Fully Distributive Evaluation Protocol: Code

Vivek Sharma

Jan 11, 2020

1 Code: Basic

1.1 First Step:

Server S_1 holds x_2, k_2, x_3, k_3 and computes $h_1 = x_2 \cdot k_3 + x_3 \cdot k_2 + x_2 \cdot k_2$
Server S_2 holds x_1, k_1, x_3, k_3 and computes $h_2 = x_1 \cdot k_3 + x_3 \cdot k_1 + x_3 \cdot k_3$
Server S_3 holds x_1, k_1, x_2, k_2 and computes $h_3 = x_1 \cdot k_2 + x_2 \cdot k_1 + x_1 \cdot k_1$

1.2 Phase 2: Interactive computation of π_{23} protocol

1. Each server, at this point have locally computed their shares, which was the multiplication of two vectors.

2. Server 1 randomly chooses a value $c \in \mathbb{Z}_3^m$ and each bit of value is converted to it's 2-bit representation to form c_0 and c_1 respectively.

3. Meanwhile, Server 1, 2 and 3 runs sub-protocol for m instances(m is the length of additive share and also the value of c, which is with server 1.

   For $1 \leq j \leq m$:

   Each server $s_i, i \in 1, 2, 3$ share their input $h_{i,j}$ [Note: $h_{i,j}$ is the input of server s_i in j^{th} iteration]

   Compute combined XOR of their input: $comb := h_1 \oplus h_2 \oplus h_3$

   Multiply one part of c(c_0), with comb and other part (c_1) with $\neg comb$ and XOR both the result, this forms d_0.

   To compute d_1, XOR the c_0 and $\neg c_1$, and multiply the result with the XOR of secret share of the servers.

   The final result $d = d_0, d_1 \in \{0, 1\}^2$ is converted back into $\mathbb{Z}_3$

At the end of this phase, Server 1 has $c \in \{0,1\}^m$ and Server 2 has received the output $d \in \mathbb{Z}_3$. The combination of values with Server one and two (i.e. c and d) yields the additive mod 3 of the secret share of the inputs by the Servers. Mathematically $c+d = h_1+h_2+h_3(mod 3)$