# Lecture 4: PKE with CCA1 Security

## Based on “Advanced Topics in Cryptography [J.Katz]

Shengli Liu

slliu@sjtu.edu.cn

Lab of Cryptography and Information Security
Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200240, China

## Definition (IND-CCA1/CCA2 Security)

$\forall$ stateful PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\mathbf{Pr}\left[\begin{array}{l} (pk, sk) \leftarrow \mathsf{Gen}(1^\kappa);\ (s, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk); \\ b \leftarrow \{0,1\}, c^* \leftarrow \mathsf{Enc}_{pk}(m_b; r^*);\ b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}_{\neq c^*}(sk,\cdot)}(s, c) \end{array} : b' = b\right] = 1/2 \pm \mathsf{negl}(\kappa).$$

# CCA1/CCA2 Security of PKE

## Definition (IND-CCA1/CCA2 Security)

$\forall$ stateful PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,

$$\Pr\left[\begin{array}{l} (pk, sk) \leftarrow \mathsf{Gen}(1^\kappa);\ (s, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk); \\ b \leftarrow \{0,1\}, c^* \leftarrow \mathsf{Enc}_{pk}(m_b; r^*);\ b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}_{\neq c^*}(sk,\cdot)}(s, c) \end{array} : b' = b\right] = 1/2 \pm \mathsf{negl}(\kappa).$$

i.e., $|\Pr[\mathcal{A}$ **wins in the CCA1/CCA2 game**$] - 1/2| = |\Pr[b = b'] - 1/2| = \mathsf{negl}(\kappa)$,

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | $\xrightarrow{pk}$ | |
| | $\xleftarrow{c}$ | decryption queries (Decryption Phase I) |
| $m/\perp \leftarrow \mathsf{Dec}(sk, c)$ | $\xrightarrow{m/\perp}$ | |
| $b \leftarrow \{0,1\}$ | $\xleftarrow{(m_0, m_1)}$ | choose $m_0, m_1$(Challenge Phase) |
| $c^* \leftarrow \mathsf{Enc}_{pk}(m_b; r^*)$ | $\xrightarrow{c^*}$ | |
| | $\xleftarrow{c(c \neq c^*)}$ | decryption queries (Decryption Phase II) |
| $m/\perp \leftarrow \mathsf{Dec}(sk, c)$ | $\xrightarrow{m/\perp}$ | |
| $b' = b?$ | $\xleftarrow{b'}$ | guess $b'$(Guessing Phase) |

## Definition (IND-CCA1/CCA2 Security)

$\forall$ PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,
$2|\Pr[b = b'] - 1/2| = |\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]| = 2\mathsf{negl}(\kappa) = \mathsf{negl}(\kappa)$,
i.e., $\left|\Pr\left[\mathbf{Exp}^{CCA}_{PKE,\mathcal{A}}(1) = 1\right] - \Pr\left[\mathbf{Exp}^{CCA}_{PKE,\mathcal{A}}(0) = 1\right]\right| = \mathsf{negl}(\kappa)$.

$\underline{\mathbf{Exp}^{CCA1/CCA2}_{PKE,\mathcal{A}}(b)}$:

$$
\begin{array}{|ccc|}
\hline
\mathcal{C} & & \mathcal{A} \\
(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa) & \xrightarrow{\quad pk \quad} & \\
 & \xleftarrow{\quad c \quad} & \text{decryption queries} \\
m/\perp \leftarrow \mathsf{Dec}(sk, c) & \xrightarrow{\quad m/\perp \quad} & \\
 & \xleftarrow{\quad (m_0, m_1) \quad} & \text{choose } m_0, m_1 \\
c^* \leftarrow \mathsf{Enc}_{pk}(m_b; r^*) & \xrightarrow{\quad c^* \quad} & \\
 & \xleftarrow{\quad c (c \neq c^*) \quad} & \text{decryption queries} \\
m/\perp \leftarrow \mathsf{Dec}(sk, c) & \xrightarrow{\quad m/\perp \quad} & \\
\mathsf{Return}(b') & \xleftarrow{\quad b' \quad} & \text{guess } b' \\
\hline
\end{array}
$$

## $\mathcal{P}, \mathcal{NP}$ and $\mathcal{NP}$-Complete

Language $L$: a set of strings, i.e., $L \subseteq \{0,1\}^*$.

$L \in \mathcal{P}$: $\exists$ Poly-Time(PT) Turing machine $M$ s. t. $\forall x \in \{0,1\}^*$,

$$x \in L \Leftrightarrow M(x) = 1.$$

$L \in \mathcal{NP}$: $\exists$ Poly-Time(PT) Turing machine $M$ s. t. $\forall x \in \{0,1\}^*$,

$$x \in L \Leftrightarrow \exists w_x \in \{0,1\}^{poly(|x|)}, \text{ s. t. } M(x, w_x) = 1.$$

$L_1 \leq_p L_2$: language $L_1$ is poly-time reducible to language $L_2$ if $\exists f$ s.t.

(1) $f$ is poly-time computable;
(2) $x \in L_1 \Leftrightarrow f(x) \in L_2$.

Note 1: $L_1 \leq_p L_2,\ L_2 \in \mathcal{P} \Rightarrow L_1 \in \mathcal{P}$;
Note 2: $L_1 \leq_p L_2,\ L_2 \in \mathcal{NP} \Rightarrow L_1 \in \mathcal{NP}$;

$\mathcal{NP}$-Complete: $L'$ is $\mathcal{NP}$-Complete if

(1) $L' \in \mathcal{NP}$;
(2) $L \leq_p L', \forall L \in \mathcal{NP}$.

Example: (1) SAT: the language of all satisfiable CNF formulae; (2) $\{ G : G$ is a graph which contains a Hamilton cycle$\}$. (3) $\{ G = (V, E) : G$ is a 3-colorable graph$\}$.

Interactive Proof (IP) system for language $L$.

- consisting of Prover P and ppt Verifier V.
- $\mathsf{P}(x) \rightleftharpoons \mathsf{V}(x)$: common input $x \in L$ and interactions between P and V. Finally V will output 0/1.
- Completeness. $\forall x \in L$, $(\mathsf{P}(x) \rightleftharpoons \mathsf{V}(x)) = 1$.
- Soundness. $\forall x \notin L$, $\forall \tilde{\mathsf{P}}$, $\left(\tilde{\mathsf{P}}(x) \rightleftharpoons \mathsf{V}(x)\right) = 0$ with high probability.

Note. $\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{IP}$, $\mathsf{IP} = \mathsf{PSPACE}$.

A pair of ppt algorithms (P, V) is a non-interactive zero-knowledge (NIZK) proof system for a language $L \in NP$ if:

Completeness. $\forall x \in L$ and its witness $w_x$,

$$\mathbf{Pr}\left[r \leftarrow \{0,1\}^{poly(\kappa)}; \pi \leftarrow \mathsf{P}(r, x, w_x) : \mathsf{V}(r, x, \pi) = 1\right] = 1.$$

Soundness. $\forall x \notin L$, $\forall \widetilde{\mathsf{P}}$(even all-powerful $\widetilde{\mathsf{P}}$), the following is negl (in $\kappa$):

$$\mathbf{Pr}\left[r \leftarrow \{0,1\}^{poly(\kappa)}; \tilde{\pi} \leftarrow \widetilde{\mathsf{P}}(r, x) : \mathsf{V}(r, x, \tilde{\pi}) = 1\right] = negl(\kappa).$$

Zero-knowledge. There exists a ppt simulator Sim s.t. $\forall x \in L$, (with $|x| = \kappa$) and $\forall$ witness $w_x$ for $x$, the following distributions are computationally indistinguishable:

$$\{r \leftarrow \{0,1\}^{poly(\kappa)}; \pi \leftarrow \mathsf{P}(r, x, w_x) : (r, x, \pi)\} \approx_c \{(\tilde{r}, \tilde{\pi}) \leftarrow \mathsf{Sim}(x) : (\tilde{r}, x, \tilde{\pi})\}$$

Note. The requirement that P is ppt is due to cryptographic applications.

A pair of ppt algorithms (P, V) is an adaptive non-interactive zero-knowledge (aNIZK) proof system for a language $L \in NP$ if:

Completeness. $\forall x \in L$ and its witness $w_x$,

$$\Pr\left[r \leftarrow \{0,1\}^{poly(\kappa)}; \pi \leftarrow \mathsf{P}(r, x, w_x) : \mathsf{V}(r, x, \pi) = 1\right] = 1.$$

Adaptive Soundness. $\forall \widetilde{\mathsf{P}}$(even all-powerful $\widetilde{\mathsf{P}}$), the following is negligible in $\kappa$:

$$\Pr\left[r \leftarrow \{0,1\}^{poly(\kappa)}; (x, \tilde{\pi}) \leftarrow \widetilde{\mathsf{P}}(r) : \mathsf{V}(r, x, \tilde{\pi}) = 1 \wedge x \in \{0,1\}^{\kappa} \setminus L\right]$$
$$= negl(\kappa).$$

Adaptive Zero-knowledge. There exists a ppt stateful simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ s.t. and $\forall$ stateful ppt $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the following distributions are computationally indistinguishable:

$$\{r \leftarrow \{0,1\}^{poly(\kappa)}; (x \in L, w_x) \leftarrow \mathcal{A}_1(r); \pi \leftarrow \mathsf{P}(r, x, w_x); b \leftarrow \mathcal{A}_2(r, x, \pi) : b = 1\}$$
$$\approx_c \{(\tilde{r} \leftarrow \mathsf{Sim}_1(1^{\kappa}); (x \in L, w_x) \leftarrow \mathcal{A}_1(\tilde{r}); \tilde{\pi} \leftarrow \mathsf{Sim}_2(\tilde{r}, x); b \leftarrow \mathcal{A}_2(\tilde{r}, x, \tilde{\pi}) : b = 1\}$$

**Building blocks:** CPA secure PKE $=$ (KeyGen, Enc, Dec) with perfect correctness and aNIZK (P, V).

**Noar-Yung's PKE$'$ Construction**

KeyGen'$(1^\kappa)$: $(pk_1, sk_1) \leftarrow$ KeyGen$(1^\kappa)$;
$(pk_2, sk_2) \leftarrow$ KeyGen$(1^\kappa)$;
$r \leftarrow \{0,1\}^{poly(\kappa)}$;
$pk = (pk_1, pk_2, r)$; $sk = (sk_1)$; Return $(pk, sk)$.

Enc'$(pk, m)$: $c_1 \leftarrow$ Enc$(pk_1, m; w_1)$;
$c_2 \leftarrow$ Enc$(pk_2, m; w_2)$;
$\pi \leftarrow$ P$(r, (c_1, c_2), (m, w_1, w_2))$.
Return $(c_1, c_2, \pi)$.

Dec'$(sk, c)$: $sk_1 := sk$; $(c_1, c_2, \pi) := c$
If V$(r, (c_1, c_2), \pi) = 0$
Return$(\perp)$;
Else $m' \leftarrow$ Dec$(sk_1, c_1)$;
Return $m'$.

aNIZK for $L \in \mathcal{NP}$

$$L := \{(c_1, c_2) \mid \exists (m, w_1, w_2) \text{ s.t. } c_1 = \mathsf{Enc}(pk_1, m; w_1) \wedge c_2 = \mathsf{Enc}(pk_2, m; w_2)\}.$$

Game 0: $= \mathbf{Exp}^{CCA1-0}_{PKE',\mathcal{A}}(\kappa)$

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $\underline{r \leftarrow \{0,1\}^{poly(\kappa)}}; pk = (pk_1, pk_2, r); sk = sk_1$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(sk, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption queries |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0,\ \ m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(sk_1, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $c^* \leftarrow \mathsf{Enc}'(pk, m_0; r^*)$ | | |
| $\{\ \ c_1^* \leftarrow \mathsf{Enc}(pk_1, m_0; w_1^*);$ | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_0; w_2^*);$ | | |
| $\underline{\pi^* \leftarrow \mathsf{P}(r, (c_1^*, c_2^*), (m_0, w_1^*, w_2^*))}\}$ | $\xrightarrow{c^*=(c_1^*,c_2^*,\pi^*)}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

Game 1: $|\Pr[\text{Game } 1 = 1] - \Pr[\text{Game } 0 = 1]| = \mathsf{Adv}^{ZK}_{aNIZK}(\kappa)$ (adaptive ZK).

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $\underline{r \leftarrow \mathsf{Sim}_1(1^\kappa)}; pk = (pk_1, pk_2, r); sk = sk_1$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(sk, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption querie |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0,\ \ m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(sk_1, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $c^* \leftarrow \mathsf{Enc}'(pk, m_0; r^*)$ | | |
| $\{$ $c_1^* \leftarrow \mathsf{Enc}(pk_1, m_0; w_1^*)$; | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_0; w_2^*)$; | | |
| $\underline{\pi^* \leftarrow \mathsf{Sim}_2(r, (c_1^*, c_2^*))}\}$ | $\xrightarrow{c^*=(c_1^*,c_2^*,\underline{\pi^*})}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

$|\Pr[\text{Game } 1 = 1] - \Pr[\text{Game } 0 = 1]| = \mathsf{Adv}^{ZK}_{aNIZK}(\kappa)$ (adaptive ZK).

| $\mathcal{C}'$ | | $\mathcal{A}'/\mathcal{C}$ |
|---|---|---|
| $r \leftarrow \{0,1\}^{poly(\kappa)}$ | | $(pk_1, sk_1), (pk_2, sk_2) \leftarrow$ Key |
| $r \leftarrow \mathsf{Sim}_1(1^\kappa)$ | $\xrightarrow{r}$ | $pk = (pk_1, pk_2, \underline{r}); sk = sk_1$ |
| | | $m'/\perp \leftarrow \mathsf{Dec}'(sk, c)$ |
| | | $\{\quad$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0,\ \ m$ |
| | | Else $m' \leftarrow \mathsf{Dec}(sk_1, c_1)\}$ |
| | | $c^* \leftarrow \mathsf{Enc}'(pk, m_0; r^*)$ |
| | | $\{\quad c_1^* \leftarrow \mathsf{Enc}(pk_1, m_0; w_1^*);$ |
| $\pi^* \leftarrow \mathsf{P}(r, (c_1^*, c_2^*), (m_0, w_1^*, w_2^*))$ | $\xleftarrow{((c_1^*, c_2^*), (m_0, w_1^*, w_2^*))}$ | $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_0; w_2^*);$ |
| $\pi^* \leftarrow \mathsf{Sim}_2(r, (c_1^*, c_2^*))$ | $\xrightarrow{\pi^*}$ | $c^* = (c_1^*, c_2^*, \underline{\pi^*})\}$ |
| | | Return$(b')$ |

Game 2: $|\Pr[\text{Game } 2 = 1] - \Pr[\text{Game } 1 = 1]| \leq \mathsf{Adv}^{cpa}_{\mathsf{PKE}}(\kappa)$.

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $r \leftarrow \mathsf{Sim}_1(1^\kappa); pk = (pk_1, pk_2, r); sk = sk_1$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(sk, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption queries |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0$, $m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(sk_1, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| $c^* \leftarrow \mathsf{Enc}'(pk, \cdot; r^*)$ | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $\{$ $c_1^* \leftarrow \mathsf{Enc}(pk_1, m_0; w_1^*)$; | | |
| $\underline{c_2^*} \leftarrow \mathsf{Enc}(pk_2, \underline{m_1}; w_2^*)$; | | |
| $\pi^* \leftarrow \mathsf{Sim}_2(r, (c_1^*, c_2^*))\}$ | $\xrightarrow{c^*=(c_1^*,\underline{c_2^*},\pi^*)}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

Game 3: $|\Pr[\text{Game } 3 = 1] - \Pr[\text{Game } 2 = 1]| \le \mathsf{Adv}^{ZK}_{aNIZK}(\kappa) + \mathsf{Adv}^{sound}_{aNIZK}(\kappa)$.

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $r \leftarrow \mathsf{Sim}_1(1^\kappa); pk = (pk_1, pk_2, r); \underline{sk = sk_2}$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(\underline{sk}, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption queries |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0$, $m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(\underline{sk_2}, c_2)\}$ | $\xrightarrow{m'/\perp}$ | |
| $c^* \leftarrow \mathsf{Enc}'(pk, \cdot; r^*)$ | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $\{$ $c_1^* \leftarrow \mathsf{Enc}(pk_1, m_0; w_1^*)$; | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_1; w_2^*)$; | | |
| $\pi^* \leftarrow \mathsf{Sim}_2(r, (c_1^*, c_2^*))\}$ | $\xrightarrow{c^*=(c_1^*,c_2^*,\pi^*)}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

**Fake$_i$:** the event that $\mathcal{A}$ submits $(c_1, c_2, \pi)$ in Game $i$ to the decryption oracle with

$$(\mathsf{Dec}(sk_1, c_1) \neq \mathsf{Dec}(sk_2, c_2)) \ \wedge \ (\mathsf{V}(r, (c_1; c_2), \pi) = 1)\,.$$

- $\Pr[\mathsf{Fake}_3] = \Pr[\mathsf{Fake}_2]$;
- Game $3|\neg\mathsf{Fake}_3$ = Game $2|\neg\mathsf{Fake}_2$.
- So $|\Pr[\text{Game } 3 = 1] - \Pr[\text{Game } 2 = 1]| \leq \Pr[\mathsf{Fake}_2]$.

**Lemma (Shoup, Difference Lemma)**

Let $A, B, C$ be events. If $\Pr[A|\neg C] = \Pr[B|\neg C]$, then $|\Pr[A] - \Pr[B]| \leq \Pr[C]$.

**Proof.**

$$\Pr[A] = \Pr[A \wedge C] + \Pr[A \wedge \neg C]\,.$$
$$\Pr[B] = \Pr[B \wedge C] + \Pr[B \wedge \neg C]\,.$$
$$\Pr[A] - \Pr[B] = \Pr[A \wedge C] - \Pr[B \wedge C]\,.$$
$$|\Pr[A] - \Pr[B]| = |\Pr[A \wedge C] - \Pr[B \wedge C]| = |(\Pr[A|C] - \Pr[B|C])\Pr[C]| \leq \Pr[C]\,.$$

□

- $|\Pr[\text{Game } 3 = 1] - \Pr[\text{Game } 2 = 1]| \le \Pr[\text{Fake}_2]$.
- $\Pr[\text{Fake}_2] = \Pr[\text{Fake}_1]$. $\mathcal{A}$ has the same view before the challenge phase in both Game 1 and Game 2.
- $|\Pr[\text{Fake}_1] - \Pr[\text{Fake}_0]| \le \text{Adv}_{aNIZK}^{ZK}(\kappa)$.
- $\Pr[\text{Fake}_0] = \text{Adv}_{aNIZK}^{sound}(\kappa)$.

Hence
$|\Pr[\text{Game } 3 = 1] - \Pr[\text{Game } 2 = 1]| \le \Pr[\text{Fake}_2] \le \text{Adv}_{aNIZK}^{ZK}(\kappa) + \text{Adv}_{aNIZK}^{sound}(\kappa)$.

Game 4: $|\Pr[\text{Game } 4 = 1] - \Pr[\text{Game } 3 = 1]| \leq \mathsf{Adv}_{\mathsf{PKE}}^{CPA}(\kappa)$.

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $r \leftarrow \mathsf{Sim}_1(1^\kappa); pk = (pk_1, pk_2, r); sk = sk_2$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(sk, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption queries |
| $\{\ \ \text{If } \mathsf{V}(r, (c_1, c_2), \pi) = 0,\ \ m' := \perp$ | | |
| $\text{Else } m' \leftarrow \mathsf{Dec}(sk_2, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| $c^* \leftarrow \mathsf{Enc}'(pk, m_1; r^*)$ | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $\{\ \ \underline{c_1^*} \leftarrow \mathsf{Enc}(pk_1, \underline{m_1}; w_1^*);$ | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_1; w_2^*);$ | | |
| $\pi^* \leftarrow \mathsf{Sim}_2(r, (c_1^*, c_2^*))\}$ | $\xrightarrow{c^*=(\underline{c_1^*},c_2^*,\pi^*)}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

Game 5: $|\Pr[\text{Game } 5 = 1] - \Pr[\text{Game } 4 = 1]| \le \mathsf{Adv}^{ZK}_{aNIZK}(\kappa)$.

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $\underline{r \leftarrow \{0,1\}^{poly(\kappa)}}; pk = (pk_1, pk_2, r); sk = sk_2$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(\underline{sk}, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption queries |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0$, $m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(sk_2, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| $c^* \leftarrow \mathsf{Enc}'(pk, m_1; r^*)$ | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $\{$ $c_1^* \leftarrow \mathsf{Enc}(pk_1, m_1; w_1^*)$; | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_1; w_2^*)$; | | |
| $\underline{\pi^* \leftarrow \mathsf{P}(r, (c_1^*, c_2^*), (m_1, w_1^*, w_2^*))}\}$ | $\xrightarrow{c^*=(c_1^*,c_2^*,\underline{\pi^*})}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

Game 6: $|\Pr[\text{Game } 6 = 1] - \Pr[\text{Game } 5 = 1]| \leq \mathsf{Adv}^{sound}_{aNIZK}(\kappa)$.

| $\mathcal{C}$ | | $\mathcal{A}$ |
|---|---|---|
| $(pk_1, sk_1), (pk_2, sk_2) \leftarrow \mathsf{KeyGen}(1^\kappa)$ | | |
| $r \leftarrow \{0,1\}^{poly(\kappa)}; pk = (pk_1, pk_2, r); \underline{sk = sk_1}$ | $\xrightarrow{pk=(pk_1,pk_2,r)}$ | |
| $m'/\perp \leftarrow \mathsf{Dec}'(\underline{sk}, c)$ | $\xleftarrow{c=(c_1,c_2,\pi)}$ | decryption querie |
| $\{$ If $\mathsf{V}(r, (c_1, c_2), \pi) = 0$, $m' := \perp$ | | |
| Else $m' \leftarrow \mathsf{Dec}(sk_1, c_1)\}$ | $\xrightarrow{m'/\perp}$ | |
| $c^* \leftarrow \mathsf{Enc}'(pk, m_1; r^*)$ | $\xleftarrow{(m_0,m_1)}$ | choose $m_0, m_1$ |
| $\{$ $c_1^* \leftarrow \mathsf{Enc}(pk_1, m_1; w_1^*)$; | | |
| $c_2^* \leftarrow \mathsf{Enc}(pk_2, m_1; w_2^*)$; | | |
| $\pi^* \leftarrow \mathsf{P}(r, (c_1^*, c_2^*), (m_1, w_1^*, w_2^*))\}$ | $\xrightarrow{c^*=(c_1^*,c_2^*,\pi^*)}$ | |
| $\mathsf{Return}(b')$ | $\xleftarrow{b'}$ | guess $b'$ |

$$\left|\Pr\left[\textbf{Exp}^{CCA}_{PKE,\mathcal{A}}(1)=1\right]-\Pr\left[\textbf{Exp}^{CCA}_{PKE,\mathcal{A}}(0)=1\right]\right|$$
$$=\left|\Pr\left[\textbf{Game 6}=1\right]-\Pr\left[\textbf{Game 0}=1\right]\right|$$
$$\leq 3\mathsf{Adv}^{ZK}_{aNIZK}(\kappa)+2\mathsf{Adv}^{CPA}_{\mathsf{PKE}}(\kappa)+2\mathsf{Adv}^{sound}_{aNIZK}(\kappa)=\mathsf{negl}(\kappa).$$

**Theorem**

**The Noar-Yung scheme PKE' is NOT secure against adaptive chosen- ciphertext attacks (in general).** *More precisely, for any semantically-secure encryption scheme PKE=(KeyGen, Enc, Dec) there exists an adaptively-secure NIZK proof system (P', V') such that the resulting Noar-Yung construction is demonstrably insecure against adaptive chosen- ciphertext attacks.*

**Proof.**

Let (P, V) be an aNIZK used in Noar-Yung scheme. Then (P', V') is also an aNIZK.

- $\mathsf{P}'(r,(c_1,c_2),(m,w_1,w_2))$: Return $\mathsf{P}(r,(c_1,c_2),(m,w_1,w_2))||0$.
- $\mathsf{V}'(r,(c_1,c_2),\pi||0)$: Return $\mathsf{V}(r,(c_1,c_2),\pi)$.

If (P', V') is used in PKE', then $\mathcal{A}$ can always submit $(c_1^*,c_2^*,\pi^*||1)$ to the decryption oracle and succeed with probability 1.