

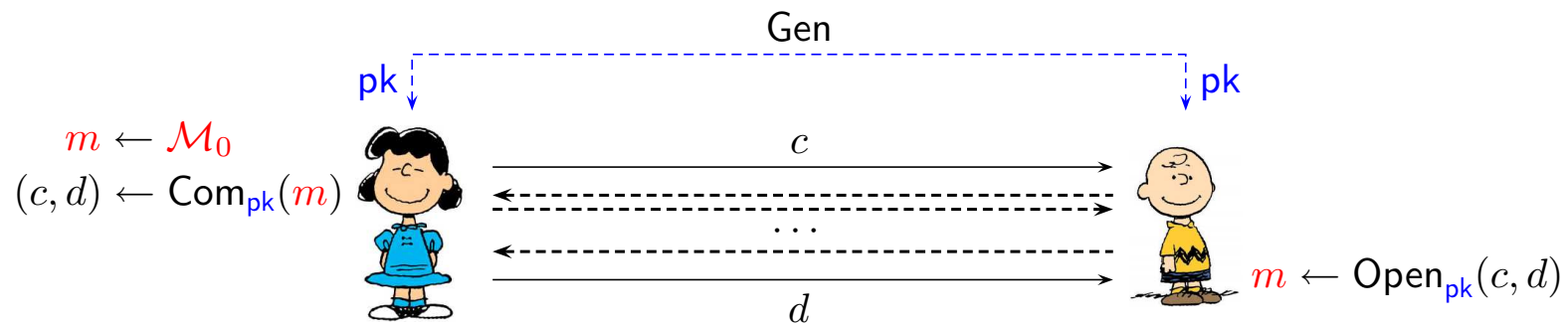
MTAT.07.003 CRYPTOLOGY II

Commitment Schemes

Sven Laur
University of Tartu

Formal Syntax

Canonical use case



- ▷ A randomised key generation algorithm Gen outputs a *public parameters* pk that must be authentically transferred all participants.
- ▷ A commitment function $\text{Com}_{pk} : \mathcal{M} \rightarrow \mathcal{C} \times \mathcal{D}$ takes in a *plaintext* and outputs a corresponding *digest* c and decommitment string d .
- ▷ A commitment can be opened with $\text{Open}_{pk} : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{M} \cup \{\perp\}$.
- ▷ The commitment primitive is *functional* if for all $pk \leftarrow \text{Gen}$ and $m \in \mathcal{M}$:

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

Binding property

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

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

where the challenge game is following

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

Collision resistant hash functions

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

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

where the challenge game is following

$$\mathcal{G}^{\mathcal{A}} \left[\begin{array}{l} h \xleftarrow{u} \mathcal{H} \\ (m_0, m_1) \leftarrow \mathcal{A}(h) \\ \text{if } m_0 = m_1 \text{ then } \mathbf{return} \ 0 \\ \text{else } \mathbf{return} \ [h(m_0) \stackrel{?}{=} h(m_1)] \end{array} \right.$$

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} :

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

where

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

Any cryptosystem is a commitment scheme

Setup:

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

Commitment:

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

$$\begin{cases} c \leftarrow \text{Enc}_{pk}(m; r) , \\ d \leftarrow (m, r) . \end{cases}$$

Opening:

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

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 $\mathbf{pk} \xleftarrow{u} \{0, 1\}^n$.

Let $f : \{0, 1\}^k \rightarrow \{0, 1\}^n$ be a pseudorandom generator.

Commitment:

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

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

Opening:

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

Security guarantees

If $f : \{0, 1\}^k \rightarrow \{0, 1\}^n$ 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

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

and thus

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

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}

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

where

$$\mathcal{G}^{\mathcal{A}} \left[\begin{array}{l} y \leftarrow_{\mathcal{U}} \mathbb{G} \\ x \leftarrow \mathcal{A}(y) \\ \mathbf{return} [g^x \stackrel{?}{=} y] \end{array} \right.$$

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 $\text{pk} \leftarrow (g, y)$.

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 $(pk, sk) \leftarrow \text{Gen}^*$ such that

- ▷ the distribution of public parameters pk coincides with the original setup;
- ▷ there exists an efficient extraction function $\text{Extr}_{sk} : \mathcal{C} \rightarrow \mathcal{M}$ such that for any t -time adversary $\text{Adv}^{\text{ext}}(\mathcal{A}) = \Pr [\mathcal{G}^{\mathcal{A}} = 1] \leq \varepsilon$ where

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

Equivocability

A commitment scheme is *equivocable* if there exists

- ▷ a modified setup procedure $(pk, sk) \leftarrow \text{Gen}^*$
- ▷ a modified fake commitment procedure $(\hat{c}, \sigma) \leftarrow \text{Com}_{sk}^*$
- ▷ an efficient equivocation algorithm $\hat{d} \leftarrow \text{Equiv}_{sk}(\hat{c}, \sigma, m)$

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

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

- ▷ 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}

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

where

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

```
[ pk ← Gen
  repeat
    |  $m_i \leftarrow \mathcal{A}$ 
    |  $(c, d) \leftarrow \text{Com}_{\text{pk}}(m_i)$ 
    | Give  $(c, d)$  to  $\mathcal{A}$ 
  until  $m_i = \perp$ 
  return  $\mathcal{A}$ 
```

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

```
[  $(\text{pk}, \text{sk}) \leftarrow \text{Gen}^*$ 
  repeat
    |  $(c, \sigma) \leftarrow \text{Com}_{\text{sk}}^*, m_i \leftarrow \mathcal{A}$ 
    |  $d \leftarrow \text{Equiv}_{\text{sk}}(c, \sigma, m_i)$ 
    | Give  $(c, d)$  to  $\mathcal{A}$ 
  until  $m_i = \perp$ 
  return  $\mathcal{A}$ 
```

A famous example

The Pedersen is perfectly equivocal commitment.

- ▷ **Setup.** Generate $x \leftarrow \mathbb{Z}_q^*$ and set $y \leftarrow g^x$.
- ▷ **Fake commitment.** Generate $s \leftarrow \mathbb{Z}_q$ and output $\hat{c} \leftarrow g^s$.
- ▷ **Equivocation.** To open \hat{c} , compute $r \leftarrow (s - m) \cdot x^{-1}$.

Proof

- ▷ Commitment value c has uniform distribution.
- ▷ 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

$$\text{Com}_{\text{pk}}(m_1) \cdot \text{Com}_{\text{pk}}(m_2) \equiv \text{Com}_{\text{pk}}(m_1 \otimes m_2) ,$$

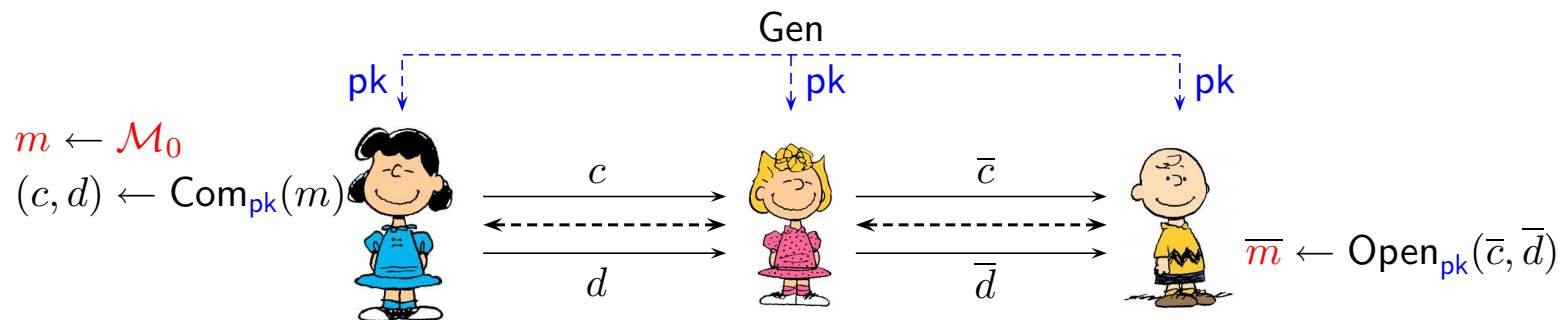
where the distributions coincide even if $\text{Com}_{\text{pk}}(m_1)$ is fixed.

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 \bar{c} to a message \bar{m} that is related to original message m .

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

Formal definition

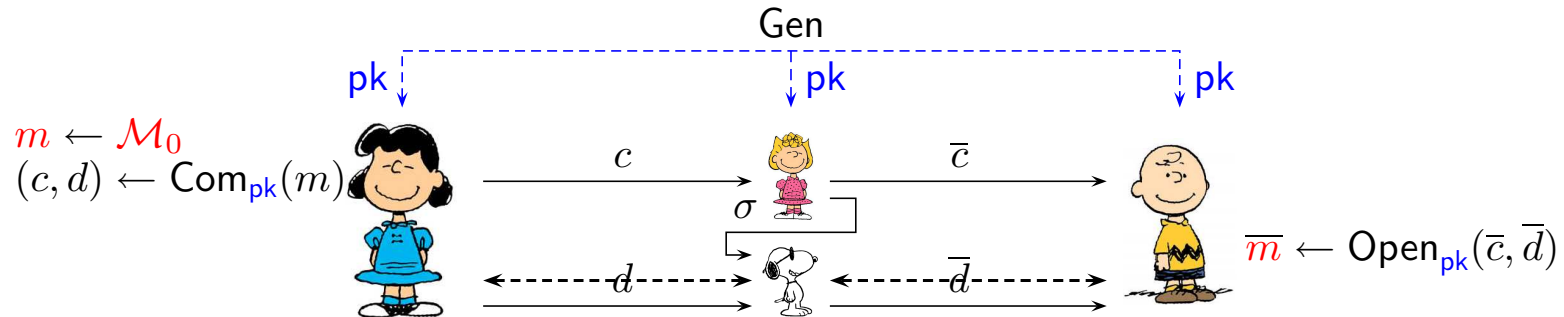
 \mathcal{G}_0^A

```
pk ← Gen
 $\mathcal{M}_0 \leftarrow \mathcal{A}(\text{pk})$ 
 $m \leftarrow \mathcal{M}_0$ 
 $(c, d) \leftarrow \text{Com}_{\text{pk}}(m)$ 
 $\pi(\cdot), \hat{c}_1, \dots, \hat{c}_n \leftarrow \mathcal{A}(c)$ 
 $\hat{d}_1, \dots, \hat{d}_n \leftarrow \mathcal{A}(d)$ 
if  $c \in \{\hat{c}_1, \dots, \hat{c}_n\}$  then return 0
 $\hat{m}_i \leftarrow \text{Open}_{\text{pk}}(\hat{c}_i, \hat{d}_i)$  for  $i = 1, \dots, n$ 
return  $\pi(m, \hat{m}_1, \dots, \hat{m}_n)$ 
```

 \mathcal{G}_1^A

```
pk ← Gen
 $\mathcal{M}_0 \leftarrow \mathcal{A}(\text{pk})$ 
 $m \leftarrow \mathcal{M}_0, \bar{m} \leftarrow \mathcal{M}_0$ 
 $(\bar{c}, \bar{d}) \leftarrow \text{Com}_{\text{pk}}(\bar{m})$ 
 $\pi(\cdot), \hat{c}_1, \dots, \hat{c}_n \leftarrow \mathcal{A}(\bar{c})$ 
 $\hat{d}_1, \dots, \hat{d}_n \leftarrow \mathcal{A}(\bar{d})$ 
if  $c \in \{\hat{c}_1, \dots, \hat{c}_n\}$  then return 0
 $\hat{m}_i \leftarrow \text{Open}_{\text{pk}}(\hat{c}_i, \hat{d}_i)$  for  $i = 1, \dots, n$ 
return  $\pi(m, \hat{m}_1, \dots, \hat{m}_n)$ 
```


Non-malleability wrt commitment

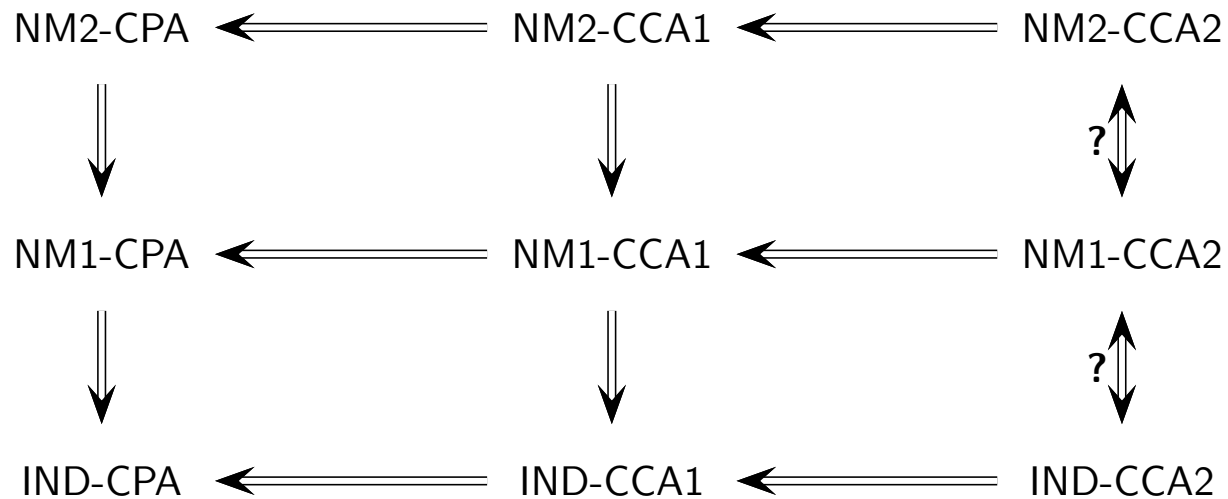


A commitment scheme is non-malleable wrt commitment if an adversary \mathcal{A}_1 who knows the input distribution \mathcal{M}_0 cannot alter the commitment value c on the fly so that

- ▷ an unbounded adversary \mathcal{A}_2 cannot open the altered commitment value \bar{c} to a message \bar{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 \bar{c} .

Homological classification



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.