Secure Multi-Party Computation: Millionaires' Problem, Coin Tossing Problem and Federated Learning

Ziqin Li

System Research Association @ School of Cyber Science and Engineering, Sichuan University
NEXTLAB @ Sichuan University

September 13, 2024

1. Coin Tossing Problem

2. Millionaires’ Problem

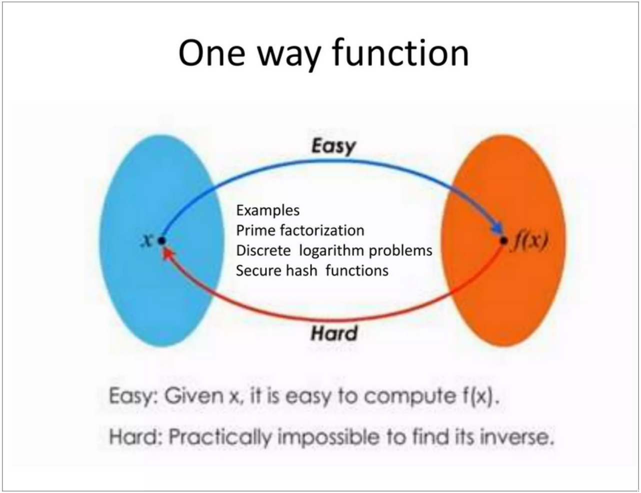
3. Federated Learning

Let's begin with a simple problem...

Consider that I'm making a bet with Junyu over the phone: let's say a bet of $100, and then we decide to use, say, a coin flip to decide who should pay who the $100. Here's the problem: how do we flip this coin?

The problem can be rephrased: how do we (2 parties) generate a unbiased random bit?
Original post by Manuel Blum[Blu83]. (Turing Award, 1995)

One way function



One way function

Examples
Prime factorization
Discrete logarithm problems
Secure hash functions

Easy: Given x, it is easy to compute f(x).

Hard: Practically impossible to find its inverse.

Figure: One way function

Solution by Blum[Blu83]

1. A flip coin: 0/1
2. choose a very big number: S, 0 -> odd / 1 -> even
3. A send to B H(S)
4. B decide to flip or not to flip it
5. A send S to B, proof H(S)

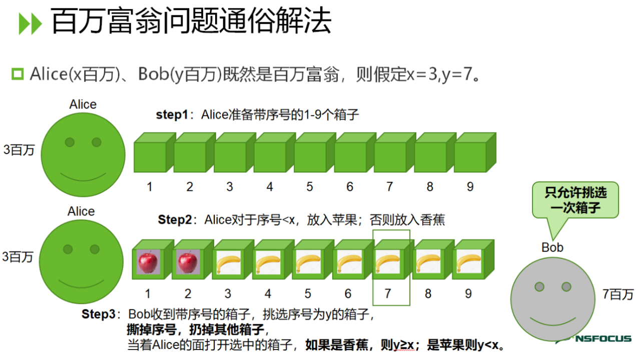
Another simple problem

Consider that I'm currently competing with Junyu to see who has more money, but neither of us wants anyone to know how much we have. (Though the reality is probably that we're comparing who has less money)

The question can be rephrased as: determine whether the inequality $a \geq b$ is true or false without revealing the actual values of a a and b.

Original post by Andrew Yao[Yao82]. (Turing Award, 2000)

A simple solution



百万富翁问题通俗解法

□ Alice(x百万)、Bob(y百万)既然是百万富翁，则假定x=3,y=7。

step1：Alice准备带序号的1-9个箱子

Step2：Alice对于序号<x，放入苹果；否则放入香蕉

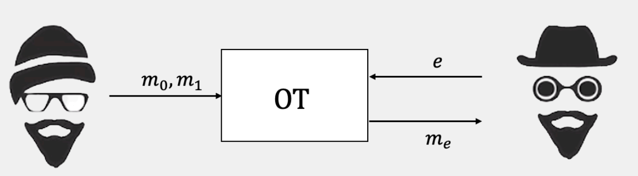
Step3：Bob收到带序号的箱子，挑选序号为y的箱子，**撕掉序号，扔掉其他箱子**，当着Alice的面打开选中的箱子，**如果是香蕉，则y≥x；是苹果则y<x。**

只允许挑选一次箱子

Figure: solution, but not rigorous

How to guarantee Bob will drop other box?
work in face-to-face situation, but not online

Oblivious Transfer



Question: How to construct a 1-out-of-N OT from 1-out-of-2 OT?

Trivial method: *cela va sans dire*
Non-trivial method: O(logn) by [NP99]

Construct 1-out-of-2 OT by trapdoor function

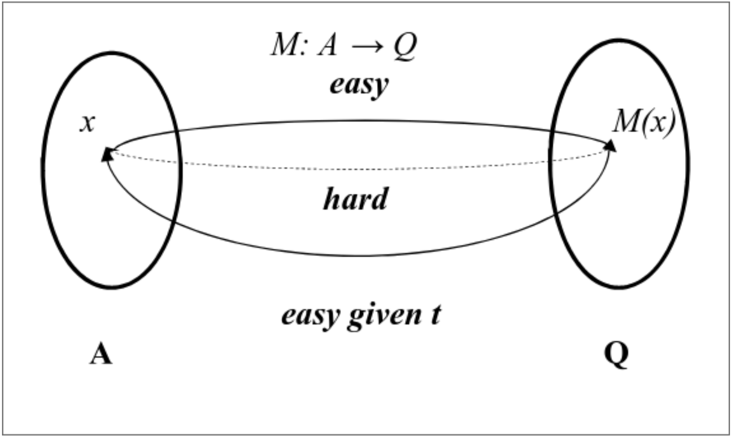


Figure: trapdoor function: eg. prime factor

Construct 1-out-of-2 OT by trapdoor function

1. A: $(f_1, t_1), (f_2, t_2) \leftarrow G$
2. B: $key \leftarrow G$
3. B: send $C = f_s(key)$
4. A: $key_1 = f_1^{-1}(C, t_1), key_2 = f_2^{-1}(C, t_2)$
5. A: $c_1 = Enc(m_1, key_1), c_2 = Enc(m_2, key_2)$
6. B: $m_s = Dec(c_s, key)$

It is easy to demonstrate that the construction is easily expandable to 1-out-of-N OT.

Expand & Disclaimer

1. The existence of such one-way functions is still an open conjecture. (P=NP)
2. A more general primitive for solving SMPC problem is GC (garbled curcuit)[GMW19, Yao82]
3. By the way, W in GMW is Avi Wigderson (Turing Award, 2023).
4. The best results on the fair coin-flip problem come from the [MNS09].
5. You could find the detailed and rigorous proof of security of Yao's protocol and GMW's protocol from [AL17] and [LP09].
6. For an overview of SMPC, see Lindell's survey[Lin20].

Applications

Secure Statistical Analysis

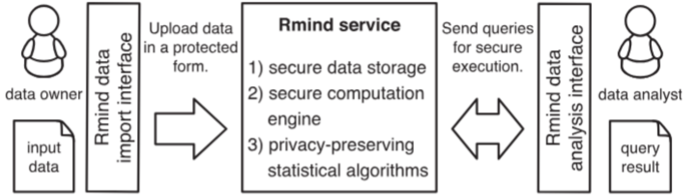

data owner
input data
Rmind data import interface
Upload data in a protected form.
Rmind service
1) secure data storage
2) secure computation engine
3) privacy-preserving statistical algorithms
Send queries for secure execution.
Rmind data analysis interface
data analyst
query result

Fig. 1. The main components and interfaces to RMIND.

Figure: Rmind: A Tool for Cryptographically Secure Statistical Analysis[BKLS16]

Applications

Federated Learning

1. Classic federated learning[MMR+17] use non-cryptographic security method like FedAvg.
2. MiniONN[LJLA17] use AHE to finish linear computation, and secret sharing & garbled curcuit to do nonlinear computation.
3. CryptoNets[GBDL+16] use leveled FHE for all computation.

Applications

Federated Learning

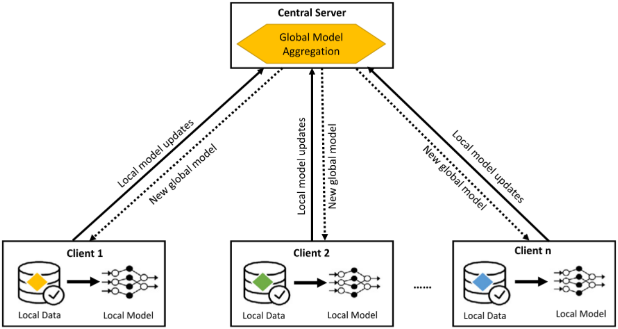


Fig. 1. Central federated learning.

Figure: Picture references: [KWL^{+}22]

Applications

PMT for Password Reuse Detection

PMT: Private Membership Test

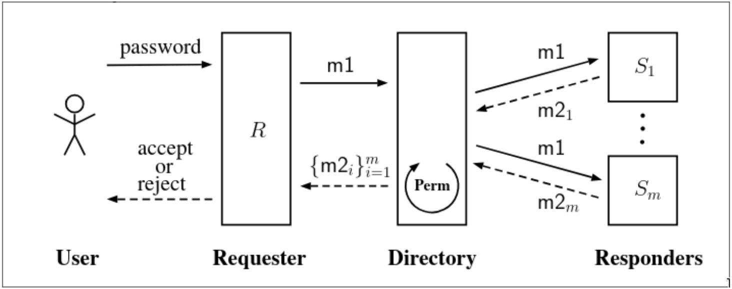


Figure: Architecture of the framework shown in [WR18]

References I

Gilad Asharov and Yehuda Lindell.
A full proof of the bgw protocol for perfectly secure multiparty computation.
Journal of Cryptology, 30(1):58–151, 2017.

Dan Bogdanov, Liina Kamm, Sven Laur, and Ville Sokk.
Rmind: a tool for cryptographically secure statistical analysis.
IEEE Transactions on Dependable and Secure Computing, 15(3):481–495, 2016.

Manuel Blum.
Coin flipping by telephone a protocol for solving impossible problems.
ACM SIGACT News, 15(1):23–27, 1983.

References II

Tung Chou and Claudio Orlandi.
The simplest protocol for oblivious transfer.
In *Progress in Cryptology–LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23-26, 2015, Proceedings 4*, pages 40–58. Springer, 2015.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing.
Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy.
In *International conference on machine learning*, pages 201–210. PMLR, 2016.

References III

Oded Goldreich, Silvio Micali, and Avi Wigderson.
How to play any mental game, or a completeness theorem for protocols with honest majority.
In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328. 2019.

Renuga Kanagavelu, Qingsong Wei, Zengxiang Li, Haibin Zhang, Juniarto Samsudin, Yechao Yang, Rick Siow Mong Goh, and Shangguang Wang.
Ce-fed: Communication efficient multi-party computation enabled federated learning.
Array, 15:100207, 2022.

Yehuda Lindell.
Secure multiparty computation.
Communications of the ACM, 64(1):86–96, 2020.

References IV

Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan.
Oblivious neural network predictions via minionn transformations.
In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 619–631, 2017.

Yehuda Lindell and Benny Pinkas.
A proof of security of yao’s protocol for two-party computation.
Journal of cryptology, 22:161–188, 2009.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas.
Communication-efficient learning of deep networks from decentralized data.
In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

References V

Tal Moran, Moni Naor, and Gil Segev.
An optimally fair coin toss.
In *Theory of Cryptography Conference*, pages 1–18. Springer, 2009.

Moni Naor and Benny Pinkas.
Oblivious transfer and polynomial evaluation.
In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254, 1999.

Manuel B Santos, Armando N Pinto, and Paulo Mateus.
Quantum and classical oblivious transfer: A comparative analysis.
IET Quantum Communication, 2(2):42–53, 2021.

References VI

Ke Coby Wang and Michael K Reiter.
How to end password reuse on the web.
arXiv preprint arXiv:1805.00566, 2018.

Andrew C Yao.
Protocols for secure computations.
In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

Acknowledge & Ads

1. Thank Prof. Lan for her guidance to me on SMPC ,academic writing and fact check this slide.
2. Thank Prof. Gao for his guidance in academic writing, how to do real research, and his continuous support in personal and association affairs.

Thanks for Listening!