Oblivious Transfer

Suppose Alice has two messages \( m_0 \) and \( m_1 \)

- Suppose Bob has a bit \( b \)
- Bob wants to have \( m_b \)

Constraints:

- Bob does not want Alice to know \( b \)
  - Or, equivalently, which \( m_b \) he wants
- Alice does not want Bob to know both \( m_0 \) and \( m_1 \)
1-2 Oblivious Transfer

(The RSA-based version)

Alice generates an RSA key: $N$, public $e$, private $d$

- **A**
  - msgs $m_0$, $m_1$
  - random $x_0$, $x_1$
  - $t_0 = m_0 + (q-x_0)^d$
  - $t_1 = m_1 + (q-x_1)^d$

- **B**
  - bit $b$
  - random $k$
  - $q = k^e + x_b \pmod{N}$
  - $t_b - k$
  - $t_b - k = m_b$

$q = k^e + x_b \pmod{N}$
1-N Oblivious Transfer

- Alice has $N$ messages
- Bob has an index $i$
- Bob wants to receive $i$-th message without Alice learning $i$
- Alice wants Bob to receive only one message

Related to private information retrieval

- Added database’s privacy requirement
K-N Oblivious Transfer

- Alice has N messages
- Bob wants K of those messages without Alice learning which
- Alice wants Bob to receive only K messages

Two possibilities:
- messages requested simultaneously (non-adaptive)
- messages requested sequentially (adaptively)
  - can depend on previous requests
The Millionaires Problem

(Andrew Yao, 1982)

Alice and Bob are both millionaires

- Alice has $I$ million dollars
- Bob has $J$ million dollars
- Alice and Bob both want to know who's richer
- But they don't want the other to know how much money they have
- For simplicity, assume $1 \leq I, J \leq 4$
The Protocol

(The RSA-based version)

Alice generates an RSA key: N, public e, private d

\( N, e \)

Bob receives \( P, R_1, \ldots, R_4 \):

If \( R_J = x \) mod \( P \)

then \( I \geq J \) (o/w \( I < J \))
Alice generates an RSA key: $N$, public $e$, private $d$

Bob receives $P, R_1, ... , R_4$:

If $R_J = x \mod P$, then $I \geq J$ (o/w $I < J$)

$Z_1 = (C - J + 1)^d \mod P$
$Z_2 = (C - J + 2)^d \mod P$
$Z_3 = (C - J + 3)^d \mod P$
$Z_4 = (C - J + 4)^d \mod P$
Given a publicly known function $F$ of $N$ inputs and producing $N$ outputs:

- $F(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$

Suppose $N$ parties, each party $i$ with a private value $a_i$:

- Goal: compute $F(a_1, \ldots, a_n) = (r_1, \ldots, r_n)$
- Each party $i$ wants to know $r_i$
- No party want others to learn their private value
Secure Multiparty Computation

Oblivious Transfer as a secure multiparty computation:
• Function $F(<m_0,m_1>,b) = (\text{nil},m_b)$
  • Alice has $<m_0,m_1>$, Bob has $b$
  • Bob wants $m_b$ (don’t care about Alice)

Millionaires Problem as a secure multiparty computation:
• Function $F(I,J) = (\text{Alice},\text{Alice})$ if $I \geq J$
  $$= (\text{Bob},\text{Bob})$$ if $I < J$
  • Alice has $I$, Bob has $J$
  • Alice and Bob want to know who’s richer