Formal model of security protocols

Véronique Cortier

Outline

An appropriate datastructure : Terms

Given a signature $\mathcal{F}$ of symbols with an arity

e.g. $\{\mathsf{enc}, \mathsf{enca}, \mathsf{pair}, a, b, c, n_a, n_b\}$

and a set $\mathcal{X}$ of variables,

An appropriate datastructure : Terms

Given a signature $\mathcal{F}$ of symbols with an arity

e.g. $\{\text{enc}, \text{enca}, \text{pair}, a, b, c, n_a, n_b\}$

and a set $\mathcal{X}$ of variables,

the set of terms $T(\mathcal{F}, \mathcal{X})$ is inductively defined as follows :

- constants terms (e.g. a, b, c, n_a, n_b) are terms
- variables are terms
- $f(t_1, \ldots, t_n)$ is a term whenever $t_1, \ldots, t_n$ are terms.

Intuition : from words to trees.

An appropriate datastructure : Terms

Given a signature $\mathcal{F}$ of symbols with an arity

e.g. $\{\text{enc}, \text{enca}, \text{pair}, a, b, c, n_a, n_b\}$

and a set $\mathcal{X}$ of variables,

the set of terms $T(\mathcal{F}, \mathcal{X})$ is inductively defined as follows :

- constants terms (e.g. a, b, c, n_a, n_b) are terms
- variables are terms
- $f(t_1, \ldots, t_n)$ is a term whenever $t_1, \ldots, t_n$ are terms.

Intuition : from words to trees.

$\rightarrow$ There exists automata on trees instead of (classical) automata on words, see e.g. TATA http ://tata.gforge.inria.fr/

Messages

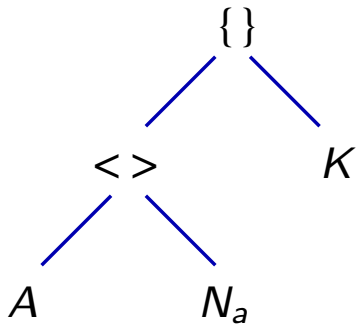
Messages are abstracted by terms.

Agents : $a, b, \ldots$ Nonces : $n_1, n_2, \ldots$ Keys : $k_1, k_2, \ldots$

Cyphertext : $\mathrm{enc}(m, k)$ Concatenation : $\mathrm{pair}(m_1, m_2)$

Example : The message $\{A, N_a\}_K$ is represented by :

$\mathrm{enc}(\mathrm{pair}(A, N_a), K)$



Intuition : only the structure of the message is kept.

Substitution

$$\sigma = \{x_1 \mapsto M_1, \ldots, x_n \mapsto M_n\}$$

The application of a substitution to a term is defined as follows.

$$\begin{aligned} \sigma(x) &= x && \text{if } x \notin \{x_1, \ldots, x_n\} \\ \sigma(f(t_1, \ldots, t_n)) &= f(\sigma(t_1), \ldots, \sigma(t_n)) \end{aligned}$$

We will write $t\sigma$ instead of $\sigma(t)$.

Substitution

$$\sigma = \{x_1 \mapsto M_1, \ldots, x_n \mapsto M_n\}$$

The application of a substitution to a term is defined as follows.

$$\begin{array}{rcll} \sigma(x) & = & x & \text{if } x \notin \{x_1, \ldots, x_n\} \\ \sigma(f(t_1, \ldots, t_n)) & = & f(\sigma(t_1), \ldots, \sigma(t_n)) & \end{array}$$

We will write $t\sigma$ instead of $\sigma(t)$.

Example :

Encryption-Decryption properties

$$\mathsf{dec}(\mathsf{enc}(x, y), y) = x$$

$$\pi_1(\langle x, y \rangle) = x$$

$$\pi_2(\langle x, y \rangle) = y$$

$$\mathsf{deca}(\mathsf{enca}(x, \mathsf{pub}(y)), y) = x$$

Equational theory

More generally, the cryptographic primitives are modeled by an equational theory.

Definition

An equational theory $=_E$ is a relation on terms that is closed under substitutions of terms for variables and closed by context.

- $u =_E v$ implies $u\sigma =_E v\sigma$ (for any σ)
- $u_1 =_E v_1, \ldots, u_n =_E v_n$ implies $f(u_1, \ldots, u_n) =_E f(v_1, \ldots, v_n)$ (for any $f \in \mathcal{F}$)

Equational theory

More generally, the cryptographic primitives are modeled by an equational theory.

Definition

An equational theory $=_E$ is a relation on terms that is closed under substitutions of terms for variables and closed by context.

- $u =_E v$ implies $u\sigma =_E v\sigma$ (for any σ)
- $u_1 =_E v_1, \ldots, u_n =_E v_n$ implies $f(u_1, \ldots, u_n) =_E f(v_1, \ldots, v_n)$ (for any $f \in \mathcal{F}$)

Example :

Other examples of theories

EXclusive Or

$$\begin{array}{rcl@{\qquad}rcl} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & x \oplus y & = & y \oplus x \\ x \oplus x & = & 0 & x \oplus 0 & = & x \end{array}$$

Diffie-Hellmann

$$\exp(\exp(z, x), y) = \exp(\exp(z, y), x)$$

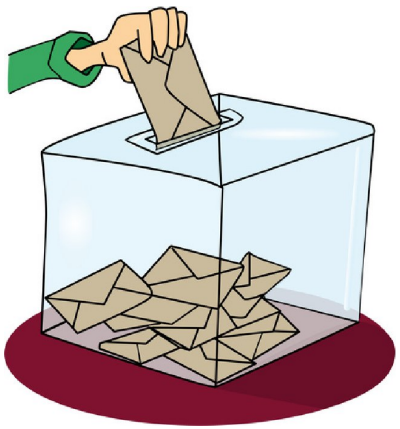
Public key encryption in ProVerif

fun pk(skey) : pkey.

fun aenc(bitstring, pkey) : bitstring.

reduc forall x : bitstring, y : skey ; adec(aenc(x,pk(y)),y) = x.

E-voting protocols



First phase :

$$V \to A : \quad \mathsf{sign}(\mathit{blind}(\mathsf{vote}, r), V)$$
$$A \to V : \quad \mathsf{sign}(\mathit{blind}(\mathsf{vote}, r), A)$$

Voting phase :

$$V \to C : \quad \mathsf{sign}(\mathsf{vote}, A)$$

...

Equational theory for blind signatures

[Kremer Ryan 05]

$$\begin{aligned} \mathsf{checksign}(\mathsf{sign}(x,y),\mathsf{pk}(y)) &= x \\ \mathsf{unblind}(\mathsf{blind}(x,y),y) &= x \\ \mathsf{unblind}(\mathsf{sign}(\mathsf{blind}(x,y),z),y) &= \mathsf{sign}(x,z) \end{aligned}$$

Outline

Syntax for processes

The grammar of processes is as follows :

$$
\begin{aligned}
P, Q, R :=\ & 0 \\
& \texttt{if } M_1 = M_2 \texttt{ then } P \texttt{ else } Q \\
& \mathsf{let}\ x = M \mathsf{in}\ P \\
& \mathsf{in}(c, x).P \\
& \mathsf{out}(c, N).P \\
& \nu n.P \\
& P \mid Q \\
& !P
\end{aligned}
$$

Inspired from the applied-pi calculus [Abadi-Fournet]

Example : Needham-Schroeder protocol

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

N_a random number generated by A.
N_b random number generated by B.

Example : Needham-Schroeder protocol

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

N_a random number generated by A.
N_b random number generated by B.

We need to model two processes :

- one corresponding to the role of A
- one corresponding to the role of B

Role of A

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

$P_A(\mathrm{priv}_A, \mathrm{pub}_B, A, B, N_A) :=$

Role of A

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

$$P_A(\mathsf{priv}_A, \mathsf{pub}_B, A, B, N_A) := \mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(A, N_A), \mathsf{pub}_B))\,;$$

Role of A

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

$$\begin{aligned} P_A(\mathrm{priv}_A, \mathrm{pub}_B, A, B, N_A) := \\ &\mathrm{out}(c, \mathrm{enca}(\mathrm{pair}(A, N_A), \mathrm{pub}_B)); \\ &\mathrm{in}(c, x{:}\ \mathit{bitstring}); \end{aligned}$$

Role of A

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

$$
\begin{aligned}
P_A(\mathrm{priv}_A, \mathrm{pub}_B, A, B, N_A) := \\
&\mathrm{out}(c, \mathrm{enca}(\mathrm{pair}(A, N_A), \mathrm{pub}_B)) ; \\
&\mathrm{in}(c, x\colon \textit{bitstring}) ; \\
&\mathrm{let}\ z\colon \textit{bitstring} = \mathrm{deca}(x, \mathrm{priv}_A)\ \mathrm{in}
\end{aligned}
$$

Role of A

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

$$\begin{aligned} P_A(\mathsf{priv}_A, \mathsf{pub}_B, A, B, N_A) := \\ &\mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(A, N_A), \mathsf{pub}_B)); \\ &\mathsf{in}(c, x\text{: } \textit{bitstring}); \\ &\mathsf{let}\ z\text{: } \textit{bitstring} = \mathsf{deca}(x, \mathsf{priv}_A)\ \mathsf{in} \\ &\mathtt{if}\ N_A = \pi_1(z)\ \mathtt{then}\ \mathsf{let}\ y = \pi_2(z)\ \mathsf{in} \end{aligned}$$

Role of A

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

$P_A(\mathrm{priv}_A, \mathrm{pub}_B, A, B, N_A) :=$
$\quad \mathrm{out}(c, \mathrm{enca}(\mathrm{pair}(A, N_A), \mathrm{pub}_B))\,;$
$\quad \mathrm{in}(c, x\colon \mathit{bitstring})\,;$
$\quad \mathrm{let}\ z\colon \mathit{bitstring} = \mathrm{deca}(x, \mathrm{priv}_A)\ \mathrm{in}$
$\quad \mathtt{if}\ N_A = \pi_1(z)\ \mathtt{then}\ \mathrm{let}\ y = \pi_2(z)\ \mathrm{in}$
$\quad \mathrm{out}(c, \mathrm{enca}(y, \mathrm{pub}_B)).$

Role of B

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

$$P_B(\mathrm{priv}_B, \mathrm{pub}_A, A, B, N_B) := \mathrm{in}(c, x);$$

Role of B

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

$$P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) := \mathsf{in}(c, x); \mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in}$$

Role of B

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

$$\begin{aligned} P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) := \\ \mathsf{in}(c, x); \\ \mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in} \\ \mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A)); \end{aligned}$$

Role of B

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

$$\begin{aligned} P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) := \\ \mathsf{in}(c, x); \\ \mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in} \\ \mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A)); \\ \mathsf{in}(c, z); \end{aligned}$$

Role of B

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

$$\begin{aligned} P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) := \\ &\mathsf{in}(c, x); \\ &\quad \mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in} \\ &\quad\ \mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A)); \\ &\quad\ \mathsf{in}(c, z); \\ &\quad\ \texttt{if}\ \mathsf{deca}(z, \mathsf{priv}_B) = N_B\ \texttt{then}\ 0\ \texttt{else}\ 0. \end{aligned}$$

Complete process

Then, the complete process representing the Needham-Schroeder protocol is :

$P :=$

$$!(\text{new } N_A; P_A(\text{priv}_A, \text{pub}(\text{priv}_B), A, B, N_A)) \mid$$
$$!(\text{new } N_B; P_B(\text{priv}_B, \text{pub}(\text{priv}_A), A, B, N_B))$$

Complete process

Then, the complete process representing the Needham-Schroeder protocol is :

$$
\begin{aligned}
P := \quad & \\
& \text{new } \ \mathsf{priv}_A; \text{new } \ \mathsf{priv}_B; \\
& \mathsf{out}(c, \mathsf{pub}(\mathsf{priv}_A)); \mathsf{out}(c, \mathsf{pub}(\mathsf{priv}_B)); \\
& \quad !(\text{new } N_A; P_A(\mathsf{priv}_A, \mathsf{pub}(\mathsf{priv}_B), A, B, N_A)) \mid \\
& \quad !(\text{new } N_B; P_B(\mathsf{priv}_B, \mathsf{pub}(\mathsf{priv}_A), A, B, N_B))
\end{aligned}
$$

Complete process

Then, the complete process representing the Needham-Schroeder protocol is :

$$
\begin{aligned}
P := \quad & \\
& \text{new } \ \text{priv}_A; \text{new } \ \text{priv}_B; \\
& \text{out}(c, \text{pub}(\text{priv}_A)); \text{out}(c, \text{pub}(\text{priv}_B)); \\
& \quad !(\text{new } N_A; P_A(\text{priv}_A, \text{pub}(\text{priv}_B), A, B, N_A)) \mid \\
& \quad !(\text{new } N_B; P_B(\text{priv}_B, \text{pub}(\text{priv}_A), A, B, N_B))
\end{aligned}
$$

satisfied ?

Complete process - continued

Better :

$$
\begin{aligned}
P := \quad & \text{new } \mathrm{priv}_A; \text{new } \mathrm{priv}_B; \text{new } \mathrm{priv}_C; \\
& \mathrm{out}(c, \mathrm{pub}(\mathrm{priv}_A)); \mathrm{out}(c, \mathrm{pub}(\mathrm{priv}_B)); \mathrm{out}(c, \mathrm{priv}_C); \\
& \quad !(\text{new } N_A; P_A(\mathrm{priv}_A, \mathrm{pub}(\mathrm{priv}_B), A, B, N_A)) \mid \\
& \quad !(\text{new } N_A; P_A(\mathrm{priv}_A, \mathrm{pub}(\mathrm{priv}_C), A, C, N_A)) \mid \\
& \quad !(\text{new } N_B; P_B(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_A), A, B, N_B)) \mid \\
& \quad !(\text{new } N_B; P_B(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_C), B, C, N_B))
\end{aligned}
$$

Complete process - continued

Better :

$P :=$

$$\begin{aligned}
&\text{new } \mathrm{priv}_A; \text{new } \mathrm{priv}_B; \text{new } \mathrm{priv}_C;\\
&\mathrm{out}(c, \mathrm{pub}(\mathrm{priv}_A)); \mathrm{out}(c, \mathrm{pub}(\mathrm{priv}_B)); \mathrm{out}(c, \mathrm{priv}_C);\\
&\quad !(\text{new } N_A; P_A(\mathrm{priv}_A, \mathrm{pub}(\mathrm{priv}_B), A, B, N_A)) \mid\\
&\quad !(\text{new } N_A; P_A(\mathrm{priv}_A, \mathrm{pub}(\mathrm{priv}_C), A, C, N_A)) \mid\\
&\quad !(\text{new } N_B; P_B(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_A), A, B, N_B)) \mid\\
&\quad !(\text{new } N_B; P_B(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_C), B, C, N_B))
\end{aligned}$$

and also

$$\begin{aligned}
&!(\text{new } N_A; P_A(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_A), B, A, N_A)) \mid\\
&!(\text{new } N_A; P_A(\mathrm{priv}_B, \mathrm{pub}(\mathrm{priv}_C), B, C, N_A)) \mid\\
&\ldots
\end{aligned}$$

What to remember when modeling a protocol ?

1. A process for each role of the protocol
   - identify the initial knowledge of the agent (here, $N_A, \mathsf{priv}_A, \mathsf{pub}_B, A, \ldots$)
   - identify the values learned during the protocol, modeled with variables

What to remember when modeling a protocol ?

1. A process for each role of the protocol
   - identify the initial knowledge of the agent (here, $N_A, \mathsf{priv}_A, \mathsf{pub}_B, A, \ldots$)
   - identify the values learned during the protocol, modeled with variables

Don't think you're done !

What to remember when modeling a protocol ?

1. A process for each role of the protocol
   - identify the initial knowledge of the agent (here, $N_A, \mathrm{priv}_A, \mathrm{pub}_B, A, \ldots$)
   - identify the values learned during the protocol, modeled with variables

   Don't think you're done !

2. Initial knowledge of the intruder
   - identify the public data, should be sent to the network
   - identify the private data of the corrupted agents

What to remember when modeling a protocol ?

1. A process for each role of the protocol
   - identify the initial knowledge of the agent (here, $N_A, \mathrm{priv}_A, \mathrm{pub}_B, A, \ldots$)
   - identify the values learned during the protocol, modeled with variables

   Don't think you're done !

2. Initial knowledge of the intruder
   - identify the public data, should be sent to the network
   - identify the private data of the corrupted agents

3. Finally, the complete process
   - put all the roles together
   - don't forget roles where an honest agent talks with a corrupted one !

How to express that a protocol is secure ? (in ProVerif)

Secrecy

Pretty simple :

query attacker(s)

How to express authentication ?

$\rightarrow$ a correspondence property : if B finishes a session, thinking he has talked to A with session nonce N_b then A has also finished a session, thinking she has talked to B with session nonce N_b.

Syntax - enriched

$$
\begin{aligned}
P, Q, R := \; & 0 \\
& \texttt{if } M_1 = M_2 \texttt{ then } P \texttt{ else } Q \\
& \mathsf{in}(c, x).P \\
& \mathsf{out}(c, N).P \\
& \nu n.P \\
& !P \\
& \mathsf{event}(p(u_1, \ldots, u_n)).P
\end{aligned}
$$

where p is a predicate of arity n.

Example : Needham-Schroeder (continued)

Role of A - with events

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

$P_A(\mathsf{priv}_A, \mathsf{pub}_B, A, B, N_A) :=$

$\mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(A, N_A), \mathsf{pub}_B));$
$\mathsf{in}(c, x);$
$\quad \mathsf{let}\ (= N_A, y) = \mathsf{deca}(x, \mathsf{priv}_A)\ \mathsf{in}$
$\quad \mathsf{out}(c, \mathsf{enca}(y, \mathsf{pub}_B));$

Role of A - with events

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

$P_A(\mathsf{priv}_A, \mathsf{pub}_B, A, B, N_A) :=$

event BeginA(A, B, N_A) ;
out(c, enca(pair(A, N_A), pub_B));
in(c, x);
    let $(= N_A, y) = \mathsf{deca}(x, \mathsf{priv}_A)$ in
    out(c, enca(y, pub_B));
event EndA(A, B, y).

Role of B

$$
\begin{array}{lll}
A & \rightarrow B: & \{A, N_a\}_{\mathsf{pub}(B)} \\
B & \rightarrow A: & \{N_a, N_b\}_{\mathsf{pub}(A)} \\
A & \rightarrow B: & \{N_b\}_{\mathsf{pub}(B)}
\end{array}
$$

$P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) :=$

$\mathsf{in}(c, x);$

$\mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in}$

$\mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A));$

$\mathsf{in}(c, z);$

$\mathtt{if}\ \mathsf{deca}(z, \mathsf{priv}_B) = N_B$

$\mathtt{then}\ 0\ \ \mathtt{else}\ 0.$

Role of B

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

$P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) :=$

$\mathsf{in}(c, x);$

$\mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in}$

$\mathsf{event}\ \mathsf{BeginB}(\mathsf{A}, \mathsf{B}, \mathsf{N_B});$

$\mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A));$

$\mathsf{in}(c, z);$

$\mathtt{if}\ \mathsf{deca}(z, \mathsf{priv}_B) = N_B$

$\mathtt{then}\ \ \mathsf{event}\ \mathsf{EndB}(\mathsf{A}, \mathsf{B}, \mathsf{y})\ \mathtt{else}\ 0.$

Role of B

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

$P_B(\mathsf{priv}_B, \mathsf{pub}_A, A, B, N_B) :=$

$\mathsf{in}(c, x);$

$\mathsf{let}\ (= A, y) = \mathsf{deca}(x, \mathsf{priv}_B)\ \mathsf{in}$

$\mathsf{event}\ \mathsf{BeginB}(\mathsf{A}, \mathsf{B}, \mathsf{N_B});$

$\mathsf{out}(c, \mathsf{enca}(\mathsf{pair}(y, N_B), \mathsf{pub}_A));$

$\mathsf{in}(c, z);$

$\mathtt{if}\ \mathsf{deca}(z, \mathsf{priv}_B) = N_B$

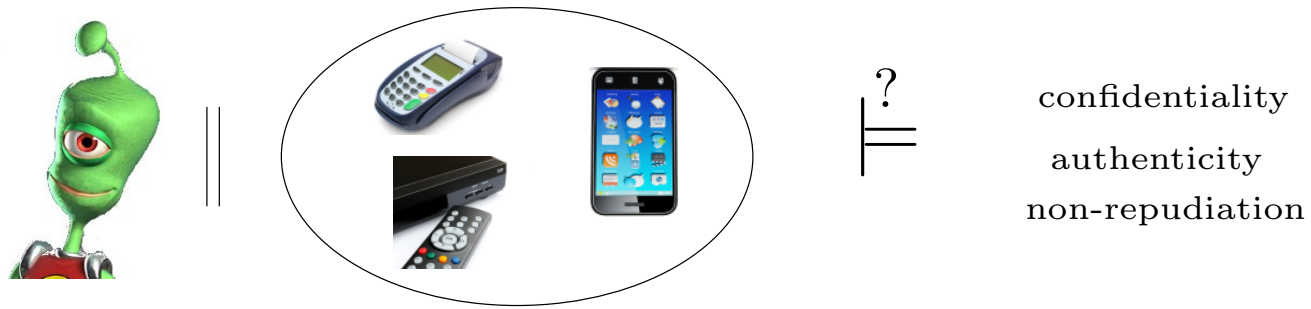
$\mathtt{then}\ \ \mathsf{event}\ \mathsf{EndB}(\mathsf{A}, \mathsf{B}, \mathsf{y})\ \mathtt{else}\ 0.$

Authentication properties :

$$\mathsf{query}\ \forall x\ \mathsf{event}\ \mathsf{EndB}(A, B, x) \quad \Rightarrow \quad \mathsf{event}\ \mathsf{BeginA}(A, B, x).$$

$$\mathsf{query}\ \forall x\ \mathsf{event}\ \mathsf{EndA}(A, B, x) \quad \Rightarrow \quad \mathsf{event}\ \mathsf{BeginB}(A, B, x).$$

How to analyse security protocols ?



Methodology

1. Proposing accurate models
   - symbolic models
   - cryptographic/computational models
2. Proving security
   - decidability/undecidability results
   - tools

Difficulty

Presence of an attacker

- may read every message sent on the net,
- may intercept and send new messages.

$\Rightarrow$ The system is infinitely branching

A simple protocol

$$A : \quad \rightarrow \quad B : \langle \mathsf{Alice}, \mathsf{k} \rangle$$
$$B : \quad \rightarrow \quad A : \langle \mathsf{Bob}, \mathsf{enc}(\mathsf{s}, \mathsf{k}) \rangle$$

A simple protocol

$$A : \quad \rightarrow \quad B : \langle \text{Alice}, \text{k} \rangle$$
$$B : \quad \rightarrow \quad A : \langle \text{Bob}, \text{enc}(\text{s}, \text{k}) \rangle$$

Question ?

Can the attacker learn the secret s ?

A simple protocol

$$A : \quad \rightarrow \quad B : \langle \mathsf{Alice}, \mathsf{k} \rangle$$
$$B : \quad \rightarrow \quad A : \langle \mathsf{Bob}, \mathsf{enc}(\mathsf{s}, \mathsf{k}) \rangle$$

Answer : Of course, Yes !

$$\dfrac{\dfrac{\langle \mathsf{Bob}, \mathsf{enc}(\mathsf{s}, \mathsf{k}) \rangle}{\mathsf{enc}(\mathsf{s}, \mathsf{k})} \qquad \dfrac{\langle \mathsf{Alice}, \mathsf{k} \rangle}{\mathsf{k}}}{\mathsf{s}}$$

Intruder abilities

Composition rules

$$\frac{u \quad v}{\mathsf{pair}(u, v)} \qquad \frac{u \quad v}{\mathsf{enc}(u, v)} \qquad \frac{u \quad v}{\mathsf{enca}(u, v)}$$

Intruder abilities

Composition rules

$$\frac{u \quad v}{\mathrm{pair}(u,v)} \qquad \frac{u \quad v}{\mathrm{enc}(u,v)} \qquad \frac{u \quad v}{\mathrm{enca}(u,v)}$$

Decomposition rules

$$\frac{}{u}\ u \in T \qquad \frac{\mathrm{pair}(u,v)}{u} \qquad \frac{\mathrm{pair}(u,v)}{v}$$

$$\frac{\mathrm{enc}(u,v) \quad v}{u} \qquad \frac{\mathrm{enca}(u,\mathrm{pub}(v)) \quad \mathrm{priv}(v)}{u}$$

Deducibility relation

Deducibility relation

A term u is deducible from a set of terms T, denoted by $T \vdash u$, if there exists a prooftree witnessing this fact.

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \overset{?}{\vdash} k_1,$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3,$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3,$$

$$\dfrac{\dfrac{k_1 \quad k_1}{\mathsf{pair}(k_1, k_1)} \qquad \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1))}{k_3}$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3, S \stackrel{?}{\vdash} \mathsf{pair}(a, k_3),$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \overset{?}{\vdash} k_1, S \overset{?}{\vdash} k_3, S \overset{?}{\vdash} \mathsf{pair}(a, k_3),$$

$$\dfrac{\dfrac{k_1 \quad k_1}{\mathsf{pair}(k_1, k_1)} \qquad \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1))}{k_3}$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3, S \stackrel{?}{\vdash} \mathsf{pair}(a, k_3),$$

$$\dfrac{\dfrac{\dfrac{k_1 \quad k_1}{\mathsf{pair}(k_1, k_1)} \quad \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1))}{k_3} \quad a}{\mathsf{pair}(a, k_3)}$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3, S \stackrel{?}{\vdash} \mathsf{pair}(a, k_3), S \stackrel{?}{\vdash} k_4,$$

Examples

$$S = \left\{ \begin{array}{l} \mathsf{enc}(\mathsf{pair}(\mathsf{pair}(a, k_3), k_4), \\ \mathsf{pair}(k_1, k_2)), \\ a, \\ k_1, \\ \mathsf{enc}(k_3, \mathsf{pair}(k_1, k_1)) \end{array} \right\}$$

$$S \stackrel{?}{\vdash} k_1, S \stackrel{?}{\vdash} k_3, S \stackrel{?}{\vdash} \mathsf{pair}(a, k_3), S \stackrel{?}{\vdash} k_4, S \stackrel{?}{\vdash} \mathsf{pair}(a, k_4)$$

Decision of the intruder problem

Given A set of messages S and a message m

Question Can the intruder learn m from S that is $S \vdash m$?

This problem is decidable in polynomial time. (left as exercice)

Decision of the intruder problem

Given A set of messages S and a message m

Question Can the intruder learn m from S that is $S \vdash m$?

This problem is decidable in polynomial time. (left as exercice)

Lemma (Locality)

If there is a proof of $S \vdash m$ then there is a proof that only uses the subterms of S and m.

Outline

How to decide security for unlimited sessions ?

→ In general, it is undecidable !

(i.e. there exists no algorithm for checking e.g. secrecy)

How to prove undecidability ?

How to decide security for unlimited sessions ?

$\rightarrow$ In general, it is undecidable !

(i.e. there exists no algorithm for checking e.g. secrecy)

How to prove undecidability ?

Post correspondence problem (PCP)

input $\{(u_i, v_i)\}_{1 \leq i \leq n}$, $u_i, v_i \in \Sigma^*$

output $\exists k, i_1, \ldots, i_k \quad u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$

Example : $\{(bab, b), (ab, aba), (a, baba)\}$

Solution ?

How to decide security for unlimited sessions ?

$\rightarrow$ In general, it is undecidable !

(i.e. there exists no algorithm for checking e.g. secrecy)

How to prove undecidability ?

Post correspondence problem (PCP)

input $\{(u_i, v_i)\}_{1 \leq i \leq n}$, $u_i, v_i \in \Sigma^*$

output $\exists k, i_1, \ldots, i_k \quad u_{i_1} \cdots u_{i_k} = v_{i_1} \cdots v_{i_k}$

Example : $\{(bab, b), (ab, aba), (a, baba)\}$

Solution ? $\rightarrow$ Yes, 1,2,3,1.

$bababbab$

$bababbab$

How to encode PCP in protocols ?

Given $\{(u_i, v_i)\}_{1 \leq i \leq n}$, we construct the following protocol P :

$$\begin{aligned} A &\to B : \quad \{\langle \overline{u_1}, \overline{v_1} \rangle\}_{K_{ab}}, \ldots, \{\langle \overline{u_k}, \overline{v_k} \rangle\}_{K_{ab}} \\ B : \{\langle x, y \rangle\}_{K_{ab}} &\to A : \quad \{\langle \overline{x, u_1}, \overline{y, v_1} \rangle\}_{K_{ab}}, \{s\}_{\{\langle \overline{x,u_1}, \overline{x,u_1} \rangle\}_{K_{ab}}}, \\ &\qquad \ldots, \{\langle \overline{x, u_k}, \overline{y, v_k} \rangle\}_{K_{ab}}, \{s\}_{\{\langle \langle \overline{x,u_k}, \overline{x,u_k} \rangle\}_{K_{ab}}} \end{aligned}$$

where $\overline{a_1 \cdot a_2 \cdots a_n}$ denotes the term $\langle \cdots \langle \langle a_1, a_2 \rangle, a_3, \rangle \ldots a_n \rangle$.

How to encode PCP in protocols ?

Given $\{(u_i, v_i)\}_{1 \leq i \leq n}$, we construct the following protocol P :

$$\begin{aligned} A &\rightarrow B : \quad \{\langle \overline{u_1}, \overline{v_1} \rangle\}_{K_{ab}}, \ldots, \{\langle \overline{u_k}, \overline{v_k} \rangle\}_{K_{ab}} \\ B : \{\langle x, y \rangle\}_{K_{ab}} &\rightarrow A : \quad \{\langle \overline{x}, \overline{u_1}, \overline{y}, \overline{v_1} \rangle\}_{K_{ab}}, \{s\}_{\{\langle \overline{x}, \overline{u_1}, \overline{x}, \overline{u_1} \rangle\}_{K_{ab}}}, \\ &\qquad \ldots, \{\langle \overline{x}, \overline{u_k}, \overline{y}, \overline{v_k} \rangle\}_{K_{ab}}, \{s\}_{\{\langle \langle \overline{x}, \overline{u_k}, \overline{x}, \overline{u_k} \rangle\}_{K_{ab}}} \end{aligned}$$

where $\overline{a_1 \cdot a_2 \cdots a_n}$ denotes the term $\langle \cdots \langle \langle a_1, a_2 \rangle, a_3, \rangle \ldots a_n \rangle$.

Then there is an attack on P iff there is a solution to the Post Correspondence Problem with entry $\{(u_i, v_i)\}_{1 \leq i \leq n}$.