# Multi-Party Replicated Secret Sharing over a Ring with Applications to Privacy-Preserving Machine Learning

Alessandro Baccarini, Marina Blanton, Chen Yuan

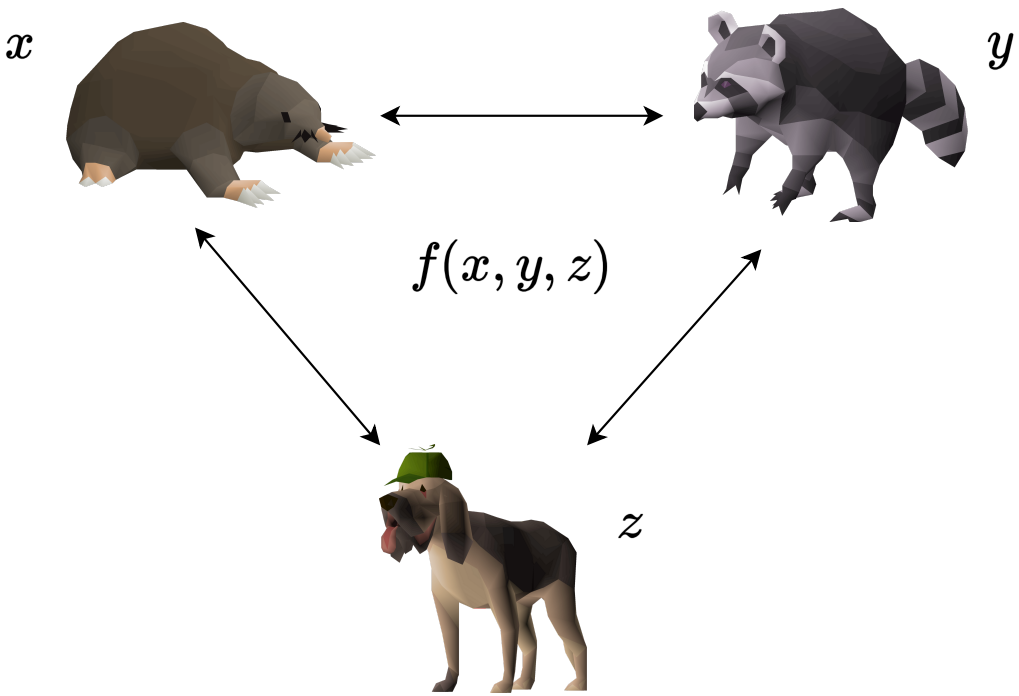
Department of Computer Science and Engineering
University at Buffalo

{anbaccar,mblanton,chyuan}@buffalo.edu

July 11, 2023

# Introduction to secure multi-party computation (SMC)

*Secure multi-party computation* – multiple participants jointly evaluate a function on secret inputs




- No information is disclosed other than the output
- Replaces a trusted third party
- $n$ parties with corruption threshold $t$
- Many applications

# Secret sharing techniques

- Shamir secret sharing over field $\mathbb{F}_p$
  - Frequent modular reduction
  - Doesn't work over a ring
- Replicated secret sharing (RSS) over ring $\mathcal{R}$
  - Efficient
  - Easy to conceptualize
  - Good performance-security trade-off
  - Built-in type compatibility (`uint32_t` and `uint64_t`)
- Setting: semi-honest, honest majority ($t < n/2$)

# Contributions

- RSS framework for $n$ parties over a ring (arbitrary and $\mathbb{Z}_{2^k}$)
- Quantized neural network optimizations
- Extensive benchmarks against field implementation and state-of-the-art

# RSS framework

- Split $x$ into $\binom{n}{t}$ shares, distribute $\binom{n-1}{t}$ shares per party
  - e.g. For 5 parties: 10 total shares, 6 per party
- $[x]_k \implies$ secret shared private value over $\mathbb{Z}_{2^k}$
- Building blocks can be done over any ring (multiplication, reconstruction, input)
- Random bit generation RandBit [DEF+19] vs. edaBit [EGK+20]
- Composite protocols (over $\mathbb{Z}_{2^k}$):
  - $[a_{k-1}]_k \leftarrow \mathsf{MSB}([a]_k)$ [CDH10, DEF+19]
  - $[a/2^m]_k \leftarrow \mathsf{TruncPr}([a]_k, m)$ [DEK20]
  - $[a]_{k'} \leftarrow \mathsf{Convert}([a]_k, k, k')$ [DEF+19, $\mathsf{BitDec}([a_k])$]

# RSS framework

- Split $x$ into $\binom{n}{t}$ shares, distribute $\binom{n-1}{t}$ shares per party
  - e.g. For 5 parties: 10 total shares, 6 per party
- $[x]_k \implies$ secret shared private value over $\mathbb{Z}_{2^k}$
- Building blocks can be done over any ring (multiplication, reconstruction, input)
- Random bit generation RandBit [DEF+19] vs. edaBit [EGK+20]
- Composite protocols (over $\mathbb{Z}_{2^k}$):
  - $[a_{k-1}]_k \leftarrow \mathrm{MSB}([a]_k)$ [CDH10, DEF+19]
  - $[a/2^m]_k \leftarrow \mathrm{TruncPr}([a]_k, m)$ [DEK20]
  - $[a]_{k'} \leftarrow \mathrm{Convert}([a]_k, k, k')$ [DEF+19, BitDec($[a_k]$)]

# Quantized neural networks

- Neural network, but smaller ($\mathbb{R} \to$ 8-bit integers)
- Weights, inputs, biases have *scales* ($m$) and *zero points* ($z$)

$$y = \sum_{i=1}^{N} x_i w_i + b \implies \bar{y} \approx z_3 + \frac{m_1 m_2}{m_3} \cdot \sum_{i=1}^{N} \left( (\bar{x}_i - z_1) \cdot (\bar{w}_i - z_2) + \bar{b} \right)$$

  and "clamp" $\bar{y}$ to $0 \leq \bar{y} \leq 255$
- Certain activation functions can be incorporated into $m_3, z_3$ for free (e.g. ReLU6)
- [DEK20]'s procedure: fixed-point multiplication, followed by truncation and clamp result

# Quantized neural networks

- Neural network, but smaller ($\mathbb{R} \to$ 8-bit integers)
- Weights, inputs, biases have *scales* ($m$) and *zero points* ($z$)

$$y = \sum_{i=1}^{N} x_i w_i + b \implies \bar{y} \approx z_3 + \boxed{\frac{m_1 m_2}{m_3}} \cdot \sum_{i=1}^{N} \left( (\bar{x}_i - z_1) \cdot (\bar{w}_i - z_2) + \bar{b} \right)$$

  and "clamp" $\bar{y}$ to $0 \leq \bar{y} \leq 255$
- Certain activation functions can be incorporated into $m_3, z_3$ for free (e.g. ReLU6)
- [DEK20]'s procedure: fixed-point multiplication, followed by truncation and clamp result

**Problem**

Uses $k = 72$ to accommodate for the 63-bit truncation.

# Quantized neural networks (cont'd)

## Solution

Fold scales into clamping operation, and compute a much smaller truncation at the end of each layer.

- "Rearranging" terms:

$$0 \leq \left( \bar{y} = \frac{m_3}{m_1 m_2} z_3 + \sum_{i=1}^{N} \left( (\bar{x}_i - z_1) \cdot (\bar{w}_i - z_2) + \bar{b} \right) \right) \leq \frac{255 m_3}{m_1 m_2},$$

- Bitlength will grow $\implies$ truncate down to 8 bits.
- Lowers ring size by over a factor of two! $(72 \rightarrow 30)^*$
- Updated model parameters can be distributed by the model owner
- No impact on accuracy

*Theoretically can be lowered further

# Summary of results

– RSS offers compelling advantages over field-based equivalents for 3, 5, 7 parties (10-33× improvement)

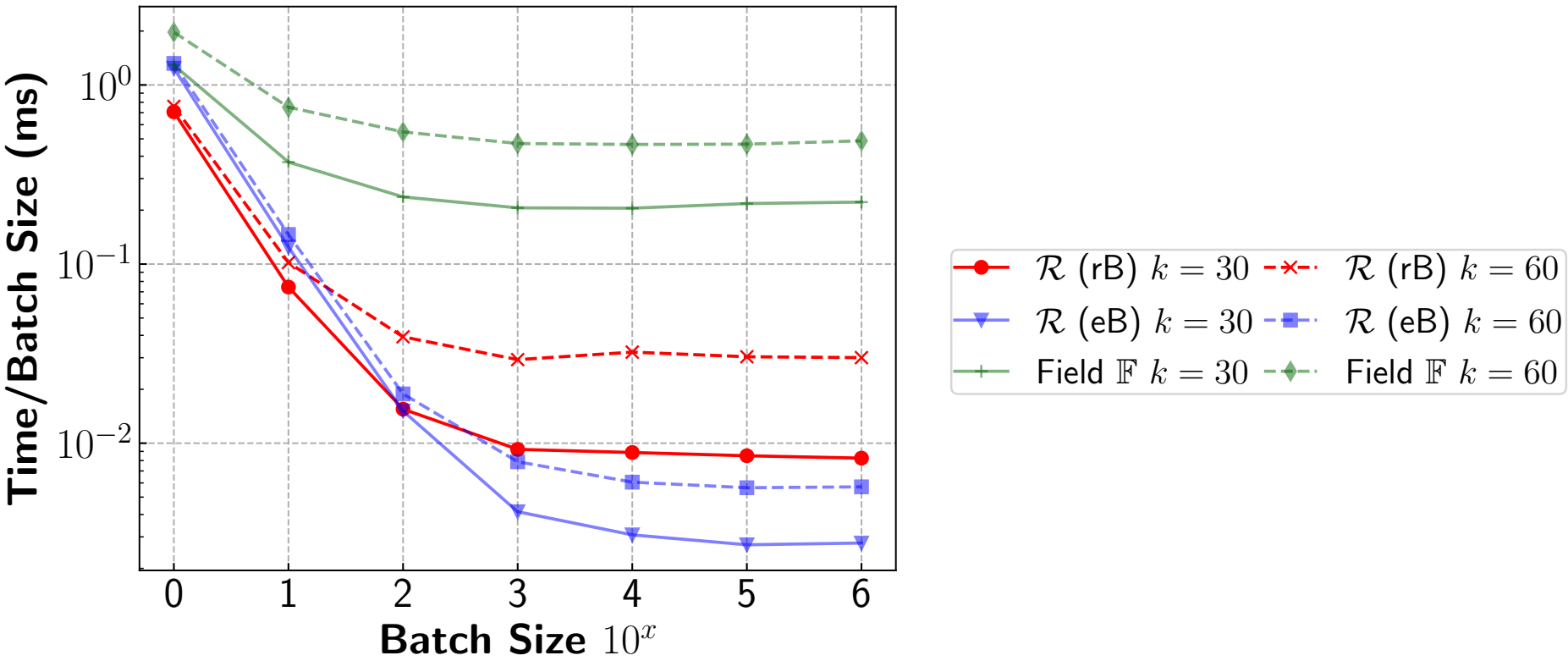



Figure: 3PC MSB([$a$])

# Summary of results

- On-par performance with 3PC ring frameworks in (quantized) neural network applications (MobileNets)

| | | Ours | | | | MP-SPDZ $\mathbb{Z}_{2^k}$, [DEK20] | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | 0.25 | 0.5 | 0.75 | 1.0 | 0.25 | 0.5 | 0.75 | 1.0 |
| $\rho$ | 128 | 3.19 | 6.47 | 9.92 | 13.3 | 3.19 | 6.26 | 9.88 | 14.0 |
| | 160 | 4.94 | 10.0 | 15.1 | 20.7 | 4.15 | 8.17 | 13.6 | 19.3 |
| | 192 | 7.17 | 14.3 | 22.0 | 29.7 | 5.00 | 11.0 | 17.8 | 26.7 |
| | 224 | 9.71 | 19.9 | 30.0 | 40.9 | 6.57 | 14.1 | 23.1 | 34.9 |

Table: 3PC quantized MobileNets

- Future work:
  - Integration to the PICCO compiler [ZSB13]
  - Floating point compatibility [RBS+22, ABZS13]

# Thank you!

Questions?

# References


M. Aliasgari, M. Blanton, Y. Zhang, and A. Steele.

Secure computation on floating point numbers.

In *Network and Distributed System Security Symposium (NDSS)*, 2013.

O. Catrina and S. De Hoogh.

Improved primitives for secure multiparty integer computation.

In *International Conference on Security and Cryptography for Networks (SCN)*, pages 182–199, 2010.

I. Damgård, D. Escudero, T. Frederiksen, M. Keller, P. Scholl, and N. Volgushev.

New primitives for actively-secure MPC over rings with applications to private machine learning.

In *IEEE Symposium on Security and Privacy*, pages 1102–1120, 2019.

A. Dalskov, D. Escudero, and M. Keller.

Secure evaluation of quantized neural networks.

*Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2020(4):355–375, 2020.

# References


D. Escudero, S. Ghosh, M. Keller, R. Rachuri, and P. Scholl.

Improved primitives for MPC over mixed arithmetic-binary circuits.

In *Advances in Cryptology – CRYPTO*, pages 823–852, 2020.

D. Rathee, A. Bhattacharya, R. Sharma, D. Gupta, N. Chandran, and A. Rastogi.

SecFloat: Accurate floating-point meets secure 2-party computation.

In *IEEE Symposium on Security and Privacy*, pages 1553–1553, 2022.

Y. Zhang, A. Steele, and M. Blanton.

PICCO: A general-purpose compiler for private distributed computation.

In *ACM Conference on Computer and Communications Security (CCS)*, pages 813–826, 2013.