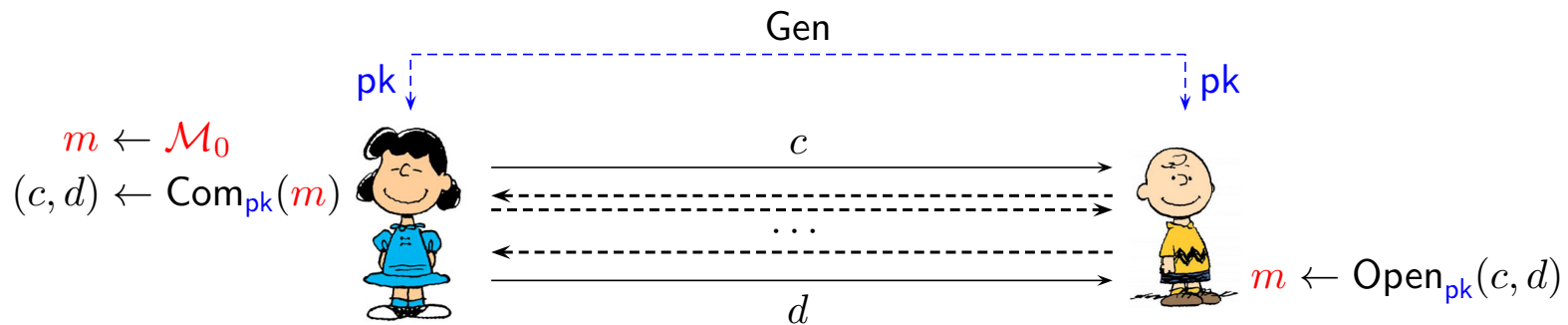# MTAT.07.003 CRYPTOLOGY II

Commitment Schemes

Sven Laur
University of Tartu

Formal Syntax

Canonical use case

$\triangleright$ A randomised key generation algorithm Gen outputs a *public parameters* pk that must be authentically transferred all participants.

$\triangleright$ A commitment function $\mathsf{Com}_{\mathsf{pk}} : \mathcal{M} \to \mathcal{C} \times \mathcal{D}$ takes in a *plaintext* and outputs a corresponding *digest* c and decommitment string d.

$\triangleright$ A commitment can be opened with $\mathsf{Open}_{\mathsf{pk}} : \mathcal{C} \times \mathcal{D} \to \mathcal{M} \cup \{\bot\}$.

$\triangleright$ The commitment primitive is *functional* if for all $\mathsf{pk} \leftarrow \mathsf{Gen}$ and $m \in \mathcal{M}$:

$$\mathsf{Open}_{\mathsf{pk}}(\mathsf{Com}_{\mathsf{pk}}(m)) = m \ .$$

Binding property

A commitment scheme is (t, ε)-*binding* if for any t-time adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{bind}}(\mathcal{A}) = \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] \leq \varepsilon \ ,$$

where the challenge game is following

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

$$\left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{Gen} \\[4pt] (c, d_0, d_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\[4pt] m_i \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_i) \textbf{ for } i = 0, 1 \\[4pt] \text{if } m_0 = \bot \text{ or } m_1 = \bot \text{ then } \textbf{return } 0 \\[4pt] \text{else } \textbf{return } \neg[m_0 \overset{?}{=} m_1] \end{array} \right.$$

Collision resistant hash functions

A function family $\mathcal{H}$ is (t, ε)-*collision resistant* if for any t-time adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H}}(\mathcal{A}) = \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] \leq \varepsilon \ ,$$

where the challenge game is following

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

$$
\begin{bmatrix}
h \xleftarrow{u} \mathcal{H} \\
(m_0, m_1) \leftarrow \mathcal{A}(h) \\
\text{if } m_0 = m_1 \text{ then } \textbf{return } 0 \\
\text{else } \textbf{return } [h(m_0) \overset{?}{=} h(m_1)]
\end{bmatrix}
$$

Hash commitments

Let $\mathcal{H}$ be (t, ε)-collision resistant hash function family. Then we can construct a binding commitment:

▷ The setup algorithm returns $h \xleftarrow{u} \mathcal{H}$ as a public parameter.

▷ To commit m, return $h(m)$ as digest and m as a decommitment string.

▷ The message m is a valid opening of c if $h(m) = c$.

Usage

▷ Integrity check for files and file systems in general.

▷ Minimisation of memory footprint in servers:
1. A server stores the hash $c \leftarrow h(m)$ of an initial application data m.
2. Data is stored by potentially malicious clients.
3. Provided data m' is correct if $h(m') = c$.

Hiding property

A commitment scheme is (t, ε)-*hiding* if for any t-time adversary $\mathcal{A}$:

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

where

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

$$\begin{aligned}
&\mathsf{pk} \leftarrow \mathsf{Gen} \\
&(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\
&(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_0) \\
&\textbf{return } \mathcal{A}(c)
\end{aligned}$$

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

$$\begin{aligned}
&\mathsf{pk} \leftarrow \mathsf{Gen} \\
&(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\
&(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_1) \\
&\textbf{return } \mathcal{A}(c)
\end{aligned}$$

Any cryptosystem is a commitment scheme

Setup:

Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$ and delete sk and output pk.

Commitment:

To commit m, sample necessary randomness $r \leftarrow \mathcal{R}$ and output:

$$\begin{cases} c \leftarrow \mathsf{Enc_{pk}}(m; r) \ , \\ d \leftarrow (m, r) \ . \end{cases}$$

Opening:

A tuple (c, m, r) is a valid decommitment of m if $c = \mathsf{Enc_{pk}}(m; r)$.

Security guarantees

If a cryptosystem is (t, ε)-IND-CPA secure and functional, then the resulting commitment scheme is (t, ε)-hiding and perfectly binding.

⋄ We can construct commitment schemes from the ElGamal and Goldwasser-Micali cryptosystems.

⋄ For the ElGamal cryptosystem, one can create public parameters pk without the knowledge of the secret key sk.

⋄ The knowledge of the secret key sk allows a participant to extract messages from the commitments.

⋄ The extractability property is useful in security proofs.

Simple Commitment Schemes

Modified Naor commitment scheme

Setup:

Choose a random n-bit string $\mathsf{pk} \xleftarrow{u} \{0,1\}^n$.
Let $f : \{0,1\}^k \to \{0,1\}^n$ be a pseudorandom generator.

Commitment:

To commit $m \in \{0,1\}$, generate $d \leftarrow \{0,1\}^k$ and compute digest

$$c \leftarrow \begin{cases} f(d), & \text{if } m = 0 \ , \\ f(d) \oplus \mathsf{pk}, & \text{if } m = 1 \ . \end{cases}$$

Opening:

Given (c, d) check whether $c = f(d)$ or $c = f(d) \oplus \mathsf{pk}$.

Security guarantees

If $f : \{0,1\}^k \to \{0,1\}^n$ is (t,ε)-secure pseudorandom generator, then the modified Naor commitment scheme is $(t, 2\varepsilon)$-hiding and 2^{2k-n}-binding.

Proof

Hiding claim is obvious, since we can change $f(d)$ with uniform distribution. For the binding bound note that

$$|\mathcal{PK}_{\text{bad}}| = \#\{\mathsf{pk} : \exists d_0, d_1 : f(d_0) \oplus f(d_1) = \mathsf{pk}\} \leq 2^{2k}$$
$$|\mathcal{PK}_{\text{all}}| = \#\{0,1\}^n = 2^n$$

and thus

$$\mathsf{Adv}^{\text{bind}}(\mathcal{A}) \leq \Pr\left[\mathsf{pk} \in \mathcal{PK}_{\text{bad}}\right] \leq 2^{2k-n} \ .$$

Discrete logarithm

Let $\mathbb{G} = \langle g \rangle$ be a q-element group that is generated by a single element g. Then for any $y \in \mathbb{G}$ there exists a minimal value $0 \leq x \leq q$ such that

$$g^x = y \quad \Leftrightarrow \quad x = \log_g y \ .$$

A group $\mathbb{G}$ is (t, ε)-secure DL group if for any t-time adversary $\mathcal{A}$

$$\mathsf{Adv}^{\mathsf{dl}}_{\mathbb{G}}(\mathcal{A}) = \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] \leq \varepsilon \ ,$$

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

$$\begin{bmatrix} y \xleftarrow{u} \mathbb{G} \\ x \leftarrow \mathcal{A}(y) \\ \textbf{return } [g^x \overset{?}{=} y] \end{bmatrix}$$

Pedersen commitment scheme

Setup:

Let q be a prime and let $\mathbb{G} = \langle g \rangle$ be a q-element DL-group.
Choose y uniformly from $\mathbb{G} \setminus \{1\}$ and set pk $\leftarrow (g, y)$.

Commitment:

To commit $m \in \mathbb{Z}_q$, choose $r \xleftarrow{u} \mathbb{Z}_q$ and output

$$\begin{cases} c \leftarrow g^m y^r \ , \\ d \leftarrow (m, r) \ . \end{cases}$$

Opening:

A tuple (c, m, r) is a valid decommitment for m if $c = g^m y^r$.

Security guarantees

Assume that $\mathbb{G}$ is (t, ε)-secure discrete logarithm group. Then the Pedersen commitment is perfectly hiding and (t, ε)-binding commitment scheme.

Proof

▷ HIDING. The factor y^r has uniform distribution over $\mathbb{G}$, since $y^r = g^{xr}$ for $x \neq 0$ and $\mathbb{Z}_q$ is simple ring: $x \cdot \mathbb{Z}_q = \mathbb{Z}_q$.

▷ BINDING. A valid double opening reveals a discrete logarithm of y:

$$g^{m_0} y^{r_0} = g^{m_1} y^{r_1} \quad \Leftrightarrow \quad \log_g y = \frac{m_1 - m_0}{r_0 - r_1} \ .$$

Note that $r_0 \neq r_1$ for valid double opening. Hence, a double opener $\mathcal{A}$ can be converted to a discrete logarithm finder.

Other Useful Properties

Extractability

A commitment scheme is (t, ε)-*extractable* if there exists a modified setup procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^*$ such that

▷ the distribution of public parameters pk coincides with the original setup;

▷ there exists an efficient extraction function $\mathsf{Extr}_{\mathsf{sk}} : \mathcal{C} \rightarrow \mathcal{M}$ such that for any t-time adversary $\mathsf{Adv}^{\mathsf{ext}}(\mathcal{A}) = \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] \leq \varepsilon$ where

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

$$
\begin{array}{|l}
(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^* \\
(c, d) \leftarrow \mathcal{A}(\mathsf{pk}) \\
\text{if } \mathsf{Open}_{\mathsf{pk}}(c, d) = \perp \text{ then } \textbf{return } 0 \\
\text{else } \textbf{return } \neg[\mathsf{Open}_{\mathsf{pk}}(c, d) \overset{?}{=} \mathsf{Extr}_{\mathsf{sk}}(c)]
\end{array}
$$

Equivocability

A commitment scheme is *equivocable* if there exists

▷ a modified setup procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^*$

▷ a modified fake commitment procedure $(\hat{c}, \sigma) \leftarrow \mathsf{Com}^*_{\mathsf{sk}}$

▷ an efficient equivocation algorithm $\hat{d} \leftarrow \mathsf{Equiv}_{\mathsf{sk}}(\hat{c}, \sigma, m)$

such that

▷ the distribution of public parameters pk coincides with the original setup;

▷ fake commitments $\hat{c}$ are indistinguishable from real commitments

▷ fake commitments $\hat{c}$ can be opened to arbitrary values

$$\forall m \in \mathcal{M}, (\hat{c}, \sigma) \leftarrow \mathsf{Com}^*_{\mathsf{sk}}, \hat{d} \leftarrow \mathsf{Equiv}_{\mathsf{sk}}(\hat{c}, \sigma, m) : \mathsf{Open}_{\mathsf{pk}}(\hat{c}, \hat{d}) \equiv m \ .$$

▷ opening fake and real commitments are indistinguishable.

Formal security definition

A commitment scheme is (t, ε)-*equivocable* if for any t-time adversary $\mathcal{A}$

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

where

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

$$
\begin{array}{l}
\mathsf{pk} \leftarrow \mathsf{Gen} \\[1ex]
\textbf{repeat} \\[1ex]
\quad \left| \begin{array}{l}
m_i \leftarrow \mathcal{A} \\[1ex]
(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m_i) \\[1ex]
\text{Give } (c, d) \text{ to } \mathcal{A}
\end{array} \right. \\[1ex]
\textbf{until } m_i = \bot \\[1ex]
\textbf{return } \mathcal{A}
\end{array}
$$

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

$$
\begin{array}{l}
(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^* \\[1ex]
\textbf{repeat} \\[1ex]
\quad \left| \begin{array}{l}
(c, \sigma) \leftarrow \mathsf{Com}_{\mathsf{sk}}^*, \ m_i \leftarrow \mathcal{A} \\[1ex]
d \leftarrow \mathsf{Equiv}_{\mathsf{sk}}(c, \sigma, m_i) \\[1ex]
\text{Give } (c, d) \text{ to } \mathcal{A}
\end{array} \right. \\[1ex]
\textbf{until } m_i = \bot \\[1ex]
\textbf{return } \mathcal{A}
\end{array}
$$

A famous example

The Pedersen is perfectly equivocable commitment.

$\triangleright$ **Setup.** Generate $x \leftarrow \mathbb{Z}_q^*$ and set $y \leftarrow g^x$.

$\triangleright$ **Fake commitment.** Generate $s \leftarrow \mathbb{Z}_q$ and output $\hat{c} \leftarrow g^s$.

$\triangleright$ **Equivocation.** To open $\hat{c}$, compute $r \leftarrow (s - m) \cdot x^{-1}$.

Proof

$\triangleright$ Commitment value c has uniform distribution.

$\triangleright$ For fixed c and m, there exists a unique value of r.

Equivocation leads to perfect simulation of (c, d) pairs.

Homomorphic commitments

A commitment scheme is $\otimes$-*homomorphic* if there exists an efficient coordinate-wise multiplication operation $\cdot$ defined over $\mathcal{C}$ and $\mathcal{D}$ such that

$$\mathsf{Com}_{\mathsf{pk}}(m_1) \cdot \mathsf{Com}_{\mathsf{pk}}(m_2) \equiv \mathsf{Com}_{\mathsf{pk}}(m_1 \otimes m_2) \ ,$$
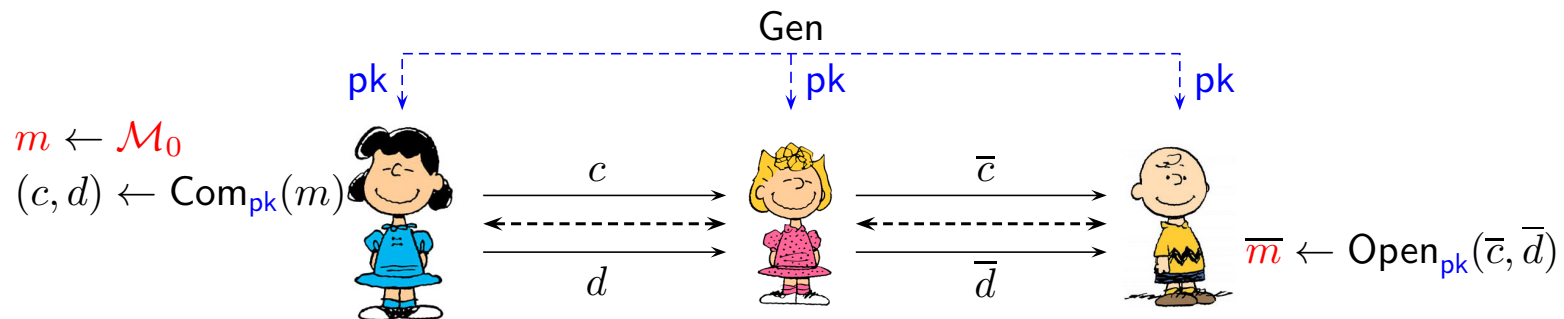
where the distributions coincide even if $\mathsf{Com}_{\mathsf{pk}}(m_1)$ is fixed.

Examples

▷ ElGamal commitment scheme

▷ Pedersen commitment scheme

Active Attacks

Non-malleability wrt opening

A commitment scheme is non-malleable wrt. opening if an adversary who knows the input distribution $\mathcal{M}_0$ cannot alter commitment and decommitment values c, d on the fly so that

▷ $\mathcal{A}$ cannot *efficiently* open the altered commitment value $\overline{c}$ to a message $\overline{m}$ that is related to original message m.

Commitment c does not help the adversary to create other commitments.

Formal definition

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

$$
\begin{array}{l}
\mathsf{pk} \leftarrow \mathsf{Gen} \\
\mathcal{M}_0 \leftarrow \mathcal{A}(\mathsf{pk}) \\
m \leftarrow \mathcal{M}_0 \\
(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(m) \\
\pi(\cdot), \hat{c}_1, \ldots, \hat{c}_n \leftarrow \mathcal{A}(c) \\
\hat{d}_1, \ldots \hat{d}_n \leftarrow \mathcal{A}(d) \\
\text{if } c \in \{\hat{c}_1, \ldots, \hat{c}_n\} \text{ then } \textbf{return } 0 \\
\hat{m}_i \leftarrow \mathsf{Open}_{\mathsf{pk}}(\hat{c}_i, \hat{d}_i) \textbf{ for } i = 1, \ldots, n \\
\textbf{return } \pi(m, \hat{m}_1, \ldots, \hat{m}_n)
\end{array}
$$

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

$$
\begin{array}{l}
\mathsf{pk} \leftarrow \mathsf{Gen} \\
\mathcal{M}_0 \leftarrow \mathcal{A}(\mathsf{pk}) \\
m \leftarrow \mathcal{M}_0, \overline{m} \leftarrow \mathcal{M}_0 \\
(\overline{c}, \overline{d}) \leftarrow \mathsf{Com}_{\mathsf{pk}}(\overline{m}) \\
\pi(\cdot), \hat{c}_1, \ldots, \hat{c}_n \leftarrow \mathcal{A}(\overline{c}) \\
\hat{d}_1, \ldots \hat{d}_n \leftarrow \mathcal{A}(\overline{d}) \\
\text{if } c \in \{\hat{c}_1, \ldots, \hat{c}_n\} \text{ then } \textbf{return } 0 \\
\hat{m}_i \leftarrow \mathsf{Open}_{\mathsf{pk}}(\hat{c}_i, \hat{d}_i) \textbf{ for } i = 1, \ldots, n \\
\textbf{return } \pi(m, \hat{m}_1, \ldots, \hat{m}_n)
\end{array}
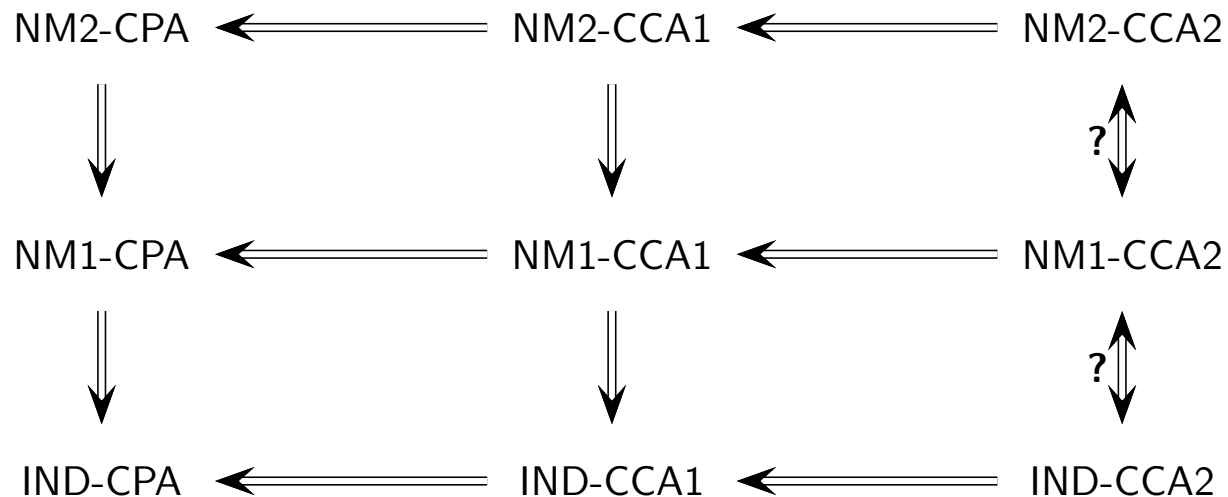$$

Non-malleability wrt commitment
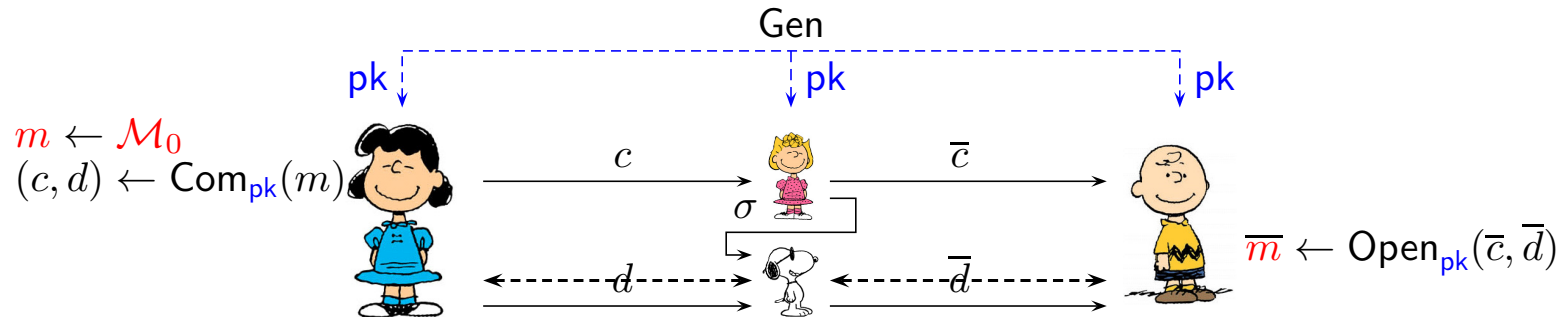
A commitment scheme is non-malleable wrt commitment if an adversary $\mathcal{A}_1$ who knows the input distribution $\mathcal{M}_0$ cannot alter the commitment value c on the fly so that

▷ an unbounded adversary $\mathcal{A}_2$ cannot open the altered commitment value $\overline{c}$ to a message $\overline{m}$ that is related to original message m.

Commitment c does not help the adversary to create other commitments even if some secret values are leaked after the creation of c and $\overline{c}$.

Homological classification

NM2-CPA $\Longleftarrow$ NM2-CCA1 $\Longleftarrow$ NM2-CCA2

$\Downarrow$ $\Downarrow$ **?** $\Updownarrow$

NM1-CPA $\Longleftarrow$ NM1-CCA1 $\Longleftarrow$ NM1-CCA2

$\Downarrow$ $\Downarrow$ **?** $\Updownarrow$

IND-CPA $\Longleftarrow$ IND-CCA1 $\Longleftarrow$ IND-CCA2

Can we define decommitment oracles such that the graph depicted above captures relations between various notions where

▷ NM1-XXX denotes non-malleability wrt opening,

▷ NM2-XXX denotes non-malleability wrt commitment.