Fundamentals of Cryptography: Problem Set 12

Due Wednesday Dec 25, 3PM

Problem 1 (5pt) 1-out-of-t Oblivious Transfer 1-out-of-t OT is a natural generalization of the standard oblivious transfer. In 1-out-of-t OT, the sender is given t equal-length messages $m_1, \dots, m_t \in \{0,1\}^\ell$, the receiver is given an index $x \in \{1, \dots, t\}$, the protocol let the receiver learn m_x, without revealing any other information.

Correctness After the protocol, the receiver always output m_x.

Receiver's Security against Semi-honest Sender The sender doesn't learn anything about the receiver's index. Let $\mathsf{View}_S((m_1, \dots, m_t), x)$ denote the view of the sender when the sender is given t messages $m_1, \dots, m_t$, the receiver is given an index $x \in \{1, \dots, t\}$. There exists an efficient simulator Sim_S such that

$$\mathsf{View}_S((m_1, \dots, m_t), x) \approx \mathsf{Sim}_S(m_1, \dots, m_t)\,.$$

The security is perfect, statistical, or computational, if the above two distributions are perfectly, statistically, or computationally indistinguishable.

Sender's Security against Semi-honest Receiver The receiver doesn't learn anything about the other messages of the sender. Let $\mathsf{View}_R((m_1, \dots, m_t), x)$ denote the view of the receiver when the sender is given t messages $m_1, \dots, m_t$, the receiver is given an index $x \in \{1, \dots, t\}$. There exists an efficient simulator Sim_R such that

$$\mathsf{View}_R((m_1, \dots, m_t), x) \approx \mathsf{Sim}_R(m_x, x)\,.$$

The security is perfect, statistical, or computational, if the above two distributions are perfectly, statistically, or computationally indistinguishable.

Show how to construct a (semi-honest) 1-out-of-t OT protocol based on a given 1-out-of-2 OT (i.e., the standard OT) protocol. You can use $\mathsf{OT.Sim}_S$ and $\mathsf{OT.Sim}_R$ to denote the simulators of the 1-out-of-2 OT protocol.

Problem 2 (5pt) Information-Theoretic Garbled Circuits In Yao's garbled circuits, the garbling algorithm works as follows. Given the circuit C, for each wire $i \in [n]$, samples two random labels $L_{i,0}, L_{i,1}$. Output $L_{1,0}, L_{1,1}, \dots, L_{n_{\text{in}},0}, L_{n_{\text{in}},1}$ as the input labels. Output the garbled circuit $\tilde{C}$, which consists of two parts

- For each $i \in \{n_{\text{in}} + 1, \dots, n\}$, say the gate function is g, and the gate takes wires j_1, j_2 as inputs, output a random shuffle of

$$\mathsf{Enc}(L_{j_1,0}, \mathsf{Enc}(L_{j_2,0}, L_{i,g(0,0)})), \mathsf{Enc}(L_{j_1,0}, \mathsf{Enc}(L_{j_2,1}, L_{i,g(0,1)})),$$
$$\mathsf{Enc}(L_{j_1,1}, \mathsf{Enc}(L_{j_2,0}, L_{i,g(1,0)})), \mathsf{Enc}(L_{j_1,1}, \mathsf{Enc}(L_{j_2,1}, L_{i,g(1,1)})).$$

- For each output wire $i \in \{n - n_{\text{out}} + 1, \dots, n\}$, output a random shuffle of

$$\mathsf{Enc}(L_{i,0}, 0), \mathsf{Enc}(L_{i,1}, 1).$$

  Moreover, we can assume w.l.o.g. that the i-th wire is not the input of any gate, then we can set $L_{i,0} = 0, L_{i,1} = 1$. And there is no need to output $\mathsf{Enc}(L_{i,0}, 0), \mathsf{Enc}(L_{i,1}, 1)$.

Here Enc denotes the encryption function of an authenticated encryption scheme.

Part A (0pt). In this part, we remove the dependency of authenticated encryption.

Let $L_{i,b}[0]$ denote the first bit of $L_{i,b}$ and let $L_{i,b}[1{:}]$ denote the rest of $L_{i,b}$. The garbling algorithm additionally samples a random mask bit $\alpha_i \in \{0,1\}$ for each $i \in \{1, \dots, n - n_{\text{out}}\}$, and sets $L_{i,b}[0] = b \oplus \alpha_i$. ($L_{i,b}[1{:}]$ are still randomly sampled.) That is

$$L_{i,0} = (\alpha_i, L_{i,0}[1{:}]), \qquad L_{i,1} = (1 \oplus \alpha_i, L_{i,1}[1{:}]).$$

For each $i \in \{n_{\text{in}} + 1, \dots, n\}$, the garbling algorithm outputs

$$\begin{aligned}
c^i_{0,0} &= \mathsf{Enc}(L_{j_1,\alpha_{j_1}}[1{:}], \quad \mathsf{Enc}(L_{j_2,\alpha_{j_2}}[1{:}], \quad L_{i,g(\alpha_{j_1},\alpha_{j_2})})),\\
c^i_{0,1} &= \mathsf{Enc}(L_{j_1,\alpha_{j_1}}[1{:}], \quad \mathsf{Enc}(L_{j_2,\alpha_{j_2}\oplus 1}[1{:}], L_{i,g(\alpha_{j_1},\alpha_{j_2}\oplus 1)})),\\
c^i_{1,0} &= \mathsf{Enc}(L_{j_1,\alpha_{j_1}\oplus 1}[1{:}], \mathsf{Enc}(L_{j_2,\alpha_{j_2}}[1{:}], \quad L_{i,g(\alpha_{j_1}\oplus 1,\alpha_{j_2})})),\\
c^i_{1,1} &= \mathsf{Enc}(L_{j_1,\alpha_{j_1}\oplus 1}[1{:}], \mathsf{Enc}(L_{j_2,\alpha_{j_2}\oplus 1}[1{:}], L_{i,g(\alpha_{j_1}\oplus 1,\alpha_{j_2}\oplus 1)})).
\end{aligned}$$

Here Enc is the encryption function of a CPA-secure encryption scheme (or even an encryption scheme that has indistinguishable multiple encryptions in the presence of an eavesdropper). Note that, the four entries are shuffled by $(\alpha_{j_1}, \alpha_{j_2})$.

Show how to evaluate this modified garbled circuit.

Part B. Show how to further modify the garbling scheme, so that it only uses *perfect* secure encryption schemes (e.g., one-time pad).

As we have discussed, prefect security requires longer key-length. How long the input label is? Which class of circuits can it efficiently garble?

Problem 3 (8pt) Garbled Circuit Optimizations In this problem, we consider a few more optimizations of Yao's garbled circuits.

- *Color bit*: For each wire, a permute bit α_i is sampled by the garbler, the color bit $\tilde{x}_i = x_i \otimes \alpha_i$ is revealed to the evaluator.

- *Row reduction*: For every gate, one of the four ciphertexts in its table is set to be zero. In other word, the label of the output wire is not randomly sampled.

- *FreeXOR*: A global random $\Delta \in \{0,1\}^\lambda$ is sampled by the garbler. For each wire, its labels L^i_0, L^i_1 satisfies $L^i_1 = L^i_0 \oplus \Delta$. Therefore, the table of every XOR gate is empty.

After applying these optimizations, the garbling algorithm becomes

- Sample random $\Delta \in \{0,1\}^\lambda$ such that $\text{lsb}(\Delta) = 1$.

- For each input wire i, sample $L^i_0 \in \{0,1\}^\lambda$.

- For any wire, we always define $\alpha_i = \text{lsb}(L_0^i)$, $L_1^i = L_1^i \oplus \Delta$.
- For each gate g, let i, j, k denote its input/output wires' indexes.
  - If g is XOR, let $L_0^k = L_0^i \oplus L_0^j$. g has an empty table.
  - If g is not XOR, set L_0^k such that $L^k_{g(\alpha_i,\alpha_j)+\alpha_k} = H(L^i_{\alpha_i}, L^j_{\alpha_j})$. The table consists of three ciphertexts

$$
\begin{aligned}
c^k_{0,1} &= H(L^i_{\alpha_i}, L^j_{\alpha_j \oplus 1}) &&\oplus L^k_{g(\alpha_i,\alpha_j\oplus 1)},\\
c^k_{1,0} &= H(L^i_{\alpha_i \oplus 1}, L^j_{\alpha_j}) &&\oplus L^k_{g(\alpha_i\oplus 1,\alpha_j)},\\
c^k_{1,1} &= H(L^i_{\alpha_i \oplus 1}, L^j_{\alpha_j \oplus 1}) &&\oplus L^k_{g(\alpha_i\oplus 1,\alpha_j\oplus 1)}.
\end{aligned}
$$

- The garbled circuit $\tilde{C}$ consists of tables of all gates and permute bits of all output wires.

Part A. How the evaluation algorithm works?

Part B. Prove the security, while H is modeled as a random oracle. You probably need to program H in some intermediate hybrid world, but in the ideal world, programing is unnecessary.