Commitments
Commitments
Commitments
Commitments

M

r_A

"Commit"

r_B

"Decommit"

"M"
Commitments

- Correctness:
- Binding
- Hiding

\[ \Pr \left[ \text{Decommit} \mathcal{C}_R(\tau, r_A) = M \right] = \text{negl} \]

\[ \Pr \left[ \text{Decommit} \mathcal{C}_R(\tau, r_A^2) = M_2 \right] = \text{negl} \]

\[ \text{Decommit} \mathcal{C}_R(\tau, r_A) = M', \quad \text{when} \quad \tau \leftarrow \text{Commit} \mathcal{C}_R(M; r_A, r_B) \]

\[ \text{Decommit} \mathcal{C}_R(\tau, r_A^2) = M_2, \quad \text{when} \quad \tau \leftarrow \text{Commit} \mathcal{C}_R^*(r_A, r_B) \]

\[ \text{Game}_b. \]

\[ \Pr \left[ R = 1 \mid \text{Game}_b \right] - \Pr \left[ R = 1 \mid \text{Game}_1 \right] = \text{negl} \]
Examples

• If \((g, g^x)\) a commitment to \(x\)?

• \(Ct = E(k, m)\) for a symmetric key encryption \(E\)
Examples

In practice, we use:

- To commit to message $M$, choose random, fixed-length $r$, send $H(r || M)$
- To open commitment, send $r, M$
- BINDING: Sender cannot find another $M'$ to open.
HIDING: $H(M||r)$ “hides” $M$

$C \xrightarrow{c} RO \xrightarrow{e} R^*$

$c = H(M||r)$

CH

$C = H(M||r)$ or $H(M||r||r)$

$R_0$

<table>
<thead>
<tr>
<th>$\infty \cdots 0$</th>
<th>$r_0 \cdots 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty \cdots 1$</td>
<td>$r_0 \cdots 01$</td>
</tr>
</tbody>
</table>

"lazy sampling"
HIDING: $H(M||r)$ "hides" $M$

$\text{ch} \xRightarrow{RO} R^*$

$(r) \xrightarrow{c} \text{"lazy sampling"}$

$\Pr [R^* \text{'s first query is } (M_0||r) \text{ or } (M_1||r)] = \frac{1}{2^{|r|}}$

(second)

$\Pr [\text{second query is } (M_0||r) \text{ or } (M_1||r)] = \frac{1}{2^{|r|}-1}$

$n^{th}$ query is $(M_0||r)$ or $(M_1||r) = \frac{1}{2^{|r|}-n}$

$|r|=n.$
$\Pr[ R^x \text{ ever queries table on } (M, k) \text{ or } (N, k)] = \text{negl}(n)$

Examples

- Is $(g, g^x)$ a commitment to $x$?

- $\text{Ct} = E(k, m)$ for a symmetric key encryption $E$

  Binding: $x < \text{order of group}$

  Hiding: $\begin{pmatrix} x_0, x_1 \\ g^{x_0}, g^{x_1} \end{pmatrix}$

  Not necessarily binding

  (OTP is not binding)

  $k \oplus m_1 = k_2 \oplus m_2$. 
Examples

In practice, we use:

- To commit to message $M$, choose random, fixed-length $r$, send $H(r \ || \ M)$
- To open commitment, send $r$, $M$
- Receiver cannot fully recover $M$.
- Sender cannot find another $M'$ to open.
Pedersen Commitments

- Public parameters: (p, g, h)
  - p: large prime (1024 bit)
  - g: generator
  - h: $g^a$ for hidden a

- Protocol
  - To commit to $x$, C chooses random $r$ and sends $(g^x h^r)$ to R.
  - To open, C sends $x$ and $r$ to R.

- Benefits:
  - One can prove many things about the committed value without opening it
Pedersen Commitments

• Unconditionally hiding
  • Given a commitment \( c \), every value \( x \) is equally likely to be the value committed in \( c \).
  • For example, given \( x, r, \) and any \( x' \), there exists \( r' \) such that \( g^x h^r = g^{x'} h^{r'} \), in fact \( r = (x-x') a^{-1} + r \mod q \).
Pedersen Commitments

• Computationally binding

  • Suppose committer sent $g^x h^r \mod p$ for some $(x, r)$

  • Now it finds $x' \neq x$ and $r'$ such that $c = g^{x'} h^{r'}$.

  • This means that the sender "knows" $\log_g(h) = (x'-x) \cdot (r-r')^{-1}$.

  • This means: assuming DL is hard, the sender cannot open the commitment to a different value.
Application: Coin Tossing

- Alice and Bob want to decide on something by tossing a coin over a phone. How to do this securely?

- Solution: Alice commits to a random bit $b_A \leftarrow \{0, 1\}$, and sends $\text{Com}(b_A; r)$ to Bob

- Bob selects a random bit $b_B \leftarrow \{0, 1\}$ and sends it to Alice

- Alice decommits $b_A$

- Alice and Bob output $b_A \text{xor} b_B$