CS 600.442 – Modern Cryptography November 9; 2016

Lecture 17: Chosen Ciphertext Security (II)

Instructor: Abhishek Jain *Scribe: Aarushi Goel*

1 Chosen-Ciphertext Attacks

In the last class we discussed why IND-CPA security might not be sufficient for all real world attacks. We need to consider a stronger security model for cases when the adversary has access to a decryption oracle. We have two security definitions in the Chosen-Ciphertext Attack model, where the adversary can make decryption queries over ciphertexts of its choice (No decryption query c should be equal to the challenge ciphertext c^*):

- CCA-1: The adversary is allowed to make decryption queries only before the challenge ciphertext query.
- CCA-2: The adversary is allowed to make decryption queries before and after the challenge ciphertext query.

In this lecture, we will first revisit the definition of IND-CCA-2 Security and then construct a CCA-2 secure Public key encryption.

2 CCA-2 Security

We begin by defining the challenge experiment $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(b, z)$ for an adversary in the CCA-2 Security model.

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

- $\mathsf{st} = z$
- $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- 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 $b' \leftarrow \mathcal{A}(pk, c^*, \mathsf{st})$

Definition 1 ***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* $\mu(.)$*, s.t. for all auxiliary inputs* $z \in \{0,1\}^*$*:*

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

A CCA-1 secure encryption scheme does not necessarily guarantee security in the CCA-2 model. This is mainly because in CCA-2, the challenge ciphertext is known to the adversary before the second decryption query phase. Thus, the adversary may be able to "maul" the challenge ciphertext into another ciphertext and then request decryption in the second phase. This is called *malleability attack*.
Such attacks can be prevented, if we make the encryption *non-malleable*, i.e., ensure that the adversary's decryption query is "independent" of (instead of just being different from) the challenge ciphertext.

3 CCA-2 Secure Public-Key Encryption

The first construction of CCA-2 secure encryption scheme was given by Dolev-Dwork-Naor. The following cryptographic primitives are required for this construction:

- An IND-CPA secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$
- An adaptive NIZK proof $(\mathsf{K}, \mathsf{P}, \mathsf{V})$
- A strongly unforgeable one-time signature (OTS) scheme $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$. Recall, that for a strongly unforgeable signature scheme we require, that it should be computationally hard for an adversary to come up with a new signature on a message, even if a signature corresponding to that message is already known to him. We assume, without loss of generality that, verification keys in OTS scheme are of length n.

Remark. Note that apart from the primitives used in the construction a CCA-1 secure encryption scheme, we also require a signature scheme. This is mainly required to cater to the additional requirement of non-malleability of the encryption scheme in the CCA-2 model.

3.1 Construction

Assuming we have an IND-CPA secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, an adaptive NIZK proof $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ and a strongly unforgeable OTS scheme $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$, we construct an encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows:

$\boxed{\mathsf{Gen}'(1^n)}$**: Execute the following steps:**

- Compute CRS for NIZK:

$$\sigma \leftarrow \mathsf{K}(1^n)$$

- Compute $2n$ key pairs of IND-CPA encryption scheme:

$$(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$$

where $j \in \{0,1\}$, $i \in [n]$.

- Output $pk' = (\begin{bmatrix} pk_1^0 & pk_2^0 & \dots & pk_n^0 \\ pk_1^1 & pk_2^1 & \dots & pk_n^1 \end{bmatrix}, \sigma)$, $sk' = \begin{bmatrix} sk_1^0 \\ sk_1^1 \end{bmatrix}$

$\boxed{\mathsf{Enc}'(pk', m)}$**: Execute the following steps:**

- Compute key pair for OTS scheme:

$$(SK, VK) \leftarrow \mathsf{Setup}(1^n)$$

- Let $VK = VK_1, ...VK_n$. For every $i \in [n]$, encrypt m using $pk_i^{VK_i}$ and randomness r_i:

$$c_i \leftarrow \mathsf{Enc}(pk_i^{VK_i}, m; r_i)$$

- Compute proof that each c_i encrypts the same message:

$$\pi \leftarrow \mathsf{P}(\sigma, x, w)$$

where $x = (\{pk_i^{VK_i}\}, \{c_i\})$, $w = (m, \{r_i\})$ and $R(x, w) = 1$ iff every c_i encrypts the same message m.

- Sign everything:

$$\Phi \leftarrow \mathsf{Sign}(SK, M)$$

where $M = (\{c_i\}, \pi)$

- Output $c' = (VK, \{c_i\}, \pi, \Phi)$

$\boxed{\mathsf{Dec}'(sk', c')}$**: Execute the following steps:**

- Parse $c' = (VK, \{c_i\}, \pi, \Phi)$
- Let $M = (\{c_i\}, \pi)$
- Verify the signature: Output $\perp$ if

$$\mathsf{Verify}(VK, M, \Phi) = 0$$

- Verify the NIZK proof: Output $\perp$ if

$$\mathsf{V}(\sigma, x, \pi) = 0$$

where $x = (\{pk_i^{VK_i}\}, \{c_i\})$

- Else, decrypt the first ciphertext component:

$$m' \leftarrow \mathsf{Dec}(sk_1^{VK_1}, c_1)$$

- Output m'

Remark. Note that key pair for the signature scheme is not generated in $\mathsf{Gen}'(.)$, because we want to construct a public key encryption scheme. If the key pair for signature scheme, were to be generated in $\mathsf{Gen}'(.)$, the signing key SK, would have to be kept hidden. As a result (because of the structure of ciphertext in this construction), given only the public key, not everybody would be able to encrypt messages, which would make it a secret key encryption scheme.

3.2 Security

Theorem 1 *The encryption scheme presented above, is CCA-2 secure if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is an IND-CPA secure encryption scheme,* $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ *is an adaptively-secure NIZK proof system, and* $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ *is a strongly- unforgeable OTS scheme.*

Proof. We begin by outlining the intuition to argue security of the above construction. Consider the decryption queries in the second phase, i.e., after the adversary receives the challenge ciphertext C^*. Let $C \neq C^*$ be a decryption query. Then the following two cases are possible:

- **Case 1:** $VK = VK^*$
  The verification key VK in C and the verification key VK^* in C^* are same.
  $\Rightarrow (\{c_i^*\}, \pi^*, \Phi^*) \neq (\{c_i\}, \pi, \Phi)$
  If this is the case, then we have been able to generate different signatures corresponding to the same verification key and thus, can break the strong unforgeability of the OTS scheme.
- **Case 2:** $VK \neq VK^*$
  In this case, VK and VK^* must differ in atleast one position $\ell \in [n]$:
  - Answer decryption query using the secret key $sk_\ell^{VK_i}$
  - Knowledge of secret keys $sk_i^{VK_i^*}$, for $i \in [n]$ is not required.
  - Reduce to IND-CPA security of underlying encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

We now construct the following hybrids, to prove security of the above construction in CCA-2 attack model.

$\mathbf{H_0} := \mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(0, z)$ (Honest Encryption of m_0)

- $\sigma \leftarrow \mathsf{K}(1^n)$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0,1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$, $sk' = (sk_1^0, sk_1^1)$.
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$,
  if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$,
  return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$
- $(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n)$, $VK^* = VK_1^*, ..., VK_n^*$
- $c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_0; r_i^*)$
- $\pi^* \leftarrow \mathsf{P}(\sigma, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), w^* = (m_0, \{r_i^*\}))$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$

- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^*, z)$, if $c \neq c^*$ and $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- Output $\mathcal{A}(pk', c^*, z)$

$\mathbf{H_1}$: Compute CRS σ in public key and proof π in challenge ciphertext using NIZK simulator

- $\boxed{(\sigma, \tau) \leftarrow \mathcal{S}_0(1^n)}$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0, 1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$, $sk' = (sk_1^0, sk_1^1)$.
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$
- $(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n)$, $VK^* = VK_1^*, ..., VK_n^*$
- $c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_0; r_i^*)$
- $\boxed{\pi^* \leftarrow \mathcal{S}_1(\sigma, \tau, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), w^* = (m_0, \{r_i^*\}))}$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^*, z)$, if $c \neq c^*$ and $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- Output $\mathcal{A}(pk', c^*, z)$

$\mathbf{H_2}$: Choose VK^* in the beginning during Gen'

- $(\sigma, \tau) \leftarrow \mathcal{S}_0(1^n)$
- $\boxed{(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n),\ VK^* = VK_1^*, ..., VK_n^*}$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0, 1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$, $sk' = (sk_1^0, sk_1^1)$.
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$

- $c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_0; r_i^*)$
- $\pi^* \leftarrow \mathcal{S}_1(\sigma, \tau, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), w^* = (m_0, \{r_i^*\}))$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^*, z)$, if $c \neq c^*$ and $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- Output $\mathcal{A}(pk', c^*, z)$

$\mathbf{H_3}$:

- $(\sigma, \tau) \leftarrow \mathcal{S}_0(1^n)$
- $(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n)$, $VK^* = VK_1^*, ..., VK_n^*$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0, 1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$:
  - If $VK = VK^*$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = sk_i^{\overline{VK_i^*}}$, $i \in [n]$, where $\overline{VK_i^*} = 1 - VK_i^*$.
    If $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_\ell^{\overline{VK_\ell^*}}, c_\ell)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$
- $c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_0; r_i^*)$
- $\pi^* \leftarrow \mathcal{S}_1(\sigma, \tau, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), w^* = (m_0, \{r_i^*\}))$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^*z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $c \neq c^*$:
  - If $VK = VK^*$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = sk_i^{\overline{VK_i^*}}$, $i \in [n]$, where $\overline{VK_i^*} = 1 - VK_i^*$.
    If $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_\ell^{\overline{VK_\ell^*}}, c_\ell)$

- Output $\mathcal{A}(pk', c^*, z)$

$\mathbf{H_4}$: Change every c_i^* in C^* to be encryption of m_1

- $(\sigma, \tau) \leftarrow \mathcal{S}_0(1^n)$
- $(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n)$, $VK^* = VK_1^*, ..., VK_n^*$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0,1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$:
  - If $VK = VK^*$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = sk_i^{\overline{VK_i^*}}$, $i \in [n]$, where $\overline{VK_i^*} = 1 - VK_i^*$.
    If $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_\ell^{\overline{VK_\ell^*}}, c_\ell)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$
- $\boxed{c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_1; r_i^*)}$
- $\pi^* \leftarrow \mathcal{S}_1(\sigma, \tau, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), \boxed{w^* = (m_1, \{r_i^*\})})$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M^* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^* z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $c \neq c^*$:
  - If $VK = VK^*$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = sk_i^{\overline{VK_i^*}}$, $i \in [n]$, where $\overline{VK_i^*} = 1 - VK_i^*$.
    If $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_\ell^{\overline{VK_\ell^*}}, c_\ell)$
- Output $\mathcal{A}(pk', c^*, z)$

$\mathbf{H_5}$:= $\mathbf{Expt}_{\mathcal{A}}^{\mathsf{CCA2}}(1, z)$ (Honest Encryption of m_1)

- $\boxed{\sigma \leftarrow \mathsf{K}(1^n)}$
- $(pk_i^j, sk_i^j) \leftarrow \mathsf{Gen}(1^n)$ for $j \in \{0,1\}$, $i \in [n]$.
- $pk' = (\{pk_i^0, pk_i^1\}, \sigma)$, $sk' = (sk_1^0, sk_1^1)$.

- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', z)$, if $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- $(m_0, m_1) \leftarrow \mathcal{A}(pk, z)$
- $(SK^*, VK^*) \leftarrow \mathsf{Setup}(1^n)$, $VK^* = VK_1^*, ..., VK_n^*$
- $c_i^* \leftarrow \mathsf{Enc}(pk_i^{VK_i^*}, m_1; r_i^*)$
- $\pi^* \leftarrow \mathsf{P}(\sigma, x^* = (\{pk_i^{VK_i^*}\}, \{c_i^*\}), w^* = (m_1, \{r_i^*\}))$
- $\Phi^* \leftarrow \mathsf{Sign}(SK^*, M* = (\{c_i\}, \pi))$
- $c^* = (VK^*, \{c_i^*\}, \pi^*, \Phi^*)$
- On receiving a decryption query $c = (VK, \{c_i\}, \pi, \Phi)$ from $\mathcal{A}(pk', c^*, z)$, if $c \neq c^*$ and $\mathsf{Verify}(VK, M = (\{c_i\}, \pi), \Phi) = 1$ and $\mathsf{V}(\sigma, x = (\{pk_i^{VK_i}\}, \{c_i\}), \pi) = 1$, return $\mathsf{Dec}(sk_1^{VK_1}, c_1)$
- Output $\mathcal{A}(pk', c^*, z)$

We now argue indistinguishability of the above hybrids:

- $\mathbf{H_0} \approx \mathbf{H_1}$: Since the only difference between the two hybrids is that in H_1, CRS σ and proof π are computed using NIZK simulator. The indistinguishability of these hybrids follows from the Zero Knowledge property of NIZK.
- $\mathbf{H_1} \approx \mathbf{H_2}$: From an adversary's point of view, generating VK^* early or later does not change the distribution.
- $\mathbf{H_2} \approx \mathbf{H_3}$: We argue indistinguishability of these hybrids as follows:
  - Case 1: The protocol is aborted.
    We claim that the probability of aborting is negligible. By the definition of CCA-2, $c \neq c^*$. So if $VK = VK^*$, then it must be that $(\{c_i\}, \pi, \Phi) \neq (\{c_i^*\}, \pi^*, \Phi^*)$. Now, if $\mathsf{Verify}(VK, (\{c_i\}, \pi), \Phi) = 1$, then we can break strong unforgeability of the OTS scheme.
  - Case 2: The protocol is not aborted.
    Let ℓ be the position s.t. $VK_\ell \neq VK_\ell^*$. Note that the only difference in H_2 and H_3 in this case might be the answers to the decryption queries of adversary. In particular, in H_2, we decrypt c_1 in c using $sk_1^{VK_1}$. In contrast, in H_3, we decrypt c_ℓ in c using $sk_\ell^{\overline{VK}_\ell^*}$. Now, from soundness of NIZK, it follows that except with negligible probability, all the $c_i's$ in c encrypt the same message. Therefore decrypting c_ℓ instead of c_1 does not change the answer.
- $\mathbf{H_3} \approx \mathbf{H_4}$: Indistinguishability of these hybrids follows from the IND-CPA security if underlying PKE $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$

- $\mathbf{H_4 \approx H_5}$: Combining the above steps, we get $H_0 \approx H_3$. Indistinguishability of these hybrids (H_4 and H_5) can be argued in a similar manner (in the reverse order).

Combining the above we get $H_0 \approx H_5$.
Hence, Encryption of m_0 is computationally indistinguishable from the encryption of m_1 in the CCA-2 model. ■