### Efficient Primitive Protocols for Sharemind

Bingsheng Zhang<sup>1,2</sup>

<sup>1</sup>Cybernetica AS, Estonia <sup>2</sup>University of Tartu, Estonia

Estonian Theory Days in Elva, 11.06.2010-13.06.2010



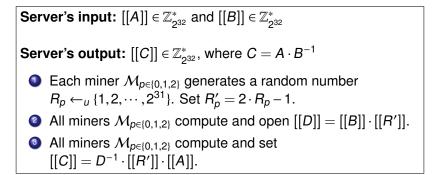
・ロト ・ 御 ト ・ ヨ ト ・ ヨ ト … ヨ

- Division in  $\mathbb{Z}^*_{2^{32}}$  Multiplication in  $\mathbb{Z}^*_{2^{32}}$
- High degree Conjunction
- Random Shuffle Protocol



# Division in $\mathbb{Z}^*_{2^{32}}$





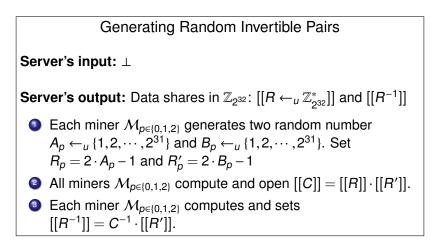
◆□▶ ◆□▶ ◆三▶ ◆三▶ ◆□▶

### The total protocol costs 3 rounds.

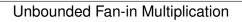


# Multiplication in $\mathbb{Z}^*_{2^{32}}$





#### The total protocol costs 2 rounds.



**Server's input:** Data shares in  $\mathbb{Z}_{2^{32}}^*$ :  $[[X_1]], \cdots, [[X_k]]$ 

**Server's output:** Data shares in  $\mathbb{Z}_{2^{32}}^*$ :  $[[\prod_{i=1}^k X_i]]$ 

All miners M<sub>p∈{0,1,2}</sub> generate random invertible pairs ([[R<sub>0</sub>]], [[R<sub>0</sub><sup>-1</sup>]]), ··· , ([[R<sub>k</sub>]], [[R<sub>k</sub><sup>-1</sup>]]) by using sub-protocol in previous section.

Seach miner  $\mathcal{M}_{p \in \{0,1,2\}}$  computes  $B = \prod_{i=1}^{k} A_i \quad (= R_0 \cdot \prod_{i=1}^{k} X_i \cdot R_k^{-1}).$ 

■ All miners  $\mathcal{M}_{p \in \{0,1,2\}}$  compute  $[[S]] = [[R_0^{-1}]] \cdot B \cdot [[R_k]].$ 

### The total protocol costs 3+2 rounds.

## High degree Conjunction

◆□▶ ◆□▶ ◆三▶ ◆三▶ ◆□▶

**Server's input:**  $[[X_1]], \dots, [[X_k]] \ (X_i \in \{0, 1\})$ 

Server's output:  $[[Y]] = [[X_1 \land \cdots \land X_k)]]$ 

- All miners  $\mathcal{M}_{p \in \{0,1,2\}}$  computes  $[[S]] = \sum_{i=1}^{k} [[X_i]]$ .
- ② All miners *M*<sub>p∈{0,1,2}</sub> call Equal sub-protocol to check if [[*S*]] = k and return the result bit as [[*Y*]].

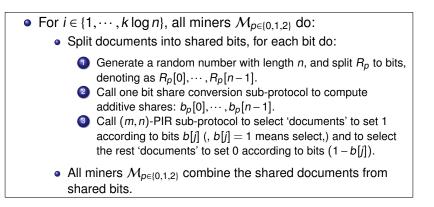
This protocol was improved by Margus Niitsoo's comments. It takes the same rounds as equality check protocol, which is 7 rounds. In theory, it is  $O(\log \log k)$  rounds protocol, where k is the degree.



### Outline

## Random Shuffle Protocol

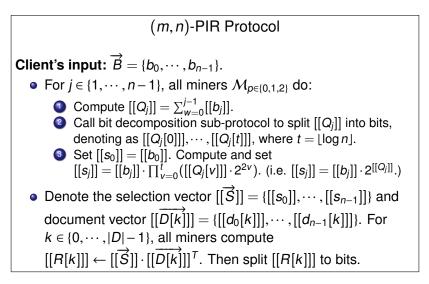
### Outline Random Shuffle Protocol



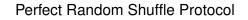
Outline

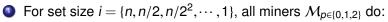
 $O(k \log^2 n)$  rounds, given that (m, n)-PIR sub-protocol takes log *n* rounds.  $O(n \log n)$  computation, given that (m, n)-PIR sub-protocol costs O(n) computation regardless *m*.

### Outline Random Shuffle Protocol



Outline





- Create array A<sub>p</sub>[i] = 0. Pick i/2 positions randomly, and set them to 1. Now A<sub>p</sub>[i] can be regarded as random number with hamming weight exactly i/2.
- **2** For u = 0, 1, 2, miner  $M_u$  shares  $A_u[i]$  bitwisely:  $[[b[0]]], \dots, [[b[i-1]]].$ 
  - Call (m, n)-PIR sub-protocol to select 'documents' to set 1 according to bits b[j] (, b[j] = 1 means select,) and to select the rest 'documents' to set 0 according to bits (1 b[j]).
- O All miners M<sub>p∈{0,1,2}</sub> will execute recursively in a parallel for sets 0 and 1 for next round.



Outline

### Thank You! Questions?