

Secure Computation

Proof of Honest But Curious:

Perfect world characterizations:

- @ case where $\rightarrow F$ knows who is corrupted
 1 Adversary 1 Honest otherwise it is trivial
 For P_i corrupted P_j not:
 1) F receives x_i for P_i
 2) F receives x_j from P_j
 3) returns $f_i(x_i, x_j)$ to P_i
 4) if P_i HALT then stop
 5) else return $f_j(x_i, x_j)$ to P_j

We say that a protocol Π securely emulates F if
 $\forall A$ against Π , $\exists S$ against F st:

$\text{Real}_{\Pi, A} \approx \text{Ideal}_{F, S}$
 A real & protocol non secure st: real \neq ideal result

wait for proof thing for example

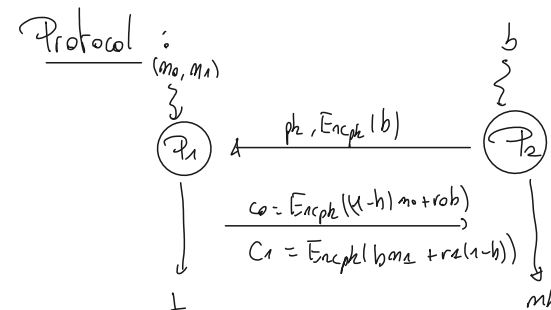
Oblivious Transfer

$$\text{Goal: } \begin{cases} f_1(m_0, m_1, b) = 1 \\ f_2(m_0, m_1, b) = m_b \end{cases}$$

$\Rightarrow P_2$ asks for m_b to P_1 that doesn't know
 Which one P_1 is sending"

\Rightarrow Based on Homomorphic encryption's properties:

$$\begin{cases} C_1, C_2 = \text{Enc}_{pk}(m_1 + m_2) \\ a \cdot C_1 = [\text{Enc}_{pk}(m_1)]^a \end{cases}$$



Note: $\rightarrow C$ and C_1 are computed without knowing b

$$\rightarrow m_b = \text{Dec}_{sk}(C_b)$$

\rightarrow is honest but curious security
 (proof by simulation)

Used for

Yao's Garbled Circuits

$$\text{Goal: } \begin{cases} P_1 \text{ has } x_1 \\ P_2 \text{ has } x_2 \end{cases}$$

$\rightarrow P_2$ need to compute $f_2(x_1, x_2)$ st
 • P_1 learns nothing about x_2
 • P_2 learns nothing about x_1

$$\rightarrow P_1 \text{ has } f_1(x_1, x_2) = 1$$

Note: The fact that $f_1 = 1$ is not an issue
 and can be converted using the following technique

- $f'_1((x_1, r), x_2) = 1$ for a random r and
- $f'_2((x_1, r), x_2) = (f_1(x_1, x_2) \oplus r, f_2(x_1, x_2))$
 \rightarrow and send back $f_1(x_1, x_2) \oplus r$ to P_1 at the end

Procedure:

P_1

P_2

Obfuscation: From f_2 that is public do

i) input wire $\rightarrow k_0, k_1$
 (output wire $\rightarrow 0, 1$ is
 the value to read)
 ii) gate $\rightarrow \text{Enc}_{k_0}, \text{Enc}_{k_1}(g(a, 1 || 0^t))$
 + keys?
 \rightarrow only one pair of key can decrypt
 iii) shuffle

Send
obfuscated
 f_2

Send key:

from x_1 compute the
 k_i (either 0 or 1)

k_i, i

P_2 needs to input the
 keys at the spots of his input...
 Can't just ask P_1 otherwise he
 compromises his key... \Rightarrow OT

for each bit b_i of the key x_2 :

$$\begin{array}{c} \text{Enc}_{k_i}(b_i), pk \\ \hline C_0, C_1 \text{ with } m_0 = k_i^0 \\ m_1 = k_i^1 \end{array}$$

\rightarrow has k_i^b without compromising

can compute solution of $f_2(x_1, x_2)$