Computationally Sound Symbolic Secrecy with Hash Functions.

Véronique Cortier, Steve Kremer, Ralf Küsters and Bogdan Warinschi

Secrecy Properties

Formal models : property on traces

A data s is secret if the adversary can not produce s.

Secrecy Properties

Formal models : property on traces

A data s is secret if the adversary can not produce s.

Concrete model : indistinguishability

The secrecy of s is defined through the following game:

- Two nonces n_0 and n_1 are randomly generated;
- The adversary interacts with the protocol where s is instantiated with n_b, $b \in \{0, 1\}$;
- We give the pair (n_0, n_1) to the adversary;
- The adversary gives b',

The data s is secret if $\mathsf{Pr}[\mathsf{Exp}^1 = 1] - \mathsf{Pr}[\mathsf{Exp}^0 = 1]$ is a negligible function.

Previous result: Soundness of secrecy properties

Setting: pairing, asymmetric encryption and signatures

Theorem [Esop'05]

If $\Pi \models^c \mathsf{AccessNonce}^c(i,j)$ then $n(a_i,j,s)$ remains secret in Π for the indistinguishability property.

i.e. secrecy (expressed on traces) implies indistinguishability

where $\mathsf{AccessNonce}(i,j)$ *is a special predicate which says that the nonce* $n(a_i,j,s)$ *must remain secret for any session* s *in which honest agents are involved.*

Corollary [Esop'05]

If $\Pi \models^f \mathsf{AccessNonce}^f(i,j)$ then $n(a_i,j,s)$ remains secret in Π for the indistinguishability property.

This does not work in general

Example:

$$A \rightarrow B : h(s)$$

s is weakly secret but not indistinguishable to an attacker.

This does not work in general

Example:

$$A \rightarrow B : h(s)$$

s is weakly secret but not indistinguishable to an attacker.

Goal: A sound (and faithful) formal criterion for indistinguishability.

Contributions

1. Design of a new formal secrecy property
2. Proof of its soundness and its faithfulness w.r.t. indistinguishability
3. Mapping from concrete to formal trace in our new setting:
   - pairing
   - asymmetric encryption
   - hashes (random oracle model)

Patterns

Definition of what is "visible" to an intruder.

Given $S = \{M_1, M_2, \ldots, M_k\}$ and T, we define

$$\mathsf{Pat}_T(S) = \{\mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2), \ldots, \mathsf{Pat}_T^S(M_k)\} \quad \text{with}$$

Patterns

Definition of what is "visible" to an intruder.

Given $S = \{M_1, M_2, \ldots, M_k\}$ and T, we define

$$\mathsf{Pat}_T(S) = \{\mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2), \ldots, \mathsf{Pat}_T^S(M_k)\} \quad \text{with}$$

$$\mathsf{Pat}_T^S(a) = \begin{cases} a & \text{if } S, T \vdash a \\ \square & \text{otherwise} \end{cases}$$

$$\mathsf{Pat}_T^S(\langle M_1, M_2 \rangle) = \langle \mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2) \rangle$$

Patterns

Definition of what is "visible" to an intruder.

Given $S = \{M_1, M_2, \ldots, M_k\}$ and T, we define

$$\mathsf{Pat}_T(S) = \{\mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2), \ldots, \mathsf{Pat}_T^S(M_k)\} \quad \text{with}$$

$$\mathsf{Pat}_T^S(a) = \begin{cases} a & \text{if } S, T \vdash a \\ \square & \text{otherwise} \end{cases}$$

$$\mathsf{Pat}_T^S(\langle M_1, M_2 \rangle) = \langle \mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2) \rangle$$

$$\mathsf{Pat}_T^S(\{M\}^r_{\mathsf{ek}(a)}) = \begin{cases} \{\mathsf{Pat}_T^S(M)\}^r_{\mathsf{ek}(a)} & \text{if } S, T \vdash \mathsf{dk}(a) \\ \{\mathsf{Pat}_T^S(M)\}^r_{\mathsf{ek}(a)} & \text{or if } r \in \mathsf{Rand}_{adv} \\ \square & \text{otherwise} \end{cases}$$

Patterns

Definition of what is "visible" to an intruder.

Given $S = \{M_1, M_2, \ldots, M_k\}$ and T, we define

$$\mathsf{Pat}_T(S) = \{\mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2), \ldots, \mathsf{Pat}_T^S(M_k)\} \quad \text{with}$$

$$\mathsf{Pat}_T^S(a) = \begin{cases} a & \text{if } S, T \vdash a \\ \square & \text{otherwise} \end{cases}$$

$$\mathsf{Pat}_T^S(\langle M_1, M_2 \rangle) = \langle \mathsf{Pat}_T^S(M_1), \mathsf{Pat}_T^S(M_2) \rangle$$

$$\mathsf{Pat}_T^S(\{M\}^r_{\mathsf{ek}(a)}) = \begin{cases} \{\mathsf{Pat}_T^S(M)\}^r_{\mathsf{ek}(a)} & \text{if } S, T \vdash \mathsf{dk}(a) \\ \{\mathsf{Pat}_T^S(M)\}^r_{\mathsf{ek}(a)} & \text{or if } r \in \mathsf{Rand}_{adv} \\ \square & \text{otherwise} \end{cases}$$

$$\mathsf{Pat}_T^S(h(M)) = \begin{cases} h(\mathsf{Pat}_T^S(M)) & \text{if } S, T \vdash M \\ \square & \text{otherwise} \end{cases}$$

Pattern-based secrecy definition

Π protocol

$X^j_{A_i}$ nonce variable occurring in some role A_i.

S set of sent messages

s session number

$$\overline{S} = S \cup \{M \mid \{M\}^r_{\mathsf{ek}(a)} \text{ subterm of } S, r \in \mathsf{Rand}_{adv}\}$$

Pattern-based secrecy definition

Π protocol
$X^j_{A_i}$ nonce variable occurring in some role A_i.
S set of sent messages
s session number

$$\overline{S} = S \cup \{M \mid \{M\}^r_{\mathsf{ek}(a)} \text{ subterm of } S, r \in \mathsf{Rand}_{adv}\}$$

Definition: $X^j_{A_i}$ is secret in Π, written $\Pi \models^f \mathsf{SecNonce}(i, j)$, if:

$$n^{a_i,j,s} \text{ does not occur in } \mathsf{Pat}_{n^{a_i,j,s}}(\overline{S}) \qquad \forall S \; \forall s$$

Examples

- $\phi_1 = \{h(\langle n_b, n' \rangle)\} = \overline{\phi_1}$. Then $\mathsf{Pat}_{n_b}(\overline{\phi_1}) = \{\Box\}$
  $\longrightarrow n_b$ is intuitively hidden by n'.

Examples

- $\phi_1 = \{h(\langle n_b, n' \rangle)\} = \overline{\phi_1}$. Then $\mathsf{Pat}_{n_b}(\overline{\phi_1}) = \{\Box\}$
  $\longrightarrow$ n_b is intuitively hidden by n'.

- $\phi_2 = \{h(\langle n_b, \{n'\}^r_{\mathsf{ek}(a)} \rangle), n'\} = \overline{\phi_2}$ where $r \notin \mathsf{Rand}_{adv}$.
  $\mathsf{Pat}_{n_b}(\overline{\phi_2}) = \{\Box, n'\}$.
  $\longrightarrow$ The encryption of n' does hide n_b.

Examples

- $\phi_1 = \{h(\langle n_b, n' \rangle)\} = \overline{\phi_1}$. Then $\mathsf{Pat}_{n_b}(\overline{\phi_1}) = \{\Box\}$
  $\longrightarrow n_b$ is intuitively hidden by n'.

- $\phi_2 = \{h(\langle n_b, \{n'\}^r_{\mathsf{ek}(a)} \rangle), n'\} = \overline{\phi_2}$ where $r \notin \mathsf{Rand}_{adv}$.
  $\mathsf{Pat}_{n_b}(\overline{\phi_2}) = \{\Box, n'\}$.
  $\longrightarrow$ The encryption of n' does hide n_b.

- $\phi_3 = \{h(\langle n_b, \{n'\}^r_{\mathsf{ek}(a)} \rangle)\}$ where $r \in \mathsf{Rand}_{adv}$.
  Then $\overline{\phi_3} = \phi_3 \cup \{n'\}$ and $\mathsf{Pat}_{n_b}(\overline{\phi_3}) = \{h(\langle n_b, \{n'\}^r_{\mathsf{ek}(a)} \rangle), n'\}$.

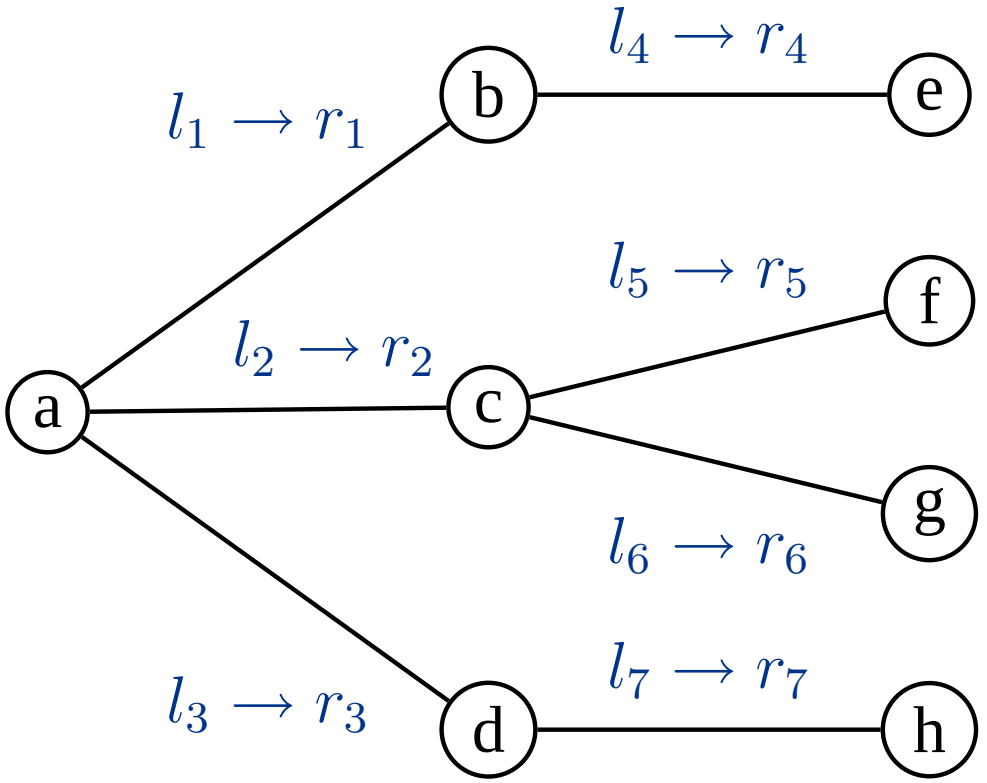
Formal Model

- agents A_i, a_i, nonces $X^j_{A_i}, n(a_i, j, s)$, garbage G
- pairing $\langle m_1, m_2 \rangle$
- asymmetric encryption $\{m\}^l_{\mathsf{ek}(a)}$
- hash $h(m)$

Formal Model

- agents A_i, a_i, nonces $X^j_{A_i}, n(a_i, j, s)$, garbage G
- pairing $\langle m_1, m_2 \rangle$
- asymmetric encryption $\{m\}^l_{\mathsf{ek}(a)}$
- hash $h(m)$

A protocol is a finite set of roles of the form:



Example

Needham-Schroeder-Lowe protocol

$$
\begin{aligned}
A \rightarrow B : &\quad \{N_a, A\}_{\mathsf{ek}(B)} \\
B \rightarrow A : &\quad \{N_a, N_b, B\}_{\mathsf{ek}(A)} \\
A \rightarrow B : &\quad \{N_b\}_{\mathsf{ek}(B)}
\end{aligned}
$$

Example

Needham-Schroeder-Lowe protocol

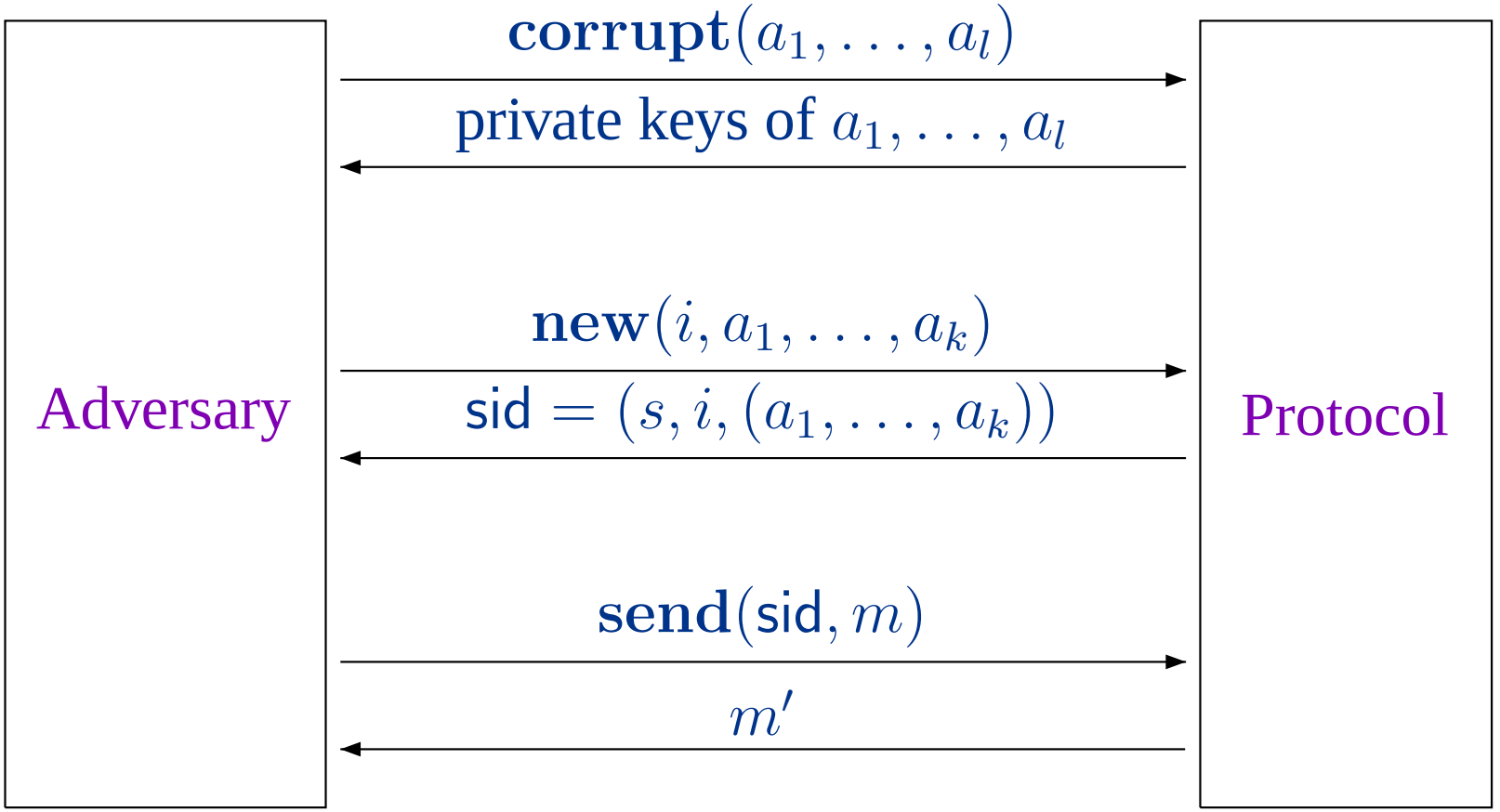
$$
\begin{aligned}
A \to B : &\quad \{N_a, A\}_{\mathsf{ek}(B)} \\
B \to A : &\quad \{N_a, N_b, B\}_{\mathsf{ek}(A)} \\
A \to B : &\quad \{N_b\}_{\mathsf{ek}(B)}
\end{aligned}
$$

$$
\begin{aligned}
\Pi(1) \;=\; & (\mathsf{init} \to \{X^1_{A_1}, A_1\}^{\mathsf{ag}(1)}_{\mathsf{ek}(A_2)}), \\
& (\{X^1_{A_1}, X^1_{A_2}, A_2\}^{L}_{\mathsf{ek}(A_1)} \to \{X^1_{A_2}\}^{\mathsf{ag}(1)}_{\mathsf{ek}(A_2)})
\end{aligned}
$$

$$
\begin{aligned}
\Pi(2) \;=\; & (\{X^1_{A_1}, A_1\}^{L_1}_{\mathsf{ek}(A_2)} \to \{X^1_{A_1}, X^1_{A_2}, A_2\}^{\mathsf{ag}(1)}_{\mathsf{ek}(A_1)}), \\
& (\{X^1_{A_2}\}^{L_2}_{\mathsf{ek}(A_2)} \to \mathsf{stop})
\end{aligned}
$$

Variables are local to each role and each session.

Network


Adversary
$\mathbf{corrupt}(a_1, \ldots, a_l)$
private keys of $a_1, \ldots, a_l$
$\mathbf{new}(i, a_1, \ldots, a_k)$
$\mathsf{sid} = (s, i, (a_1, \ldots, a_k))$
$\mathbf{send}(\mathsf{sid}, m)$
m'
Protocol

Formal Intruder Deduction Rules

$$\frac{S \vdash m_1 \quad S \vdash m_2}{S \vdash \langle m_1 , m_2 \rangle} \qquad \frac{S \vdash \langle m_1 , m_2 \rangle}{S \vdash m_i}\, i \in \{1,2\}$$

$$\frac{S \vdash \mathsf{ek}(b) \quad S \vdash m}{S \vdash \{m\}^{\mathsf{adv}(i)}_{\mathsf{ek}(b)}}\, i \in \mathbb{N} \qquad \frac{S \vdash \{m\}^{l}_{\mathsf{ek}(b)} \quad S \vdash \mathsf{dk}(b)}{S \vdash m}$$

Hypotheses on the Implementation

- encryption : IND-CCA2
  $\longrightarrow$ the adversary cannot distinguish between $\{n_0\}_k$ and $\{n_1\}_k$ even if he has access to encryption and decryption oracles.

Hypotheses on the Implementation

- encryption : IND-CCA2
  $\longrightarrow$ the adversary cannot distinguish between $\{n_0\}_k$ and $\{n_1\}_k$ even if he has access to encryption and decryption oracles.

- hash : Random Oracle Model

Hypotheses on the Implementation

- encryption : IND-CCA2
  $\longrightarrow$ the adversary cannot distinguish between $\{n_0\}_k$ and $\{n_1\}_k$ even if he has access to encryption and decryption oracles.
- hash : Random Oracle Model
- parsing :
  - each bit-string has a label which indicates his type (identity, nonce, key, hash, ...)
  - one can retrieve the (public) encryption key from an encrypted message.

First result: correspondence between traces

Theorem 1:

> Every concrete trace is the image of a valid formal trace, except with negligible probability.

Proof: Adaptation of [Esop05] for hash functions in the Random Oracle Model

First result: correspondence between traces

Theorem 1:

> Every concrete trace is the image of a valid formal trace, except with negligible probability.

Proof: Adaptation of [Esop05] for hash functions in the Random Oracle Model

Corollary:

> Let Π be protocol, $\phi^s \subseteq \mathsf{SymbTr}$ an arbitrary predicate on formal traces and $\phi^c \subseteq \mathsf{ConcTr}$ an arbitrary predicate on concrete traces such that $c(\phi^s) \subseteq \phi^c$.
>
> Then $\Pi \models^s \phi^s$ implies $\Pi \models^c \phi^c$.

Applications : authentication,...

Main result

Theorem 2

$$\Pi \models^{f} \mathsf{SecNonce}^{f}(i,j) \quad \text{iff} \quad \Pi \models^{c} \mathsf{SecNonce}^{c}(i,j)$$

Remark: Our formal secrecy definition is both sufficient and necessary for indistinguishability in the computational world.

Necessary condition

Lemma 1: n does not occur in $\mathsf{Pat}_n(\overline{S})$ if and only if

- $\overline{S} \not\vdash n$
- and $\forall M$ subterm of S such that $\overline{S} \vdash M$, $\forall p$ such that $M|_p = n$,
  - either there exists an encryption along p
  - or $\exists p' < p$ such that
    1. $M|_{p'} = h(M')$ and
    2. $\phi, n \not\vdash M'$.

Necessary condition

Lemma 1: n does not occur in $\mathsf{Pat}_n(\overline{S})$ if and only if

- $\overline{S} \not\vdash n$
- and $\forall M$ subterm of S such that $\overline{S} \vdash M$, $\forall p$ such that $M|_p = n$,
  - either there exists an encryption along p
  - or $\exists p' < p$ such that
    1. $M|_{p'} = h(M')$ and
    2. $\phi, n \not\vdash M'$.

Thus if n occurs in $\mathsf{Pat}_n(\overline{S})$, there exists M such that

- $\overline{S} \vdash M \rightarrow$ there exists a (efficient) recipe to obtain M from S
- There exists a recipe to obtain M from S, n that does not work with S, n'.

Example: $S = \{\langle h(\langle n_b, n' \rangle), n' \rangle\}$

Sufficient condition: difficulties

Suppose $\mathcal{A}$ breaks the indistinguishability game. We have to construct $\mathcal{B}$ that breaks encryption.

$\mathcal{B}$ generates two nonces n_0 and n_1 and simulates the protocol.

Suppose n_b appears in clear in $M[n_b]$.

- To send $\{M[n_b]\}_{\mathsf{ek}(a)}$, $\mathcal{B}$ computes $M[n_0]$ and $M[n_1]$ and uses the left-right oracle
- To send $h(M[n_b])$, $\mathcal{B}$ generates a new random value.

Sufficient condition: difficulties

Suppose $\mathcal{A}$ breaks the indistinguishability game. We have to construct $\mathcal{B}$ that breaks encryption.

$\mathcal{B}$ generates two nonces n_0 and n_1 and simulates the protocol.

Suppose n_b appears in clear in $M[n_b]$.

- To send $\{M[n_b]\}_{\mathsf{ek}(a)}$, $\mathcal{B}$ computes $M[n_0]$ and $M[n_1]$ and uses the left-right oracle
- To send $h(M[n_b])$, $\mathcal{B}$ generates a new random value.

Then $\mathcal{A}$ is given n_0, n_1 and can still queries the random oracle (controlled by $\mathcal{B}$).

$\mathcal{B}$ must be able to answer consistently the hash queries.

Example: $S = \{\{h(n_b)\}_{\mathsf{ek}(a)}\}$ and the hash of n_0 is queried.

Sufficient condition: soundness result

- We define an magical execution where encrypted message $\{m\}_{\mathsf{ek}(a)}$ are replaced by $\{r_m\}_{\mathsf{ek}(a)}$ where r_m is a random value of equal length.

  Lemma 2: If the encryption scheme is IND-CCA then

  $$\Pr\left[\,\mathsf{Exec}_{\mathcal{A},\Pi}(\eta) = 1\,\right] - \Pr\left[\,\mathsf{Exec}^{o}_{\mathcal{A},\Pi}(\eta) = 1\,\right]$$

  is negligible.

Sufficient condition: soundness result

- We define an magical execution where encrypted message $\{m\}_{\mathsf{ek}(a)}$ are replaced by $\{r_m\}_{\mathsf{ek}(a)}$ where r_m is a random value of equal length.

  Lemma 2: If the encryption scheme is IND-CCA then

  $$\Pr\left[\mathsf{Exec}_{\mathcal{A},\Pi}(\eta) = 1\right] - \Pr\left[\mathsf{Exec}^{o}_{\mathcal{A},\Pi}(\eta) = 1\right]$$

  is negligible.

- Using Theorem 1, we obtain that any concrete trace of $\mathsf{Exec}^{o}_{\mathcal{A},\Pi}$ is the image of a valid formal trace thus verifies the pattern-based secrecy property.

Sufficient condition: soundness result

- We define an magical execution where encrypted message $\{m\}_{\mathsf{ek}(a)}$ are replaced by $\{r_m\}_{\mathsf{ek}(a)}$ where r_m is a random value of equal length.

  Lemma 2: If the encryption scheme is IND-CCA then

  $$\Pr\left[\mathsf{Exec}_{\mathcal{A},\Pi}(\eta) = 1\right] - \Pr\left[\mathsf{Exec}^{o}_{\mathcal{A},\Pi}(\eta) = 1\right]$$

  is negligible.

- Using Theorem 1, we obtain that any concrete trace of $\mathsf{Exec}^{o}_{\mathcal{A},\Pi}$ is the image of a valid formal trace thus verifies the pattern-based secrecy property.

- We conclude using Lemma 2 that hash queries can indeed be answered consistently.

Further work

- equivalent definition of the pattern-based secrecy property: equivalence-based definition?
- Decidability result / decision procedures in the formal models
- Other cryptographic primitives: symmetric encryption, signatures, ...
- ...