COT6427 – Secret Sharing

Assignment 2 - MPC (addition and multiplication gates)

Written by:

Christopher Foley
Z15092976

Academic Year: 2017-2018

Introduction

Secure multi-party communication can be defined as a problem involving n players who each have a secret. They need to compare/share their secrets in such a way as to not reveal their secret or discover the secrets of others. One method involves submitting the secrets to a trusted third party, who will then perform the computation and reveal the result to all parties. This practice is used in the "real world" in a number of cases. Specifically, market prices of goods and services, college quality studies, genome studies and fare calculation.[1] The use of a secure third party can be eliminated if all parties agree on one player (perhaps lowest/highest id) to perform calculations.

Process

Each player will calculate a share relative to their secret, relative to a given modulus, and then via secured channels transmit their share to the computation party.

For purposes of this example we will use a modulus of 31 and a threshold of 3 with 5 parties involved. The values were chosen for simplicity of the pencil/paper calculations.

We will create an MPC gate which will add and multiply values before disclosing the secret shares to all users.

We begin with all players calculating and exchanging their share with respect to their own secret:

$$P_1: f_1(x) = \alpha_1 + a_1 x_1 + a_2 x_2^{\,2} + \ldots + a_n x_n^{\,n}$$

$$P_2: f_2(x) = \alpha_2 + b_1 x_1 + b_2 x_2^{\,2} + \ldots + b_n x_n^{\,n}$$

.

.

.

$$P_n: f_n(x) = \alpha_n + \beta_1 x_1 + \beta_2 x_2^{\,2} + \ldots + \beta_n x_n^{\,n}$$

It is clear that if the circuit simply adds the shares, then the resulting shares will contain the shares of the secret.

$$f_1(x) + f_2(x) + \ldots + f_n(x) = (\alpha_1 + \alpha_2 + \ldots + \alpha_n) + (a_1 + b_1 + \ldots \beta_1)\, x + \ldots + (a_n + b_n + \ldots \beta_n) x^n$$

When multiplication occurs, the resulting polynomial will be a polynomial of degree n^2 (where n is

1 "Protecting Privacy with Secure Multi-Party Computation", https://www.newamerica.org/oti/blog/protecting-privacy-secure-multi-party-computation/

the threshold - 1). To correct this we must reduce the degree of the polynomial. We will demonstrate in the following example:

As stated previously, we will assume 5 players, threshold of 3, $\mathbb{Z}_{31}$.

$$f(x)=3+x+x^2$$
$$g(x)=7+x+x^2$$
$$h(x)=5+x^2$$

The goal will be to share $(f(x)+g(x))\times h(x)$ as a final secret. Each player computes their share with respect to the speficied polynomial and the computations performed as specified in the share.

	P_1	P_2	P_3	P_4	P_5
f(x)	5	9	15	23	2
g(x)	9	13	19	27	6
h(x)	6	9	14	21	30
f(x)+g(x)	14	22	3	19	8
$(f(x)+g(x))\times h(x)$	22	12	11	27	23

The final row of the above calculations contain a new secret that we will then split and divide shares among the participants. However we must first reduce the polynomial from a 4th degree polymomial (square of the threshold) to share to a quadratic (threshold – 1). To do this we use the final shares as a secret with the given threshold and a random polynomial. For simplicity we will use the following “random” polynomials.

$$k_1(x)=22+x+x^2$$
$$k_2(x)=12+x+x^2$$
$$k_3(x)=11+x+x^2$$
$$k_4(x)=27+x+x^2$$
$$k_5(x)=23+x+x^2$$

Each player will evaluate their polynomial with respect to the ids and exchange the resulting values to form a matrix similar to the following:

$$\begin{bmatrix} k_1(1) & k_1(2) & k_1(3) & k_1(4) & k_1(5) \\ k_2(1) & k_2(2) & k_2(3) & k_2(4) & k_2(5) \\ k_3(1) & k_3(2) & k_3(3) & k_3(4) & k_3(5) \\ k_4(1) & k_4(2) & k_4(3) & k_4(4) & k_4(5) \\ k_5(1) & k_5(2) & k_5(3) & k_5(4) & k_5(5) \end{bmatrix} \rightarrow \begin{bmatrix} 24 & 28 & 3 & 11 & 21 \\ 14 & 18 & 24 & 1 & 11 \\ 13 & 17 & 23 & 0 & 10 \\ 29 & 2 & 8 & 16 & 26 \\ 25 & 29 & 4 & 12 & 22 \end{bmatrix}$$

The first column of secrets ($k_1(1)$, $k_2(1)$, $k_3(1)$, $k_4(1)$, $k_5(1)$) will be sent to the first player, $k_{1..5}(2)$ sent to the second player and so on through $k_{1..5}(5)$ to player 5. These matrices of secrets will then be multiplied by the first row of the inverse Vandermonde matrix by each player to discover a new secret γ_n to replace the prior secret in their private store. This gives us the following:

$$\gamma_1 = (5*24+21*14+10*13+26*29+1*25) \bmod 31 = 21$$
$$\gamma_1 = (5*28+21*18+10*17+26*02+1*29) \bmod 31 = 25$$
$$\gamma_1 = (5*03+21*24+10*23+26*08+1*04) \bmod 31 = 0$$
$$\gamma_1 = (5*11+21*01+10*00+26*16+1*12) \bmod 31 = 8$$
$$\gamma_1 = (5*21+21*11+10*10+26*26+1*22) \bmod 31 = 18$$

These individual values, [21, 25, 0, 8 18] are shares of a quadratic polynomial which may then be used to identify the new secret through Lagrange Interpolation, as shown below.

$$\left(\frac{2-0}{2-1}\right)\left(\frac{3-0}{3-1}\right)(21)+\left(\frac{1-0}{1-2}\right)\left(\frac{3-0}{3-2}\right)(25)$$
$$\left(\frac{2}{1}\right)\left(\frac{3}{2}\right)(21)+\left(\frac{1}{-1}\right)\left(\frac{3}{1}\right)(25)$$
$$(3)(21)+(-3)(25)$$
$$63-75=-12 \bmod 31 = 19$$

Which is the expected secret from $(f(x)+g(x))\times h(x)$ or $(3+7)*5 ¿= 50 \bmod 31 = 19$.