Lecture 22: CCA Security

Recall: Public-Key Encryption

- **Syntax:**
  - $\mathsf{Gen}(1^n) \rightarrow (pk, sk)$
  - $\mathsf{Enc}(pk, m) \rightarrow c$
  - $\mathsf{Dec}(sk, c) \rightarrow m'$ or $\perp$

  All algorithms are polynomial time

- **Correctness:** For every m, $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$, where $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

IND-CPA Security

Definition (IND-CPA Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is indistinguishably secure under chosen plaintext attack (IND-CPA) if for all n.u. PPT adversaries $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t.:

$$\Pr\left[\begin{array}{c} (pk, sk) \xleftarrow{\$} \mathsf{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n, pk), \\ b \xleftarrow{\$} \{0,1\} \end{array} : \mathcal{A}\left(pk, \mathsf{Enc}\left(m_b\right)\right) = b\right] \leqslant \frac{1}{2} + \mu(n)$$

IND-CPA Security

Definition (IND-CPA Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is indistinguishably secure under chosen plaintext attack (IND-CPA) if for all n.u. PPT adversaries $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t.:

$$\Pr\left[\begin{array}{c} (pk, sk) \stackrel{\$}{\leftarrow} \mathsf{Gen}(1^n), \\ (m_0, m_1) \leftarrow \mathcal{A}(1^n, pk), \\ b \stackrel{\$}{\leftarrow} \{0,1\} \end{array} : \mathcal{A}\left(pk, \mathsf{Enc}\left(m_b\right)\right) = b \right] \leqslant \frac{1}{2} + \mu(n)$$

1. IND-CPA for one-message implies IND-CPA for multiple messages

Question

What if an adversary finds a decryption box? Is IND-CPA security still enough?

Security against Chosen-Ciphertext Attacks (CCA)

- Augment the IND-CPA security experiment

Security against Chosen-Ciphertext Attacks (CCA)

- Augment the IND-CPA security experiment
- Adversary can make decryption queries over ciphertext of its choice

Security against Chosen-Ciphertext Attacks (CCA)

- Augment the IND-CPA security experiment
- Adversary can make decryption queries over ciphertext of its choice
- **CCA-1**: Decryption queries only before challenge ciphertext query

Security against Chosen-Ciphertext Attacks (CCA)

- Augment the IND-CPA security experiment
- Adversary can make decryption queries over ciphertext of its choice
- **CCA-1**: Decryption queries only before challenge ciphertext query
- **CCA-2**: Decryption queries before and after challenge ciphertext query

Security against Chosen-Ciphertext Attacks (CCA)

- Augment the IND-CPA security experiment
- Adversary can make decryption queries over ciphertext of its choice
- **CCA-1**: Decryption queries only before challenge ciphertext query
- **CCA-2**: Decryption queries before and after challenge ciphertext query
- No decryption query c should be equal to challenge ciphertext c^*

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$

CCA-1 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$

CCA-1 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Output $b' \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Output $b' \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

CCA-1 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Output $b' \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

Definition (IND-CCA-1 Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CCA-1 secure if for all n.u. PPT adversaries $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t. for all auxiliary inputs $z \in \{0,1\}^*$:

$$\left|\Pr\left[\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(1, z) = 1\right] - \Pr\left[\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA1}}(0, z) = 1\right]\right| \leqslant \mu(n)$$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):

CCA-2 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$

CCA-2 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
    - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
    - $m \leftarrow \mathsf{Dec}(sk, c)$
    - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$

CCA-2 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$
  - If $c = c^*$, output reject

CCA-2 Security

$\textbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$
  - If $c = c^*$, output reject
  - $m \leftarrow \mathsf{Dec}(sk, c)$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$
  - If $c = c^*$, output reject
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$

CCA-2 Security

$\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$:

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Decryption query phase 1(repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, \mathsf{st})$
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, \mathsf{st})$
- $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
- Decryption query phase 2 (repeated poly times):
  - $c \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$
  - If $c = c^*$, output reject
  - $m \leftarrow \mathsf{Dec}(sk, c)$
  - $\mathsf{st} = (\mathsf{st}, m)$
- Output $b' \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

CCA-2 Security (contd.)

Definition (IND-CCA-2 Security)

A public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CCA-1 secure if for all n.u. PPT adversaries $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t. for all auxiliary inputs $z \in \{0,1\}^*$:

$$\left| \Pr\left[\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(1, z) = 1 \right] - \Pr\left[\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(0, z) = 1 \right] \right| \leqslant \mu(n)$$

Construction: CCA-1 Secure Public-Key Encryption

Theorem

Assuming NIZKs in the CRS model and IND-CPA secure public-key encryption, there exists IND-CCA-1 secure public-key encryption

Construction: CCA-1 Secure Public-Key Encryption

Theorem

Assuming NIZKs in the CRS model and IND-CPA secure public-key encryption, there exists IND-CCA-1 secure public-key encryption

Think: Proof?

Construction [Naor-Yung]

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA encryption scheme.
Let $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ be an adaptive NIZK.

Construction of $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$:

Construction [Naor-Yung]

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA encryption scheme.
Let $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ be an adaptive NIZK.

Construction of $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$:

- $\mathsf{Gen}'(1^n)$: For $i \in [2]$, compute $(pk_i, sk_i) \leftarrow \mathsf{Gen}(1^n)$. Compute $\sigma \leftarrow \mathsf{K}(1^n)$. Set $pk' = (pk_1, pk_2, \sigma)$, $sk' = sk_1$.

Construction [Naor-Yung]

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA encryption scheme.
Let $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ be an adaptive NIZK.

Construction of $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$:

- $\mathsf{Gen}'(1^n)$: For $i \in [2]$, compute $(pk_i, sk_i) \leftarrow \mathsf{Gen}(1^n)$. Compute $\sigma \leftarrow \mathsf{K}(1^n)$. Set $pk' = (pk_1, pk_2, \sigma)$, $sk' = sk_1$.
- $\mathsf{Enc}'(pk', m)$: For $i \in [2]$, compute $c_i \leftarrow \mathsf{Enc}(pk_i, m; r_i)$. Compute $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ where $x = (pk_1, pk_2, c_1, c_2)$, $w = (m, r_1, r_2)$ and $R(x, w) = 1$ iff c_1 and c_2 encrypt same message m.

Construction [Naor-Yung]

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA encryption scheme.
Let $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ be an adaptive NIZK.

Construction of $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$:

- $\mathsf{Gen}'(1^n)$: For $i \in [2]$, compute $(pk_i, sk_i) \leftarrow \mathsf{Gen}(1^n)$. Compute $\sigma \leftarrow \mathsf{K}(1^n)$. Set $pk' = (pk_1, pk_2, \sigma)$, $sk' = sk_1$.
- $\mathsf{Enc}'(pk', m)$: For $i \in [2]$, compute $c_i \leftarrow \mathsf{Enc}(pk_i, m; r_i)$. Compute $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ where $x = (pk_1, pk_2, c_1, c_2)$, $w = (m, r_1, r_2)$ and $R(x, w) = 1$ iff c_1 and c_2 encrypt same message m.
- $\mathsf{Dec}'(sk', c')$: If $\mathsf{V}(\sigma, \pi) = 0$, output $\perp$. Else output $\mathsf{Dec}(sk_1, c_1)$.

Security (Hybrids)

- H_0: (Honest) Encryption of m_0

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator
- H_2: Change c_2 in challenge ciphertext to encryption of m_1

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator
- H_2: Change c_2 in challenge ciphertext to encryption of m_1
- H_3: Change decryption key sk' to sk_2

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator
- H_2: Change c_2 in challenge ciphertext to encryption of m_1
- H_3: Change decryption key sk' to sk_2
- H_4: Change c_1 in challenge ciphertext to encryption of m_1

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator
- H_2: Change c_2 in challenge ciphertext to encryption of m_1
- H_3: Change decryption key sk' to sk_2
- H_4: Change c_1 in challenge ciphertext to encryption of m_1
- H_5: Change decryption key sk' to sk_1

Security (Hybrids)

- H_0: (Honest) Encryption of m_0
- H_1: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator
- H_2: Change c_2 in challenge ciphertext to encryption of m_1
- H_3: Change decryption key sk' to sk_2
- H_4: Change c_1 in challenge ciphertext to encryption of m_1
- H_5: Change decryption key sk' to sk_1
- H_6: Compute CRS σ in public key and proof π in challenge ciphertext honestly. This experiment is same as (honest) encryption of m_1.

Indistinguishability of Hybrids

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE
- $H_2 \approx H_3$: Only difference might be in the answers to decryption queries of adversary. But from soundness of NIZK, it follows that except with negligible probability, in each decryption query $c = (c_1, c_2)$, c_1 and c_2 encrypt same message. Therefore decrypting c_2 instead of c_1 does not change the answer.

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE
- $H_2 \approx H_3$: Only difference might be in the answers to decryption queries of adversary. But from soundness of NIZK, it follows that except with negligible probability, in each decryption query $c = (c_1, c_2)$, c_1 and c_2 encrypt same message. Therefore decrypting c_2 instead of c_1 does not change the answer.
- $H_3 \approx H_4$: IND-CPA security of underlying PKE

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE
- $H_2 \approx H_3$: Only difference might be in the answers to decryption queries of adversary. But from soundness of NIZK, it follows that except with negligible probability, in each decryption query $c = (c_1, c_2)$, c_1 and c_2 encrypt same message. Therefore decrypting c_2 instead of c_1 does not change the answer.
- $H_3 \approx H_4$: IND-CPA security of underlying PKE
- $H_4 \approx H_5$: Same proof as in $H_2 \approx H_3$

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE
- $H_2 \approx H_3$: Only difference might be in the answers to decryption queries of adversary. But from soundness of NIZK, it follows that except with negligible probability, in each decryption query $c = (c_1, c_2)$, c_1 and c_2 encrypt same message. Therefore decrypting c_2 instead of c_1 does not change the answer.
- $H_3 \approx H_4$: IND-CPA security of underlying PKE
- $H_4 \approx H_5$: Same proof as in $H_2 \approx H_3$
- $H_5 \approx H_6$: ZK property of NIZK

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: IND-CPA security of underlying PKE
- $H_2 \approx H_3$: Only difference might be in the answers to decryption queries of adversary. But from soundness of NIZK, it follows that except with negligible probability, in each decryption query $c = (c_1, c_2)$, c_1 and c_2 encrypt same message. Therefore decrypting c_2 instead of c_1 does not change the answer.
- $H_3 \approx H_4$: IND-CPA security of underlying PKE
- $H_4 \approx H_5$: Same proof as in $H_2 \approx H_3$
- $H_5 \approx H_6$: ZK property of NIZK

Combining the above, we get $H_0 \approx H_6$.