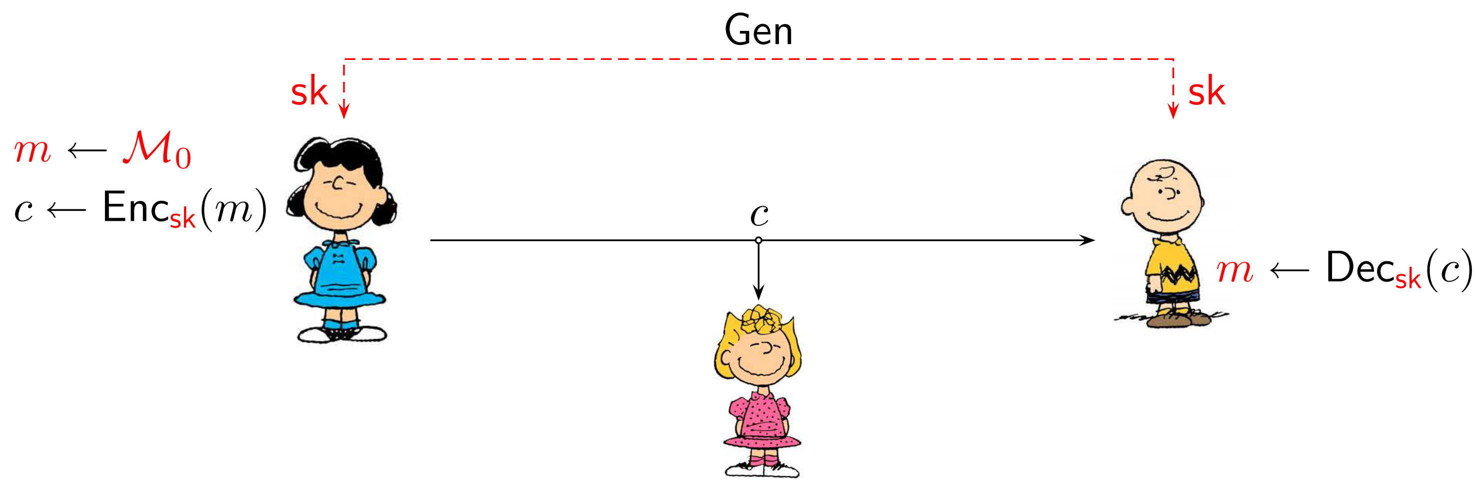
Security of Cryptosystems

Sven Laur
swen@math.ut.ee

University of Tartu

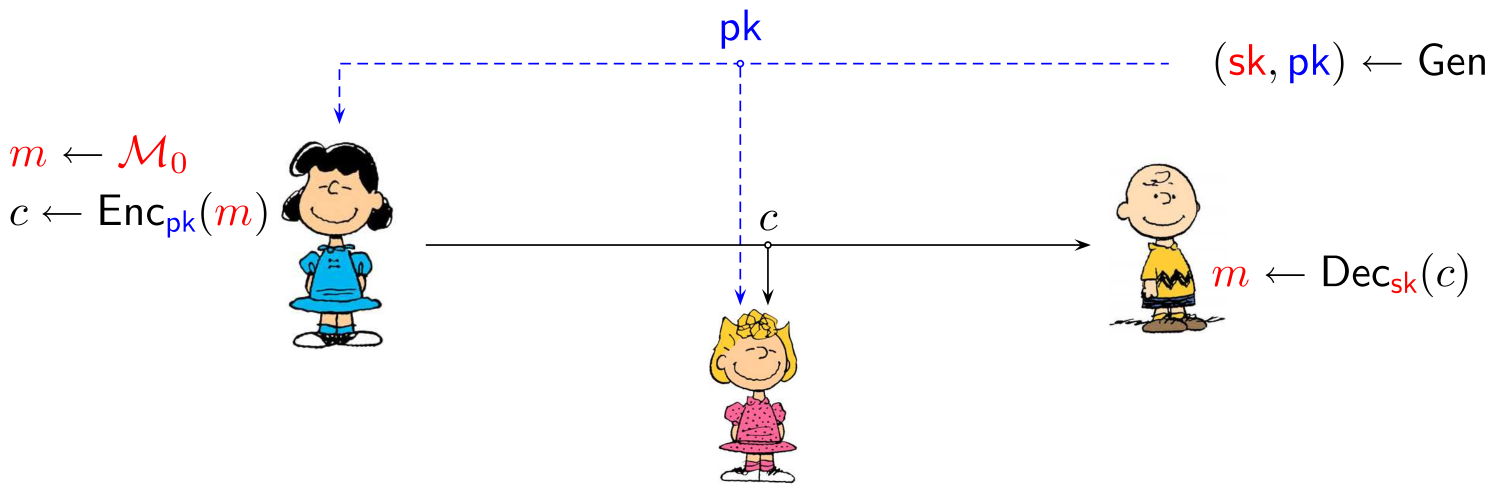
Formal Syntax

Symmetric key cryptosystem



- A randomised key generation algorithm outputs a secret key sk that must be transferred privately to the sender and to the receiver.
- A randomised encryption algorithm $\mathsf{Enc}_{\mathsf{sk}} : \mathcal{M} \rightarrow \mathcal{C}$ takes in a plaintext and outputs a corresponding ciphertext.
- A decryption algorithm $\mathsf{Dec}_{\mathsf{sk}} : \mathcal{C} \rightarrow \mathcal{M} \cup \{\bot\}$ recovers the plaintext or a special abort symbol $\bot$ to indicate invalid ciphertexts.

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 $\mathsf{Enc}_{\mathsf{pk}} : \mathcal{M} \to \mathcal{C}$ takes in a plaintext and outputs a corresponding ciphertext.
- ▷ A decryption algorithm $\mathsf{Dec}_{\mathsf{sk}} : \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ 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 $d = e^{-1} \mod \phi(N)$.
4. Output $\mathsf{sk} = (p, q, e, d)$ and $\mathsf{pk} = (N, e)$.

Encryption and decryption:

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

$$\mathsf{Enc}_{\mathsf{pk}}(m) = m^e \mod N \qquad \mathsf{Dec}_{\mathsf{sk}}(c) = c^d \mod N \ .$$

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 $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is (t, ε)-IND-CPA secure if for all t-time adversaries $\mathcal{A}$:

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

where the security games are defined as follows

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

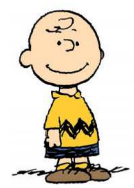
Semantic security against adaptive influence

$m \leftarrow \mathcal{M}_0$

pk →

$\mathcal{M}_0$ ←

$\text{Enc}_{\text{pk}}(m)$ →



Given
- pk
- $\mathcal{M}_0$
- $\text{Enc}_{\text{pk}}(m)$

Charlie tries to guess $g(m)$

$m \leftarrow \mathcal{M}_0$

pk →

$\mathcal{M}_0$ ←

Given
- pk

Charlie tries to guess $g(m)$

Formal definition

Consider following games:

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

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

The true guessing advantage is

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

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, ε)-IND-CPA secure, then for all $(t - t_g - 2t_m)$-time adversaries $\mathcal{A}$:

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

Note that

- The function g might be randomised.
- The function g 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 can predict the value of g well in SEM-CPA game. Now consider a new IND-CPA adversary $\mathcal{A}$:

1. $\mathcal{A}$ forwards pk to $\mathcal{B}$ who describes the distribution $\mathcal{M}_0$ to $\mathcal{A}$.
2. $\mathcal{A}$ independently samples $m_0 \leftarrow \mathcal{M}_0$ and $m_1 \leftarrow \mathcal{M}_0$.
3. $\mathcal{A}$ forwards $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b)$ to $\mathcal{A}$.
4. $\mathcal{B}$ outputs its guess guess to $\mathcal{A}$ who
   - outputs 1 if guess $= g(m_0)$,
   - outputs 0 if guess $\neq g(m_0)$.

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-CPA security games and $\mathcal{G}_0$ and $\mathcal{G}_1$ IND-SEM security games. Then note

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

where

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

An example of IND-CPA secure cryptosystem

ElGamal cryptosystem

Combine the Diffie-Hellman key exchange protocol

Alice		**Bob**
$x \leftarrow \mathbb{Z}_{\vert\mathbb{G}\vert}$	$\xrightarrow{y=g^x}$	$k \leftarrow \mathbb{Z}_{\vert\mathbb{G}\vert}$
	$\xleftarrow{g^k}$	
$g^{xk} = (g^k)^x$		$g^{xk} = (g^x)^k$

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

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

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

Decisional Diffie-Hellman Assumption (DDH)

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

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

where the security games are defined as follows

$$\mathcal{G}_0^{\mathcal{A}} \qquad\qquad \mathcal{G}_1^{\mathcal{A}}$$

$$\left[\begin{array}{l} x, k \leftarrow \mathbb{Z}_q \\ \text{return } \mathcal{A}(g, g^x, g^k, g^{xk}) \end{array}\right. \qquad \left[\begin{array}{l} x, k, c \leftarrow \mathbb{Z}_q \\ \text{return } \mathcal{A}(g, g^x, g^k, g^c) \end{array}\right.$$

The Diffie-Hellman key exchange protocol is secure under the DDH assumption, as Charlie cannot tell the difference between g^{xk} and g^c.

DDH $\Rightarrow$ IND-CPA

Theorem. Let $\mathbb{G}$ be a (t, ε)-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 $\mathcal{A}$:

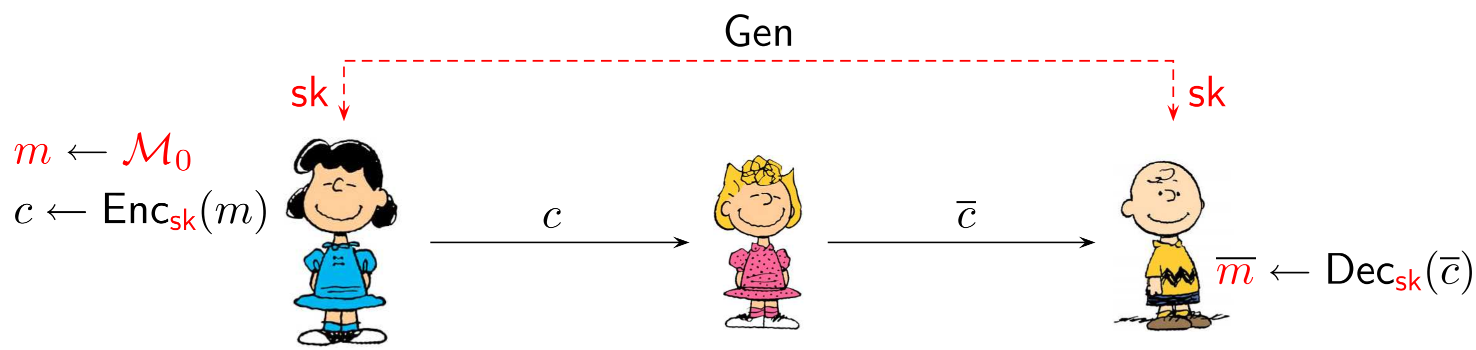
1. Given (g, g^x, g^k, z), set pk $= g^x$ and $(m_0, m_1) \leftarrow \mathcal{B}(\mathsf{pk})$.
2. Toss a fair coin $b \leftarrow \{0, 1\}$ and set $c = (g^k, m_b z)$.
3. If $b \stackrel{?}{=} \mathcal{A}(c)$ return 1 else output 0.

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 form b.

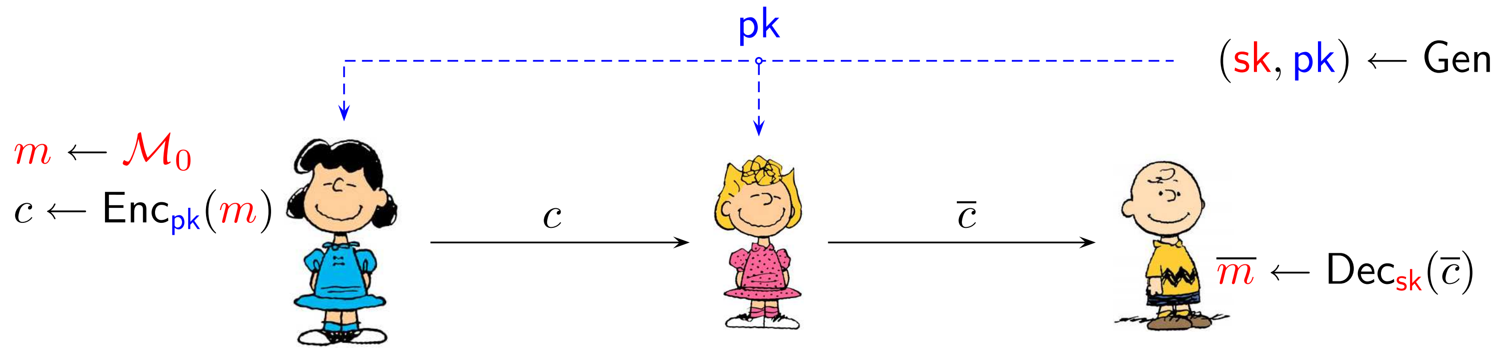
Ciphertext modification attacks

Symmetric key cryptosystem



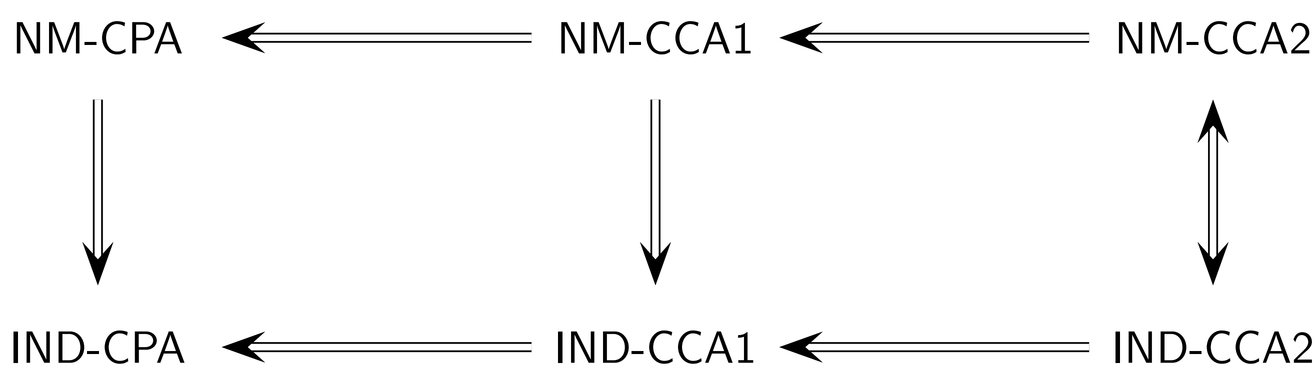
- ▷ A malicious participant may control the communication network and alter the ciphertexts to bypass various security checks.
- ▷ A malicious participant may interact with a key holder and use him or her as an encryption or decryption oracle.
- ▷ A non-malleable encryption detects modifications in ciphertexts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

- ▷ semantic security that follows IND-CPA security,
- ▷ safety against improper usage that follows form IND-CCA1 security,
- ▷ non-malleability of ciphertexts that follows form 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 attack (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, ε)-IND-CCA1 secure if for all t-time adversaries $\mathcal{A}$:

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

where the security games are defined as follows

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

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

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 sk $= (p, q)$ and pk $= N$.

Encryption and decryption:

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

$$\mathsf{Enc}_{\mathsf{pk}}(m) = m^2 \mod N \qquad \mathsf{Dec}_{\mathsf{sk}}(c) = \sqrt{c} \mod N \ .$$

Lunchtime attack

1. Choose $x \leftarrow \mathbb{Z}_N$ and set $c \leftarrow m^2 \mod N$.
2. Compute decryption $\overline{x} \leftarrow \mathcal{O}_1(c)$.
3. If $\overline{x} \neq \pm x$ then
   - Compute nontrivial square root $\xi = \overline{x} \cdot x^{-1} \mod N$
   - 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, ε)-IND-CCA2 secure if for all t-time adversaries $\mathcal{A}$:

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

where the security games are defined as follows

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

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

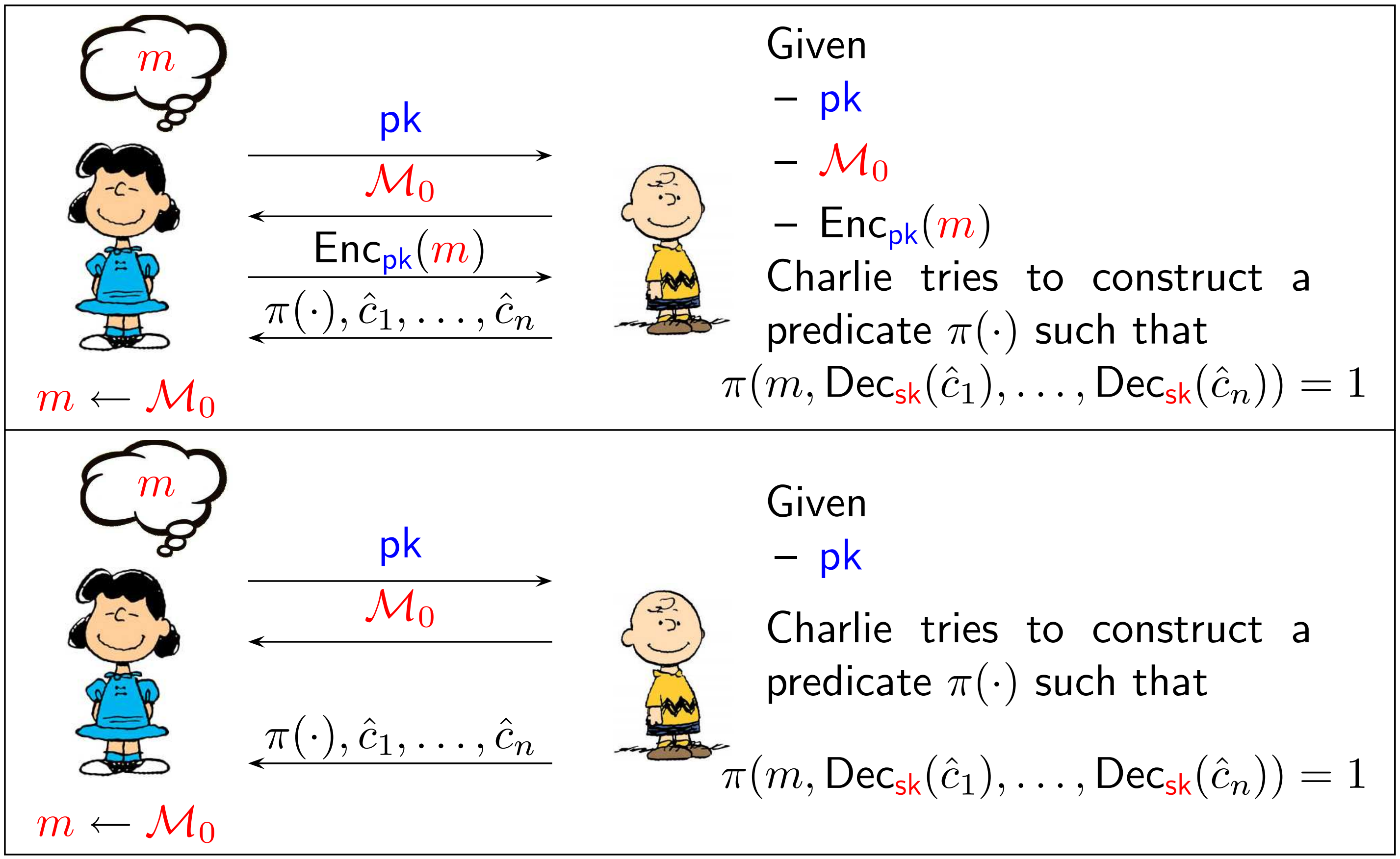
IND-CCA2 secure cryptosystems

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

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

$$\left[\begin{array}{l} (\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen} \\ \mathcal{M}_0 \leftarrow \mathcal{A}(\mathsf{pk}) \\ m \leftarrow \mathcal{M}_0 \\ \pi(\cdot), \hat{c}_1, \ldots \hat{c}_n \leftarrow \mathcal{A}(\mathsf{Enc}_{\mathsf{pk}}(m)) \\ \text{if } c \in \{\hat{c}_1, \ldots \hat{c}_n\} \text{ then return } 0 \\ \text{return } \pi(m, \mathsf{Dec}_{\mathsf{sk}}(\hat{c}_1), \ldots, \hat{c}_n) \end{array}\right.$$

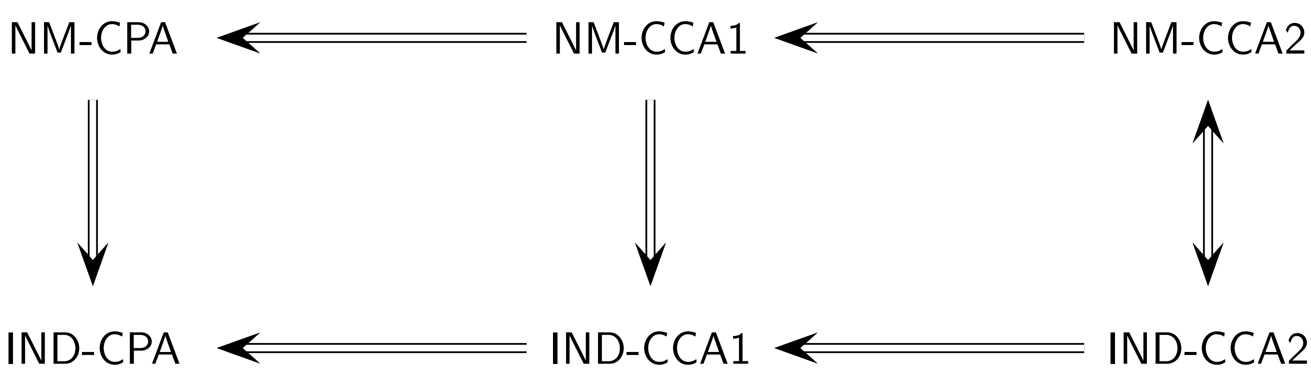
$\mathcal{G}_1^{\mathcal{A}}$

$$\left[\begin{array}{l} (\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen} \\ \mathcal{M}_0 \leftarrow \mathcal{A}(\mathsf{pk}) \\ m \leftarrow \mathcal{M}_0, \ \overline{m} \leftarrow \mathcal{M}_0 \\ \pi(\cdot), \hat{c}_1, \ldots \hat{c}_n \leftarrow \mathcal{A}(\mathsf{Enc}_{\mathsf{pk}}(\overline{m})) \\ \text{if } c \in \{\hat{c}_1, \ldots \hat{c}_n\} \text{ then return } 0 \\ \text{return } \pi(m, \mathsf{Dec}_{\mathsf{sk}}(\hat{c}_1), \ldots, \hat{c}_n) \end{array}\right.$$

The true advantage is

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

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_π-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, ε)-IND-CCA2 secure, then for all $(t - t_g - 2t_m)$-time adversaries $\mathcal{A}$:

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

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 goon in NM-CCA2 games. Then we can emulate NM-CCA2 game given access to the decryption oracle $\mathcal{O}_2$:

1. $\mathcal{A}$ forwards 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 $\mathsf{Enc}_{\mathsf{pk}}(m_b)$ to $\mathcal{B}$.
4. $\mathcal{B}$ sends $\hat{c}_1, \ldots, \hat{c}_n$ and $\pi(\cdot)$ to $\mathcal{A}$ who
   - uses $\mathcal{O}_2$ to recover $\mathsf{Dec}_{\mathsf{sk}}(\hat{c}_1), \ldots, \mathsf{Dec}_{\mathsf{sk}}(\hat{c}_n)$,
   - outputs $\pi(m_b, \mathsf{Dec}_{\mathsf{sk}}(\hat{c}_1), \ldots, \mathsf{Dec}_{\mathsf{sk}}(\hat{c}_n))$ 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

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

where

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

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