Lecture 23: CCA Security - II

Recall: Chosen-Ciphertext Attacks (CCA)

- 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^*

Recall: Chosen-Ciphertext Attacks (CCA)

- 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^*

<u>Last time</u>: Construction of CCA-1 secure PKE

Recall: Chosen-Ciphertext Attacks (CCA)

- 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^*

Last time: Construction of CCA-1 secure PKE

Today: Construction of CCA-2 secure PKE

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 $z \in \{0,1\}^*$:

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

Question

Why doesn't the CCA-1 secure construction also satisfy CCA-2 security?

Main Challenge (Intuition)

- Adversary may be able to "maul" challenge ciphertext into another ciphertext and then request decryption

Main Challenge (Intuition)

- Adversary may be able to "maul" challenge ciphertext into another ciphertext and then request decryption
- "Two-key trick" (as used in CCA-1 construction) does not work anymore

Main Challenge (Intuition)

- Adversary may be able to "maul" challenge ciphertext into another ciphertext and then request decryption
- "Two-key trick" (as used in CCA-1 construction) does not work anymore
- Must ensure that adversary's decryption query is "independent" of (and not just different from) the challenge ciphertext

Construction [Dolev-Dwork-Naor]

Ingredients:

- 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 $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$. Assume, wlog, that verification keys in OTS scheme are of length n.

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

- $\mathsf{Gen}'(1^n)$:

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:
  - Compute key pair for OTS scheme: $(SK, VK) \leftarrow \mathsf{Setup}(1^n)$.

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:
  - Compute key pair for OTS scheme: $(SK, VK) \leftarrow \mathsf{Setup}(1^n)$.
  - Let $VK = VK_1, \ldots, VK_n$. For every $i \in [n]$, encrypt m using $pk_i^{VK_i}$ and randomness r_i: $c_i \leftarrow \mathsf{Enc}\left(pk_i^{VK_i}, m; r_i\right)$

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:
  - Compute key pair for OTS scheme: $(SK, VK) \leftarrow \mathsf{Setup}(1^n)$.
  - Let $VK = VK_1, \ldots, VK_n$. For every $i \in [n]$, encrypt m using $pk_i^{VK_i}$ and randomness r_i: $c_i \leftarrow \mathsf{Enc}\left(pk_i^{VK_i}, m; r_i\right)$
  - Compute proof that each c_i encrypts the same message: $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ where $x = \left(\left\{pk_i^{VK_i}\right\}, \{c_i\}\right)$, $w = (m, \{r_i\})$ and $R(x, w) = 1$ iff every c_i encrypts the same message m.

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:
  - Compute key pair for OTS scheme: $(SK, VK) \leftarrow \mathsf{Setup}(1^n)$.
  - Let $VK = VK_1, \ldots, VK_n$. For every $i \in [n]$, encrypt m using $pk_i^{VK_i}$ and randomness r_i: $c_i \leftarrow \mathsf{Enc}\left(pk_i^{VK_i}, m; r_i\right)$
  - Compute proof that each c_i encrypts the same message: $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ where $x = \left(\left\{pk_i^{VK_i}\right\}, \{c_i\}\right)$, $w = (m, \{r_i\})$ and $R(x, w) = 1$ iff every c_i encrypts the same message m.
  - Sign everything: $\Phi \leftarrow \mathsf{Sign}(SK, M)$ where $M = (\{c_i\}, \pi)$

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

- $\mathsf{Gen}'(1^n)$:
  - Compute CRS for NIZK: $\sigma \leftarrow \mathsf{K}(1^n)$
  - Compute $2n$ key pairs of IND-CPA encryption scheme: $\left(pk_i^j, sk_i^j\right) \leftarrow \mathsf{Gen}(1^n)$, where $j \in \{0,1\}$, $i \in [n]$.
  - Output $pk' = \left(\left\{pk_i^0, pk_i^1\right\}, \sigma\right)$, $sk' = \left(sk_1^0, sk_1^1\right)$.
- $\mathsf{Enc}'(pk', m)$:
  - Compute key pair for OTS scheme: $(SK, VK) \leftarrow \mathsf{Setup}(1^n)$.
  - Let $VK = VK_1, \ldots, VK_n$. For every $i \in [n]$, encrypt m using $pk_i^{VK_i}$ and randomness r_i: $c_i \leftarrow \mathsf{Enc}\left(pk_i^{VK_i}, m; r_i\right)$
  - Compute proof that each c_i encrypts the same message: $\pi \leftarrow \mathsf{P}(\sigma, x, w)$ where $x = \left(\left\{pk_i^{VK_i}\right\}, \{c_i\}\right)$, $w = (m, \{r_i\})$ 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 $c' = (VK, \{c_i\}, \pi, \Phi)$

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

- $\mathsf{Dec}'(sk', c')$:

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

- $\mathsf{Dec}'\left(sk', c'\right)$:
    - Parse $c' = \left(VK, \{c_i\}, \pi, \Phi\right)$

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

- $\mathsf{Dec}'(sk', c')$:
  - Parse $c' = (VK, \{c_i\}, \pi, \Phi)$
  - Let $M = (\{c_i\}, \pi)$

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

- $\mathsf{Dec}'(sk', c')$:
  - Parse $c' = (VK, \{c_i\}, \pi, \Phi)$
  - Let $M = (\{c_i\}, \pi)$
  - Verify the signature: Output $\perp$ if $\mathsf{Verify}(VK, M, \Phi) = 0$

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

- $\mathsf{Dec}'(sk', c')$:
  - 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 $\mathsf{V}(\sigma, \pi) = 0$

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

- $\mathsf{Dec}'(sk', c')$:
  - 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 $\mathsf{V}(\sigma, \pi) = 0$
  - Else, decrypt the first ciphertext component: $m' \leftarrow \mathsf{Dec}\left(sk_1^{VK_1}, c_1\right)$

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

- $\mathsf{Dec}'(sk', c')$:
  - 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 $\mathsf{V}(\sigma, \pi) = 0$
  - Else, decrypt the first ciphertext component: $m' \leftarrow \mathsf{Dec}\left(sk_1^{VK_1}, c_1\right)$
  - Output m'

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query
- If verification key VK in c and verification key VK^* in challenge ciphertext c^* are same, then we can break the strong unforgeability of OTS

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query
- If verification key VK in c and verification key VK^* in challenge ciphertext c^* are same, then we can break the strong unforgeability of OTS
- If different, then VK and VK^* differ in at least one position $\ell \in [n]$:

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query
- If verification key VK in c and verification key VK^* in challenge ciphertext c^* are same, then we can break the strong unforgeability of OTS
- If different, then VK and VK^* differ in at least one position $\ell \in [n]$:
    - Answer decryption query using the secret key $sk_\ell^{VK_i}$.

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query
- If verification key VK in c and verification key VK^* in challenge ciphertext c^* are same, then we can break the strong unforgeability of OTS
- If different, then VK and VK^* differ in at least one position $\ell \in [n]$:
  - Answer decryption query using the secret key $sk_\ell^{VK_i}$.
  - Don't need to know the secret keys $sk_i^{VK_i^*}$ for $i \in [n]$

Security (Intuition)

Consider decryption queries after the challenge ciphertext query c^*:

- Let c be a decryption query
- If verification key VK in c and verification key VK^* in challenge ciphertext c^* are same, then we can break the strong unforgeability of OTS
- If different, then VK and VK^* differ in at least one position $\ell \in [n]$:
  - Answer decryption query using the secret key $sk_\ell^{VK_i}$.
  - Don't need to know the secret keys $sk_i^{VK_i^*}$ for $i \in [n]$
  - Reduce to IND-CPA security of underlying encryption scheme

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: Choose VK^* in the beginning during Gen'

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: Choose VK^* in the beginning during Gen'
- H_3: For any decryption query $c = (VK, \{c_i\}, \pi, \Phi)$:

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: Choose VK^* in the beginning during Gen'
- H_3: For any decryption query $c = (VK, \{c_i\}, \pi, \Phi)$:
  - If $VK = VK^*$ and $\mathsf{Verify}(VK, (\{c_i\}, \pi, \Phi), \Phi) = 1$, then abort.

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: Choose VK^* in the beginning during Gen'
- H_3: For any decryption query $c = (VK, \{c_i\}, \pi, \Phi)$:
  - If $VK = VK^*$ and $\mathsf{Verify}(VK, (\{c_i\}, \pi, \Phi), \Phi) = 1$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = \left\{ sk_i^{\overline{VK}_i^*} \right\}$, $i \in [n]$, where $\overline{VK}_i^* = 1 - VK_i^*$. Decrypt c by decrypting c_ℓ (instead of c_1) using $sk_\ell^{\overline{VK}_\ell^*}$.

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: Choose VK^* in the beginning during Gen'
- H_3: For any decryption query $c = (VK, \{c_i\}, \pi, \Phi)$:
  - If $VK = VK^*$ and $\mathsf{Verify}(VK, (\{c_i\}, \pi, \Phi), \Phi) = 1$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = \left\{ sk_i^{\overline{VK}_i^*} \right\}$, $i \in [n]$, where $\overline{VK}_i^* = 1 - VK_i^*$. Decrypt c by decrypting c_ℓ (instead of c_1) using $sk_\ell^{\overline{VK}_\ell^*}$.
- H_4: Change every c_i^* in c^* 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: Choose VK^* in the beginning during Gen'
- H_3: For any decryption query $c = (VK, \{c_i\}, \pi, \Phi)$:
  - If $VK = VK^*$ and $\mathsf{Verify}(VK, (\{c_i\}, \pi, \Phi), \Phi) = 1$, then abort.
  - Else, let $\ell \in [n]$ be such that VK^* and VK in c differ at position ℓ. Set $sk' = \left\{ sk_i^{\overline{VK}_i^*} \right\}$, $i \in [n]$, where $\overline{VK}_i^* = 1 - VK_i^*$. Decrypt c by decrypting c_ℓ (instead of c_1) using $sk_\ell^{\overline{VK}_\ell^*}$.
- H_4: Change every c_i^* in c^* to encryption of m_1
- H_5: 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

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

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: Generating VK^* early or later does not change the distribution

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: Generating VK^* early or later does not change the distribution
- $H_2 \approx H_3$: We argue indistinguishability as follows:

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: Generating VK^* early or later does not change the distribution
- $H_2 \approx H_3$: We argue indistinguishability as follows:
  - First, we argue that probability of aborting is negligible. Recall that $c \neq c^*$ by the definition of CCA-2. Then, if $VK = VK^*$, 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.

Indistinguishability of Hybrids

- $H_0 \approx H_1$: ZK property of NIZK
- $H_1 \approx H_2$: Generating VK^* early or later does not change the distribution
- $H_2 \approx H_3$: We argue indistinguishability as follows:
  - First, we argue that probability of aborting is negligible. Recall that $c \neq c^*$ by the definition of CCA-2. Then, if $VK = VK^*$, 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.
  - Now, conditioned on not aborting, 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.

Indistinguishability of Hybrids (contd.)

- $H_3 \approx H_4$: IND-CPA security of underlying PKE
- $H_4 \approx H_5$: ZK property of NIZK

Indistinguishability of Hybrids (contd.)

- $H_3 \approx H_4$: IND-CPA security of underlying PKE
- $H_4 \approx H_5$: ZK property of NIZK

Combining the above, we get $H_0 \approx H_5$.