MTAT.07.003 Cryptology II

Commitment Schemes

Sven Laur
University of Tartu
Formal Syntax
A randomised key generation algorithm Gen outputs a *public parameters* \( pk \) that must be authentically transferred all participants.

A commitment function \( \text{Com}_{pk} : M \rightarrow C \times D \) takes in a *plaintext* and outputs a corresponding *digest* \( c \) and decommitment string \( d \).

A commitment can be opened with \( \text{Open}_{pk} : C \times D \rightarrow M \cup \{ \perp \} \).

The commitment primitive is *functional* if for all \( pk \leftarrow \text{Gen} \) and \( m \in M \):

\[
\text{Open}_{pk}(\text{Com}_{pk}(m)) = m.
\]
**Binding property**

A commitment scheme is \((t, \varepsilon)-binding\) if for any \(t\)-time adversary \(A\):

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

where the challenge game is following:

\[
G^A
\]

\[
\begin{align*}
\text{pk} &\leftarrow \text{Gen} \\
(c, d_0, d_1) &\leftarrow A(\text{pk}) \\
\text{if } m_0 = \bot \text{ or } m_1 = \bot \text{ then return 0} \\
\text{else return } &\neg[m_0 \overset{?}{=} m_1]
\end{align*}
\]
Collision resistant hash functions

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

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

where the challenge game is following

\[
G^A
\]

\[
\begin{align*}
&h \leftarrow \mathcal{H} \\
&(m_0, m_1) \leftarrow \mathcal{A}(h) \\
&\text{if } m_0 = m_1 \text{ then return 0} \\
&\text{else return } [h(m_0) \neq h(m_1)]
\end{align*}
\]
Hash commitments

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

\begin{itemize}
  \item The setup algorithm returns \( h \leftarrow u \mathcal{H} \) as a public parameter.
  \item To commit \( m \), return \( h(m) \) as digest and \( m \) as a decommitment string.
  \item The message \( m \) is a valid opening of \( c \) if \( h(m) = c \).
\end{itemize}

Usage

\begin{itemize}
  \item Integrity check for files and file systems in general.
  \item Minimisation of memory footprint in servers:
    \begin{enumerate}
      \item A server stores the hash \( c \leftarrow h(m) \) of an initial application data \( m \).
      \item Data is stored by potentially malicious clients.
      \item Provided data \( m' \) is correct if \( h(m') = c \).
    \end{enumerate}
\end{itemize}
Hiding property

A commitment scheme is \((t, \varepsilon)\)-hiding if for any \(t\)-time adversary \(A\):

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

where

\[
\begin{align*}
\mathcal{G}_0^A & \\
& \begin{cases}
(pk \leftarrow \text{Gen}) \\
(m_0, m_1) \leftarrow A(pk) \\
(c, d) \leftarrow \text{Com}_{pk}(m_0) \\
\text{return } A(c)
\end{cases} \\
\mathcal{G}_1^A & \\
& \begin{cases}
(pk \leftarrow \text{Gen}) \\
(m_0, m_1) \leftarrow A(pk) \\
(c, d) \leftarrow \text{Com}_{pk}(m_1) \\
\text{return } A(c)
\end{cases}
\end{align*}
\]
Any cryptosystem is a commitment scheme

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

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 \((t, \varepsilon)\)-IND-CPA secure and functional, then the resulting commitment scheme is \((t, \varepsilon)\)-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 \(\text{pk}\) without the knowledge of the secret key \(\text{sk}\).
- The knowledge of the secret key \(\text{sk}\) 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 $\text{pk} \leftarrow \{0, 1\}^n$.
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

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

Opening:
Given $(c, d)$ check whether $c = f(d)$ or $c = f(d) \oplus \text{pk}$.
Security guarantees

If \( f : \{0, 1\}^k \rightarrow \{0, 1\}^n \) is \((t, \varepsilon)\)-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

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

and thus

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

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

$$g^x = y \iff x = \log_g y .$$

A group $G$ is $(t, \varepsilon)$-secure DL group if for any $t$-time adversary $A$

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

where

$$G^A \begin{cases} y \leftarrow u \in G \\ x \leftarrow A(y) \\ \text{return} \ [g^x \overset{?}{=} y] \end{cases}$$
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 $r \leftarrow \mathbb{Z}_q$ 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 $G$ is $(t, \varepsilon)$-secure discrete logarithm group. Then the Pedersen commitment is perfectly hiding and $(t, \varepsilon)$-binding commitment scheme.

Proof

▷ Hiding. The factor $y^r$ has uniform distribution over $G$, 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 $y$:

$$g^{m_0}y^{r_0} = g^{m_1}y^{r_1} \iff \log g y = \frac{m_1 - m_0}{r_0 - r_1}.$$

Note that $r_0 \neq r_1$ for valid double opening. Hence, a double opener $A$ can be converted to a solver of discrete logarithm.
Other Useful Properties
Extractability

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

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

\[
\begin{align*}
G^A & \\
& \left[ (pk, sk) \leftarrow \text{Gen}^* \\
& (c, d) \leftarrow A(pk) \\
& \text{if } \text{Open}_{pk}(c, d) = \bot \text{ then return } 0 \\
& \text{else return } \neg[\text{Open}_{pk}(c, d) \overset{?}{=} \text{Extr}_{sk}(c)]
\end{align*}
\]
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 \((t, \varepsilon)\)-equivocable if for any \(t\)-time adversary \(\mathcal{A}\)

\[
\text{Adv}^{\text{equiv}}(\mathcal{A}) = \left| \Pr \left[ G_0^A = 1 \right] - \Pr \left[ G_1^A = 1 \right] \right| \leq \varepsilon
\]

where

\[
G_0^A \\
\begin{align*}
\text{pk} &\leftarrow \text{Gen} \\
\text{repeat} &\rule{0pt}{2ex} \\
\quad m_i &\leftarrow \mathcal{A} \\
\quad (c, d) &\leftarrow \text{Com}_{\text{pk}}(m) \\
\quad \text{Give (c, d) to } \mathcal{A} &\rule{0pt}{2ex} \\
\text{until } m_i = \bot &\rule{0pt}{2ex} \\
\text{return } \mathcal{A}
\end{align*}
\]

\[
G_1^A \\
\begin{align*}
(pk, sk) &\leftarrow \text{Gen}^* \\
\text{repeat} &\rule{0pt}{2ex} \\
\quad (c, \sigma) &\leftarrow \text{Com}^*_{sk}, m_i \leftarrow \mathcal{A} \\
\quad d &\leftarrow \text{Equiv}_{sk}(c, \sigma, m) \\
\quad \text{Give (c, d) to } \mathcal{A} &\rule{0pt}{2ex} \\
\text{until } m_i = \bot &\rule{0pt}{2ex} \\
\text{return } \mathcal{A}
\end{align*}
\]
A famous example

The Pedersen is perfectly equivocable 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 \(C\) and \(D\) such that

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

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

Examples

- ElGamal commitment scheme
- Pedersen commitment scheme
Active Attacks
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

\[ \nabla A \text{ cannot efficiently open the altered commitment value } \overline{c} \text{ to a message } \overline{m} \text{ that is related to original message } m. \]

Commitment \( c \) does not help the adversary to create other commitments.
Formal definition

\[ \mathcal{G}_0^A \]
\[
\begin{align*}
&pk \leftarrow \text{Gen} \\
&M_0 \leftarrow A(pk) \\
&m \leftarrow M_0 \\
&(c, d) \leftarrow \text{Com}_pk(m) \\
&\pi(\cdot), \hat{c}_1, \ldots, \hat{c}_n \leftarrow A(c) \\
&\hat{d}_1, \ldots \hat{d}_n \leftarrow A(d) \\
&\text{if } c \in \{\hat{c}_1, \ldots, \hat{c}_n\} \text{ then return 0} \\
&m_i \leftarrow \text{Open}_{pk}(\hat{c}_i, \hat{d}_i) \text{ for } i = 1, \ldots, n \\
&\text{return } \pi(m, m_1, \ldots, m_n)
\end{align*}
\]

\[ \mathcal{G}_1^A \]
\[
\begin{align*}
&pk \leftarrow \text{Gen} \\
&M_0 \leftarrow A(pk) \\
&m \leftarrow M_0, \overline{m} \leftarrow M_0 \\
&(c, d) \leftarrow \text{Com}_pk(\overline{m}) \\
&\pi(\cdot), \hat{c}_1, \ldots, \hat{c}_n \leftarrow A(\overline{c}) \\
&\hat{d}_1, \ldots \hat{d}_n \leftarrow A(\overline{d}) \\
&\text{if } c \in \{\hat{c}_1, \ldots, \hat{c}_n\} \text{ then return 0} \\
&m_i \leftarrow \text{Open}_{pk}(\hat{c}_i, \hat{d}_i) \text{ for } i = 1, \ldots, n \\
&\text{return } \pi(m, \overline{m}_1, \ldots, \overline{m}_n)
\end{align*}
\]
A commitment scheme is non-malleable wrt commitment if an adversary \(A_1\) who knows the input distribution \(M_0\) cannot alter the commitment value \(c\) on the fly so that

\[\text{\(A_2\) 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 \(\overline{c}\).
Can we define decommitment oracles such that the graph depicted above captures relations between various notions where

- NM1-XXX denotes non-malleability w.r.t opening,
- NM2-XXX denotes non-malleability w.r.t commitment.