Efficient Perfectly Secure Computation with Optimal Resilience

CRYPTO 2021

Eleni Makri

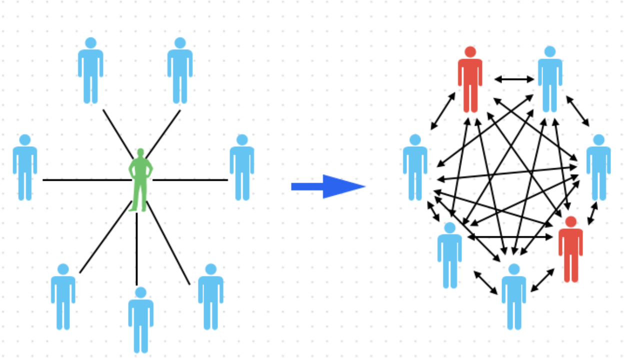
CoReLab NTUA

June 8, 2023

Outline

1. MPC
2. Introduction
3. Preliminaries
   - Shamir Secret Sharing
   - Arithmetic Circuits
   - Linear Computation
4. BGW
   - BGW Multiplication Protocol
   - GRR Multiplication Protocol
   - Honest but curious BGW
   - Malicious BGW
   - BGW VSS
   - BGW Multiplication Improved
5. [AAY21]
   - Natural Barrier
   - Weak VSS

Secure Multiparty Computation



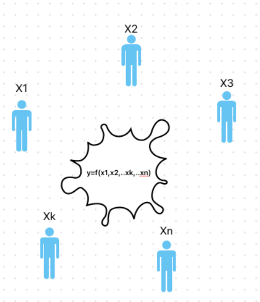
MPC: A number of parties compute a function over private inputs jointly.

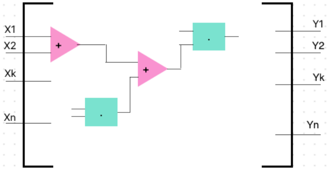
BGW Introduction

1988: Ben-Or, Goldwasser, Widgerson

- For every abstract function
- Synchronous network
- Pairwise private channels
- Threshold corruption
- Unbounded,static adversary
- Perfect Security

High Level View





Input $\Rightarrow$ (arithmetic circuit) $\Rightarrow$ Output

Shamir Secret Sharing

- Finite field $\mathbb{F}$ with $(+, \cdot)$ and $|F| > n$
- $a_1, ..., a_n \in \mathbb{F}$,distinct, non-zero

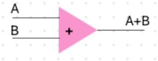
Share:
Choose $\alpha_1, ..., \alpha_t \in \mathbb{F}$ randomly.
Define $A(x) = x_0 + \alpha_1 x_1 + ... + \alpha_t x^t$, where x_0 is the secret value.

Send $share_i = A(a_i)$ for $i = 1, .., n$

Reconstruction:
Needs at least t+1 parties, with Lagrange interpolation.

Arithmetic Circuits

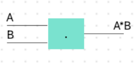
Addition Gates



- $share_i + c$
- $share_i + share'_i$
- $share_i \cdot c$

Addition of random shares:
New random share on **t-degree** polynomial

Multiplication Gates



- $share_i \cdot share'_i$

Multiplication of random shares:
New share on **2t-degree** polynomial

Linear Computation

We can compute linear functions $f(x_1, ..., x_n) = c_1 \cdot x_1 + ... + c_n \cdot x_n$ **without interaction** using only arithmetic circuit with addition gates.

Can we compute any function using addition and multiplication gates without interaction? No!

- Reduce the degree and randomize the polynomial

BGW Multiplication Protocol

- First input wire $a_1, a_2, ..., a_n$ (shares of t-deg,random poly)
- Second input wire $b_1, b_2, ..., b_n$ (shares of t-deg,random poly)
- Output wire $k_1, k_2, ..., k_n$ (shares of 2t-deg poly, not random)
- Add $r_1, r_2, ..., r_n$ (shares of 2t-deg,random,zero constant poly)

- Result: $c_1, c_2, ..., c_n$ only constant and first t terms.

GRR Multiplication Protocol

There are 2 ways to describe t degree polynomials:

1. Through their t+1 coefficients.
2. Through evaluation of polynomial on t+1 distinct points.

There is a way to map 1,2 with **linear operation**.

$k_1, ..., k_n$, shares of $A(x) = a_0 + a_1 x + ... + a_{2t} x^{2t}$ and a_0:secret

$$[a_0, a_1, ..., a_{2t}] \begin{bmatrix} 1 & 1 & ... & 1 \\ 1 & 2 & ... & 2t+1 \\ 1^2 & 2^2 & ... & 2t+1^2 \\ ... & & & \\ 1^{2t} & 2^{2t} & ... & (2t+1)^{2t} \end{bmatrix} = [A(1), A(2), ..., A(2t+1)]$$

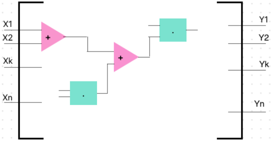
$$a_0 = \lambda_1 A(1) + \lambda_2 A(2) + ... + \lambda_{2t+1} A(2t+1)$$

GRR Multiplication Protocol

Each share k_i is actually $A(a_i)$!!

- First input wire $a_1, a_2, ..., a_n$ (shares of t-deg,random poly)
- Second input wire $b_1, b_2, ..., b_n$ (shares of t-deg,random poly)
- Output wire $k_1, k_2, ..., k_n$ (shares of 2t-deg poly, not random)
- Locally compute coefficients $\lambda_1, ..., \lambda_n$
- Each party secretly shares $k_1, k_2, ..., k_n$
  - Choose a random t degree poly with constant term k_i
- Compute $c_i = \lambda_1 k_{1i} + \lambda_2 k_{2i} + ... + \lambda_n k_{ni}$, on deg-t poly

BGW



Every party P_i:

- Has input x_1
- Secretly shares x_i and sends $x_{1i}, ..., x_{ni}$
- Initializes the circuit with subshares received
- Computes circuit
- Subshares the output $y_{1i}, ..., y_{ni}$
- Reconstructs the secret with subshares received

Malicious BGW

1. Malicious parties can send wrong shares at reconstruction
2. Some parties may not receive shares
3. Inconsistent shares throughout the protocol
4. Dealer may not act honestly

<u>Solution</u>

- Reed-Solomon Codes (up to $\dfrac{n-t}{2}$ errors)
- Broadcast complaints to dealer
- Use Bivariate Polynomials for VSS
- Players vote "Good" if their view is consistent

Bivariate Polynomials

Hide secret in bivariate polynomial of degree t

$$S(x,y) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{i,j}x^i y^j \text{ where } a_{0,0} = s$$

- Convert to univariate : $f_i(x) = S(x, a_i)$ and $g_i(y) = S(a_i, y)$
- $f_i(a_j) = S(a_j, a_i) = g_j(a_i)$
- Parties can authenticate their shares
- Use $f(x)$ as sharing and $g(y)$ for verifying

BGW VSS

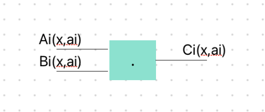
Dealer chooses $S(x,y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{i,j} x^i y^j$

Every party P_i:

1. Receives input $f_i(x) = S(x, a_i)$ and $g_i(y) = S(a_i, y)$
2. Secretly sends $f_i(a_j)$ and $g_i(a_j)$ to every P_j
3. Every P_j verifies the received shares and broadcasts complains if any
4. Dealer broadcasts the correct shares from complaints.
5. Parties vote "good" if what they saw was consistent

Communication complexity: $O(n^2)$

BGW Multiplication Improved



New Multiplication protocol:

1. Each P_i shares $f_i^a(a_j) = A(a_j, a_i), f_i^b(a_j) = B(a_j, a_i)$ and $C_i(x, a_j), C_i(a_j, y)$ to P_j
2. Each P_i proves share is correct
3. All parties compute $C_1, ..., C_n$ with $\lambda_1, ..., \lambda_n$ coefficients

Communication complexity: $O(n^2)$

Verification Protocol

Proof of correctness

1. Dealer defines $D_1(x), ..., D_t(x)$ polynomials such that :

$$C_i(x,0) = f_i^a(x) f_i^b(x) - \sum_{l=1}^{t} x^l D_l(x,0) \ (1)$$

2. Dealer shares $D_1(x), ..., D_t(x)$ with parties
3. All parties verify that (1) holds

Communication Complexity of MultGMW

Communication Complexity of GMW Multiplication Protocol

- Natural barrier in communication
- Each party shares its local multiplication
- Overall complexity is $\Omega(n \cdot comm(VSS))$
- BGW protocols did not meet this barrier
- until [AAY21] paper : $\Omega(n^2 \cdot comm(VSS))$

Weak VSS

Share a bivariate $D(x, y)$ (2t,t)-degree instead of $D_1(x), ..., D_t(x)$ polynomials.
We need **only one VSS** for this.

- VSS will be the same
- If 2t+1 parties voted good, we accepted
- Not efficient for 2t degree $f_i(x) = D(x, a_i)$ because there are t corrupted parties

Solution : Ask dealer to reveal the $g_i(y)$ whenever there is a conflict.

Weak VSS

Honest dealer: 2t+1 honest parties will have $f_i(x)$ and $g_i(y)$ correctly, reconstruction efficient

Corrupted dealer: t+1 honest parties will have correct $f_i(x)$ and $g_i(y)$

Solution : Dealer helps with reconstruction

- Dealer broadcasts $S(0, y)$ (t degree).
- Only those who have $f_i(0)$ check if $f_i(0) = S(0, a_i)$.
- Broadcast good if check was successful

Communication complexity: $O(n \cdot comm(VSS))$ and so $O(n^3)$

Round complexity: $O(depth|C|)$

Thank you!