

Secure Computation

Proof of honest But Curious :

Perfect world characterizations :

- Look @ case where \mathcal{E} knows who is corrupted
- For P_i corrupted P_j not:
- 1 Adversary 1 Honest otherwise
 - 1) \mathcal{E} receives x'_i for P_i
 - 2) \mathcal{E} 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 \mathcal{F} if
 $\forall A \text{ against } \Pi, \exists S \text{ against } \mathcal{F} \text{ s.t.}$

A real π protocol non secure \Rightarrow Ideal \mathcal{F} is non resultant

Want for proof thing for example

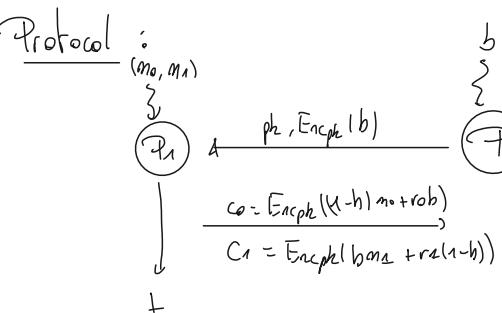
Oblivious Transfer

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_i is sending

\Rightarrow Based on Homomorphic encryption's properties:

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



Note :

- $\rightarrow c_0$ and c_1 are computed without knowing b
- $\rightarrow m_b = \text{Decsk}(c_b)$
- \rightarrow is honest but curious security (proof by simulation)

Used for

Yao's Garbled Circuits

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

$\rightarrow P_2$ need to compute $f_2(x_1, x_2)$ s.t.

- $\bullet P_1$ learns nothing about x_2
- $\bullet P_2$ learns nothing about x_1

$\rightarrow P_1$ 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

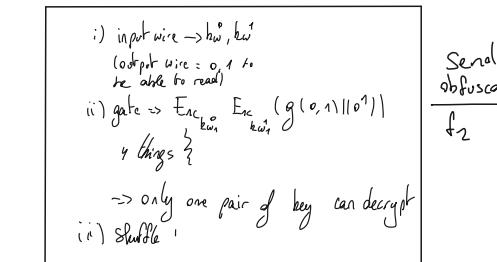
- $\blacktriangleright f'_1((x_1, r), x_2) = \perp$ for a random r and
- $\blacktriangleright f'_2((x_1, r), x_2) = (f_1(x_1, x_2) \oplus r, f_2(x_1, x_2))$
 \rightarrow and send back $f_1(x_1, x_2) \oplus r$ to P_1 at the end

Procedure:

P_1

P_2

Obfuscation: From f_2 that is public do



Send key:

from x_2 compute the
 $k_{i,1}$ (either 0 or 1)

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 :

$\text{Encpk}(h_i), pk$

C, C_1 with $m_0 = k_0^b$
 $m_1 = k_1^b$ \rightarrow has k_i^b without compromising

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