1 Review: Indistinguishable Security for Public-Key Encryption

**Definition 1 (Indistinguishable Security for PKE)** We say a public-key encryption scheme \((\text{Gen}, \text{Enc}, \text{Dec})\) satisfies IND-SEC (Indistinguishable security), if for all \((m_0, m_1)\), the following two distributions are computationally indistinguishable:

\[
\{(pk, sk) \leftarrow \text{Gen}(k) : pk \circ \text{Enc}(pk, m_0)\} \approx_c \{(pk, sk) \leftarrow \text{Gen}(k) : pk \circ \text{Enc}(pk, m_1)\}
\]

where \(k\) is the security parameter.

**Remark 1** Although RSA may look like a PKE satisfying definition 1, one needs to note that RSA is a deterministic encryption scheme. Since the adversary holds the public key, it can encrypt those two messages by itself. Therefore, RSA is not a IND-SEC PKE.

2 Trapdoor One-way Function

2.1 Definition of Trapdoor OWF

**Definition 2 (Trapdoor One-way Function)** We say a family of collections of functions \(\{F_n\}\), where \(F_n = \{f_i : D_i \rightarrow R_i\}_{i=1}^{I_n}\), is a trapdoor one-way functions if:

- **Function Sampler**: there exists a PPT generator \(G\) which takes the security parameter \(n\) as input and outputs \((i, t)\) where \(i \in [I_n]\) and \(t\) is a trapdoor associated with \(f_i \in F_n\).

- **There exists a PPT algorithm** \(\text{Com}\) such that for all security parameter \(n\) and for all \(i, x \in D_i\), \(\text{Com}(n, i, x) = f_i(x)\).

- **Input Sampler**: there exists a PPT sampler \(S\) such that for all \(n, i\), \(S(n, i)\) will return a uniformly random element in \(D_i\). We will write \(x \xleftarrow{} D_i\) to represents that \(x\) is chosen uniformly from \(D_i\).

- **For all PPT adversary** \(A\),

\[
\Pr[(i, t) \leftarrow G(n), x \xleftarrow{} D_i, y = f_i(x) : A(i, y) = x] \leq \text{negl}(n)
\]

- **Invertible with trapdoor**: there exists a PPT algorithm \(B\), which is given \((i, t, y)\) where \((i, t) \leftarrow G(n)\), such that \(B(i, t, y) = x\) if \(y = f_i(x)\) and \(B(i, t, y) = \bot\) if \(y \notin R_i\).

**Remark 2** Note that, for each security parameter \(n\), \(F_n\) is a collection of functions while for OWF/OWP, each security parameter just corresponds to one function. If \(|F_n| = 1\), since \(f \in F_n\) can be efficiently inverted when given the trapdoor, an adversary can simply hardcore the trapdoor in itself.

The last requirement implies that for each \(f_i \in F_n\), \(f_i\) is a one-to-one mapping.
2.2 Construction of PKE using Trapdoor OWF

In this part, we will give a construction of PKE based on a trapdoor one-way permutation. Suppose \( \{F_n\} \) is a trapdoor one-way permutation. We use \( G \) to denote the function sampler of \( \{F_n\} \) and \( B \) for the PPT algorithm which takes the trapdoor \( t \) as input and inverts \( \{F_n\} \). We define (Gen, Enc, Dec) as following:

- Gen: it takes the security parameter \( n \) as input. Gen first calls \( G(n) = (i, t) \). Then, it sets \( sk = t \) and \( pk = (i, f_i, h_i) \) where \( f_i \in F_n \) and \( h_i \) is a hardcore predicate for \( f_i \). (Recall that each OWP has a hardcore predicate.) Finally Gen\( (n) \) outputs \( (pk, sk) \).
- Enc: it takes a one-bit message \( m \) and a public key \( pk = (i, f_i, h_i) \) as input. Enc first randomly samples \( x \sim D_i \). Then output \( c = (c_1, c_2) = (f_i(r), m \oplus h_i(r)) \).
- Dec: it takes a cipher-text \( c = (c_1, c_2) \) and a secret key \( sk = t \) as input. Dec first uses \( sk = t \) to invert \( c_1 \) by using \( B \). Suppose the output is \( r \). Then compute \( h_i(r) \) and output \( c_2 \oplus h_i(r) \).

Now we show the above construction is a PKE satisfying IND-SEC.

**Proof.**

For correctness, it follows from the properties of the trapdoor one-way permutation \( \{F_n\} \).

Now consider the security property. Note that \( pk = (i, f_i, h_i) \). We only need to show that, for all \((m_0, m_1)\),

\[
\{(i, f_i, h_i), t \sim Gen(n), r \sim D_i \mid (i, f_i, h_i) \circ (f_i(r), m_0 \oplus h_i(r))\}
\]

\[\approx_c \{(i, f_i, h_i), t \sim Gen(n), r \sim D_i \mid (i, f_i, h_i) \circ (f_i(r), m_1 \oplus h_i(r))\}\]

Consider the following 4 hybrids:

\[
H_0 := \{(i, f_i, h_i), t \sim Gen(n), r \sim D_i : (i, f_i, h_i) \circ (f_i(r), m_0 \oplus h_i(r))\}
\]

\[
H_1 := \{(i, f_i, h_i), t \sim Gen(n), r \sim D_i, b \sim \{0, 1\} : (i, f_i, h_i) \circ (f_i(r), m_0 \oplus b)\}
\]

\[
H_2 := \{(i, f_i, h_i), t \sim Gen(n), r \sim D_i, b \sim \{0, 1\} : (i, f_i, h_i) \circ (f_i(r), m_1 \oplus b)\}
\]

\[
H_3 := \{(i, f_i, h_i), t \sim Gen(n), r \sim D_i : (i, f_i, h_i) \circ (f_i(r), m_1 \oplus h_i(r))\}
\]

By the property of the hardcore predicate, any PPT adversary is not able to distinguish between \( h_i(r) \) and a uniformly random bit \( b \). Thus \( H_0 \approx_c H_3 \). Similarly, \( H_2 \approx_c H_3 \). Since \( b \) is uniformly random, \( m_0 \oplus b \) is also uniformly random (and independent with \( i, f_i, h_i, f_i(r) \)). Similarly, \( m_1 \oplus b \) is uniformly random. Thus \( H_1 \) and \( H_2 \) are identical. Therefore, \( H_0 \approx_c H_3 \). It is exactly what we need.

2.3 RSA implies Trapdoor OWP

In this part, we show that RSA assumption implies a trapdoor one-way permutation. To this end, we will show the correspondences between RSA assumption and a trapdoor one-way permutation.

We construct a trapdoor one-way permutation as following:

- Function Sampler: \( G \) first generates two different primes \( p, q \) and compute \( N = pq \). Then, randomly sample \( e \in \mathbb{Z}_{\phi(N)}^* \) and compute \( d \) such that \( ed = 1 \mod N \). Finally, \( G \) outputs \((i, t) = ((N, e), d)\).
- For each \( i = (N, e), f_i(x) = x^e \mod N \). It is easy to see that \( f_i(x) \) can be efficiently computed.

14-2
• Input Sampler: note that $D_i = \mathbb{Z}_N^*$. Thus there exists a PPT algorithm to sample a random element from $D_i$.
• By RSA assumption, for all PPT adversary $A$,
  \[
  \Pr[((N, e), d) \leftarrow G(n), x \leftarrow D_i, y = x^e \mod N : A(N, e, y) = x] \leq \text{negl}(n)
  \]
• Invertible with trapdoor: we construct $B$ as following: $B$ takes $((N, e), d, y)$ as input and outputs $y^d = x^{de} = x \mod N$.

Note that the input space and the output space are both $\mathbb{Z}_N^*$. Thus, it gives us a construction of a trapdoor one-way permutation.

3 Construction of PKE using LWE Assumption

3.1 Review: Decisional Learning with Error Assumption

The decisional learning with error (DLWE) assumption states that the following two distributions are computationally indistinguishable:

\[
\{s \leftarrow (\mathbb{Z}_q)^{n \times 1}, A \leftarrow (\mathbb{Z}_q)^{m \times n}, e \sim \text{Error}^{m \times 1} : (A, As + e)\} 
\approx_c \{A \leftarrow (\mathbb{Z}_q)^{m \times n}, u \leftarrow (\mathbb{Z}_q)^{n \times 1} : (A, u)\}
\]

Here $\text{Error}$ is the error distribution which is roughly a Gaussian Distribution. We write $e \sim \text{Error}^{m \times 1}$ to represents that $e$ is sampled following the distribution $\text{Error}^{m \times 1}$.

3.2 PKE construction based on DLWE Assumption

In this part, we will give a construction of PKE based on DLWE Assumption. We define $(\text{Gen}, \text{Enc}, \text{Dec})$ as following:

• $\text{Gen}$: it takes the security parameter $n$ as input. $\text{Gen}$ randomly samples $s \leftarrow (\mathbb{Z}_q)^{n \times 1}$, $A \leftarrow (\mathbb{Z}_q)^{m \times n}$, $e \sim \text{Error}^{m \times 1}$. Then, compute $b = As + e$. Let $pk = (A, b)$ and $sk = s$. Finally, $\text{Gen}$ outputs $(pk, sk) = ((A, b), s)$.

• $\text{Enc}$: it takes a one-bit message $m$ and a public key $pk = (A, b)$ as input. $\text{Enc}$ first randomly samples $x \leftarrow \{0, 1\}^{m \times 1}$. Then output $c = (c_1, c_2) = (x^T A, x^T b + mq/2)$.

• $\text{Dec}$: it takes a cipher-text $c = (c_1, c_2)$ and a secret key $sk = s$ as input. $\text{Dec}$ first computes $c_2 - c_1 s$. If the result is close to 0, then output 0. Otherwise, output 1

Now we give a proof sketch that above construction is a PKE with IND-SEC.

Proof.
For correctness, since the error vector $e$ is close to 0 with all but a negligible probability. Therefore, the scalar $x^T e$ is also close to 0 (compared with $q/2$). Thus, for $c = (c_1, c_2) = (x^T A, x^T b + mq/2)$,

\[
\begin{align*}
c_2 - c_1 s &= x^T b + mq/2 - x^T As \\
&= x^T (As + e) + mq/2 - x^T As \\
&= x^T e + mq/2
\end{align*}
\]

If $m = 0$, then $c_2 - c_1 s$ is close to 0. Otherwise, it is close to 1. Thus, $\text{Dec}$ successfully decrypts the message with all but a negligible probability.

For security, consider the following hybrids. For $(m_0, m_1)$,

\[
\begin{align*}
H_0 &= \{(A, b), s\} \gets \text{Gen}(n), x \gets \{0, 1\}^{m \times 1} : (A, b) \circ (x^T A, x^T b + m_0 q/2) \\
H_1 &= \{(A, b), s\} \gets \text{Gen}(n), x \gets \{0, 1\}^{m \times 1}, u \gets \mathbb{Z}_q^{m \times 1} : (A, u) \circ (x^T A, x^T u + m_0 q/2) \\
H_2 &= \{(A, b), s\} \gets \text{Gen}(n), x \gets \{0, 1\}^{m \times 1}, u \gets \mathbb{Z}_q^{m \times 1} : (A, u) \circ (x^T A, x^T u + m_1 q/2) \\
H_3 &= \{(A, b), s\} \gets \text{Gen}(n), x \gets \{0, 1\}^{m \times 1} : (A, b) \circ (x^T A, x^T b + m_1 q/2)
\end{align*}
\]

We first show that $H_0 \approx_c H_1$. Suppose there is some PPT adversary $A$ which can distinguish $H_0$ and $H_1$ with some non-negligible advantage. We will then construct an adversary $B$ to break the DLWE assumption. Recall that, in the DLWE experiment, $B$ will takes a pair $(A, w)$ as input. $B$ works as following:

1. $B$ uses $(A, w)$ as the public key $pk$ and then encrypts $m_0$. Let $c = \text{Enc}(pk, m_0)$.
2. $B$ calls the adversary $A$ with input $(pk, c)$. Then output the result of $A$.

Note that, if $w$ is $b$, then the distribution of the input of $A$ is the same as $H_0$. If $w$ is $u$, then the distribution of the input of $A$ is the same as $H_1$. Therefore, $B$ has the same advantage to win the DLWE experiment as $A$ does to distinguish $H_0$ and $H_1$. It contradicts with the DLWE assumption.

Thus, $H_0 \approx_c H_1$. Similarly, we have $H_2 \approx_c H_3$.

As for $H_1$ and $H_2$, the proof idea is to show the distribution of $c_2 = x^T u + m_0 q/2$ is statistically indistinguishable with a uniform bit even given $u$ and $x^T A$. The proof relies on the Leftover Hash Lemma. By symmetry, the distribution of $x^T u + m_1 q/2$ is also statistically indistinguishable with a uniform bit when given $u$ and $x^T A$. Thus $H_1$ and $H_2$ are statistically indistinguishable.

Remark 3 We correct the mistake in class where the encryption function chooses $x \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times 1}$. In this case, $x^T e$ is uniformly random in $\mathbb{Z}_q$. It thus does not satisfy our requirement that $x^T e$ is close to 0 with all but a negligible probability. The correct version is choosing $x \overset{\$}{\leftarrow} \{0, 1\}^{m \times 1}$.