Breaking the $O(mn)$ Bit Barrier: MPC with Static Adversary

Mahnush Movahedi  
Jared Saia  
University of New Mexico

MURI Final Review Briefing  
June 13, 2012

Helix MURI Project -- http://helix.cs.virginia.edu
Multiparty Computation (MPC)

- $n$ players participate in an auction
- They want not to reveal their inputs to each other or any external party

Q: How can they determine the highest bid without revealing any information about the other bids?
MPC: Formal Definition

Given:
- \( n \) players
- Each player \( i \) has a private input
- Function \( f \) over \( n \) inputs, known to all players

Goals:
- All players learn the value \( f(x_1, x_2, \ldots, x_n) \)
- The inputs remain as private as possible

Typically, assume \( n/3 \) are bad

No more information revealed than what can be determined by the output
Applications

- Auctions
- Threshold cryptography
- Anonymous message transmission
- Information aggregation

A group can sign/read a document collectively, but not individually

Messages can be broadcast to a network but originator remains anonymous
Applications as Functions

- **Auctions**
  \[ f = \max(x_1, x_2, \ldots, x_n) \]

- **Threshold cryptography**
  \[ f = M^s \mod pq \]
  1) \( M, p, q \) are parameters of the function;
  2) \( s \) is the \( y \) intercept of a degree \((d-1)\) function with points given by the \( x_i \) values.

- **Information aggregation**
  \[ f = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n} - \left( \frac{\sum_{i=1}^{n} x_i}{n} \right)^2} \]
The Helix system learns, regenerates, and improves.

MPC provides a completely decentralized and private means to do this at the network level.

\[ f = \text{a repair algorithm based on input data from the nodes in the Helix network} \]
Previous Work

- Create a circuit based on function $f$
- Assume the circuit has $m$ gates
  - Each player sends $O(mn)$ messages
  - Each player performs $O(mn)$ computation.
Current algorithms to solve MPC are not resource-efficient.
Our Contribution

- Much improved computation & message cost
- Assume the circuit has m gates
  - Each player sends $\tilde{O}(\frac{m+n}{n} + \sqrt{n})$ messages
  - Each player performs $\tilde{O}(\frac{m+n}{n} + \sqrt{n})$ computation.
- We solve MPC \textit{w.h.p.} meaning
  \[ 1 - O(1/n^k) \text{ for any fixed } k \]
Algorithm Overview

- Make critical use of a quorum:
  - Has $\theta(\log n)$ players
  - Less than $1/3$ are bad
  - Each gate is computed by a quorum

- Preserve privacy
  - Masking inputs & gate outputs with random numbers
  - Random number are known collectively via verifiable secret sharing
Verifiable Secret Sharing

- 4 people want to open a safe collectively,
- Each knows only one digit of the password.

The secret can be reconstructed from 2/3 shares,
- Fewer shares reveal nothing,
- Shares define a unique secret.
Tools Used

- Can get all processors to agree on $n$ quorums \textit{w.h.p.} [KS 11]
  - Previous result from this MURI

- Verifiable secret sharing algorithm [BGW 88]

- \textsc{Heavy-Weight-MPC} algorithm [BGW 88]
Algorithm Overview

- Translate function $f$ to circuit $C$
- Build network $G$ based on $C$
  - Gates $\rightarrow$ Internal nodes
  - Inputs $\rightarrow$ Input nodes
  - Wire $\rightarrow$ Edges
- Build quorums
- Assign each quorum to a node
The Algorithm

- Input commitment using VSS

- Random number generation

Generate $R_G$ jointly
Computation of a Gate

1) $G$ Shares of $R_G$
   \[
   \begin{align*}
   &X + R_x \\
   \text{Shares of } R_x
   \end{align*}
   \]
   \[
   \begin{align*}
   &Y + R_y \\
   \text{Shares of } R_y
   \end{align*}
   \]

2) \[
   G(X, Y) + R_G
   \]
   \[
   G(X, Y) + R_G
   \]
Propagating Output

- Output reconstruction
- Output propagation
Conclusion

- MPC provides means for helix system computations
- We have created a scalable algorithm to perform MPC
- Our algorithm:
  - needs less resources.
  - tolerates $n/3$ bad players.