# Weak PRF Protocol: Pseudocode

Vivek Sharma

Jan 11, 2020

## 1 Fully Distributive Evaluation Protocol

The key is structured as a vector in $\mathbb{Z}_2$

The protocol is divided into three phases:

### 1.1 Phase 1:

Each server $S_i$ holds replicated additive shares of key $k_i \in \mathbb{Z}_n^2$ and $x_i \in \mathbb{Z}_n^2$ and computes h, which is the multiplication of key and input over $\mathbb{Z}_2$. This computation is performed locally.

### 1.2 Phase 2:

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: $comb := h_1 \oplus h_2 \oplus h_{13}$

   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 $d \in \mathbb{Z}_3$

### 1.3 Phase 3:

Explanation: