Final - Siddarth Ijju

1.

$\mathcal{L}_1$: View($m \in \mathbb{Z}_n$):

$k \leftarrow \mathbb{Z}_n$
$c = (k+m) \bmod n$
return c

$\mathcal{L}_{hyb-1}$: View($m \in \mathbb{Z}_n$):

$c = A(m)$
return c

A: ($m \in \mathbb{Z}_n$)

$k \leftarrow \mathbb{Z}_n$
$c = (k+m) \bmod n$
return c

$\mathcal{L}_{hyb-2}$: View($m \in \mathbb{Z}_n$):

$c = A(m)$
return c

A: ($m \in \mathbb{Z}_n$)

$c \leftarrow \mathbb{Z}_n$
return c

Since c was originally $(k+m) \bmod n$, and $\mathbb{Z}_n$ has the same domain, it is possible to replace $(k+m) \bmod n$ with a randomly selected $\mathbb{Z}_n$, since $k \leftarrow \mathbb{Z}_n$.

$\mathcal{L}_2$: View($m \in \mathbb{Z}_n$):

$c \leftarrow \mathbb{Z}_n$
return c

$\mathcal{L}_1 \equiv \mathcal{L}_{hyb-1} \equiv \mathcal{L}_{hyb-2} \equiv \mathcal{L}_2$ $\therefore$ QED

2.

$\mathcal{L}^{F'}_{prf\text{-}real}$

$k \leftarrow \{0,1\}^{\lambda}$

Query $(x \in \{0,1\}^{in})$:

return $F'(k, x)$

$\mathcal{L}^{F'}_{prf\text{-}rand}$

Query $(x \in \{0,1\}^{in})$:

$z \leftarrow \{0,1\}^{out}$

return z

A

pick $x_1, x_2 \in \{0,1\}^{in}$ arbitrarily so that $x_1 \neq x_2$

z_1 = Query (x_1)

z_2 = Query (x_2)

z_3 = Query $(x_1 \oplus x_2)$

return $z_3 \stackrel{?}{=} z_1 \oplus z_2$

When A is linked to $\mathcal{L}^{F'}_{prf\text{-}real}$, the library will choose a key k. Then z_1 is set to $F'(k, x_1)$ and z_2 is set to $F'(k, x_2)$. Since $z_1 \oplus z_2 = x_1 \oplus x_2 \,||\, x_1 \oplus x_2 = F'(k, x_1 \oplus x_2) = z_3$, the output of A is always 1 and $Pr[A \circ \mathcal{L}^{F'}_{prf\text{-}real} \Rightarrow 1] = 1$

When A is linked to $\mathcal{L}^{F'}_{prf\text{-}rand}$, the responses of the two calls to Query will be chosen uniformly and independently because different arguments to Query were used. Consider the moment when the ~~second~~ third call to Query is about to happen. z_2, x_1, x_2, and z_1 have all been uniformly determined and A will output 1 only if z_3 is exactly $x_1 \oplus x_2 \,||\, x_1 \oplus x_2$. This happens with probability $1/2^{\lambda}$, therefore $Pr[A \circ \mathcal{L}^{F'}_{prf\text{-}rand} \Rightarrow 1] = 1/2^{\lambda}$. The advantage of A is therefore $[1 - 1/2^{\lambda}]$ which is non negligible.

3.

Sender (A)

Input: $x_1, \ldots x_n \in \{0,1\}^L$

(n cards represented by binary strings of length L)

Choose $k_0 = 0^L$

for $(j = 1 \ldots n)$

choose $k_j \leftarrow \{0,1\}^L$

select PRF F ——F——→

1 out of 2 OT with inputs x_0, x_1

$x_0 = k_0 \oplus \ldots \oplus k_{j-1} \oplus x_j$

$x_1 = k_j$

$r_0, r_1 \leftarrow_R \{0,1\}^n$,

$w_i = F^{-1}(x_i)$

←—— x_i, x_0 or x_1 ——

——$r_0, r_1 \oplus w_i \oplus x_i$ for $i = 0, 1$.——→

Reciever (B)

Input: $i \in \{1 \ldots n\}$

select x_1 or x_0 at random in $\{0,1\}^n$ (inputs) and $x_i = F(w)$, $w \leftarrow \{0,1\}^n$

↑

choose x_0 or x_1 depending on inputs of sender

Reciever learns k_j for all $j \neq i$ and $k_0 \oplus k_1 \ldots \oplus k_{i-1} \oplus x_i$ and thus can recover i.

4. Yao's Garbled Circuit Evaluations w/ multiple optimizations

XOR Gate

	# of ciphertexts	generator # of hash function evals	evaluator # of hash function evals
traditional	4	4	4
point & permute	4	4	1
GRR3	0	0	0
GRR2	2	4	1
free-XOR	0	0	0
half gates coupled with free-XOR	0	0	0

AND Gate

	# of ciphertexts	generator # of hash function evals	evaluator # of hash function evals
traditional	4	4	4
point & permute	4	4	1
GRR3	3	4	1
GRR2	2	4	1
free-XOR	4	4	1
half gates coupled with free-XOR	2	4	2

5.

Setup

Two parties Alice & Bob (A & B)

A runs g to generate G_p using the multiplication modulo p.

$R \leftarrow \{0,1\}$

Commit (b) (b = chosen bit)

$c = g^{b \cdot R} \bmod p$

$s \leftarrow \{0,1\}$

return $(c, (s, b))$

Open $(c, (s,b))$

If $c = g^{b \cdot R} \bmod p$, return $m = b$, and $m = \perp$ otherwise