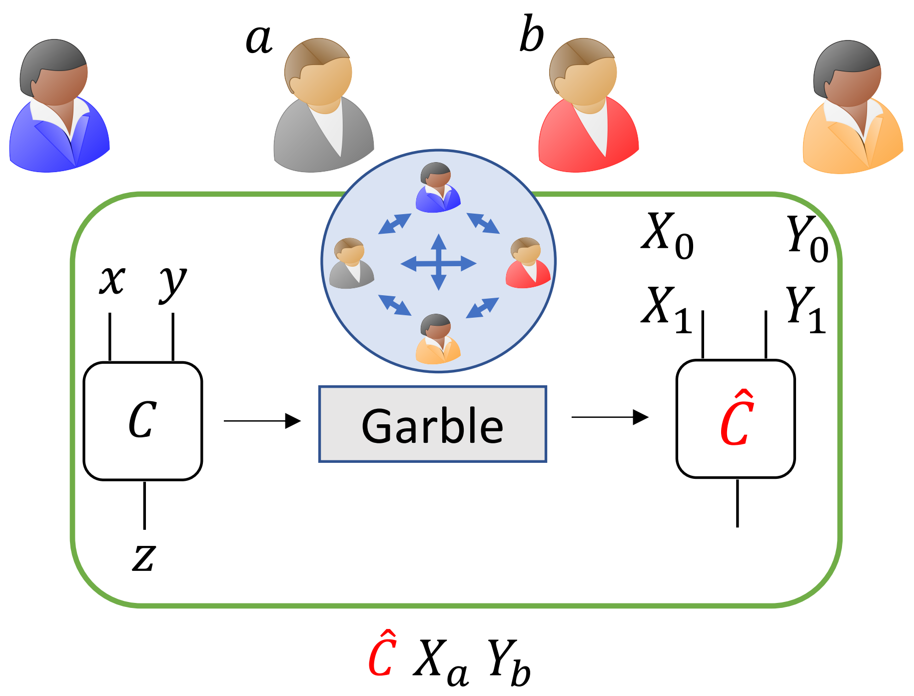
Evaluation Phase



Any non-constant round MPC works, since garbling algorithm can be computed by constant depth circuit

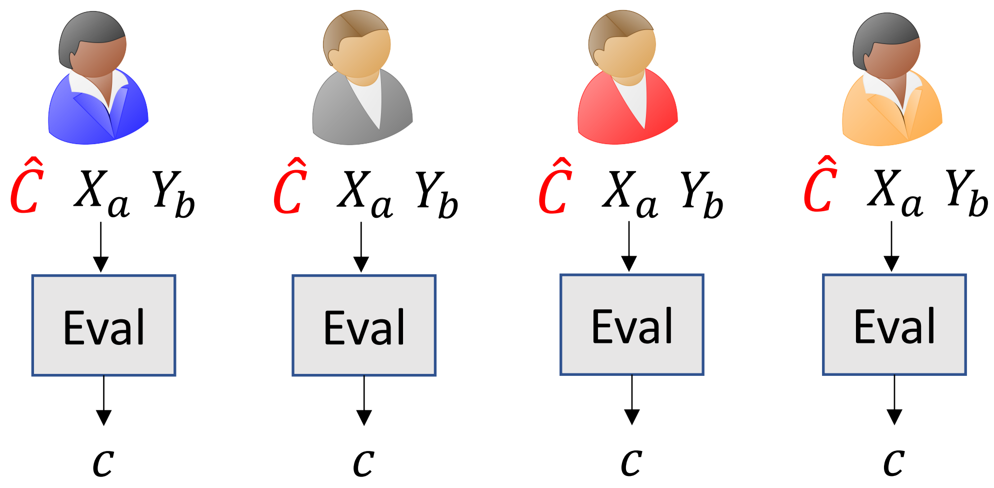
Template For Multiparty Garbling[BMR90]

Garbling Phase



Any non-constant round MPC works, since garbling algorithm can be computed by constant depth circuit

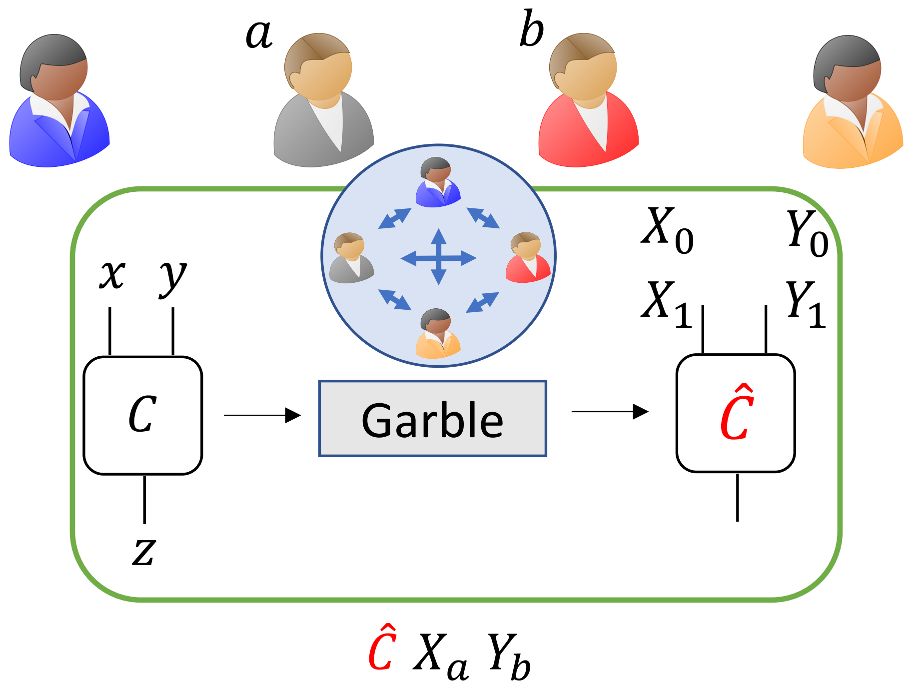
Evaluation Phase



MPC ensures adversary only learns $\hat{C}$ and one label per wire

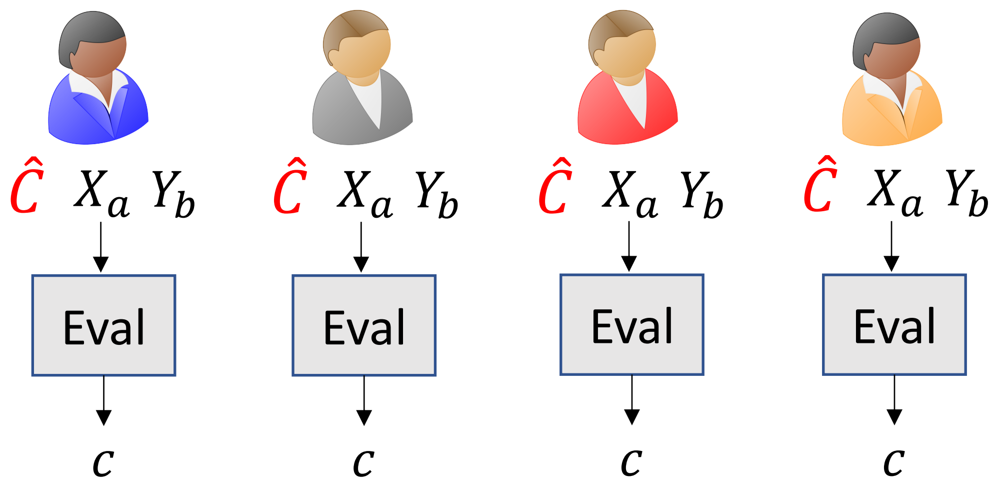
Template For Multiparty Garbling[BMR90]

Garbling Phase



Any non-constant round MPC works, since garbling algorithm can be computed by constant depth circuit

Evaluation Phase



MPC ensures adversary only learns $\hat{C}$ and one label per wire

Inefficient due to non-black box use of encryption

Avoiding Non-Black Box Use Of Encryption

Parties locally evaluate the PRF used in encryption [DI05]

$|\widehat{C}|$ is linear in n

Avoiding Non-Black Box Use Of Encryption

Parties locally evaluate the PRF used in encryption [DI05]

$|\widehat{C}|$ is linear in n

Parties locally compute additive sharing of ciphertext from additive sharing of the key and message i.e., $E([k],[m]) \rightarrow [E(k,m)]$ [BLO17]

$[x] \rightarrow$ secret sharing of x

MPC protocols work with secret shares

Avoiding Non-Black Box Use Of Encryption

Parties locally evaluate the PRF used in encryption [DI05]

$|\widehat{C}|$ is linear in n

Parties locally compute additive sharing of ciphertext from additive sharing of the key and message i.e., $E([k],[m]) \rightarrow [E(k,m)]$ [BLO17]

$[x] \rightarrow$ secret sharing of x

MPC protocols work with secret shares

$|\widehat{C}|$ is independent of n

Avoiding Non-Black Box Use Of Encryption

Parties locally evaluate the PRF used in encryption [DI05]

$|\widehat{C}|$ is linear in n

Parties locally compute additive sharing of ciphertext from additive sharing of the key and message i.e., $E([k],[m]) \rightarrow [E(k,m)]$ [BLO17]

$[x] \rightarrow$ secret sharing of x

MPC protocols work with secret shares

$|\widehat{C}|$ is independent of n

$O(n^2|C|)$ communication due to dishonest majority MPC

Avoiding Non-Black Box Use Of Encryption

Parties locally evaluate the PRF used in encryption [DI05]

$|\widehat{C}|$ is linear in n

Parties locally compute additive sharing of ciphertext from additive sharing of the key and message i.e., $E([k],[m]) \rightarrow [E(k,m)]$ [BLO17]

$[x] \rightarrow$ secret sharing of x

MPC protocols work with secret shares

$|\widehat{C}|$ is independent of n

$O(n^2|C|)$ communication due to dishonest majority MPC

Can we leverage techniques from $O(|C|)$ communication honest majority, non-constant round MPC?

Achieving $O(|C|)$ Communication

Parties locally compute threshold sharing of ciphertext from threshold sharing of key, message, and randomness i.e., $E([k],[m];[r]) \rightarrow [E(k,m;r)]$

Based on Learning Parity with Noise (LPN)

Achieving $O(|C|)$ Communication

Parties locally compute threshold sharing of ciphertext from threshold sharing of key, message, and randomness i.e., $E([k],[m];[r]) \rightarrow [E(k,m;r)]$

Based on Learning Parity with Noise (LPN)

$|\widehat{C}|$ is independent of n

Compatible with honest majority MPC

Achieving $O(|C|)$ Communication

Parties locally compute threshold sharing of ciphertext from threshold sharing of key, message, and randomness i.e., $E([k],[m];[r]) \rightarrow [E(k,m;r)]$

Based on Learning Parity with Noise (LPN)

$|\widehat{C}|$ is independent of n

Compatible with honest majority MPC

Packed secret sharing [FY92] - Standard technique in $O(|C|)$ communication non-constant round MPC protocols [BGJK21, GSY21, GPS21, GPS22]

- Pack $O(n)$ secrets into a single sharing $\Longrightarrow$ Reduces communication cost by a factor of n

Achieving $O(|C|)$ Communication

Parties locally compute threshold sharing of ciphertext from threshold sharing of key, message, and randomness i.e., $E([k],[m];[r]) \rightarrow [E(k,m;r)]$

Based on Learning Parity with Noise (LPN)

$|\widehat{C}|$ is independent of n

Compatible with honest majority MPC

Packed secret sharing [FY92] - Standard technique in $O(|C|)$ communication non-constant round MPC protocols [BGJK21, GSY21, GPS21, GPS22]

- Pack $O(n)$ secrets into a single sharing $\Longrightarrow$ Reduces communication cost by a factor of n
- Utilize existing techniques for MPC over packed sharings

Achieving $O(|C|)$ Communication

Parties locally compute threshold sharing of ciphertext from threshold sharing of key, message, and randomness i.e., $E([k],[m];[r]) \rightarrow [E(k,m;r)]$

Based on Learning Parity with Noise (LPN)

$|\widehat{C}|$ is independent of n

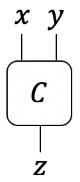
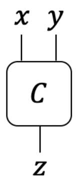
Compatible with honest majority MPC

Packed secret sharing [FY92] - Standard technique in $O(|C|)$ communication non-constant round MPC protocols [BGJK21, GSY21, GPS21, GPS22]

- Pack $O(n)$ secrets into a single sharing $\Longrightarrow$ Reduces communication cost by a factor of n
- Utilize existing techniques for MPC over packed sharings
- Efficiently computing $[r]$ requires developing new subprotocols, building on prior works [CCXY18]

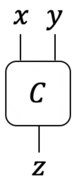
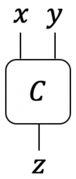
Evaluation Of Semi-Honest Secure Protocol

$n = 512$ $t = 127$ 2 threads per party

		Runtime	Per Party Communication
 $\equiv$ AES-128, $\lvert C \rvert = 36{,}663$	Total	126.28 s	66.84 MB
 $\equiv$ SHA-256, $\lvert C \rvert = 114{,}107$	Total	481.88 s	240.44 MB

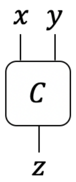
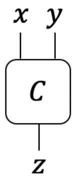
Evaluation Of Semi-Honest Secure Protocol

$n = 512$ $t = 127$ 2 threads per party

		Runtime	Per Party Communication
 $\equiv$ AES-128, $\lvert C\rvert = 36{,}663$	Total	126.28 s	66.84 MB
	Circuit Dependent	15.52 s	8.1 MB
 $\equiv$ SHA-256, $\lvert C\rvert = 114{,}107$	Total	481.88 s	240.44 MB
	Circuit Dependent	40.8 s	27.07 MB

Evaluation Of Maliciously Secure Protocol

$n = 512$ $t = 127$ 2 threads per party

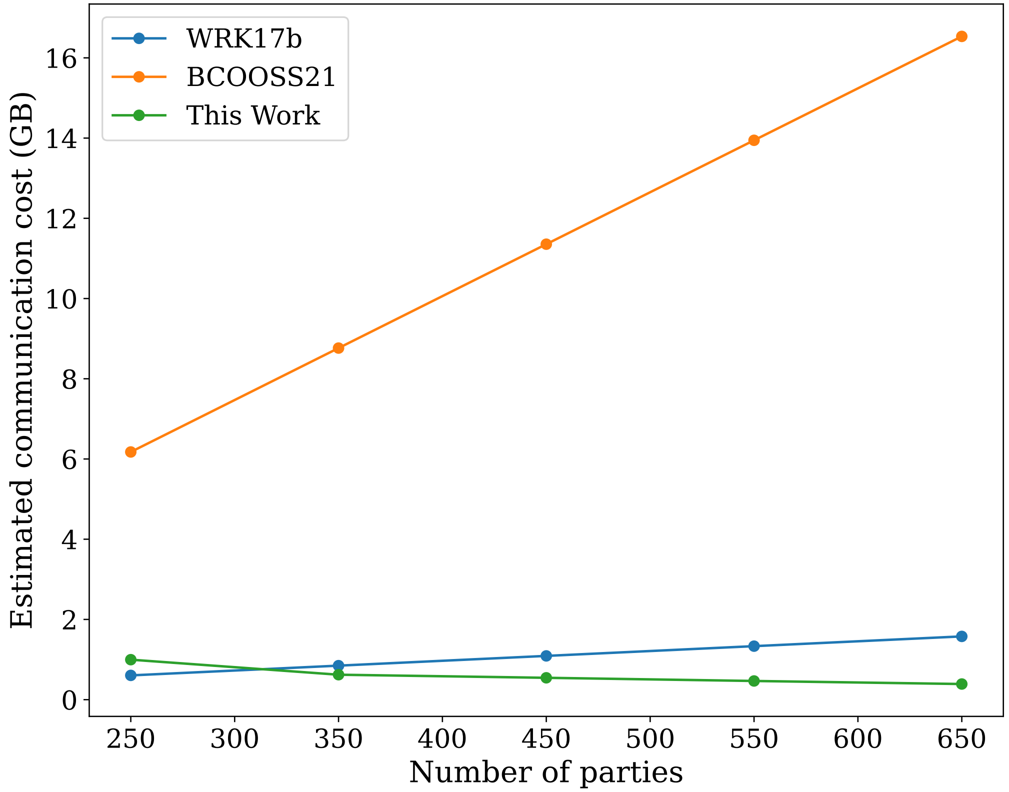
		Computation Time	Per Party Communication
 $\equiv$ AES-128 $\lvert C \rvert = 36{,}663$	Total	≈220 s	≈334 MB
	Circuit Dependent	≈18 s	≈42 MB
 $\equiv$ SHA-256 $\lvert C \rvert = 114{,}107$	Total	≈811 s	≈1230 MB
	Circuit Dependent	≈67 s	≈155 MB

Thank You

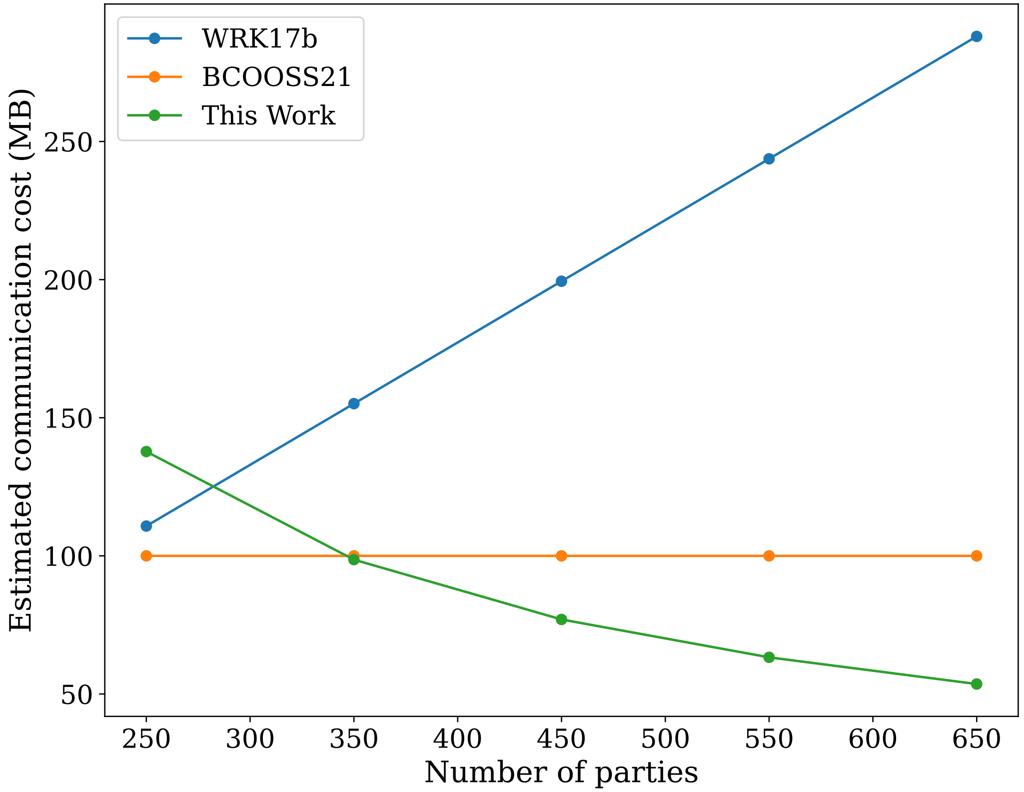
ia.cr/2023/099

github.com/adishegde/scalable_garbling

Appendix: Comparison To Prior Works



Circuit Independent



Circuit Dependent

Comparison of per party communication when garbling AES-128