

Multiple Secret Leaders

Authors here

June 22, 2023

1 Some Notation or Smthng

1. We will have n parties
2. We will have k leaders elected
3. We will have a “bid” published by a user $i \in [n]$ be denoted as b_i
4. We will denote a commitment as comm_i
5. We will denote some generic CRHF as h
6. We will say Enc_{TFHE} and Dec_{TFHE} for TFHE encoding and decoding respectively

2 Sketching shtuff out

Say we want to elect k leaders (maybe with or without repetition) secretly out of n parties, we can run SSLE k times but that’d be painful. Let’s not do that.

2.1 Simple sorting

A simple approach with runtime (assuming $k \leq n$) $O(k + n \log n)$ is to use sorting. At a high level, the idea is to somehow randomly sort the n parties then elect the top k parties in the sort. To do this we can use TFHE in a very similar way to the SSLE paper.

1. **Setup:** Same exact setup with TFHE in SSLE. Setup TFHE and distribute shares. Each party also has some secret s_i and commitment $\text{comm}_i = \text{comm}(s_i)$.
2. **Publish Bid:** Each party samples $r_i \leftarrow U$ and publish $b_i = (\text{Enc}_{\text{TFHE}}(r_i), \text{Enc}_{\text{TFHE}}(\text{comm}_i))$.

```
 $\alpha \leftarrow \text{ROM}(b_1, b_2, \dots)$ 
for  $i \in [n]$  do
   $b'_i \leftarrow (\text{Enc}_{\text{TFHE}}(r_i) - \alpha, \text{Enc}_{\text{TFHE}}(\text{comm}_i))$ 
 $\text{sorted} = [b'_{j_1}, b'_{j_2}, \dots] \leftarrow \text{Sort}_{\text{TFHE}}([b'_1, b'_2, \dots])$ 
return first  $k$  elements of  $\text{sorted}$ 
```

Figure 1: Simple Sorting Evaluation

```

Input:  $L$ , set of  $(\text{bid}, \text{level})$  pairs. New bid and level,  $(b, \ell)$ 
Output: New list  $L$  with remaining evaluations
if there is some for some  $b', (b', \ell) \in L$  then
     $\alpha \leftarrow \text{ROM}(b, b', \ell)$ 
     $\bar{b} \leftarrow \text{game}(b, b')$ 
     $L' \leftarrow L \setminus \{ (b', \ell) \}$ 
    return  $\text{EvalStream}(L', (\bar{b}, \ell + 1))$ 
else
     $L' \leftarrow L \cup \{ (b, \ell) \}$ 
    return  $L'$ 

```

Figure 2: Simple Sorting Evaluation

3. **Evaluation:** This is done by one party whom may even be an external party. See [fig. 1](#) for details
4. **Decoding:** At least t parties publish their threshold decryptions of the return from the evaluation (TODO: is this how TFHE works...)
5. **Proving:** For a party i to prove that they are the selected leader for the j th slot, they prove that they know the secret to produce comm_i where comm_i is the decrypted commitment for the i th slot

2.2 Distributed/ Streaming Simple Sorting

Assume that n, k are a power of 2. We will keep a lot of the same from sorting but now look at selection as a sort of tournament. For completeness, we will write out each step again

1. **Setup:** Same exact setup with TFHE in SSLE. Setup TFHE and distribute shares. Each party also has some secret s_i and commitment $\text{comm}_i = \text{comm}(s_i)$.
2. **Publish Bid:** Each party samples $r_i \leftarrow U$ and publish $b_i = (\text{Enc}_{\text{TFHE}}(r_i), \text{Enc}_{\text{TFHE}}(\text{comm}_i))$.
3. **Evaluation:** This step is different than that in the simple sorting version ([section 2.1](#)). Now, we can evaluate in a streaming fashion and even distribute evaluation (though we'll not go into details about the distributed implementation).
4. **Decoding:** At least t parties publish their threshold decryptions of the return from the evaluation (TODO: is this how TFHE works...)
5. **Proving:** For a party i to prove that they are the selected leader for the j th slot, they prove that they know the secret to produce comm_i where comm_i is the decrypted commitment for the i th slot

References