Scalable Mixed-Mode MPC

Radhika Garg[2], Kang Yang[3], Jonathan Katz[1], Xiao Wang[2]

1

UNIVERSITY OF MARYLAND

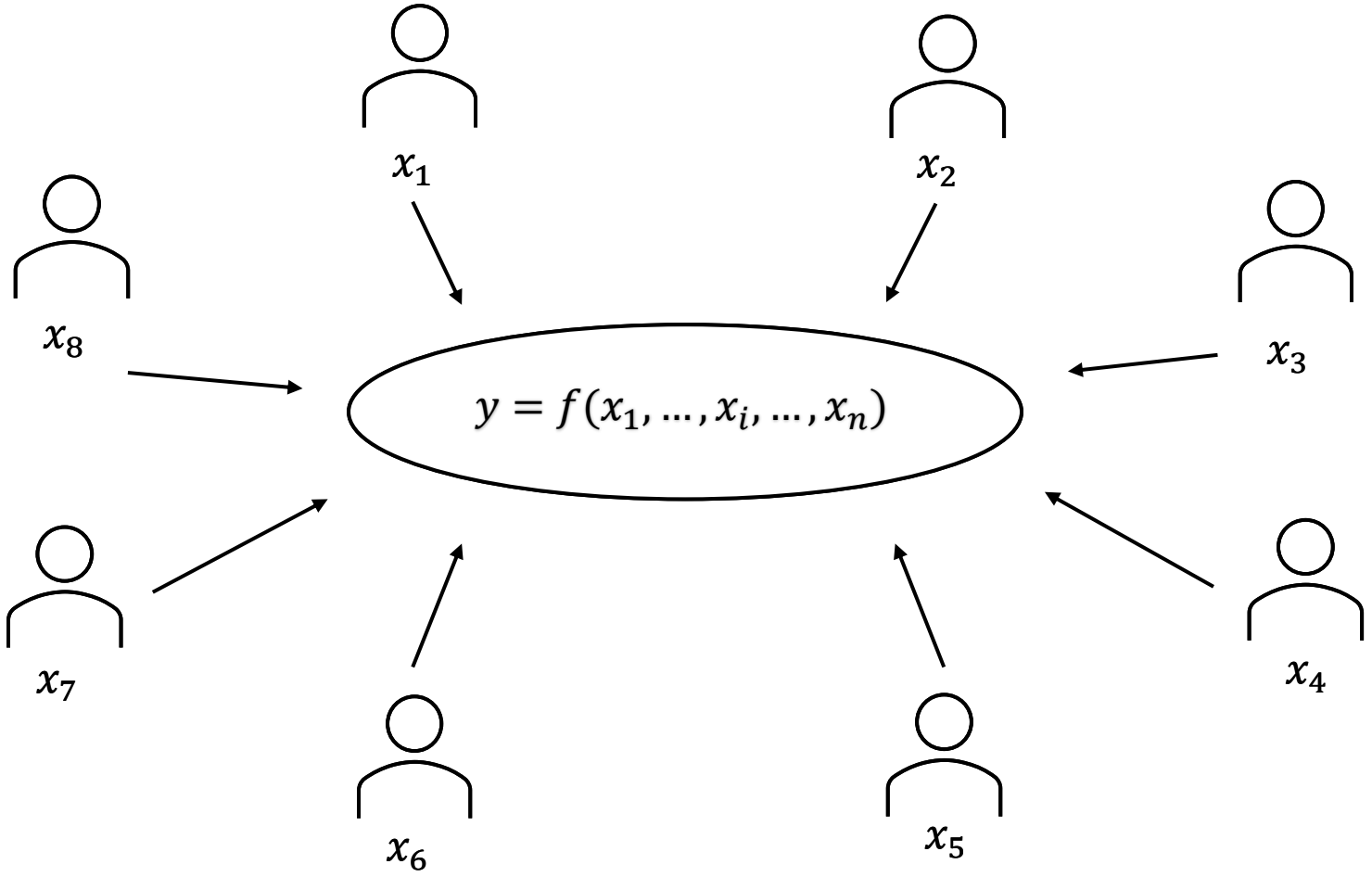
2

Northwestern University

3

State Key Laboratory of Cryptology

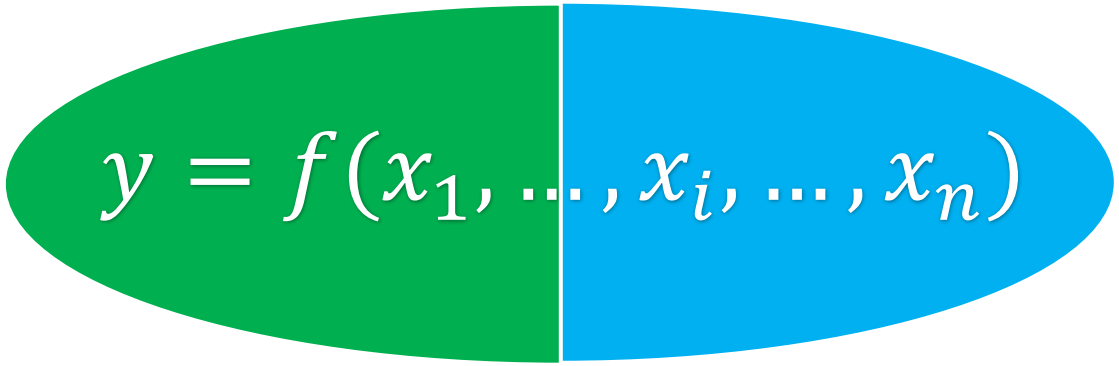
Multi-Party Computation



Mixed-Mode MPC

Arithmetic

$+,*$



Boolean

$\oplus, \&$

P_i has x_i such that $x = \Sigma_{\{i=1\}}^{n} x_i$

P_i has x_i such that $x = \bigoplus_{\{i=1\}}^{n} x_i$

Mixed-Mode MPC Bottleneck

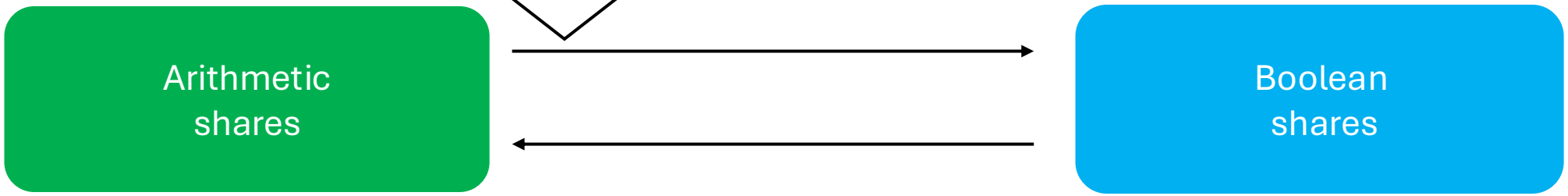
```
def biomatch(data, sample):
    dist = euclidian_distance(data, sample, data.size(), sample.size())
    min_dist = min(dist, N)
    return min_dist
```

With prior state-of-the-art solutions, 99% of the time and communication is due to the conversion protocols.

Prior Work

- Most works [PSSY21, BCD+20] are catered towards a limited number of parties or [KPPS23] require honest majority setting.

Can we do better than pairwise communication?



Our Contribution

- Multi-party private table lookup protocol
- Multi-party secret-share conversion
  - Boolean to Arithmetic
  - Arithmetic to Boolean
- Linear complexity multi-party garbled circuits

Homomorphic Encryption

$$\boldsymbol{F\big(Enc(x), Enc(y)\big) = Enc(F(x, y))}$$

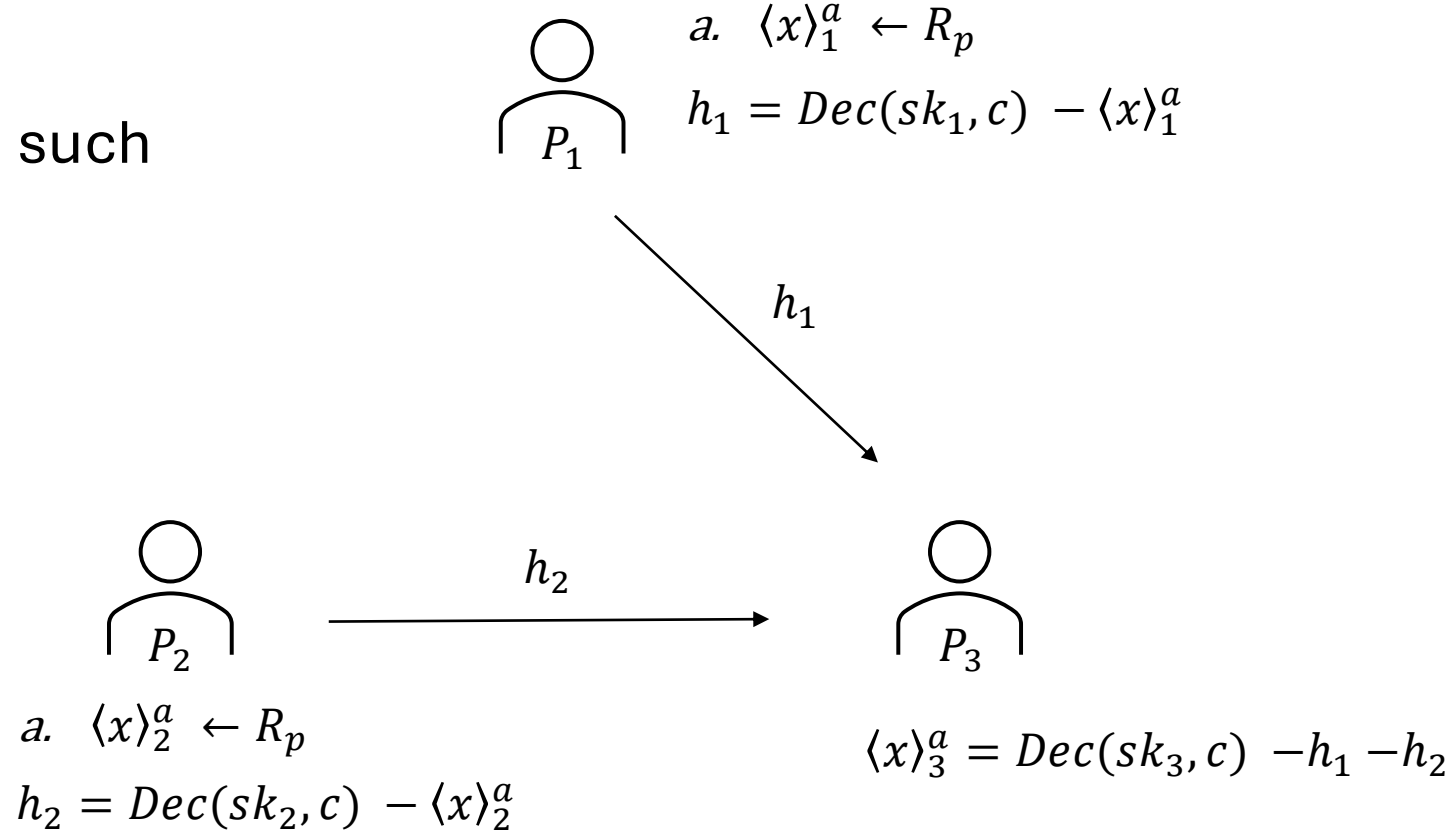
Threshold Homomorphic Encryption:

- $P_i: sk_i, pk$
- Encryption is local
- Decryption requires at least t-parties (all in our case) to be online.

Encryption to Arithmetic Shares

Given - $P_i: sk_i, c$

To obtain - $P_i: \langle x \rangle_i^a$, such that $x = Dec(sk, c)$



Multi-party private table lookup protocol

P_i: $\langle x \rangle_i^b$, T

P_i: $\langle y \rangle_i^a$, y = T[x]

P_i: $\langle r \rangle_i^b, \langle T' \rangle_i^a$ such that $T'[i] = T[r \oplus i]$

To lookup at x: $T'[x \oplus r] = T[x]$

Multi-party private table lookup protocol

T:

p_0
p_1
p_2
…
…
p_M

P_1:

T_0:

e_0
e_1
e_2
…
…
e_M

1.

2. $r_1 \leftarrow \mathbb{Z}_M$

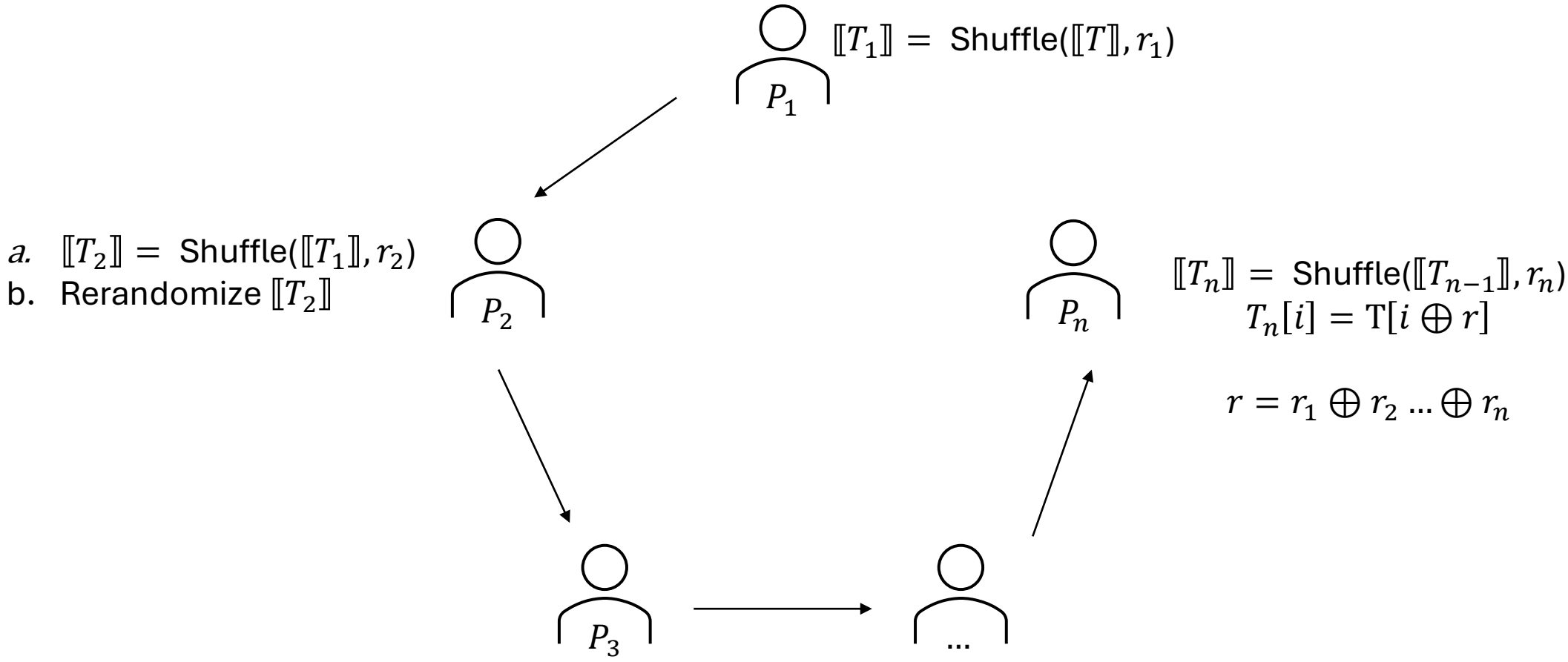
3. $T_1[i \oplus r_1] = T_o[i]$

4. Send T_1 to P_2

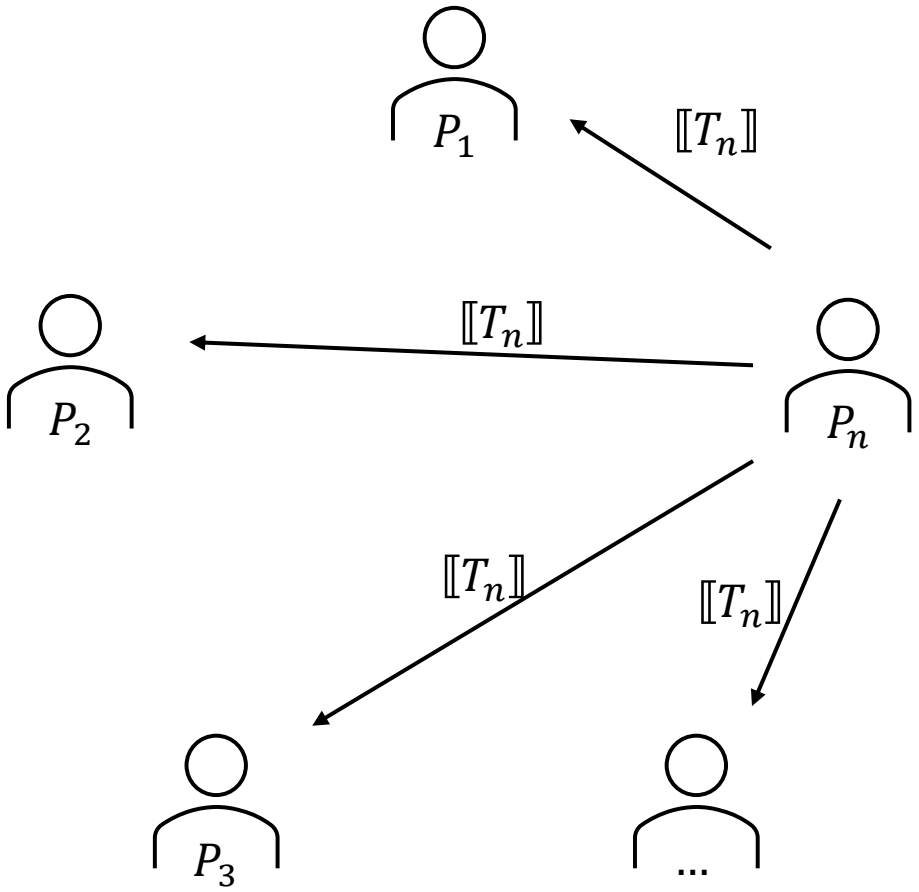
P_2:

1. $T_2[i \oplus r_2] = T_1[i]$
2. Rerandomize the entries of T_2
3. Send T_2 to P_3

Multi-party private table lookup protocol



Multi-party private table lookup protocol



Now all parties use E2A to get additive shares of all entries:

$$\langle T_n[i] \rangle^a \text{ for all } i \in [0, M]$$

Boolean to Arithmetic Share Conversion

- P_i has $\langle x \rangle_i^b$ such that $x = \bigoplus_{\{i=1\}}^{n} \langle x \rangle_i^b$ and want $\langle x \rangle_i^a$ such that $x = \Sigma_{\{i=1\}}^{n} \langle x \rangle_i^a$
- Say, x is an ℓ bit number
- We look at converting each bit first.
  - For bit = y, lookup in $T = [0\ 1]$ such that $T[y] = \mathrm{y}$.
  - When all parties use the LUT protocol to lookup at position y, they obtain $\langle T[y] \rangle_1^a = \langle y \rangle_1^a$

Boolean to Arithmetic Share Conversion

Problem: Each ciphertext is too large (more than 100 KB for RLWE-based HE)

Solution: Packing

Ciphertext space: $R_q = \mathbb{Z}_q[X]/(X^N + 1)$

Plaintext space: $R_p = \mathbb{Z}_p[X]/(X^N + 1)$

Boolean to Arithmetic Share Conversion

P_1:

0
1
0
1

$r_1[0], r_1[1]$

$[\![T_1]\!]$ =

a
b
c
d

P_2:

- Receives $[\![T_2]\!]$
- Samples $r_2 = [x, y]$

$[\![T_2]\!]$ =

a	$\sim x$		b	x		d	0	
b	$\sim x$	+	c	0	+	a	x	+
c	$\sim y$	*	d	y	*	b	0	*
d	$\sim y$		a	0		c	y	

Boolean to Arithmetic Share Conversion

- For each bit $x[i]$, all parties have $\langle x[i] \rangle^a$.
- $\langle x \rangle^a = \Sigma_{\{i=0\}}^{\{i=l-1\}} 2^i \langle x[i] \rangle^a$

- For each conversion: $O\left(l \frac{|ct|}{\frac{N}{2}}\right)$
  - N is the number of slots here.

Arithmetic to Boolean Share Conversion

P_i: $\langle x \rangle_i \Rightarrow x = \langle x \rangle_1^a + \langle x \rangle_2^a + \cdots + \langle x \rangle_n^a$

- Now that we know B2A: Generate $(\langle r \rangle^b, \langle r \rangle^a)$
- Reveal $x - r$: $\langle x \rangle^a - \langle r \rangle^a$
- Compute $x - r + \langle r \rangle^b$ using a Boolean circuit to obtain $\langle x \rangle^b$.

Problem: r is not uniform in $\mathbb{Z}_p$

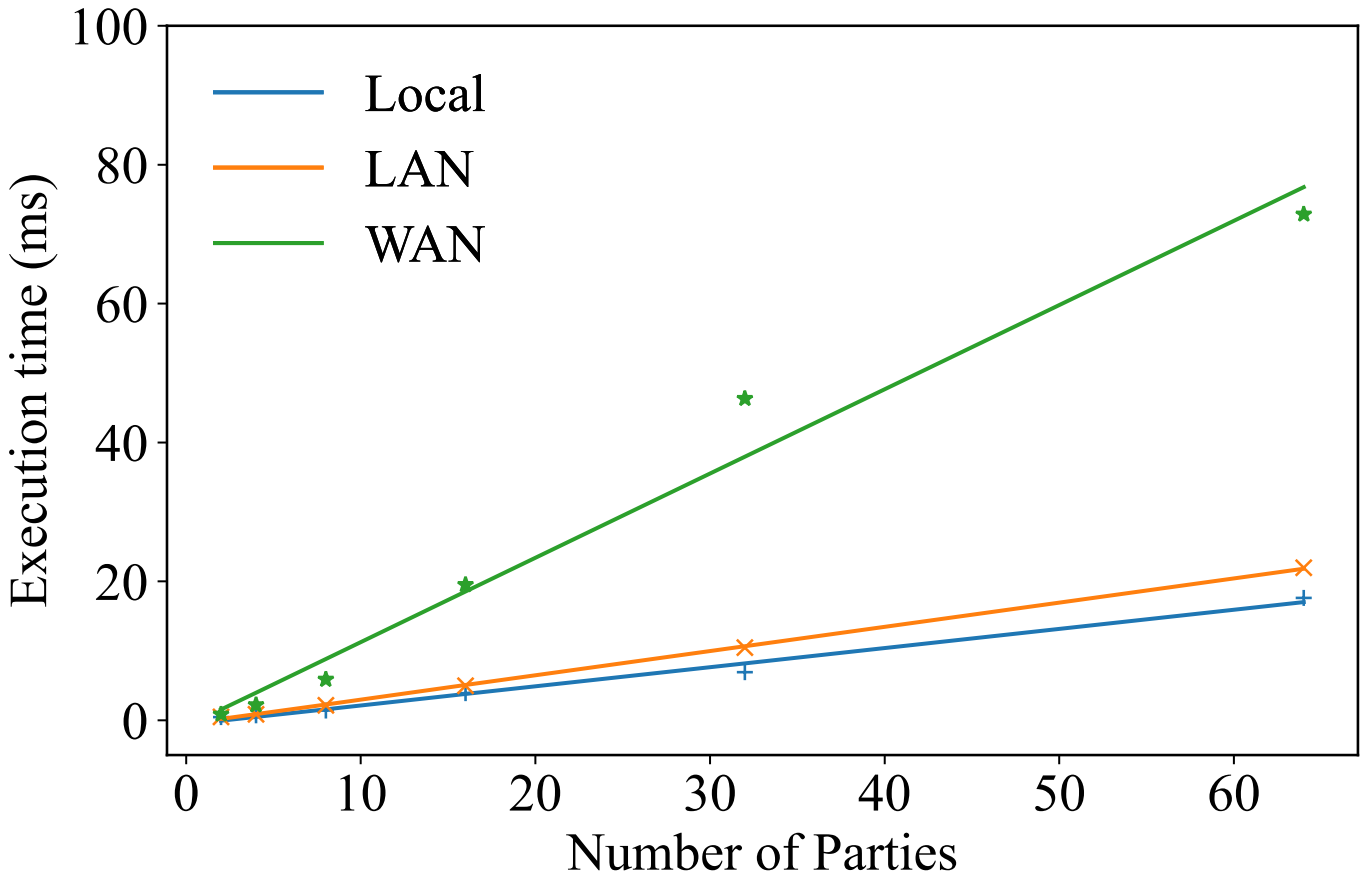
Arithmetic to Boolean Share Conversion

Question: How to generate $(\langle r \rangle^b, \langle r \rangle^a)$ such that r is uniform in $\mathbb{Z}_p$

Solution:

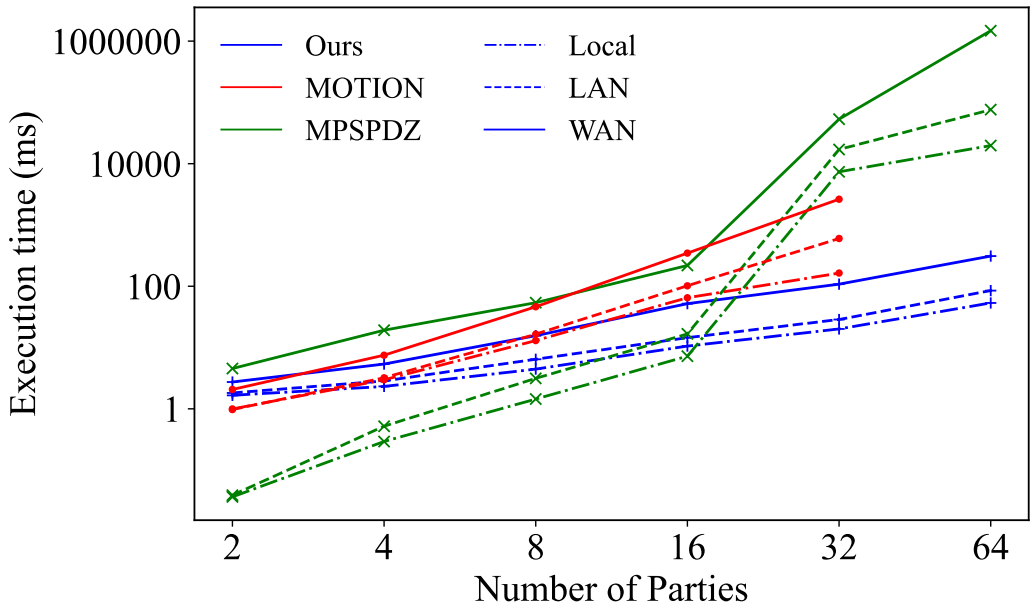
- We have $p = 2^{32} - 2^{30} + 1$
- So, the max possible $r = 1100 \ldots 0$
- From B2A, we have $\langle r[j] \rangle^b$ for all $j \in [0, \ell - 1]$:
  - Thus, check: $\langle r[\ell - 1] \rangle^b . \langle r[\ell - 2] \rangle^b . \Sigma_{\{j=0\}}^{\{j=\ell-3\}} \langle r[j] \rangle^b = 0$
- Only 2 multiplications per check.

Results

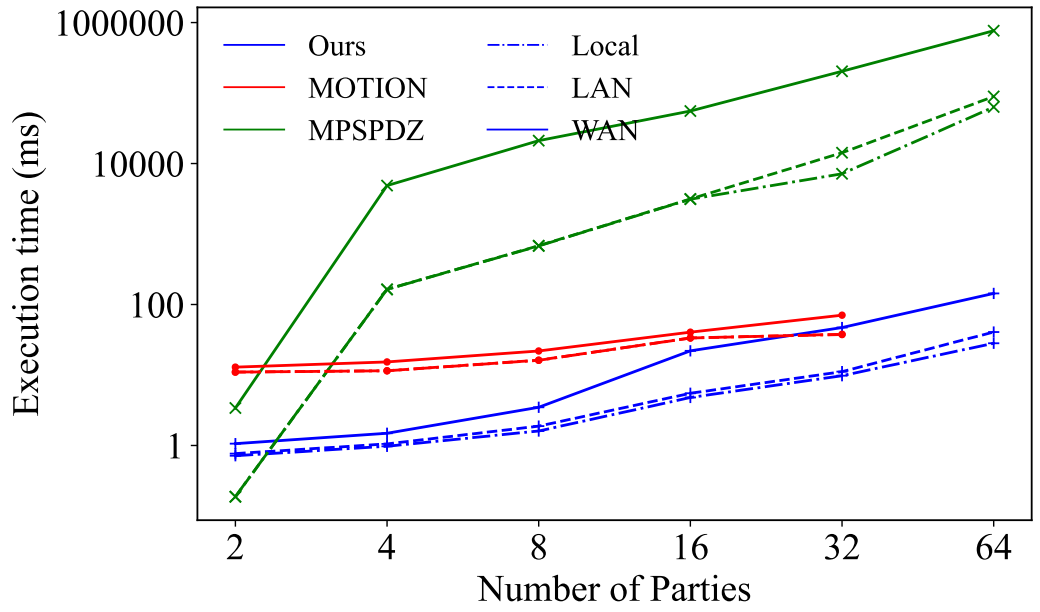


Conversion from B2A: Execution time scales linearly in the number of parties

Computation Comparison



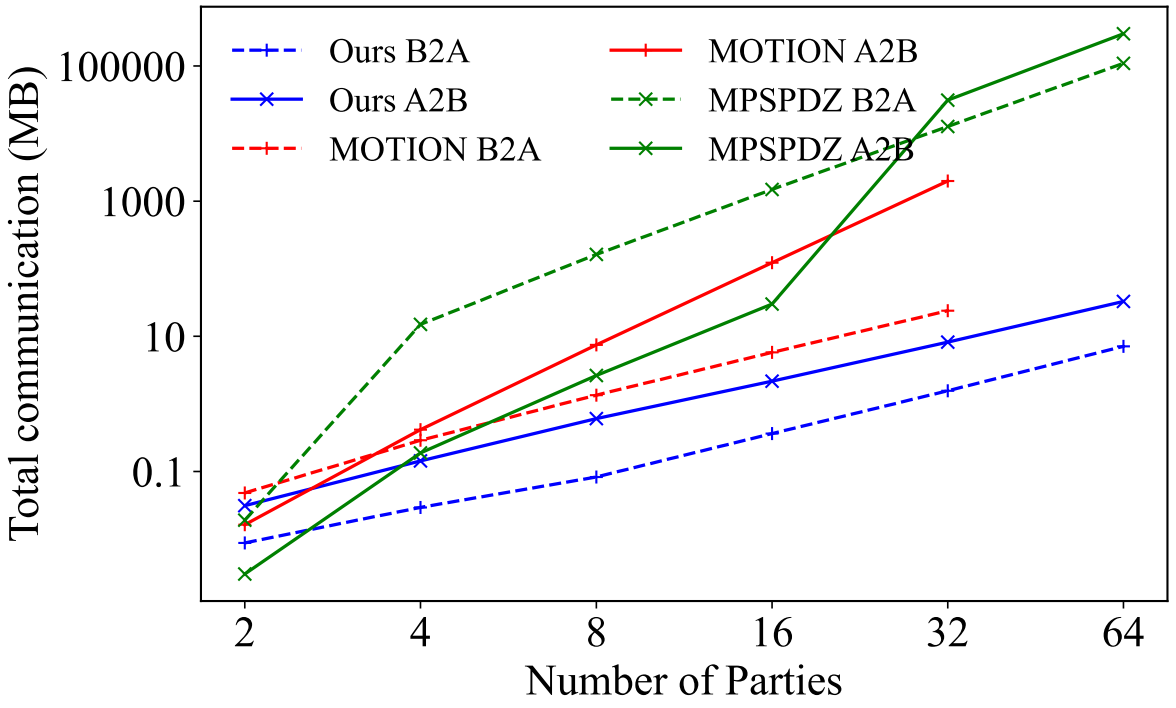
Boolean to Arithmetic



Arithmetic to Boolean

We have up to 20x (resp., 2247x) less running time than MOTION (resp., MP-SPDZ).

Communication Comparison



We require up to **1184x** (resp., **8819x**) less communication than the prior state-of-art MOTION (resp., MPSPDZ)

Multi-party Garbled Circuits Protocol

- We build upon the previous work on LWE-based Garbled Circuit to achieve constant communication per party.
- Premise:
  - For each wire w, keys $k_{w,0}, k_{w,1}$ are additively secret shared and a mask value λ_w is Boolean shared among all parties.
  - For each gate g: $\alpha, \beta \in \{0,1\}$:
    - the garbled row value = $F\left(\left\langle k_{u,\alpha}\right\rangle^a \parallel \left\langle k_{v,\beta}\right\rangle^a ;\ gid\right) + \left\langle k_{w,e_{u,v,\alpha,\beta}}\right\rangle^a$ where
    - $e_{u,v,\alpha,\beta} = g(\lambda_u \oplus \alpha, \lambda_v \oplus \beta) \oplus \lambda_w$

Multi-party Garbled Circuits Protocol

- We propose that our LUT protocol can be directly applied here.
  - We say that the table T is available as additive shares and each party sends an encryption of its shares to P_1.
  - Now, P_1 has $[\![T]\!] = [[\![k_{w,0}]\!][\![k_{w,1}]\!]]$, all parties obtain the shares of the shuffled table using the LUT protocol and lookup $\langle e_{u,v,\alpha,\beta} \rangle^b$.

R. Garg, K. Yang, J. Katz and X. Wang, "Scalable Mixed-Mode MPC," in 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024 pp. 109-109.

Thank You

https://eprint.iacr.org/2023/1700.pdf

Our implementation will be open-sourced soon