MTAT.07.003 Cryptology II

Security of Protocols

Sven Laur
University of Tartu

Primitives and protocols

Cryptographic primitives. Primitives are tailor-made constructions that have to preserve their security properties in very specific scenarios.

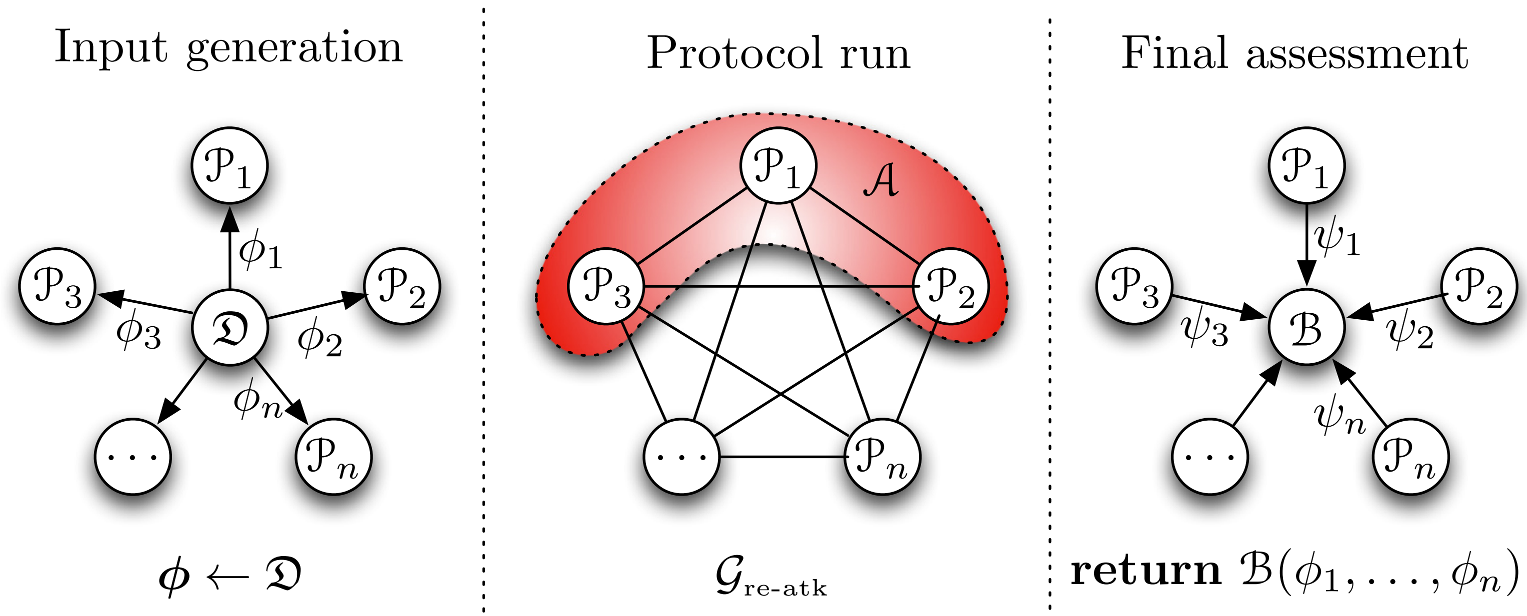
- ▷ IND-CPA cryptosystem is guaranteed to be secure *only* with respect to the simplistic games that define IND-CPA security.
- ▷ A binding commitment is secure *only* against double opening.

Cryptographic protocols. Protocols must preserve security under the wide range of conditions that are implicitly specified by security model.

- ▷ An adversary might try to learn inputs of honest parties.
- ▷ An adversary might try to change the outputs of honest parties.
- ▷ An adversary might force honest parties to compute something else.
- ▷ An adversary might try to learn his or her outputs so that honest parties learn nothing about their outputs.

Security against a specific security goal

For each specific security goal and input distribution $\mathfrak{D}$, we can construct a security game $\mathcal{G}_{\mathrm{real}}$ that models the corresponding protocol run.



Any well-defined security goal can be formalised as a predicate $\mathcal{B}(\cdot)$. It is common to model the adversary $\mathcal{A}$ as a dedicated entity in the model.

Relevant attack scenarios

No protocol can be secure against all imaginable attacks and security goals. Hence, we have to specify the answer for the following questions.

- ◇ What is tolerated adversarial behaviour?
- ◇ What type of predicates $\mathcal{B}(\cdot)$ are considered relevant?
- ◇ What is the model of communication and computations?
- ◇ In which context the protocol is executed?
- ◇ When is a plausible attack successful enough?

Common security levels. Let $\mathfrak{B}$ be the set of relevant predicates.

- ▷ If $\mathfrak{B}$ consists of all predicates then we talk about *statistical security*.
- ▷ If $\mathfrak{B}$ is a set of all t-time predicates, we talk about *computational security*.

Resilience Principle

Resilience principle

Let π_α and π_β be protocols such that any plausible attack $\mathcal{A}$ against π_α can be converted to a plausible attack against the π_β roughly with the same success rate. Then protocol π_α is as secure as π_β. We denote it $\pi_\beta \preceq \pi_\alpha$.

Ideal implementation. For any functionality $\mathcal{F}$, we can consider the ideal implementation π°, which uses *unconditionally trusted third party* $\mathcal{T}$:

1. All parties submit their inputs to a trusted party $\mathcal{T}$.
2. $\mathcal{T}$ computes and sends the desired outputs back.

Resilience principle. An ideal implementation π° is as secure as any protocol π that correctly implements the functionality $\mathcal{F}$. Any protocol $\pi \succeq \pi^\circ$ achieves maximal security level for any relevant security goal $\mathcal{B}(\cdot)$.

Ideal vs real world paradigm

Let $\mathcal{G}_{\text{id-atk}}$ and $\mathcal{G}_{\text{re-atk}}$ be the games that model the execution of an ideal and real protocols π° and π and let $\mathcal{A}^\circ$ and $\mathcal{A}$ be the corresponding real and ideal world adversaries. Then we can compare the following games.

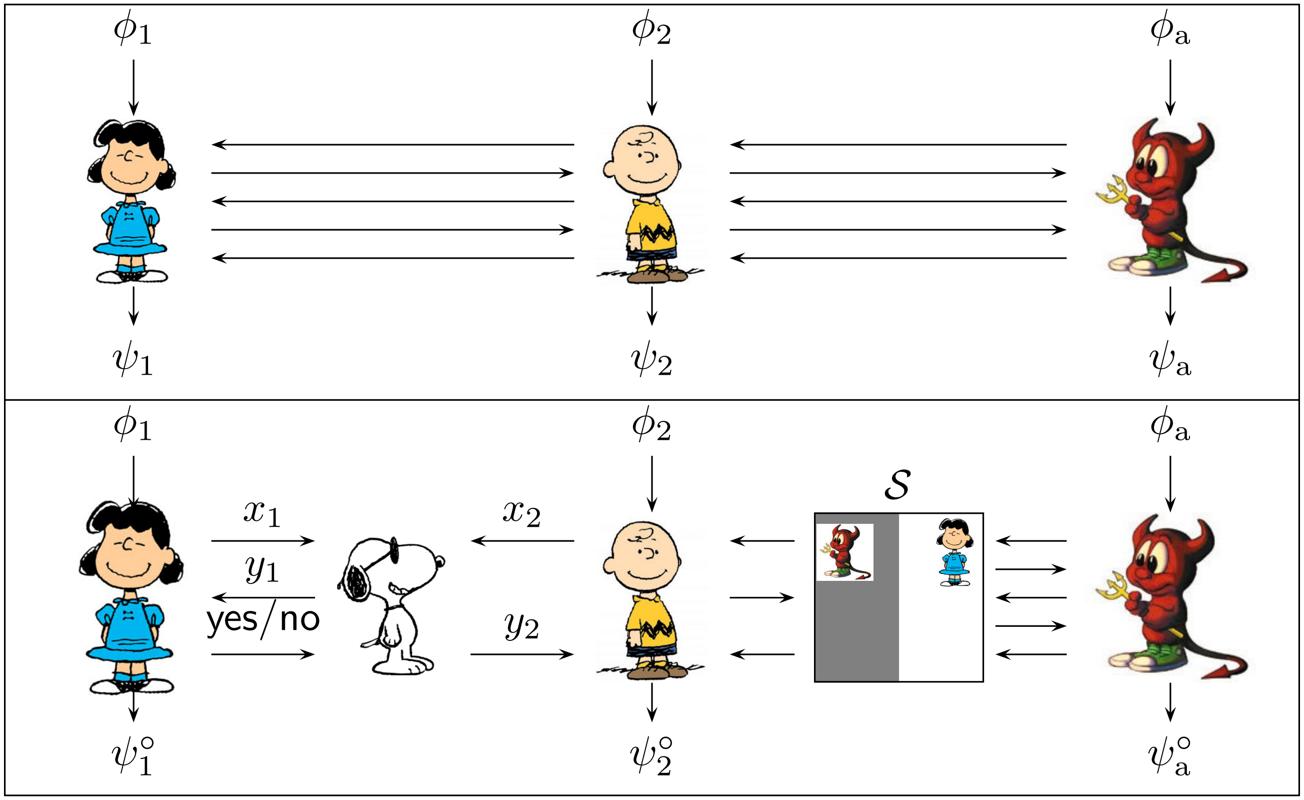
$$\mathcal{G}_{\text{ideal}}^{\mathcal{A}^\circ}\qquad\qquad\qquad\qquad \mathcal{G}_{\text{real}}^{\mathcal{A}}$$

$$\left[\begin{array}{l}\boldsymbol{\phi} \leftarrow \mathfrak{D}\\ \boldsymbol{\psi}^\circ \leftarrow \mathcal{G}_{\text{id-atk}}^{\mathcal{A}^\circ}(\boldsymbol{\phi})\\ \textbf{return } \mathcal{B}(\boldsymbol{\psi}^\circ)\end{array}\right.\qquad\qquad \left[\begin{array}{l}\boldsymbol{\phi} \leftarrow \mathfrak{D}\\ \boldsymbol{\psi} \leftarrow \mathcal{G}_{\text{re-atk}}^{\mathcal{A}}(\boldsymbol{\phi})\\ \textbf{return } \mathcal{B}(\boldsymbol{\psi})\end{array}\right.$$

Now $\pi^\circ \preceq \pi$ if for any $\mathcal{B} \in \mathfrak{B}$ and for any t_{re}-time real world adversary there exists a t_{id}-time ideal world adversary $\mathcal{A}^\circ$ such that

$$\left|\Pr\left[\mathcal{G}_{\text{real}}^{\mathcal{A}} = 1\right] - \Pr\left[\mathcal{G}_{\text{ideal}}^{\mathcal{A}^\circ} = 1\right]\right| \leq \varepsilon \enspace .$$

Simulation principle



The correspondence $\mathcal{A}, \mathcal{B} \mapsto \mathcal{A}^\circ$ is usually implemented by *simulator* $\mathcal{S}$ that act as a translator between real world adversary $\mathcal{A}$ and ideal world.

Standalone Security Model

Two Parties and Static Corruption

Formal description

Computational context. The protocol π is executed once with the inputs x_1, x_2 and auxiliary information σ_1, σ_2, i.e., $\phi_1 = (x_1, \sigma_1)$ and $\phi_2 = (x_2, \sigma_2)$. The output of honest parties is $\psi_i = (y_i, \sigma_i)$ where y_i is the protocol output.

Corruption model. Adversary can corrupt one party before the protocol. A *semihonest* adversary only observes the computations done by the corrupted party. A *malicious* adversary can alter the behaviour of the party.

Communication model. Each party sends and receives one message during a round. A maliciously corrupted party can send his or her message the honest party has sent his or her message (*rushing adversary*).

Ideal world model. Both parties submit their inputs x_1, x_2 to $\mathcal{T}$ who computes the corresponding outputs y_1, y_2. Party $\mathcal{P}_1$ gets his or her input y_1 first and *maliciously* corrupted $\mathcal{P}_1$ *can abort* the protocol after that.

Classical security definitions

Statistical security

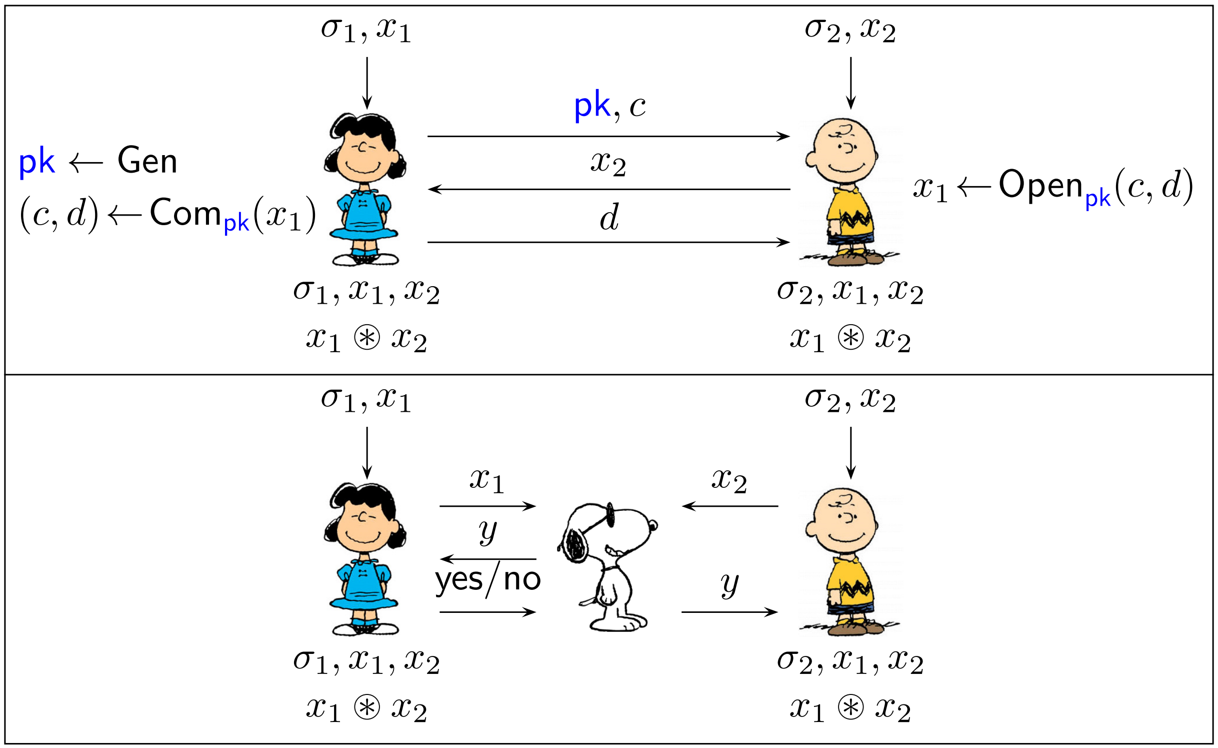
A protocol is $(t_{\mathrm{re}}, t_{\mathrm{id}}, \varepsilon)$-*secure* if for any t_{re}-time real world adversary $\mathcal{A}$ there exists a t_{id}-time ideal world adversary $\mathcal{A}^\circ$ such that for any input distribution $\mathfrak{D}$ the output distributions $\boldsymbol{\psi}$ and $\boldsymbol{\psi}^\circ$ are statistically ε-close.

Computational security

A protocol is $(t_{\mathrm{re}}, t_{\mathrm{id}}, t_{\mathrm{pred}}, \varepsilon)$-*secure* if for any t_{re}-time real world adversary $\mathcal{A}$ there exists a t_{id}-time ideal world adversary $\mathcal{A}^\circ$ such that for any input distribution $\mathfrak{D}$ the output distributions $\boldsymbol{\psi}$ and $\boldsymbol{\psi}^\circ$ are $(t_{\mathrm{pred}}, \varepsilon)$-close.

Examples

Protocol for rock-paper-scissors game



Assume that $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open})$ is perfectly binding commitment scheme. Let $x_1 \circledast x_2$ denote the outcome of the game for $x_1, x_2 \in \{0, 1, 2\}$ and $y = (x_1, x_2, x_1 \circledast x_2)$ denote the desired end result of the game.

Simulator for the first player

$\mathcal{S}^{\mathcal{P}_1^*}(\sigma_1, x_1)$

$(\mathsf{pk}, c) \leftarrow \mathcal{P}_1^*(\sigma_1, x_1)$

Use rewinding to get

$\big[\, d_0 \leftarrow \mathcal{P}_1^*(0), d_1 \leftarrow \mathcal{P}_1^*(1), d_2 \leftarrow \mathcal{P}_1^*(2)$

Compute $x_1^i \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_i)$ for $i \in \{0, 1, 2\}$.

Send 0 to $\mathcal{T}$ if none of the decommitments are valid.

Otherwise send $x_1^i \neq \perp$ to $\mathcal{T}$.

Given y form $\mathcal{T}$ store $d \leftarrow \mathcal{P}_1^*(x_2)$.

If $\mathsf{Open}_{\mathsf{pk}}(c, d) = \perp$ then order $\mathcal{T}$ to halt the computations.

Output whatever $\mathcal{P}_1^*$ outputs.

Simulator for the second player

We cannot build simulator for the second player since $\hat{x}_2$ sent to $\mathcal{P}_1$ may depend on the commitment value and the following code fragment fails

$$\mathcal{S}^{\mathcal{P}_2^*}(\sigma_2, x_2)$$

$$\left[\begin{array}{l}
\mathsf{pk} \leftarrow \mathsf{Gen} \\
(c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0) \\
\text{Send } \hat{x}_2 \leftarrow \mathcal{P}_2^*(\sigma_2, x_2, c) \text{ to } \mathcal{T}. \\
\text{Given } y \text{ from } \mathcal{T} \text{ rewind until success.} \\
\left[\begin{array}{l}
(c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1) \\
\text{If } \mathcal{P}_2^*(\sigma_2, x_2, c) \neq \hat{x}_2 \text{ repeat the cycle.}
\end{array}\right. \\
\text{Output whatever } \mathcal{P}_2^* \text{ does.}
\end{array}\right.$$

Further analysis

If commitment scheme is $(t_{\mathrm{re}}, \varepsilon)$-hiding then probabilities

$$\alpha(x_1, x_2) = \Pr\left[\mathsf{pk} \leftarrow \mathsf{Gen}, (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1) : \mathcal{P}_2^*(c) = x_2\right]$$

can vary at most ε if we alter x_1. Hence, on average after $\frac{1}{\alpha(0,x_2)-\varepsilon}$ the rewinding succeeds and the continuation of the simulation is perfect.

As the running-time must be finite, a nonzero failure probability causes statistical difference. The statistical difference comes from two sources:

- ▷ The distribution of inputs $\hat{x}_2$ submitted to $\mathcal{T}$ is different from the distribution of $\hat{x}_2$ over the real protocol runs.
- ▷ A nonzero simulation failure cause secondary difference.

Coin flipping by telephone

Gen: pk to Lucy, pk to Charlie

Lucy		Messages		Charlie
$b_1 \xleftarrow{u} \{0,1\}$				$b_2 \xleftarrow{u} \{0,1\}$
$(c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(b_1)$		$\xrightarrow{\quad c \quad}$		
		$\xleftarrow{\quad b_2 \quad}$		
		$\xrightarrow{\quad d \quad}$		$b_1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c,d)$
return $b_1 \oplus b_2$				return $b_1 \oplus b_2$

The protocol above assures that participants output a uniformly distributed bit even if one of the participants is malicious.

- ▷ If the commitment scheme is perfectly binding, then Lucy can also generate public parameters for the commitment scheme.
- ▷ If the commitment scheme is perfectly hiding, then Charlie can also generate public parameters for the commitment scheme.

Simulator for the second party

$\mathcal{S}^{\mathcal{P}_2^*}(\phi_2, y)$

$\mathsf{pk} \leftarrow \mathsf{Gen}$

For $i = 1, \ldots k$ do

$b_1 \xleftarrow[u]{} \{0, 1\}$

$(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(b_1)$

$b_2 \leftarrow \mathcal{P}_2^*(\phi_2, \mathsf{pk}, c)$

if $b_1 \oplus b_2 = y$ then

Send d to $\mathcal{P}_2^*$ and output whatever $\mathcal{P}_2^*$ outputs.

return Failure

Failure probability

$\mathcal{S}^{\mathcal{P}_2^*}(\phi_2, y)$

- $\mathsf{pk} \leftarrow \mathsf{Gen}$
- For $i = 1, \ldots k$ do
  - $b_1 \xleftarrow[u]{} \{0,1\}$
  - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(b_1)$
  - $b_2 \leftarrow \mathcal{P}_2^*(\phi_2, \mathsf{pk}, c)$
  - if $b_1 \oplus b_2 = y$ then
    - **return** Success
- **return** Failure

$\mathcal{S}_1^{\mathcal{P}_2^*}(\phi_2, y)$

- $\mathsf{pk} \leftarrow \mathsf{Gen}$
- For $i = 1, \ldots k$ do
  - $b_1 \xleftarrow[u]{} \{0,1\}$
  - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$
  - $b_2 \leftarrow \mathcal{P}_2^*(\phi_2, \mathsf{pk}, c)$
  - if $b_1 \oplus b_2 = y$ then
    - **return** Success
- **return** Failure

$\mathcal{S}_2^{\mathcal{P}_2^*}(\phi_2, y)$

- $\mathsf{pk} \leftarrow \mathsf{Gen}$
- For $i = 1, \ldots k$ do
  - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$
  - $b_2 \leftarrow \mathcal{P}_2^*(\phi_2, \mathsf{pk}, c)$
  - $b_1 \xleftarrow[u]{} \{0,1\}$
  - if $b_1 \oplus b_2 = y$ then
    - **return** Success
- **return** Failure

If commitment scheme is $(k \cdot t, \varepsilon_1)$-hiding, then for any t-time adversary $\mathcal{P}_2^*$ the failure probability

$$\Pr\left[\mathsf{Failure}\right] \leq \Pr\left[\mathcal{S}_2^{\mathcal{P}_2^*}(y) = \mathsf{Failure}\right] + k \cdot \varepsilon_1 \leq 2^{-k} + k \cdot \varepsilon_1 \enspace .$$

The corresponding security guarantee

If the output y is chosen uniformly over $\{0, 1\}$, then the last effective value of b_1 has also an almost uniform distribution: $\left|\Pr\left[b_1 = 1 \middle| \neg\text{Failure}\right] - \frac{1}{2}\right| \leq k \cdot \varepsilon_1$. Hence, for $\mathcal{P}_2^\circ = \mathcal{S}^{\mathcal{P}_2^*}$ the outputs of games

$$\mathcal{G}_{\mathrm{ideal}}^{\mathcal{P}_2^\circ}\left[\begin{array}{l}(\phi_1, \phi_2) \leftarrow \mathfrak{D}\\ y \xleftarrow[u]{} \{0, 1\}\\ \psi_1 \leftarrow (\phi_1, y)\\ \psi_2 \leftarrow \mathcal{S}_2^{\mathcal{P}_2^*(\phi_2)}\\ \textbf{return } (\psi_1, \psi_2)\end{array}\right. \qquad \mathcal{G}_{\mathrm{real}}^{\mathcal{P}_2^*}\left[\begin{array}{l}(\phi_1, \phi_2) \leftarrow \mathfrak{D}\\ \mathcal{P}_1 \text{ and } \mathcal{P}_2^* \text{ run the protocol.}\\ \psi_1 \leftarrow \mathcal{P}_1\\ \psi_2 \leftarrow \mathcal{P}_2^*\\ \textbf{return } (\psi_1, \psi_2)\end{array}\right.$$

are at most $k \cdot \varepsilon_2$ apart if the run of $\mathcal{S}_2^{\mathcal{P}_2^*}$ is successful. Consequently, the statistical distance between output distributions is at most $2^{-k} + 2k \cdot \varepsilon_1$.

Simulator for the first party

$\mathcal{S}^{\mathcal{P}_1^*}(\phi_1, y)$

- $\mathsf{pk} \leftarrow \mathsf{Gen}$, $c \leftarrow \mathcal{P}_1^*(\phi_1, \mathsf{pk})$
- Rewind $\mathcal{P}_1$ to get $d_0 \leftarrow \mathcal{P}_1^*(0)$, $d_1 \leftarrow \mathcal{P}_1^*(1)$
- $b_1^0 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_0)$, $b_1^1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_1)$
- if $\perp \neq b_1^0 \neq b_1^1 \neq \perp$ then Failure
- if $b_1^0 = \perp = b_1^1$ then
  - Send the Halt command to $\mathcal{T}$.
  - Choose $b_2 \xleftarrow[u]{} \{0, 1\}$ and re-run the protocol with b_2.
  - Return whatever $\mathcal{P}_1^*$ returns.
- if $b_1^0 = \perp$ then $b_1 \leftarrow b_1^1$ else $b_1 \leftarrow b_1^0$
- $b_2 \leftarrow b_1 \oplus y$
- Re-run the protocol with b_2
- if $b_1^{b_2} = \perp$ then Send the Halt command to $\mathcal{T}$.
- Return whatever $\mathcal{P}_1^*$ returns.

Further analysis

If the commitment scheme is (t, ε_2)-binding, then the failure probability is less than ε_2. If the output y is chosen uniformly over $\{0, 1\}$, then the value of b_2 seen by $\mathcal{P}_1^*$ is uniformly distributed.

Consequently, the output distributions of $\mathcal{S}^{\mathcal{P}_1^*}$ and $\mathcal{P}_2$ in the ideal world coincide with the real world outputs if $\mathcal{S}$ does not fail.

Resulting security guarantee

Theorem. If a commitment scheme is $(k \cdot t, \varepsilon_1)$-hiding and (t, ε_2)-binding, then for any plausible t-time real world adversary there exists $\mathrm{O}(k \cdot t)$-time ideal world adversary such that the output distributions in the real and ideal world are $\max\left\{2^{-k} + 2k \cdot \varepsilon_1, \varepsilon_2\right\}$-close.