

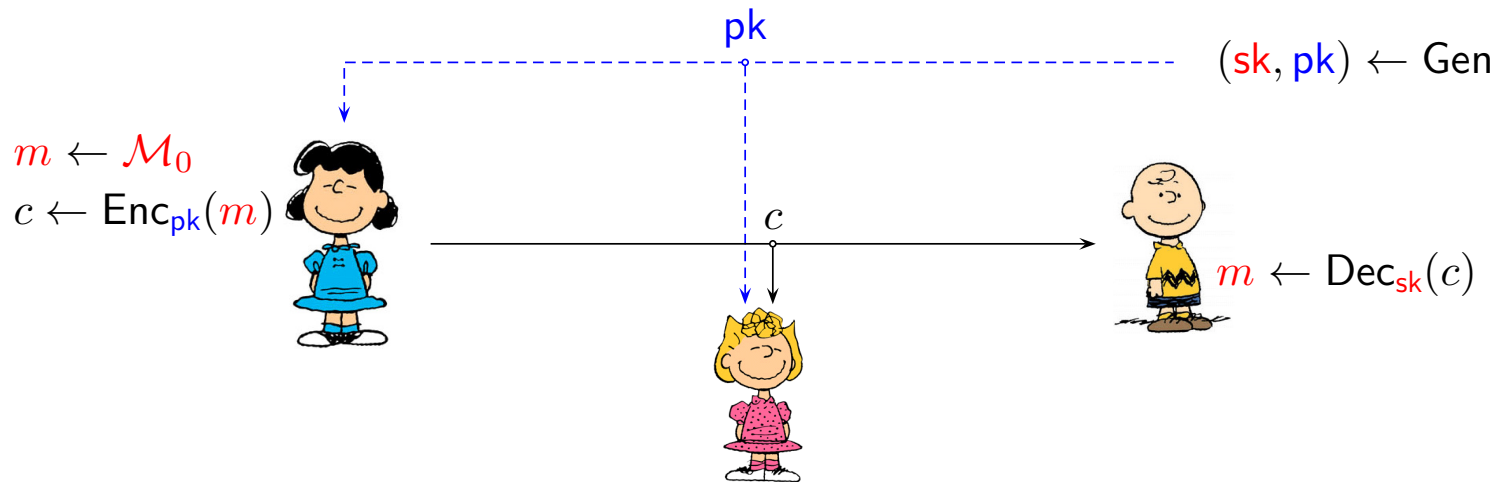
MTAT.07.003 CRYPTOLOGY II

# **Public Key Cryptosystems**

Sven Laur  
University of Tartu

# Formal Syntax

# Public key cryptosystem



- ▷ A randomised *key generation algorithm* outputs a *secret key*  $sk$  and a *public key*  $pk$ . A public key gives ability to encrypt messages.
- ▷ A randomised *encryption algorithm*  $\text{Enc}_{pk} : \mathcal{M} \rightarrow \mathcal{C}$  takes in a *plaintext* and outputs a corresponding *ciphertext*.
- ▷ A *decryption algorithm*  $\text{Dec}_{sk} : \mathcal{C} \rightarrow \mathcal{M} \cup \{\perp\}$  recovers the plaintext or a special abort symbol  $\perp$  to indicate invalid ciphertexts.

## Example. RSA-1024 cryptosystem

### Key generation Gen:

1. Choose uniformly 512-bit prime numbers  $p$  and  $q$ .
2. Compute  $N = p \cdot q$  and  $\phi(N) = (p - 1)(q - 1)$ .
3. Choose uniformly  $e \leftarrow \mathbb{Z}_{\phi(N)}^*$  and set  $d = e^{-1} \bmod \phi(N)$ .
4. Output **sk** =  $(p, q, e, d)$  and **pk** =  $(N, e)$ .

### Encryption and decryption:

$$\mathcal{M} = \mathbb{Z}_N, \quad \mathcal{C} = \mathbb{Z}_N, \quad \mathcal{R} = \emptyset$$

$$\text{Enc}_{\text{pk}}(m) = m^e \bmod N \quad \text{Dec}_{\text{sk}}(c) = c^d \bmod N .$$

# Semantic Security

## IND-CPA security

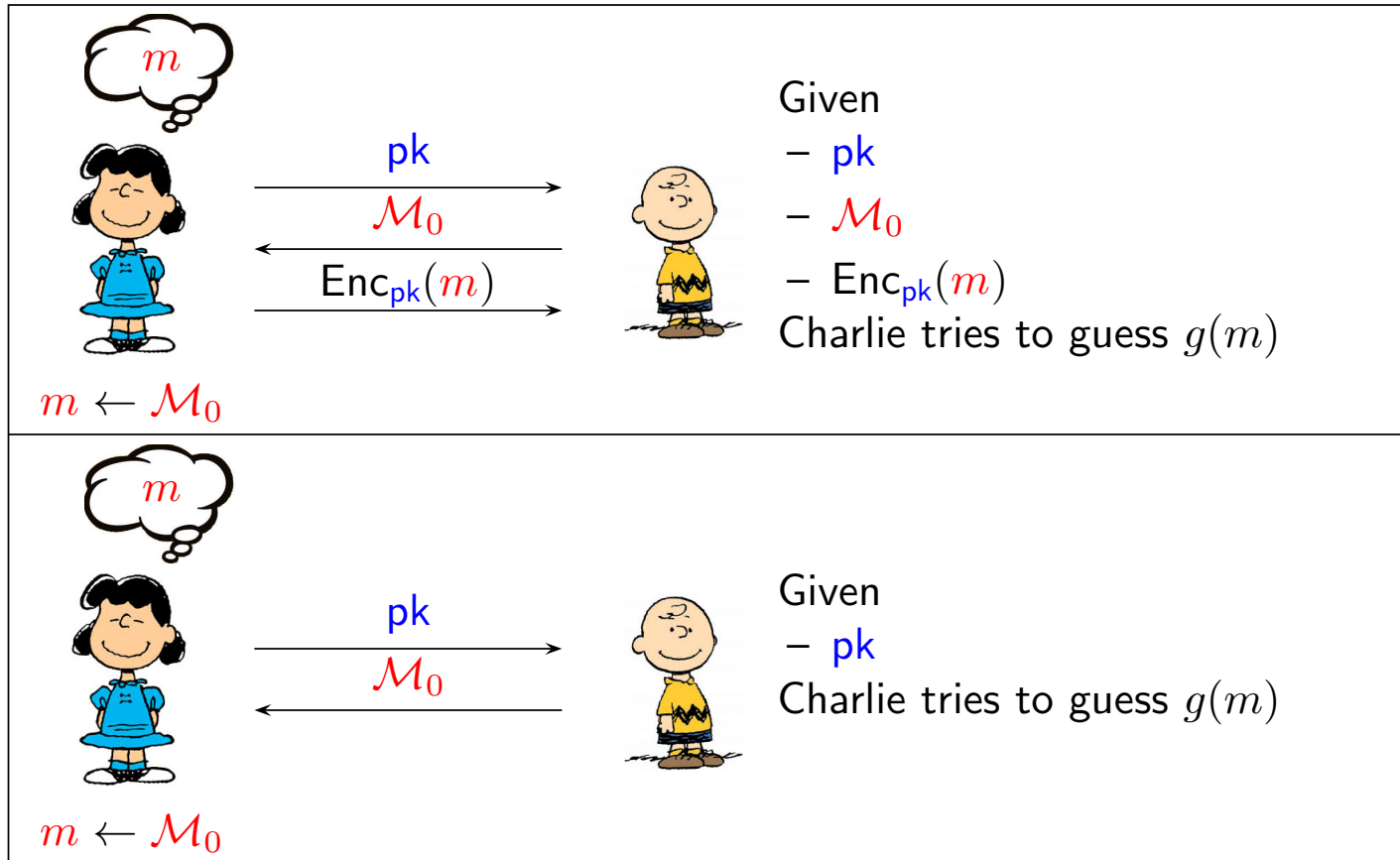
As a potential adversary  $\mathcal{A}$  can influence which messages are encrypted, we must model the corresponding effects in our attack model. A cryptosystem  $(\text{Gen}, \text{Enc}, \text{Dec})$  is  $(t, \varepsilon)$ -IND-CPA secure if for all  $t$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}^{\text{ind-cpa}}(\mathcal{A}) = |\Pr [\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr [\mathcal{G}_1^{\mathcal{A}} = 1]| \leq \varepsilon ,$$

where the security games are defined as follows

 $\mathcal{G}_0^{\mathcal{A}}$ 
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}(\text{pk}) \\ \text{return } \mathcal{A}(\text{Enc}_{\text{pk}}(m_0)) \end{array} \right.$$
 $\mathcal{G}_1^{\mathcal{A}}$ 
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}(\text{pk}) \\ \text{return } \mathcal{A}(\text{Enc}_{\text{pk}}(m_1)) \end{array} \right.$$

# Semantic security against adaptive influence



## Formal definition

Consider following games:

$$\mathcal{G}_0^{\mathcal{A}} \left[ \begin{array}{l} (\textcolor{red}{sk}, \textcolor{blue}{pk}) \leftarrow \text{Gen} \\ \mathcal{M}_0 \leftarrow \mathcal{A}(\textcolor{blue}{pk}) \\ m \leftarrow \mathcal{M}_0 \\ c \leftarrow \text{Enc}_{\textcolor{blue}{pk}}(m) \\ \textbf{return } [g(m) \stackrel{?}{=} \mathcal{A}(c)] \end{array} \right]$$

$$\mathcal{G}_1^{\mathcal{A}} \left[ \begin{array}{l} (\textcolor{red}{sk}, \textcolor{blue}{pk}) \leftarrow \text{Gen} \\ \mathcal{M}_0 \leftarrow \mathcal{A}(\textcolor{blue}{pk}) \\ m \leftarrow \mathcal{M}_0, \overline{m} \leftarrow \mathcal{M}_0 \\ \overline{c} \leftarrow \text{Enc}_{\textcolor{blue}{pk}}(\overline{m}) \\ \textbf{return } [g(m) \stackrel{?}{=} \mathcal{A}(\overline{c})] \end{array} \right]$$

The true guessing advantage is

$$\text{Adv}_g^{\text{sem}}(\mathcal{A}) = \Pr[\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr[\mathcal{G}_1^{\mathcal{A}} = 1] \ .$$



## IND-CPA $\Rightarrow$ SEM-CPA

**Theorem.** Assume that  $g$  is a  $t_g$ -time function and it is always possible to obtain a sample from  $\mathcal{M}_0$  in time  $t_m$ . Now if the cryptosystem is  $(t, \varepsilon)$ -IND-CPA secure, then for all  $(t - t_g - 2t_m)$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}_g^{\text{sem}}(\mathcal{A}) \leq \varepsilon .$$

Note that

- ▷ The function  $g$  might be randomised.
- ▷ The function  $g$  must be efficiently computable.
- ▷ The distribution  $\mathcal{M}_0$  must be efficiently samplable.

# An Example of IND-CPA Secure Cryptosystem

# ElGamal cryptosystem

Combine the Diffie-Hellman key exchange protocol

**Alice**

$$x \leftarrow \mathbb{Z}_{|\mathbb{G}|}$$

$$\xrightarrow{y=g^x}$$

$$\xleftarrow{g^k}$$

$$g^{xk} = (g^k)^x$$

**Bob**

$$k \leftarrow \mathbb{Z}_{|\mathbb{G}|}$$

$$g^{xk} = (g^x)^k$$

with one-time pad by multiplication using in  $\mathbb{G} = \langle g \rangle$  as encoding rule

$$\text{Enc}_{\text{pk}}(m) = (g^k, m \cdot g^{xk}) = (g^k, m \cdot y^k) \quad \text{for all elements } m \in \mathbb{G}$$

with a public key  $\text{pk} = y = g^x$  and a secret key  $\text{sk} = x$ .

# Decisional Diffie-Hellman Assumption (DDH)

**Definition.** We say that a  $q$ -element multiplicative group  $\mathbb{G}$  is  $(t, \varepsilon)$ -Decisional Diffie-Hellman group if for all  $t$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{A}) = |\Pr[\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr[\mathcal{G}_1^{\mathcal{A}} = 1]| \leq \varepsilon$$

where the security games are defined as follows

$$\mathcal{G}_0^{\mathcal{A}}$$
$$\left[ \begin{array}{l} x, k \leftarrow \mathbb{Z}_q \\ \mathbf{return} \mathcal{A}(g, g^x, g^k, g^{xk}) \end{array} \right]$$
$$\mathcal{G}_1^{\mathcal{A}}$$
$$\left[ \begin{array}{l} x, k, c \leftarrow \mathbb{Z}_q \\ \mathbf{return} \mathcal{A}(g, g^x, g^k, g^c) \end{array} \right]$$

The Diffie-Hellman key exchange protocol is secure under the DDH assumption, as an attacker cannot distinguish values  $g^{xk}$  and  $g^c$ .

## DDH $\Rightarrow$ IND-CPA

**Theorem.** Let  $\mathbb{G}$  be a  $(t, \varepsilon)$ -DDH group. Then the corresponding instantiation of the ElGamal cryptosystem is  $(t, 2\varepsilon)$ -IND-CPA secure.

Let  $\mathcal{B}$  be good against IND-CPA games. Then we can consider the following algorithm  $\mathcal{A}$ :

1. Given  $(g, g^x, g^k, z)$ , set  $\text{pk} = g^x$  and  $(m_0, m_1) \leftarrow \mathcal{B}(\text{pk})$ .
2. Toss a fair coin  $b \leftarrow \{0, 1\}$  and set  $c = (g^k, m_b z)$ .
3. If  $b \stackrel{?}{=} \mathcal{A}(c)$  return 1 else output 0.

We argue that this is a good strategy to win the DDH game:

- In the game  $\mathcal{G}_0$ , we simulate the bit guessing game.
- In the game  $\mathcal{G}_1$ , the guess **guess** is independent from  $b$ .

## Hybrid encryption

Assume that  $(\text{Gen}, \text{Enc}, \text{Dec})$  is a IND-CPA secure public key cryptosystem and  $(\text{Gen}^\circ, \text{Enc}^\circ, \text{Dec}^\circ)$  is a IND-CPA secure symmetric key cryptosystem. Then we can construct a hybrid IND-CPA secure cryptosystem.

**Key generation.** Output the original secret and public key  $(\text{sk}, \text{pk}) \leftarrow \text{Gen}$ .

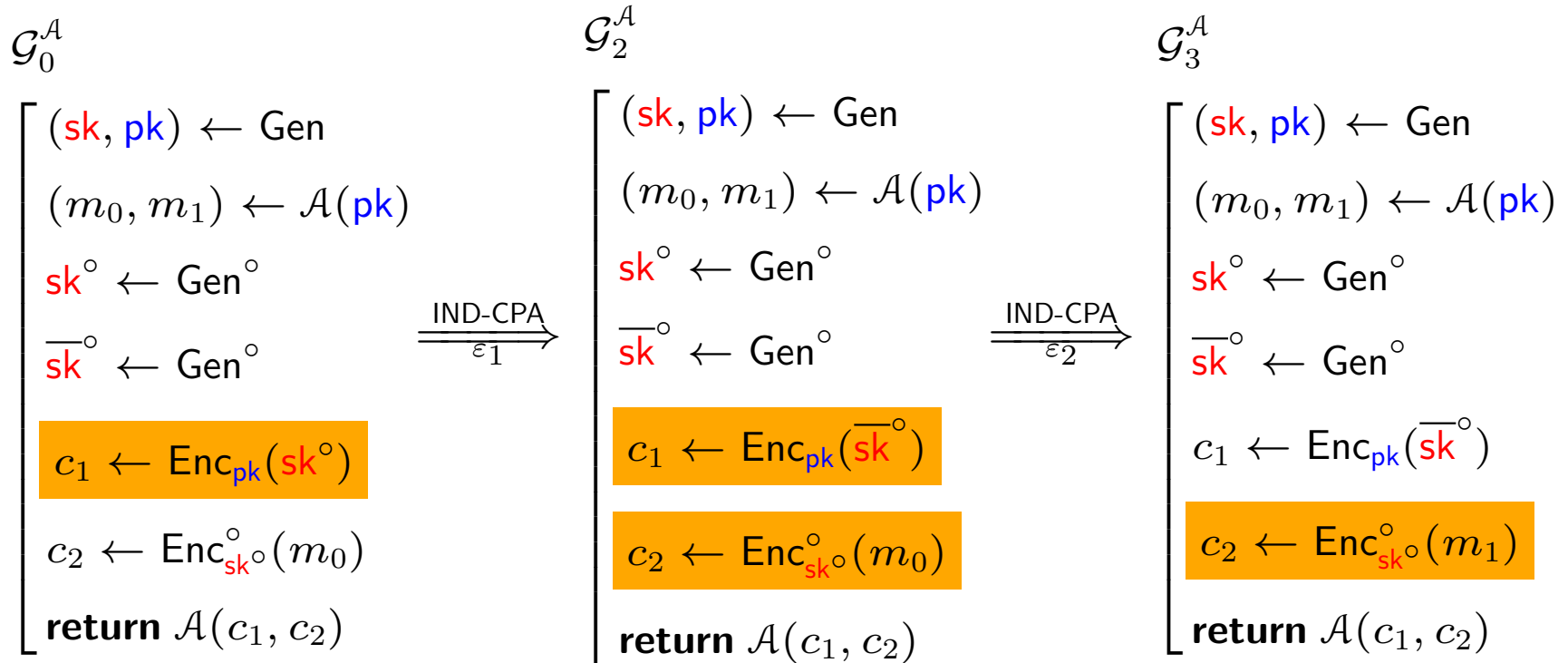
**Encryption.** For  $m \in \mathcal{M}^\circ$  generate a session key  $\text{sk}^\circ \leftarrow \text{Gen}^\circ$  and compute

$$\text{Enc}_{\text{pk}}^*(m) = (\text{Enc}_{\text{pk}}(\text{sk}^\circ), \text{Enc}_{\text{sk}^\circ}^\circ(m))$$

**Decryption.** Given  $(c_1, c_2)$  compute  $\text{sk}^\circ \leftarrow \text{Dec}_{\text{sk}}(c_1)$  and output  $\text{Dec}_{\text{sk}^\circ}^\circ(c_2)$ .

**Theorem.** The hybrid encryption is  $(t, 2\varepsilon_1 + \varepsilon_2)$ -IND-CPA secure if the public key cryptosystem is  $(t, \varepsilon_1)$ -IND-CPA secure and the symmetric key cryptosystem is  $(t, \varepsilon_2)$ -IND-CPA secure.

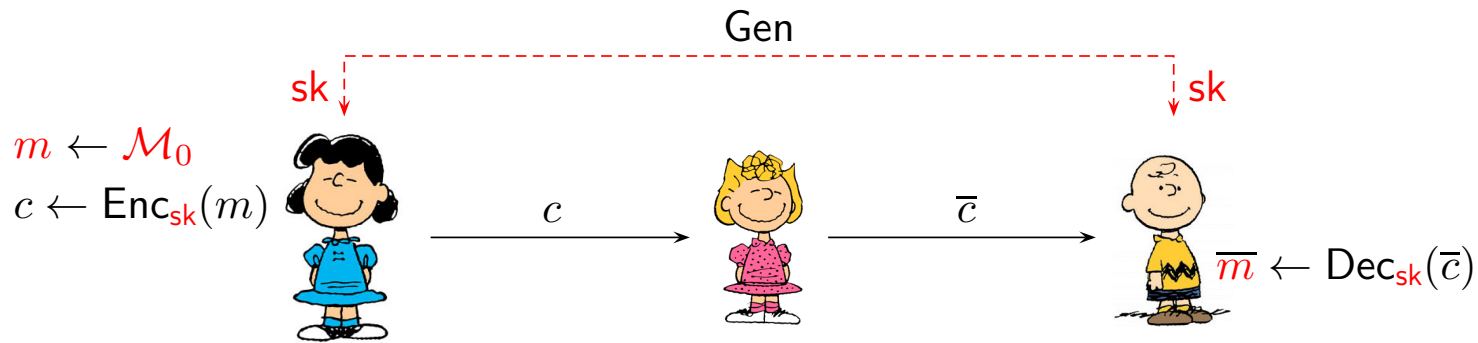
## Corresponding proof



# Ciphertext modification attacks

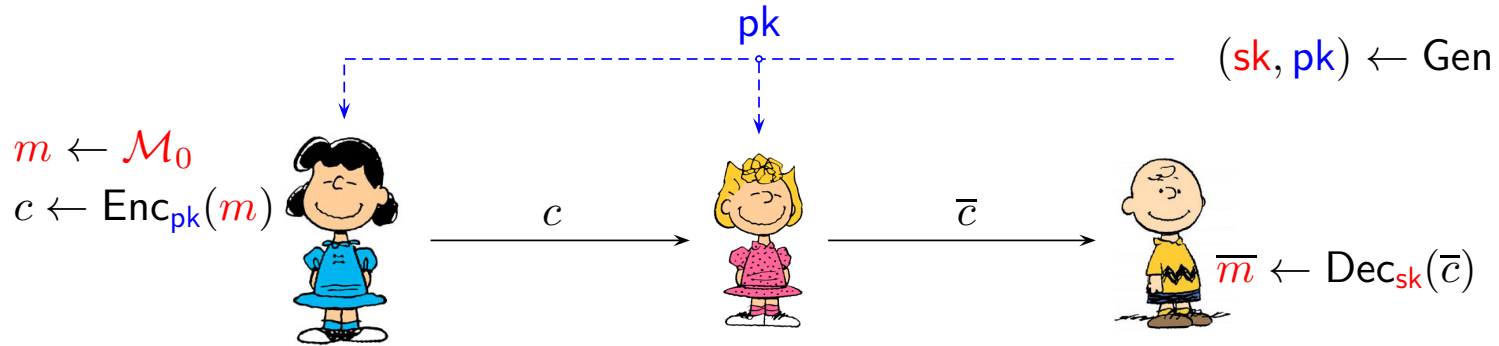


# Symmetric key cryptosystem



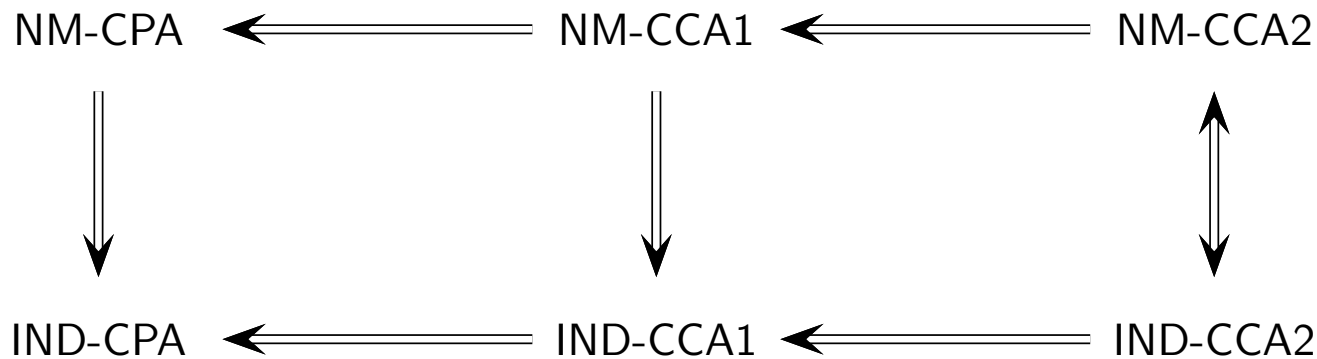
- ▷ A malicious participant may control the communication network and alter the ciphertexts to bypass various security checks.
- ▷ A malicious participant may interact with a key holder and use him or her as an *encryption* or *decryption* oracle.
- ▷ A non-malleable encryption detects modifications in ciphertexts (*authenticated encryption*) or assures that  $m$  and  $\bar{m}$  are unrelated.

# Public key cryptosystem



- ▷ Active attacks are similar for public key cryptosystems. Except there is no need for encryption oracle, since the adversary knows the public key.
- ▷ Common cryptosystems detect tampered ciphertexts with high probability and thus the adversary cannot use the decryption oracle for useful tasks.

## Homological classification



The figure above depicts the relations among various security properties of public key cryptosystems. In practice one normally needs:

- ▷ semantic security that follows IND-CPA security,
- ▷ safety against improper usage that follows from IND-CCA1 security,
- ▷ non-malleability of ciphertexts that follows from NM-CPA security.

## **Safety against improper usage**

Cleverly crafted ciphertexts or ciphertext-like messages may provide relevant information about the secret key or even reveal the secret key.

Such attacks naturally occur in:

- ▷ smart card cracking (Satellite TV, TPM-modules, ID cards)
- ▷ authentication protocols (challenge-response protocols)
- ▷ side-channel attacks (timing information, encryption failures)

### **Minimal security level:**

- ▷ Attacks reveal information only about currently known ciphertexts.

### **Affected cryptosystems:**

- Rabin cryptosystem, some versions of NTRU cryptosystem, etc.

## IND-CCA1 security

A cryptosystem is  $(t, \varepsilon)$ -IND-CCA1 secure if for all  $t$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}^{\text{ind-cca1}}(\mathcal{A}) = |\Pr [\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr [\mathcal{G}_1^{\mathcal{A}} = 1]| \leq \varepsilon ,$$

where the security games are defined as follows

$$\mathcal{G}_0^{\mathcal{A}}$$
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \textbf{return } \mathcal{A}(\text{Enc}_{\text{pk}}(m_0)) \end{array} \right.$$
$$\mathcal{G}_1^{\mathcal{A}}$$
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \textbf{return } \mathcal{A}(\text{Enc}_{\text{pk}}(m_1)) \end{array} \right.$$

and the oracle  $\mathcal{O}_1$  serves decryption queries, i.e.,  $\mathcal{O}_1(c) = \text{Dec}_{\text{sk}}(c)$ .

# Rabin cryptosystem

## Key generation Gen:

1. Choose uniformly 512-bit prime numbers  $p$  and  $q$ .
2. Compute  $N = p \cdot q$  and  $\phi(N) = (p - 1)(q - 1)$ .
3. Output  $\text{sk} = (p, q)$  and  $\text{pk} = N$ .

## Encryption and decryption:

$$\mathcal{M} = \mathbb{Z}_N, \quad \mathcal{C} = \mathbb{Z}_N, \quad \mathcal{R} = \emptyset$$
$$\text{Enc}_{\text{pk}}(m) = m^2 \mod N \quad \text{Dec}_{\text{sk}}(c) = \sqrt{c} \mod N .$$

## Lunchtime attack

1. Choose  $x \leftarrow \mathbb{Z}_N^*$  and set  $c \leftarrow x^2 \bmod N$ .
2. Compute decryption  $\bar{x} \leftarrow \mathcal{O}_1(c)$ .
3. If  $\bar{x} \neq \pm x$  then
  - Compute nontrivial square root  $\xi = \bar{x} \cdot x^{-1} \bmod N$
  - Compute a nontrivial factors  $p \leftarrow \gcd(N, \xi + 1)$  and  $q = N/p$ .
  - Output a secret key **sk** =  $(p, q)$ .
4. Continue from Step 1.

### Efficiency analysis

- Each iteration fails with probability  $\frac{1}{2}$ .
- With 80 decryption queries the failure probability is  $2^{-80}$ .

## IND-CCA2 security

A cryptosystem is  $(t, \varepsilon)$ -IND-CCA2 secure if for all  $t$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}^{\text{ind-cca1}}(\mathcal{A}) = |\Pr [\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr [\mathcal{G}_1^{\mathcal{A}} = 1]| \leq \varepsilon ,$$

where the security games are defined as follows

 $\mathcal{G}_0^{\mathcal{A}}$ 
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \text{return } \mathcal{A}^{\mathcal{O}_2(\cdot)}(\text{Enc}_{\text{pk}}(m_0)) \end{array} \right.$$
 $\mathcal{G}_1^{\mathcal{A}}$ 
$$\left[ \begin{array}{l} (\text{sk}, \text{pk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \text{return } \mathcal{A}^{\mathcal{O}_2(\cdot)}(\text{Enc}_{\text{pk}}(m_1)) \end{array} \right.$$

and oracles  $\mathcal{O}_1$  and  $\mathcal{O}_2$  serve decryption queries, i.e.,  $\mathcal{O}_1(c) = \text{Dec}_{\text{sk}}(c)$  and  $\mathcal{O}_2(c) = \text{Dec}_{\text{sk}}(c)$  for all non-challenge ciphertexts.



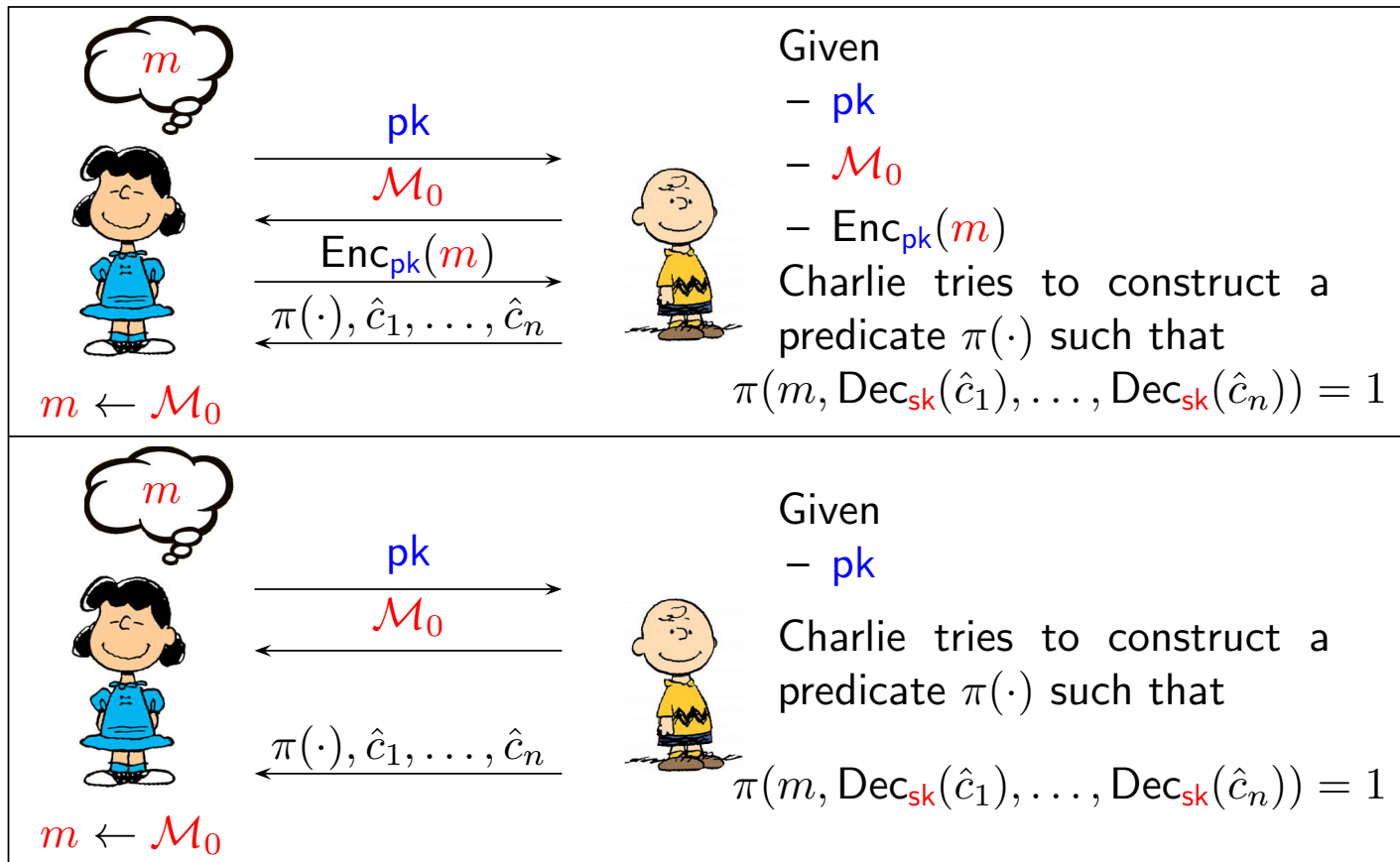
## IND-CCA2 secure cryptosystems

All known IND-CCA2 secure cryptosystems include a non-interactive proof that the creator of the ciphertexts  $c$  knows the corresponding message  $m$ :

- the RSA-OAEP cryptosystem in the random oracle model,
- the Cramer-Shoup cryptosystem in standard model,
- the Kurosawa-Desmedt key encapsulation scheme.

Non-malleability

# NM-CPA security



## Formal definition

$\mathcal{G}_0^{\mathcal{A}}$

```

[ (sk, pk) ← Gen
   $\mathcal{M}_0 \leftarrow \mathcal{A}(\text{pk})$ 
   $m \leftarrow \mathcal{M}_0$ 
   $c \leftarrow \text{Enc}_{\text{pk}}(m)$ 
   $\pi(\cdot), \hat{c}_1, \dots, \hat{c}_n \leftarrow \mathcal{A}(c)$ 
  if  $c \in \{\hat{c}_1, \dots, \hat{c}_n\}$  then return 0
  return  $\pi(m, \text{Dec}_{\text{sk}}(\hat{c}_1), \dots)$ 

```

$\mathcal{G}_1^{\mathcal{A}}$

```

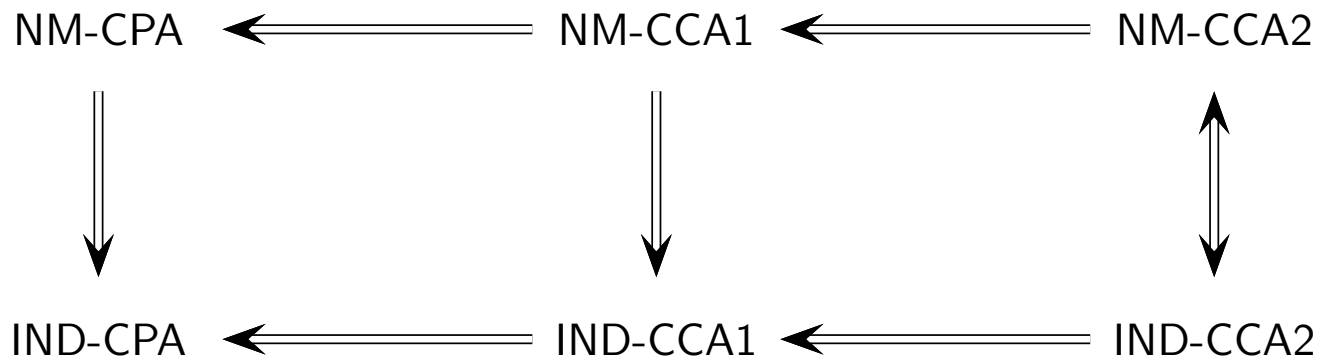
[ (sk, pk) ← Gen
   $\mathcal{M}_0 \leftarrow \mathcal{A}(\text{pk})$ 
   $m \leftarrow \mathcal{M}_0, \bar{m} \leftarrow \mathcal{M}_0$ 
   $\bar{c} \leftarrow \text{Enc}_{\text{pk}}(\bar{m})$ 
   $\pi(\cdot), \hat{c}_1, \dots, \hat{c}_n \leftarrow \mathcal{A}(\bar{c})$ 
  if  $\bar{c} \in \{\hat{c}_1, \dots, \hat{c}_n\}$  then return 0
  return  $\pi(m, \text{Dec}_{\text{sk}}(\hat{c}_1), \dots)$ 

```

The true advantage is

$$\text{Adv}^{\text{nm-cpa}}(\mathcal{A}) = |\Pr[\mathcal{G}_0^{\mathcal{A}} = 1] - \Pr[\mathcal{G}_1^{\mathcal{A}} = 1]|$$

# Homological classification



Horizontal implications are trivial.

- The adversary just gets more powerful in the row.

Downwards implications are trivial.

- A guess **guess** can be passed as a predicate  $\pi(\cdot) \equiv 0$  and  $\pi(\cdot) \equiv 1$ .

## IND-CCA2 $\Rightarrow$ NM-CC2

**Theorem.** Assume that  $\pi(\cdot)$  is always a  $t_\pi$ -time predicate and it is always possible to obtain a sample from  $\mathcal{M}_0$  in time  $t_m$ . Now if the cryptosystem is  $(t, \varepsilon)$ -IND-CCA2 secure, then for all  $(t - t_g - 2t_m)$ -time adversaries  $\mathcal{A}$ :

$$\text{Adv}^{\text{nm-cca2}}(\mathcal{A}) \leq \varepsilon .$$

Note that

- ▷ The predicate  $\pi(\cdot)$  might be randomised.
- ▷ The predicate  $\pi(\cdot)$  might have variable number of arguments.
- ▷ The predicate  $\pi(\cdot)$  must be a computationally efficient function.
- ▷ The distribution  $\mathcal{M}_0$  must be efficiently samplable.

## The corresponding proof

Let  $\mathcal{B}$  be an adversary that is good in NM-CCA2 games. Then we can emulate NM-CCA2 game given access to the decryption oracle  $\mathcal{O}_2$ :

1.  $\mathcal{A}$  forwards  $\text{pk}$  to  $\mathcal{B}$  who sends back a description of  $\mathcal{M}_0$ .
2.  $\mathcal{A}$  independently samples  $m_0 \leftarrow \mathcal{M}_0$  and  $m_1 \leftarrow \mathcal{M}_0$ .
3.  $\mathcal{A}$  forwards the challenge  $\text{Enc}_{\text{pk}}(m_b)$  to  $\mathcal{B}$ .
4.  $\mathcal{B}$  sends  $\hat{c}_1, \dots, \hat{c}_n$  and  $\pi(\cdot)$  to  $\mathcal{A}$  who
  - uses  $\mathcal{O}_2$  to recover  $\text{Dec}_{\text{sk}}(\hat{c}_1), \dots, \text{Dec}_{\text{sk}}(\hat{c}_n)$ ,
  - outputs  $\pi(m_0, \text{Dec}_{\text{sk}}(\hat{c}_1), \dots, \text{Dec}_{\text{sk}}(\hat{c}_n))$  as the final output.

### Running time

The running time of  $\mathcal{A}$  is  $t_b + t_g + 2t_m$  where  $t_b$  is the running time of  $\mathcal{B}$ .

## Further analysis by code rewriting

For clarity, let  $\mathcal{Q}_0$  and  $\mathcal{Q}_1$  denote the IND-CCA2 security games and  $\mathcal{G}_0$  and  $\mathcal{G}_1$  NM-CCA2 security games. Then note

$$\mathcal{Q}_0^{\mathcal{A}} \equiv \mathcal{G}_0^{\mathcal{B}} \quad \text{and} \quad \mathcal{Q}_1^{\mathcal{A}} \equiv \mathcal{G}_1^{\mathcal{B}}$$

where

$$\mathcal{Q}_0^{\mathcal{A}} \left[ \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \text{return } \mathcal{A}^{\mathcal{O}_2(\cdot)}(\text{Enc}_{\text{pk}}(m_0)) \end{array} \right.$$

$$\mathcal{Q}_1^{\mathcal{A}} \left[ \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{Gen} \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_1(\cdot)}(\text{pk}) \\ \text{return } \mathcal{A}^{\mathcal{O}_2(\cdot)}(\text{Enc}_{\text{pk}}(m_1)) \end{array} \right.$$