Chap 11

Mingjia Huo

Problem 11.3.

(a).

Given *one-way* public key encryption scheme (Gen,Enc,Dec) and randm oracle H, here is the construction:

1. Gen: Run $\text{Gen}(1^n)$ to obtain keys (pk, sk).
2. Encaps: Uniformly choose $m \in \{0,1\}^n$, compute $c = \text{Enc}_{pk}(m)$. And $k = H(m)$.
3. Decaps: Given c, pk, compute $k = H(\text{Dec}_{pk}(c)) = \text{Decaps}_{pk}(c)$.

Then prove it's CPA-secure.

Consider the experiment $\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n)$. For an adversary $\mathcal{A}$, let Query be the event that $\mathcal{A}$ queries m to H. Then,

$$\begin{aligned}\Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1] &= \Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1 \wedge \text{Query}] + \Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1 \wedge \overline{\text{Query}}] \\ &\leq \Pr[\text{Query}] + \Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1 \mid \overline{\text{Query}}] \times \Pr[\overline{\text{Query}}] \\ &\leq \Pr[\text{Query}] + \Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1 \mid \overline{\text{Query}}]\end{aligned}$$

Since H is a random oracle, $H(m)$ is uniformly distributed from the perspective of the adversary, which is the same with a uniform string. So

$$\Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1 \mid \overline{\text{Query}}] = \frac{1}{2}.$$

As for $\Pr[\text{Query}]$, we can prove it is negligible to n:

Use $\mathcal{A}$ to construct $\mathcal{A}'$ in the experiment of *one-way* public key encryption scheme:

- When $\mathcal{A}'$ is given (pk, c), choose a uniform $k \in \{0,1\}^n$.
- Give (pk, c, k) to $\mathcal{A}$ and then run $\mathcal{A}$. Record all $\mathcal{A}$'s queries to H, denote as set Q.
- Assume $|Q| = t$, uniform choose an item $m' \in Q$, and output m'.

If $\mathcal{A}$ queries on m, then $\mathcal{A}'$ has probability $\frac{1}{t}$ to output m. So

$$\Pr[m' = m] \geq \frac{1}{t}\Pr[\text{Query}].$$

Since $\Pr[m' = m]$ is negligible to n, $\Pr[\text{Query}]$ is also negligible to n.

Thus $\Pr[\text{KEM}^{cpa}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \text{negl}(n)$. And this scheme is CPA-secure.

(b).

Yes. Construct a scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ based on RSA.

- $\text{Gen}(1^n)$: Run $\text{GenRSA}(1^n)$ to obtain N, e, d. Here $N > 2^n$.
- Enc: $c = m^e, m \in \{0,1\}^n$.
- Dec: $m = c^d$.

Assume RSA problem is hard relative to GenRSA. Then we prove the scheme is one-way: Given $\mathcal{A}'$ for Π, construct $\mathcal{A}$ for $\text{RSA-inv}_{\mathcal{A},\text{GenRSA}}(n)$.

- $\mathcal{A}'$ is given (N, e, y).

- Run $\mathcal{A}(N, e, y)$, and get m.
- Output m.

Here,

$$\begin{aligned}\text{negl}(n) \geq &\Pr[\text{RSA-inv}_{\mathcal{A},\text{GenRSA}}(n) = 1] \\ =&\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1 \mid m \in \mathbb{Z}_N^*] \\ =&(\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1] - \Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1 \wedge m \notin \mathbb{Z}_N^*]) \times (\Pr[m \in \mathbb{Z}_N^*])^{-1} \\ \geq&(\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1] - \Pr[m \notin \mathbb{Z}_N^*]) \times (\Pr[m \in \mathbb{Z}_N^*])^{-1}.\end{aligned}$$

Since $\phi(N) = (p-1)(q-1) > \frac{1}{2}N > \frac{1}{2} \cdot 2^n$, we have $(\Pr[m \in \mathbb{Z}_N^*])^{-1} \geq \frac{1}{2}$. And we have $\Pr[m \notin \mathbb{Z}_N^*] \leq \text{negl}(n)$.

Thus

$$\begin{aligned}\text{negl}(n) \geq &(\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1] - \Pr[m \notin \mathbb{Z}_N^*]) \times (\Pr[m \in \mathbb{Z}_N^*])^{-1} \\ \geq &\frac{1}{2}(\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1] - \text{negl}(n)).\end{aligned}$$

That is

$$\Pr[\text{One-Way}_{\mathcal{A}',\Pi}(n) = 1] \leq 3\text{negl}(n).$$

So the experiment of one-way successes with probability negligible to n.

Problem 11.7. It's not CPA-secure. Give an attack of adversary $\mathcal{A}$:

- When $\mathcal{A}$ is given $(\mathbb{G}, g, q, h)$, uniformly choose $r \in \mathbb{Z}_p$, output $(m_0, m_1) = (0, r)$.
- When given ciphertext c, if $c^q \equiv 1 \mod p$, then output 0; otherwise, output 1.

Here

$$\Pr[\text{Pubk}^{cpa}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}\Pr[\mathcal{A}(\mathbb{G}, g, q, h, c) = 0 \mid b = 0] + \frac{1}{2}\Pr[\mathcal{A}(\mathbb{G}, g, q, h, c) = 1 \mid b = 1].$$

If $b = 0$, then $c = h^r + m_0$, which is a quadratic residue. And $c^q \equiv 1 \mod p$, so the output of $\mathcal{A}$ is 0. Thus,

$$\Pr[\mathcal{A}(\mathbb{G}, g, q, h, c) = 0 \mid b = 0] = 1.$$

If $b = 1$, then $c = h^r + m_1$, with probability $\frac{q+1}{p}$ to be a quadratic residue. So the probability that $c^q \not\equiv 1 \mod p$ is $\frac{q}{p} > \frac{1}{4}$.

Thus,

$$\Pr[\text{Pubk}^{cpa}_{\mathcal{A},\Pi}(n) = 1] = \frac{1}{2}\Pr[\mathcal{A}(\mathbb{G}, g, q, h, c) = 0 \mid b = 0] + \frac{1}{2}\Pr[\mathcal{A}(\mathbb{G}, g, q, h, c) = 1 \mid b = 1] > \frac{5}{8}.$$

So it's not CPA-secure.

Problem 11.8. (a).

Denote the bit of two parties as a, b, and assume $\Pr[a = 0] = p$. Let the result be value $r \in \{0, 1\}$.
Then $\Pr[r = 0] = \Pr[a = 0 \wedge b = 0] + \Pr[a = 1 \wedge b = 1] = p \cdot \frac{1}{2} + (1-p) \cdot \frac{1}{2} = \frac{1}{2}$. Also $\Pr[r = 0] = \frac{1}{2}$.
So the result is uniform.

(b).
Assume A outputs (c_1, c_2).
If b wants the result to be 0, just output $(c_1 g, c_2 h)$. So the value it decrypts is the same as b_A.
If b wants the result to be 1, just output $(\frac{1}{c_1}, \frac{g}{c_2})$. So the value it decrypts is $\frac{g}{g^{b_A}} = 1 \oplus b_A$.

(c).
We should use a CCA-secure scheme.

Definition 1 (secure coin flip protocol)**.** *A secure coin flip protocol with public key encryption scheme* $\Pi = (Gen, Enc, Dec)$ *should satisfy:* $\forall$ *adversary* $\mathcal{A}$*,*

$$\Pr[Pubk_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n).$$

Here we define the experiment $\mathrm{Pubk}_{\mathcal{A},\Pi}(n)$ with the decryption oracle:

1. Run $\mathrm{Gen}(1^n)$ to obtain (pk, sk). Randomly choose $b \in \{0,1\}$ and compute $\mathrm{Enc}_{pk}(b) = c$. Then uniformly choose the expected result bit $r \in \{0,1\}$.
2. Give (pk, c, r) and the oracle $\mathcal{O}(\cdot) = \mathrm{Dec}_{sk}(\cdot)$ to $\mathcal{A}$. But $\mathcal{A}$ can't ask the oracle to decrypt c directly.
3. $\mathcal{A}$ outputs c'.
4. If $b \oplus \mathrm{Dec}_{sk}(c') = r$, the experiment outputs 1; otherwise, outputs 0.

We prove if Π is a CCA-secure scheme, then the coin flip protocol is secure.

Proof. Construct an adversary $\mathcal{A}'$ for CCA-secure scheme based on $\mathcal{A}$ for coin flip protocol.

- $\mathcal{A}'$ is given pk and decryption oracle $\mathcal{O}(\cdot)$.
- Output $(m_0, m_1) = (0, 1)$ and get $c = \mathrm{Enc}_{pk}(m_b)$.
- Run $\mathcal{A}(pk, c, 0)$, and get c' from $\mathcal{A}$.
- Output the bit $b' = \mathcal{O}(c')$.

In this construction, $\Pr[b = b'] = \Pr[\mathrm{Dec}_{sk}(c) = \mathrm{Dec}_{sk}(c')] = \Pr[\mathcal{A}(pk, c, 0) = 1]$. Since Π is CCA-secure, so $\Pr[b = b'] \leq \frac{1}{2} + \mathrm{negl}(n)$. So

$$\Pr[Pubk_{\mathcal{A},\Pi}(n) = 1] = \Pr[\mathcal{A}(pk, c, 0) = 1] \leq \frac{1}{2} + \mathrm{negl}(n).$$

□

Problem 11.13.

Definition 2 (secure under t-multiple receivers)**.** *A public key encryption scheme $\Pi = (Gen, Enc, Dec)$ is secure under t-multiple receivers if $\forall \mathcal{A}$,*

$$\Pr[Pubk^{t-multi}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n).$$

Here, t should be $poly(n)$.

Here we define the experiment $\mathrm{Pubk}^{t-\mathrm{multi}}_{\mathcal{A},\Pi}(n)$:

- Run $\mathrm{Gen}(1^n)$ for t times, and get $(pk_i, sk_i), 1 \leq i \leq t$. Choose uniform $b \in \{0,1\}$.
- Give $pk_i, i \in [t]$ to $\mathcal{A}$. Then $\mathcal{A}$ outputs (m_0, m_1).
- Give $c = (\mathrm{Enc}_{pk_1}(m_b), \cdots, \mathrm{Enc}_{pk_t}(m_b))$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs b'. If $b = b'$, output 1; otherwise output 0.

We prove if Π is a CPA-secure scheme, then it's secure under t-multiple receivers.

Proof. Construct an adversary $\mathcal{A}'$ for CPA-secure scheme experiment based on $\mathcal{A}$ for security under t-multiple receivers experiment.

- $\mathcal{A}'$ is given pk. Uniform choose $r \in [t]$, and denote pk as pk_r.
- Run $\mathrm{Gen}(1^n)$ for $t-1$ times and get (pk_i, sk_i), with $i \in [t], i \neq u$.
- Run $\mathcal{A}(pk_1, \cdots, pk_t)$ and then $\mathcal{A}$ outputs (m_0, m_1).
- $\mathcal{A}'$ outputs (m_0, m_1) and get c.
- Then $\mathcal{A}'$ computes $c_i = \mathrm{Enc}_{pk_i}(m_0), 1 \leq i < r$ and $c_i = \mathrm{Enc}_{pk_i}(m_1), r < i \leq t$. Then give $c_i, i \in [t]$ to $\mathcal{A}$.
- Output the same bit as what $\mathcal{A}$ outputs, denoted as b'.

In this construction, $\Pr[Pubk^{cpa}_{\mathcal{A}',\Pi}(n)=1] = \Pr[b=b'] = \Pr[\mathrm{Dec}_{sk}(c_r)=m_b] = \Pr[\mathcal{A}(c_1,\cdots,c_t)=b]$. Denote the choice of r as event R, we have

$$\begin{aligned}
&\Pr[\mathcal{A}(c_1,\cdots,c_t)=b]\\
=&\sum_{r=1}^{t}\Pr[R=r]\times\Pr[\mathcal{A}(\mathrm{Enc}_{pk_1}(m_0),\cdots,\mathrm{Enc}_{pk_{r-1}}(m_0),\mathrm{Enc}_{pk_r}(m_b),\mathrm{Enc}_{pk_{r+1}}(m_1),\cdots,\mathrm{Enc}_{pk_t}(m_1))=b]\\
=&\frac{1}{t}\sum_{r=1}^{t}\frac{1}{2}\Pr[\mathcal{A}(\mathrm{Enc}_{pk_1}(m_0),\cdots,\mathrm{Enc}_{pk_{r-1}}(m_0),\mathrm{Enc}_{pk_r}(m_0),\mathrm{Enc}_{pk_{r+1}}(m_1),\cdots,\mathrm{Enc}_{pk_t}(m_1))=0]\\
&+\frac{1}{t}\sum_{r=1}^{t}\frac{1}{2}\Pr[\mathcal{A}(\mathrm{Enc}_{pk_1}(m_0),\cdots,\mathrm{Enc}_{pk_{r-1}}(m_0),\mathrm{Enc}_{pk_r}(m_1),\mathrm{Enc}_{pk_{r+1}}(m_1),\cdots,\mathrm{Enc}_{pk_t}(m_1))=1]\\
=&\frac{1}{2t}(t-1+\Pr[\mathcal{A}(\mathrm{Enc}_{pk_i}(m_1))=1]+\Pr[\mathcal{A}(\mathrm{Enc}_{pk_i}(m_0))=0])\\
=&\frac{1}{2t}(t-1+2\Pr[\mathrm{Pubk}^{t-\mathrm{multi}}_{\mathcal{A},\Pi}(n)=1]).
\end{aligned}$$

Since $\Pr[\mathcal{A}(c_1,\cdots,c_t)=b] = \Pr[Pubk^{cpa}_{\mathcal{A}',\Pi}(n)=1] \le \frac{1}{2}+\mathrm{negl}(n)$, we have

$$\frac{1}{2t}(t-1+2\Pr[\mathrm{Pubk}^{t-\mathrm{multi}}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\mathrm{negl}(n).$$

So

$$\Pr[\mathrm{Pubk}^{t-\mathrm{multi}}_{\mathcal{A},\Pi}(n)=1] \le \frac{1}{2}+\mathrm{negl}(n).$$

□

Problem 11.15. The algorithm is not CCA-secure, since it's deterministic.

The adversary $\mathcal{A}$ just gets the encryption of (m_0,m_1), denoted as (c_0,c_1). Then $\mathcal{A}$ outputs (m_0,m_1). Compare the answer to (c_0,c_1), he can know the value of b.

Problem 11.20.

Algorithm 1 Get_value(N,e,y): Return x such that $x^e \equiv y \mod N$

$LB \leftarrow 0$;
$UB \leftarrow N$; // Lower bound and upper bound of x;
for each $i \in [1,\log_2 N]$ **do**
  if $\mathcal{A}(y) == 0$ **then**
    $UB = (UB+LB)/2$;
  else
    $LB = (UB+LB)/2$;
  end if
  $y = (2^e)y$;
  if There is only one number x in $[LB,UB)$. **then**
    Break;
  end if
end for
return x;

The first i iterations decide a value t such that $\frac{t}{2^i} < x < \frac{t+1}{2^i}, 0 \le t \le 2^i-1$.

For iteration $i+1$:

- $0 < 2^i x \equiv x_i < \frac{N}{2} \mod N$: We have $2^{i+1}x \equiv 2x_i \mod N$. So $\mathrm{lsb}(2^{i+1}x \mod N) = \mathrm{lsb}(2x_i) = 0$.
- $\frac{N}{2} < 2^i x \equiv x_i < N \mod N$: We have $2^{i+1}x \equiv 2x_i - N \mod N$. So $\mathrm{lsb}(2^{i+1}x \mod N) = \mathrm{lsb}(2x_i - N) = 1$.

So

$$\frac{2t}{2^{i+1}} < x < \frac{2t+1}{2^{i+1}} \Leftrightarrow \mathrm{lsb}(2^{i+1}x \mod N) = 0,$$

$$\frac{2t+1}{2^{i+1}} < x < \frac{2t+2}{2^{i+1}} \Leftrightarrow \mathrm{lsb}(2^{i+1}x \mod N) = 1.$$

Thus the algorithm can return x, such that $x^e \equiv y \mod N$.

Problem 11.21. Given $\mathcal{A}$ for experiment $\text{RSA-half}_{\mathcal{A},\text{GenRSA}}(1^n)$, construct $\mathcal{A}'$ for $\text{RSA-lsb}_{\mathcal{A}',\text{GenRSA}}(1^n)$:

- $\mathcal{A}'$ computes $t = 2^e \mod N$.
- When $\mathcal{A}'$ is given (N, e, y), compute $y' = yt \mod N$. So $x' = 2x \mod N$. Give (N, e, y') to $\mathcal{A}$.
- Output the same as what $\mathcal{A}$ outputs.

If $\mathrm{half}(x) = 0$, then $y' = yt \equiv x^e \cdot 2^e \equiv (2x)^e \mod N$. Since $2x < N$, we have $\mathrm{lsb}(x') = 0$.
If $\mathrm{half}(x) = 1$, then $y' = yt \equiv x^e \cdot 2^e \equiv (x \cdot 2 - N)^e \mod N$, where $0 < 2x - N < N$. Then $\mathrm{lsb}(x') = \mathrm{lsb}(2x - N) = 1$.
So

$$\mathrm{half}(x) = 0 \Leftrightarrow \mathrm{lsb}(x') = 0, \quad \mathrm{half}(x) = 1 \Leftrightarrow \mathrm{lsb}(x') = 1.$$

Since x is uniform in $\mathbb{Z}_N^*$, so $(2x \mod N)$ is also uniform in $\mathbb{Z}_N^*$.
Thus we have

$$\Pr[\text{RSA-half}_{\mathcal{A},\text{GenRSA}}(1^n)] = \Pr[\text{RSA-lsb}_{\mathcal{A},\text{GenRSA}}(1^n)] \le \frac{1}{2} + \mathrm{negl}(n).$$

So $\mathrm{half}(x)$ is also a hard-core prediction for the RSA problem.