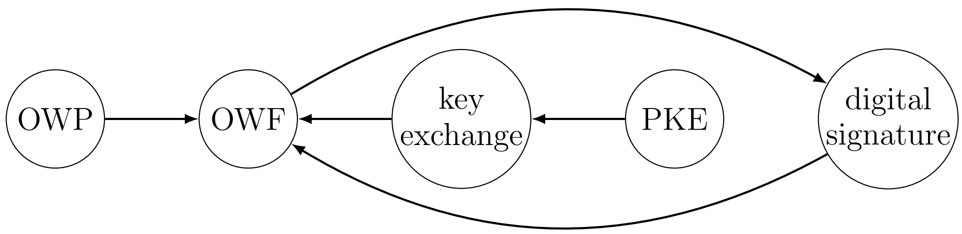
# Fundamentals of Cryptography: Final

Wednesday Jan 8, 2-4PM

**Problem 1** (b)

**Problem 2**




**Problem 5A** We should that $H$ is not a PRF. Then neither is $F$.

Choose $\ell = 2\lambda$ distinct $x_1, \dots, x_\ell \in \{0,1\}^{n/2}$. Let $\mathbf{u}_i$ denote the first row of

$$M_{1,x_{i,1}} M_{2,x_{i,2}} \cdots M_{n/2,x_{i,n/2}}.$$

Let $\mathbf{v}_j$ denote the first column of

$$M_{n/2+1,x_{j,1}} M_{n/2+2,x_{j,2}} \cdots M_{n/2+n/2,x_{j,n/2}}.$$

Then

$$\begin{aligned} H(x_i \| x_j) &= \text{first entry of } M_{1,x_{i,1}} \cdots M_{n/2,x_{i,n/2}} \cdot M_{n/2+1,x_{j,1}} \cdots M_{n/2+n/2,x_{j,n/2}} \\ &= \mathbf{u}_i^\intercal \mathbf{v}_j \end{aligned}$$

Consider a $\ell \times \ell$ matrix $M$ such that $M_{i,j} = H(x_i \| x_j)$.

$$M = \begin{bmatrix} H(x_1\|x_1) & \cdots & H(x_1\|x_\ell) \\ \vdots & \ddots & \vdots \\ H(x_\ell\|x_1) & \cdots & H(x_\ell\|x_\ell) \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_\ell \\ | & & | \end{bmatrix}^\intercal \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_\ell \\ | & & | \end{bmatrix}$$

So the rank of $M$ is no larger than $\lambda$.

But the rank of a random $\ell \times \ell$ matrix is close to $\ell$ with high probability. This allows an efficient distinguisher to distinguish between $H$ and a random function.

**Problem 5B** OT implies PKE.

Let $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be a two-message OT protocol. It is enough to construct a PKE scheme for encrypting one-bit messages. The PKE scheme can be defined as follows:

- $\mathsf{Gen}$ runs $\mathsf{OT}_1(0) \to (\mathsf{msg}_1, \pi)$. Let the first message $\mathsf{msg}_1$ be the public key, let the status $\pi$ be the secret key.

- $\mathsf{Enc}(pk, x)$ runs $\mathsf{OT}_2(\mathsf{msg}_1, (x, 0)) \to \mathsf{msg}_2$. Let $\mathsf{msg}_2$ be the ciphertext.
- $\mathsf{Dec}$ runs $\mathsf{OT}_3(\pi, \mathsf{msg}_2)$ to recover $x$.

The correctness is straight-forward.

For CPA-security, it is sufficient to show that (public key, encryption of 0) is indistinguishable from (public key, encryption of 1). Let $\mathsf{View}_E((m_0, m_1), b)$ denote the view of an external party during an execution of the OT protocol, when the sender has messages $m_0, m_1$ and the receiver has selection bit $b$.

$$
\begin{aligned}
&\text{(public key, encryption of 0)} \\
&\quad \equiv \mathsf{View}_E((0,0),0) \approx_c \mathsf{View}_E((0,0),1) \approx_c \mathsf{View}_E((1,0),1) \approx_c \mathsf{View}_E((1,0),0) \\
&\qquad \equiv \text{(public key, encryption of 1)}
\end{aligned}
$$

The first and last $\approx_c$ follow from the security against semi-honest sender. The middle $\approx_c$ follows from the security against semi-honest receiver.

**Problem 6** Share the $k$ secrets separately. More concretely, for each $\alpha \in [k]$, we will construct a secret sharing scheme such that

For any subset $T = \{i_1, \dots, i_k\}$, where $1 \le i_1 < i_2 < \cdots < i_k \le n$:

**(Correctness)** If $i^* = i_\alpha \in T$, the secret can be recovered from $(s_{i_1}, \dots, s_{i_k})$.

**(Privacy)** Otherwise, nothing about the secret can be recovered from $(s_{i_1}, \dots, s_{i_k})$.

This condition is implied by the following condition:

For any subset $T \subseteq [n]$:

**(Correctness)** If $(i^* \in T) \wedge (T \cap \{1, \dots, i^*-1\} \ge \alpha - 1) \wedge (T \cap \{i^*+1, \dots, n\} \ge k - \alpha)$, the secret can be recovered from $(s_{i_1}, \dots, s_{i_k})$.

**(Privacy)** Otherwise, nothing about the secret can be recovered from $(s_{i_1}, \dots, s_{i_k})$.

Inspired by the observation, the PoSS distribution algorithm can be constructed as follows

- For each $\alpha \in [k]$
  - Additively share $m_i$ among $m_{\alpha,L}, m_{\alpha,i*}, m_{\alpha,H}$.
  - Use an $(\alpha-1)$-out-of-$(i^*-1)$ threshold secret sharing to distribute $m_{\alpha,L}$ among shares $m_{\alpha,1}, \dots, m_{\alpha,i^*-1}$.
  - Use an $(k-\alpha)$-out-of-$(n-i^*)$ threshold secret sharing to distribute $m_{\alpha,H}$ among shares $m_{\alpha,i^*+1}, \dots, m_{\alpha,n}$.
- The $i$-th share consists of $m_{1,i}, \dots, m_{k,i}$.

**Problem 7** Construct such OT protocol recursively. Let $\Pi_1 = \Pi$. Assume $\Pi_n$ is a 2-message 1-out-of-$2^n$ OT protocol. Construct $\Pi_{n+1}$ as follows:

- Let $i$ be the selection number. The receiver parses $i = (i_0, i_{1:})$ into its most significant bit $i_0$ and the rest $i_{1:}$, runs $\Pi.\mathsf{OT}_1(i_0) \to (\mathsf{msg}_1, \pi)$, runs $\Pi_n.\mathsf{OT}_1(i_{1:}) \to (\mathsf{msg}'_1, \pi')$, sends $(\mathsf{msg}_1, \mathsf{msg}'_1)$ to the sender.
- Let $m_0, \ldots, m_{2^{n+1}-1}$ denote the sender's list of inputs. The sender runs
$$\begin{aligned}\Pi_n.\mathsf{OT}_2(\mathsf{msg}'_1, (m_0, \ldots, m_{2^n-1})) &\to \mathsf{msg}_{2,0}\\ \Pi_n.\mathsf{OT}_2(\mathsf{msg}'_1, (m_{2^n}, \ldots, m_{2^{n+1}-1})) &\to \mathsf{msg}_{2,1}\\ \Pi.\mathsf{OT}_2(\mathsf{msg}_1, (\mathsf{msg}_{2,0}, \mathsf{msg}_{2,1})) &\to \mathsf{msg}_2\end{aligned}$$
  sends $\mathsf{msg}_2$ to the receiver.
- Upon receiving $\mathsf{msg}_2$, the receiver computes
$$\begin{aligned}\Pi.\mathsf{OT}_2(\pi, \mathsf{msg}_2) &\to \mathsf{msg}_{2,i_0}\\ \Pi_n.\mathsf{OT}_2(\pi', \mathsf{msg}_{2,i_0}) &\to m_i\end{aligned}$$

For the communication complexity. Note that the second message of $\Pi$ must be at least $\ell$ bit, thus the first message of $\Pi$ is at most $\text{poly}(\lambda)$ bit.

$$\begin{aligned}&\text{communication complexity of } \Pi_n \text{ when inputs are } \ell\text{-bit long}\\ &\le \text{poly}(\lambda) + \text{communication complexity of } \Pi_{n-1} \text{ when inputs are } (\ell + \text{poly}(\lambda))\text{-bit long}\\ &\le n\,\text{poly}(\lambda) + \ell\end{aligned}$$

**Problem 8** Set $n = 5$, so 2 parties' views will be opened to the verifier. We requires that the MPC protocol is (perfectly) correct, and has semi-honest static security against $\lfloor \frac{n-1}{2} \rfloor$ corruptions. For example, BGW satisfies all the requirements, and does not rely on any assumption.

**Completeness** Obvious.

**Soundness.** Since the protocol $\Pi$ is (perfectly) correct, the prover cannot fool the verifier if $V_1, \ldots, V_5$ are the views in an honest execution.

To fool the verifier, the views $V_1, \ldots, V_5$ must not be consistent: (a) either $i$-th party is not following the protocol in the view $V_i$, for some $i$; (b) or $V_i, V_j$ do not agree with each other, for some $i, j$. In either case, the verifier will catch the prover with probability at least $1/\binom{5}{2}$. (Soundness error $1 - 1/\binom{5}{2}$.)

**Zero-knowledge.** The verifier opens the views of $\lfloor \frac{n-1}{2} \rfloor$ parties, and tries to learn information about the witness. This is essentially the same as $\lfloor \frac{n-1}{2} \rfloor$ semi-honest static corruptions. If $\Pi$ is perfectly/statistically/computationally secure against $\lfloor \frac{n-1}{2} \rfloor$ semi-honest static corruptions, then the open views can be perfectly/statistically/-computationally simulated without knowing the witness, then the ZKP protocol is perfectly/statistically/computationally zero-knowledge.

**Proof of knowledge.** In the OT hybrid model, the extractor gets $V_1, \ldots, V_5$. If the views are consistent, then $w = w_1 + \cdots + w_5$ is a witness.