# Multivariate basic function secret sharing from oblivious transfer

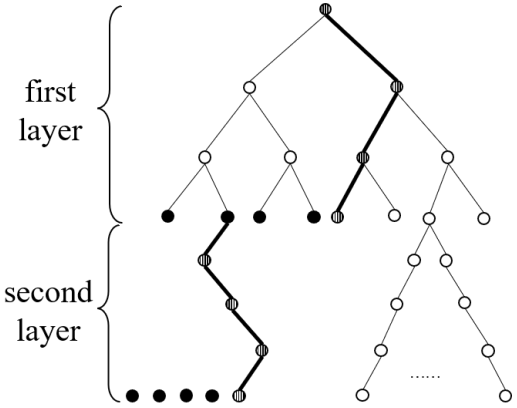
**Yanqing YAO, Fangyuan MIN**

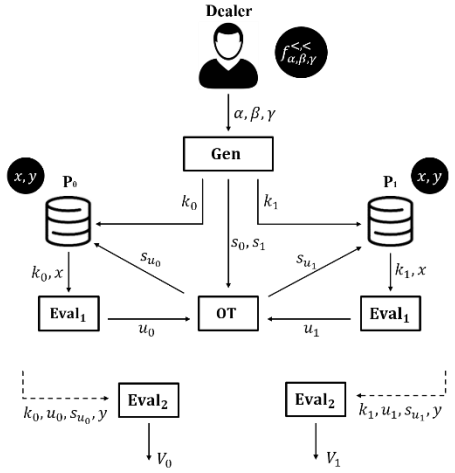
Frontiers of Computer Science, DOI: 10.1007/s11704-025-40919-y

Footnote: The two authors contributed equally to this work.

# Problems & Ideas

- Problems of existing function secret sharing schemes:
  - The function classes corresponding to the current DPF and DCF schemes are almost all unary function classes.
  - There is no efficient FFS construction for multivariate function classes.
  - The applications of FSS can be extended with the development of a multivariate scheme.
- Ideas: Multivariate basic function secret sharing scheme that employs oblivious transfer to realize the transition transfer of seeds based on “two/ multi-layer binary tree” structure.




Left: Two-layer binary tree structure of binary DCF; Right: Framework of Binary Distributed Comparison Function from OT

# Main Contributions

- Contributions:
  - A new “two-layer binary tree” structure for constructing binary DCF, where the OT protocol acts as a “bridge” to connect the layers;
  - An optimization batch computation scheme of one-time transmission of multiple seed pairs using extended OT to reduce communication cost;
  - FSS for multivariate mixed basic functions based on “multilayer binary tree” and 2-layer structure by serial and parallel methods respectively;
  - The applications of our schemes in 2-server multi-keyword PIR.

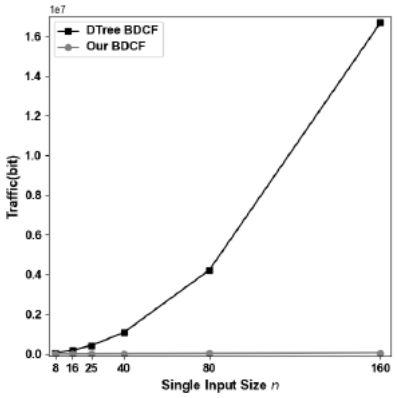



Fig. 6 Comparison of total traffic.

| Single input size | Our scheme | | | Scheme in [3] |
|---|---|---|---|---|
| | key | OT1 | OT2 | key |
| $n = 8$ | 3376 | 1024 | 1280 | $5.18 \times 10^4$ |
| $n = 16$ | 6496 | 1024 | 1280 | $1.86 \times 10^5$ |
| $n = 25$ | 10006 | 1024 | 1280 | $4.35 \times 10^5$ |
| $n = 40$ | 15856 | 1024 | 1280 | $1.08 \times 10^6$ |
| $n = 80$ | 31456 | 1024 | 1280 | $4.22 \times 10^6$ |
| $n = 160$ | 62656 | 1024 | 1280 | $1.67 \times 10^7$ |

**Table 1** Comparison of scheme communication efficiency. The unit of data in the table is bit.

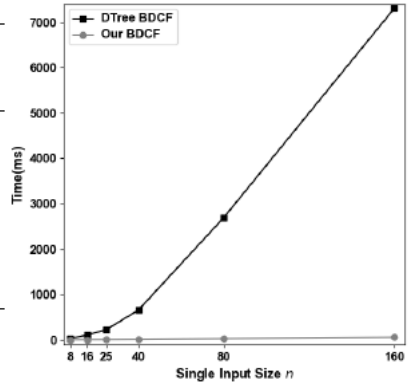



Fig. 7 Comparison of computation time.

| Single input size (bit) | Our scheme | | Scheme in [3] |
|---|---|---|---|
| | $Eval_1$ | $Eval_2$ | Eval |
| $n = 8$ | 8 | 16 | 320 |
| $n = 16$ | 16 | 32 | 1152 |
| $n = 25$ | 25 | 50 | 2700 |
| $n = 40$ | 40 | 80 | 6720 |
| $n = 80$ | 80 | 160 | 26240 |
| $n = 160$ | 160 | 320 | 103680 |

**Table 2** Comparison of computation efficiency of Eval algorithm. The data in columns 2 to 4 represents the times of AES operation required in each algorithm.

Comparison of our BDCF scheme with the BDCF scheme constructed based on the decision tree (Dtree) FSS of Boyle et al. [3].