SD04630230: 数字货币与区块链 **2023 年 12 月 7 日**

Note 1: Naor-Yung 通用转化

Lecturer: 钱宸

免责声明: 该讲义仅用于山东大学网络空间安全学院课程教学，尚未经过通常用于正式出版物的审查。仅在获得讲师的许可的情况下，可以在课堂外部分发。

1.1 Preliminaries

In this section, we first introduce some useful notations and formal definitions.

Notation: We denote probabilistic polynomial Turing machine by PPT. We denote that two distribution is computationally indistinguishable by $\mathcal{D}_0 \approx_c \mathcal{D}_1$.

Algorithms: We use $x \xleftarrow{\$} \mathsf{Alg}$ to present an algorithm randomly generating an output x, and $x := \mathsf{Alg}$ to present an algorithm deterministically generating an output x. We use $\mathcal{A}^{\mathsf{OAlg}(\cdot)}$, to present an algorithm with oracle access to OAlg.

Pseudo-code: We use **check** to check if the following condition is fulfilled; the algorithm aborts otherwise. We use **parse** $\mathsf{x} =: \mathsf{y}$ to parse y into the variable x.

Negligible Function: We denote the negligible functions with respect to the security parameter λ by $\mathsf{negl}(\lambda)$. We recall that a function f is negligible, if for all polynomial $\mathsf{p}(\cdot)$, there exists a λ_0 such that

$$\forall \lambda > \lambda_0 . \mathsf{f}(\lambda) < \frac{1}{\mathsf{p}(\lambda)}$$

Language: For any NP language $\mathcal{L}$, we denote a statement x in language $\mathcal{L}$ with witness w by $\mathsf{x} \in_{\mathsf{w}} \mathcal{L}$.

1.1.1 Public Key Encryption scheme

Definition 1.1 (Public Key Encryption)**.** *A public key encryption scheme consists of three* PPT *algorithms* $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$: *takes the security parameter* 1^λ *as input, and returns a public key* pk *and a secret key* sk.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{m}; \mathsf{r}) \to \mathsf{ct}$: *takes the public key* pk*, the message* m*, the randomness* r *as input, and returns a ciphertext* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{m}$: *takes the secret key* sk*, the ciphertext* ct *as input, and returns a message* m.

We also require the following properties:

- **Correctness:** *For all messages* $\mathsf{m} \in \mathcal{M}$ *in the message space, for all randomness* $\mathsf{r} \in \mathcal{R}$ *in the randomness space, and for all key pairs* $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*, we have*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m}; \mathsf{r})) = \mathsf{m}$$

- **<u>Semantic Security:</u>** PKE *is* ε-IND-CPA *secure, if for all two-stages* PPT *adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *with an internal state* st*, we first define the security games* $\mathsf{Game}^{\text{IND-CPA}_b}_{\mathsf{PKE},1^\lambda}(\mathcal{A})$ *as in Fig. 1.1. We say*

$$
\begin{array}{|l|}
\hline
\mathsf{Game}^{\text{IND-CPA}_b}_{\mathsf{PKE},1^\lambda}(\mathcal{A}):\\
\text{01 } (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)\\
\text{02 } (\mathsf{m}_0,\mathsf{m}_1,\mathsf{st}) \xleftarrow{\$} \mathcal{A}_0(\mathsf{pk})\\
\text{03 } \mathsf{r} \xleftarrow{\$} \mathcal{R};\ \mathsf{ct}_b := \mathsf{Enc}(\mathsf{pk},\mathsf{m}_b;\mathsf{r})\\
\text{04 } b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{st},\mathsf{ct}_b)\\
\text{05 } \textbf{return } b'\\
\hline
\end{array}
$$

Figure 1.1: This is the IND-CPA security game with bit $b \in \{0, 1\}$.

that the public-key encryption scheme PKE *is* IND-CPA *secure, if and only if*

$$\varepsilon = \left|\Pr\left[\mathsf{Game}^{\text{IND-CPA}_0}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Game}^{\text{IND-CPA}_1}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right]\right| \le \mathsf{negl}(\lambda).$$

- <u>IND-CCA1 **Security:**</u> PKE *is* ε-IND-CCA1 *secure, if for all two-stages* PPT *adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *with an internal state* st*, we first define the security games* $\mathsf{Game}^{\text{IND-CCA1}_b}_{\mathsf{PKE},1^\lambda}(\mathcal{A})$ *as in Fig. 1.2.*

$$
\begin{array}{|ll|}
\hline
\mathsf{Game}^{\text{IND-CCA1}_b}_{\mathsf{PKE},1^\lambda}(\mathcal{A}): & \textbf{Oracle } \mathsf{ODec}(\mathsf{ct})\\
\text{01 } (\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda) & \text{06 } m := \mathsf{Dec}(\mathsf{sk},\mathsf{ct})\\
\text{02 } (\mathsf{m}_0,\mathsf{m}_1,\mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\mathsf{ODec}(\cdot)}(\mathsf{pk}) & \text{07 } \textbf{return } m\\
\text{03 } \mathsf{r} \xleftarrow{\$} \mathcal{R};\ \mathsf{ct}_b := \mathsf{Enc}(\mathsf{pk},\mathsf{m}_b;\mathsf{r}) & \\
\text{04 } b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{st},\mathsf{ct}_b) & \\
\text{05 } \textbf{return } b' & \\
\hline
\end{array}
$$

Figure 1.2: This is the IND-CCA1 security game with bit $b \in \{0, 1\}$.

We say that the public-key encryption scheme PKE *is* IND-CCA1 *secure, if and only if*

$$\varepsilon = \left|\Pr\left[\mathsf{Game}^{\text{IND-CCA1}_0}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Game}^{\text{IND-CCA1}_1}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right]\right| \le \mathsf{negl}(\lambda).$$

1.1.2 Non-Interactive Zero-Knowledge Proof

Definition 1.2 (NIZK)**.** *Let* $\mathcal{L}$ *be an* NP *language, an adaptive non-interactive zero-knowledge proof system consists of three* PPT *algorithms* NIZK = (Setup, Prove, Ver) *with the following syntax*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$: *takes a security parameter* 1^λ *as input, and returns a common reference string* crs*.*
- $\mathsf{Prove}(\mathsf{crs},\mathsf{x},\mathsf{w}) \to \pi$: *takes a common reference string* crs*, a statement* x*, a witness* w *as input, and returns* π*.*
- $\mathsf{Ver}(\mathsf{crs},\mathsf{x},\pi) \to \{0,1\}$: *takes a common reference string* crs*, a statement* x*, and a proof* π *as input, and returns a result bit* $b \in \{0, 1\}$*.*

We require the following properties:

- **Completeness:** *For all statements* $\mathsf{x} \in_{\mathsf{w}} \mathcal{L}$*, for all honestly generated common reference string* $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*, we have*

$$\mathsf{Ver}(\mathsf{crs}, \mathsf{x}, \mathsf{Prove}(\mathsf{crs}, \mathsf{x}, \mathsf{w})) = 1.$$

- **Soundness:** NIZK *is* $\varepsilon_{\mathsf{snd}}$*-sound, if for all* PPT *adversary* $\mathcal{A}$*, we have*

$$\Pr\left[\begin{array}{l} \mathsf{Ver}(\mathsf{crs}, \mathsf{x}, \pi) = 1 \\ \wedge \mathsf{x} \notin \mathcal{L} \end{array} \;\middle|\; \begin{array}{l} \varepsilon_{\mathsf{snd}} = \mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda) \\ \pi \xleftarrow{\$} \mathcal{A}(\mathsf{crs}, \mathsf{x}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

- **Zero-Knowledge:** NIZK *is* $\varepsilon_{\mathsf{zk}}$*-zero-knowledge, if for all* PPT *adversary* $\mathcal{A}$ *with running time* t_{zk}*,there exists a two-stage* PPT *algorithm* $\mathsf{Sim} = (\mathsf{SimSetup}, \mathsf{SimProve})$ *with the following syntax:*
  - $\mathsf{SimSetup}(1^\lambda) \to (\mathsf{crs}, \mathsf{td})$: *takes a security parameter* 1^λ *as input, and returns a* crs *and a simulation trapdoor* td*.*
  - $\mathsf{SimProve}(\mathsf{crs}, \mathsf{x}, \mathsf{td}) \to \pi$: *takes a* crs*, a statement* x*, and a trapdoor* td *as input, and returns a simulated proof* π*.*

  We require that the simulated $\mathsf{SimSetup}$ *and* $\mathsf{SimProve}$ *are indistinguishable from the real one for any* PPT *adversary. More formally, for all* PPT *adversaries, the following two games are indistinguishable: We require that the following requirement holds*

$\mathsf{Game}^{Real}_{\mathsf{PKE},1^\lambda}(\mathcal{A})$:	$\mathsf{Game}^{Sim}_{\mathsf{PKE},1^\lambda}(\mathcal{A})$:
01 $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$	03 $\mathsf{crs} \xleftarrow{\$} \mathsf{SimSetup}(1^\lambda)$
02 **return** $\mathcal{A}^{\mathsf{OProve}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs})$	04 **return** $\mathcal{A}^{\mathsf{OSimProve}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs})$

Figure 1.3: This is the indistinguishability game between the real and simulated worlds. Note that $\mathsf{OSimProve}(\mathsf{crs}, \mathsf{x}, \mathsf{w})$ returns $\mathsf{SimProve}(\mathsf{crs}, \mathsf{x}, \mathsf{td})$ without using w.

$$\varepsilon_{\mathsf{zk}} = \left|\Pr\left[\mathsf{Game}^{Real}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Game}^{Sim}_{\mathsf{PKE},1^\lambda}(\mathcal{A}) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

1.2 Naor-Yung CCA1 construction

Let PKE be a ε-IND-CPA public-key encryption scheme, and NIZK be a $(\varepsilon_{\mathsf{zk}}, \varepsilon_{\mathsf{snd}})$-adaptive non-interactive zero-knowledge proof system. We recalled the Naor-Yung CCA1 construction that we saw during the lecture.

As in [NY90], we give the detailed construction as in Fig. 1.4

Theorem 1.3 ([NY90])**.** *The public-key encryption scheme given in Fig. 1.4 is* ε'-IND-CCA1 *secure, with*

$$\varepsilon' \leq 2\varepsilon + 4\varepsilon_{\mathsf{zk}} + 2\varepsilon_{\mathsf{snd}}$$

Proof. We give the proof following a sequence of hybrid games $(\mathbf{G}_0, \ldots, \mathbf{G}_6)$, in which $\mathbf{G}_0 = \mathsf{Game}^{\text{IND-CCA1}_0}_{\mathsf{PKE},1^\lambda}$ and $\mathbf{G}_6 = \mathsf{Game}^{\text{IND-CCA1}_1}_{\mathsf{PKE},1^\lambda}$. By arguing that $\mathbf{G}_i \approx_c \mathbf{G}_{i+1}$ for all $i \in \{0, \ldots, 5\}$, we complete the proof.

We give the detailed hybrid game description as follows. We denote by pr_i the probability that the adversary outputs 1 in the game $\mathbf{G}_i$. Note that with the above notation, we only need to prove that $|\mathsf{pr}_0 - \mathsf{pr}_6| \leq \mathsf{negl}(\lambda)$ We summarize all hybrid games in Fig. 1.5.

Alg $\mathsf{Setup}(1^\lambda)$:

$(\mathsf{pk}_0, \mathsf{sk}_0) \xleftarrow{\$} \mathsf{PKE.Setup}(1^\lambda)$
$(\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \mathsf{PKE.Setup}(1^\lambda)$
$\mathsf{crs} \xleftarrow{\$} \mathsf{NIZK.Setup}(1^\lambda)$
$\mathsf{pk} := (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{crs});\ \mathsf{sk} := \mathsf{sk}_0$

Alg $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$:

parse $\mathsf{sk}_0 =: \mathsf{sk};\ (\mathsf{ct}_0, \mathsf{ct}_1, \pi) =: \mathsf{ct}$
check $\mathsf{NIZK.Ver}(\mathsf{crs}, (\mathsf{ct}_0, \mathsf{ct}_1), \pi) = 1$
$\mathsf{m} := \mathsf{PKE.Dec}(\mathsf{sk}_0, \mathsf{ct}_0)$
return m

Alg $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$:

parse $(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{crs}) =: \mathsf{pk}$
$\mathsf{r}_0, \mathsf{r}_1 \xleftarrow{\$} \mathcal{R}$
$\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_0, \mathsf{m}; \mathsf{r}_0)$
$\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_1, \mathsf{m}; \mathsf{r}_1)$
$\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{crs}, (\mathsf{ct}_0, \mathsf{ct}_1), (\mathsf{m}, \mathsf{r}_0, \mathsf{r}_1))$
return $(\mathsf{ct}_0, \mathsf{ct}_1, \pi)$

Figure 1.4: This is CCA1 Naor-Yung construction.

$\mathbf{G}_0$: This is the initial security game with the challenge bit $b = 0$.

$\mathbf{G}_1$: This game is the same as in $\mathbf{G}_0$ except that the challenger uses Sim for simulating the proof instead of honestly generating the zero-knowledge proofs.

Since the only difference is whether using the simulator to generate the proofs, we have

$$|\mathsf{pr}_0 - \mathsf{pr}_1| \leq \varepsilon_{\mathsf{zk}}.$$

$\mathbf{G}_2$: In this game, we change the generation of ct_1. In $\mathbf{G}_2$, ct_1 is an encryption of m_1 instead of m_0.

Notice that, the adversary $\mathcal{A}$ has only access of $\mathsf{ODec}(\cdot)$ which uses only sk_0. Therefore, any adversary $\mathcal{B}$ which can distinguish $\mathbf{G}_2$ from $\mathbf{G}_1$ can also break the IND-CPA security of the underlying encryption scheme. Thus, we have

$$|\mathsf{pr}_2 - \mathsf{pr}_1| \leq \varepsilon.$$

$\mathbf{G}_3$: In $\mathbf{G}_3$, we switch the the decryption key from sk_0 to sk_1.

To analyze the probability of distinguishing $\mathbf{G}_2$ from $\mathbf{G}_3$, we define a bad event Bad. Bad happens when the adversary submits a ciphertext $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$ to the decryption oracle with $\mathsf{Dec}(\mathsf{sk}_0, \mathsf{ct}_0) \neq \mathsf{Dec}(\mathsf{sk}_1, \mathsf{ct}_1)$ and $\mathsf{Ver}(\mathsf{crs}, (\mathsf{ct}_0, \mathsf{ct}_1), \pi) = 1$. Our first observation is that the adversary's view is different in $\mathbf{G}_2$ and $\mathbf{G}_3$ only if Bad happens in $\mathbf{G}_2$. Therefore, we have $|\mathsf{pr}_2 - \mathsf{pr}_3| \leq \Pr[\mathsf{Bad}]$. Our second observation is that Bad can also happen in $\mathbf{G}_0$, $\mathbf{G}_1$ and $\mathbf{G}_2$, we denote these event by Bad_i with $i \in \{0, 1, 2\}$. We can have the following analysis:

- In $\mathbf{G}_0$ the crs is generated by an honest $\mathsf{NIZK.Setup}$ algorithm, we have $\mathsf{Bad}_0 \leq \varepsilon_{\mathsf{snd}}$.
- Since the probability of distinguishing $\mathbf{G}_0$ and $\mathbf{G}_1$ is bounded by $\varepsilon_{\mathsf{zk}}$, and Bad can be detected by the adversary himself, we have

$$\Pr[\mathsf{Bad}_1] \leq \Pr[\mathsf{Bad}_0] + \varepsilon_{\mathsf{zk}} = \varepsilon_{\mathsf{snd}} + \varepsilon_{\mathsf{zk}}$$

- The change in $\mathbf{G}_2$ happens after all decryption queries. Therefore, we have $\Pr[\mathsf{Bad}_2] = \Pr[\mathsf{Bad}_1] \leq \varepsilon_{\mathsf{snd}} + \varepsilon_{\mathsf{zk}}$.

In summary, we have

$$|\mathsf{pr}_3 - \mathsf{pr}_2| \le \varepsilon_{\mathsf{snd}} + \varepsilon_{\mathsf{zk}}.$$

$\mathbf{G}_4$: In the game $\mathbf{G}_4$, we change the message m_0 in ct_0 to ct_1.

Similar to the argument in $\mathbf{G}_2$, now the decryption oracle does not use sk_0, thus we can bound the probability of distinguishing $\mathbf{G}_3$ and $\mathbf{G}_4$ by the IND-CPA security of PKE. We have

$$|\mathsf{pr}_4 - \mathsf{pr}_3| \le \varepsilon.$$

$\mathbf{G}_5$: In the game $\mathbf{G}_5$, we change back the decryption oracle using sk_0. Similarly to $\mathbf{G}_3$, we have

$$|\mathsf{pr}_5 - \mathsf{pr}_4| \le \varepsilon_{\mathsf{snd}} + \varepsilon_{\mathsf{zk}}.$$

$\mathbf{G}_6$: In $\mathbf{G}_6$, we use the $(\mathsf{Setup}, \mathsf{Prove})$ instead of $(\mathsf{SimSetup}, \mathsf{SimProve})$. Similar to $\mathbf{G}_1$, we have

$$|\mathsf{pr}_6 - \mathsf{pr}_5| \le \varepsilon_{\mathsf{zk}}.$$

We can notice that $\mathbf{G}_6$ is exactly the same as $\mathsf{Game}^{\text{IND-CCA1}_1}_{\mathsf{PKE},1^\lambda}$. By the triangle inequality we have

$$\begin{aligned} |\mathsf{pr}_6 - \mathsf{pr}_0| &\le |\mathsf{pr}_6 - \mathsf{pr}_5| + |\mathsf{pr}_5 - \mathsf{pr}_4| + |\mathsf{pr}_4 - \mathsf{pr}_3| + |\mathsf{pr}_3 - \mathsf{pr}_2| + |\mathsf{pr}_2 - \mathsf{pr}_1| + |\mathsf{pr}_1 - \mathsf{pr}_0| \\ &\le 2\varepsilon + 4\varepsilon_{\mathsf{zk}} + 2\varepsilon_{\mathsf{snd}}. \end{aligned}$$

$\underline{\mathsf{Game}^{\text{IND-CCA1}}_{\mathsf{PKE},1^\lambda}(\mathcal{A})}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \xleftarrow{\$} \mathsf{PKE.Setup}(1^\lambda)$
$(\mathsf{pk}_1, \mathsf{sk}_1) \xleftarrow{\$} \mathsf{PKE.Setup}(1^\lambda)$
$\mathsf{crs} \xleftarrow{\$} \mathsf{NIZK.Setup}(1^\lambda)$ $\quad //\mathbf{G}_{0,6}$
$(\mathsf{crs}, \mathsf{td}) \xleftarrow{\$} \mathsf{NIZK.SimSetup}(1^\lambda)$ $\quad //\mathbf{G}_{1-5}$
$\mathsf{pk} := (\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{crs});\ \mathsf{sk} := \mathsf{sk}_0$
$(\mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_0^{\mathsf{ODec}(\cdot)}(\mathsf{pk})$
$\mathsf{r}_0, \mathsf{r}_1 \xleftarrow{\$} \mathcal{R}$
$\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_0, \mathsf{m}_0; \mathsf{r}_0)$ $\quad \mathbf{G}_{0-3}$
$\mathsf{ct}_0 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_0, \mathsf{m}_1; \mathsf{r}_0)$ $\quad \mathbf{G}_{4-6}$
$\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_1, \mathsf{m}_0; \mathsf{r}_1)$ $\quad //\mathbf{G}_{0-1}$
$\mathsf{ct}_1 \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}_1, \mathsf{m}_1; \mathsf{r}_1)$ $\quad //\mathbf{G}_{2-6}$
$\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{crs}, (\mathsf{ct}_0, \mathsf{ct}_1), (\mathsf{m}, \mathsf{r}_0, \mathsf{r}_1))$ $\quad //\mathbf{G}_{0,6}$
$\pi \xleftarrow{\$} \mathsf{SimProve}(\mathsf{crs}, \mathsf{td}, (\mathsf{ct}_0, \mathsf{ct}_1))$ $\quad //\mathbf{G}_{1-5}$
$\mathsf{ct} := (\mathsf{ct}_0, \mathsf{ct}_1, \pi)$
$b' \xleftarrow{\$} \mathcal{A}_1(\mathsf{st}, \mathsf{ct})$
return b'

$\underline{\textbf{Oracle}\ \mathsf{ODec}(\mathsf{sk}, \mathsf{ct}):}$

parse $\mathsf{sk}_0 =: \mathsf{sk};\ (\mathsf{ct}_0, \mathsf{ct}_1, \pi) =: \mathsf{ct}$
check $\mathsf{NIZK.Ver}(\mathsf{crs}, (\mathsf{ct}_0, \mathsf{ct}_1), \pi) = 1$
$\mathsf{m} := \mathsf{PKE.Dec}(\mathsf{sk}_0, \mathsf{ct}_0)$ $\quad //\mathbf{G}_{0-2,5-6}$
$\mathsf{m} := \mathsf{PKE.Dec}(\mathsf{sk}_1, \mathsf{ct}_0)$ $\quad //\mathbf{G}_{3-4}$
return m

Figure 1.5: This is a summary of all hybrid games. The code line ends with $//\mathbf{G}_i$ only appears in security game $\mathbf{G}_i$.

□

1.3 Remarks

There have been several extensions of the Naor-Yung transform since its first introduction. We give a non-exhaustive list of the improvements.

- We can notice that in the Naor-Yung encryption scheme the proof of plaintext equality is only verified by the decryptor. Therefore, we can replace the zero-knowledge proof system by a designated-verifier zero-knowledge proof system (DV-NIZK). In DV-NIZK, the proof is not publically verifiable, to check a proof we need a secret key. DV-NIZK is also called a hash proof system if the proof is deterministic. In fact, historically the first CCA2 encryption scheme without random oracle Cramer-Shoup can be viewed as Naor-Yung instantiated with a special form of hash proof system.
- We only give a proof of the CCA1 version of Naor-Yung transform, but this transform is more powerful. In fact, a few years after the first Naor-Yung scheme, people found out that if the underlying zero-knowledge proof is simulation sound, which implies non malleability, then the resulting encryption scheme can achieve CCA2 security.

References

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, MD, USA, May 14–16, 1990. ACM Press.