Communication security: Formal models and proofs

Hubert Comon

September 1, 2016

1 Introduction to protocol security

The context (I)

- credit cards
- contactless cards
- telephones
- online transactions
- cars, fridges,... Internet of Things
- Big Brother: NSA
- Biomedical applications
- ...

The context (III)

- Security protocols
- Testing is not very useful
- Hiding the code is not a good idea
- The scope of formal methods

A simple handshake protocol

$$\begin{array}{ll} A \to B: & \nu n, r.\ \mathsf{aenc}(\langle A, n\rangle, \mathsf{pk}(\mathsf{sk}_B), r) \\ B \to A: & \nu r'.\ \mathsf{aenc}(n, \mathsf{pk}(\mathsf{sk}_A), r') \end{array}$$

The formal verification problem

$$\forall \mathcal{A}. \quad \mathcal{A} \parallel P \quad \models \quad \phi$$

$$\forall \mathcal{A}. \quad \mathcal{A} \parallel P_1 \quad \sim \quad \mathcal{A} \parallel P_2$$

Universal quantification on $\mathcal{A}$: we cannot apply directly model-checking techniques.

One important issue: range of $\mathcal{A}$?

Attacker models

The DY-attacker
Messages are terms, the attacker is defined through an equation theory or an inference system

The computational attacker
Messages are bitstrings, the attacker is a probabilistic polynomial time Turing machine

Other attackers

Goals of the lecture

Verification inputs

- Cryptographic libraries
- Protocol programs
- Attacker model
- Security property

Goals of the lecture
Show how to derive the proof obligations in a parametric way, abstracting from crypto libraries, attacker models.

Focus on the semantics of protocols, for arbitrary libraries and attacker models.

Roadmap

4 successive versions of the calculus, by increasing expressiveness (we could have considered the last case only...)

1. Simple case
2. Adding events: required for agreement properties
3. Adding replication
4. Adding channel generation: required for computational semantics

Then indistinguishability properties (privacy).

2 A simple version of the process calculus

Cryptographic libraries
Syntax

- An arbitrary set of cryptographic primitives $\mathcal{F}$: hash, public-key encryption(s), symmetric encryption(s), zkp,... represented by (typed) function symbols
- At least one random generation algorithm. Random numbers are represented by *names* $n, n_1, r, ...$ out of a set $\mathcal{N}$

Terms are built over variables, function symbols and names.

Cryptographic libraries
Semantics

$\mathcal{M}$ is an interpretation domain. Typically ground or constructor terms (the DY semantics) or bitstrings (the computational semantics).

$\mathcal{M}$ includes error messages (exceptions) `Err`.

If σ is an environment (mapping from variables to $\mathcal{M}$), u is a term,

$$[\![u]\!]_\sigma^{\mathcal{M}}$$

is the interpretation of u in $\mathcal{M}$ w.r.t. σ: $\mathcal{M}$ is a (partial) $\mathcal{F}$-algebra.

The interpretation is strict:

$$u_i \in \texttt{Err} \quad \Rightarrow \quad [\![f(u_1, \ldots, u_n)]\!]_\sigma^{\mathcal{M}} \in \texttt{Err}$$

Cryptographic libraries
A possible set of function symbols

- $\mathsf{aenc}(u, pk, r)$ is (supposed to be) the asymmetric encryption of u with the public key pk and random input r.
- $\mathsf{dec}(u, \mathsf{sk})$ is (supposed to be) the decryption of u with the secret key sk

- $\mathsf{pk}(\mathsf{sk})$ is (supposed to be) the public key associated with the secret key sk
- $\langle u, v \rangle$
- $\pi_1(u), \pi_2(u)$

Cryptographic libraries
A DY model

$\mathcal{M}_{DY}$ (messages) is the least set of ground terms such that:

- $\mathcal{N} \subseteq \mathcal{M}_{DY}$
- if $u, v \in \mathcal{M}_{DY}$ then $\langle u, v \rangle \in \mathcal{M}_{DY}$
- if $k \in \mathcal{N}$ then $\mathsf{pk}(k) \in \mathcal{M}_{DY}$
- if $u \in \mathcal{M}_{DY}, k, r \in \mathcal{N}$, then $\mathsf{aenc}(u, \mathsf{pk}(k), r) \in \mathcal{M}_{DY}$.

$\mathcal{M}_{DY}$ also includes special error terms `Err` (not messages).

$$\begin{array}{rcll} \mathsf{dec}(\mathsf{aenc}(u, \mathsf{pk}(k), r), k) & \to & u & \text{For } k, r \in \mathcal{N},\ u \text{ a message} \\ \pi_1(\langle u, v \rangle) & \to & u & u, v \text{ are messages} \\ \pi_2(\langle u, v \rangle) & \to & v & u, v \text{ are messages} \end{array}$$

$$[\![u]\!]_\sigma^{\mathcal{M}_{DY}} = u\sigma\downarrow$$

Any irreducible ground term, which is not a message, is an error.

Cryptographic libraries
Computational models

- $\eta \in \mathbb{N}$ is a security parameter
- τ maps $\mathcal{N}$ to $\{0,1\}^\eta$
- $\mathcal{M}_c(\tau, \eta) \subseteq \{0,1\}^*$
- $[\![n]\!]^{\mathcal{M}_c(\tau,\eta)} = \tau(n)$
- $\mathsf{aenc}(\_, \_, \_), \mathsf{dec}(\_, \_), \mathsf{pk}(\_)$ are interpreted as a public-key encryption scheme.
- with an interpretation of pairing/projections, $\mathcal{M}_c(\tau, \eta)$ is an $\mathcal{F}$-algebra

A simple process calculus

Syntax

P ::=	0		null process (stalled)
	$\mathsf{in}(x).P$		input of x (binds x)
	$\mathsf{out}(t).P$		output of t
	$\mathsf{if}\ \mathtt{EQ}(u,v)\ \mathsf{then}\ P\ \mathsf{else}\ P$		conditional branching
	$\mathsf{let}\ y = u\ \mathsf{in}\ P$		evaluation (binds y)
	$\nu \overline{n}.P$		random generation
	$P \| P$		parallel composition

All variable occurrences are bound.

Example

The simple handshake protocol

$$A \to B: \quad \nu n, r.\ \mathsf{aenc}(\langle A, n\rangle, \mathsf{pk}(\mathsf{sk}_B), r)$$
$$B \to A: \quad \nu r'.\ \mathsf{aenc}(n, \mathsf{pk}(\mathsf{sk}_A), r')$$

$$A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_b)) = \nu n, r.\ \mathsf{out}(\mathsf{aenc}(\langle \mathsf{pk}(\mathsf{sk}_a), n\rangle, \mathsf{pk}(\mathsf{sk}_b), r)).\ \mathsf{in}(z).\ \mathsf{let}\ z_1 = \mathsf{dec}(z, \mathsf{sk}_a)\ \mathsf{in}\ \mathsf{if}\ \mathtt{EQ}(z_1, n)\ \mathsf{then}\ \mathbf{0}(Success)\ \mathsf{else}\ \mathbf{0}(Fail)$$

$$B(\mathsf{sk}_b) = \nu r'.\ \mathsf{in}(x).\mathsf{let}\ y = \mathsf{dec}(x, \mathsf{sk}_b)\ \mathsf{in}\ \mathsf{let}\ y_1 = \pi_1(y)\ \mathsf{in}\ \mathsf{let}\ y_2 = \pi_2(y)\ \mathsf{in}\ \mathsf{out}(\mathsf{aenc}(y_2, y_1, r')).\ \mathbf{0}.$$

$$\nu \mathsf{sk}_a, \mathsf{sk}_b.\quad \mathsf{out}(\langle \mathsf{pk}(\mathsf{sk}_a), \mathsf{pk}(\mathsf{sk}_b)\rangle).\quad (A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_b)) \parallel B(\mathsf{sk}_b))$$

Structural equivalence

$$\begin{aligned}
0 \parallel P &\equiv P \\
P \parallel Q &\equiv Q \parallel P \\
P \parallel (Q \parallel R) &\equiv (P \parallel Q) \parallel R \\
\nu n.P &\equiv \nu n'.P\{n \mapsto n'\} \\
\mathsf{in}(x).P &\equiv \mathsf{in}(x').P\{x \mapsto x'\} \\
\mathsf{let}\ x = u\ \mathsf{in}\ P &\equiv \mathsf{let}\ x' = u\ \mathsf{in}\ P\{x \mapsto x'\} \\
(\nu n.P) \| Q &\equiv \nu n'.(P \| Q) \qquad \text{if } n \notin \mathsf{freenames}(Q)
\end{aligned}$$

Operational semantics

States of the network are tuples (ϕ, σ, P), where

- ϕ is a *frame* of the form $\nu\overline{n}.m_1, \ldots, m_k$, where $\overline{n}$ is a set of names (used so far) and $m_1, \ldots, m_k$ is a sequence of values in $\mathcal{M}$ (that have been sent out so far)
- σ is an environment: an assignment of the free variables to values in $\mathcal{M}$
- P is a process

The semantics is a labeled transition system, whose labels are the inputs provided by the attacker (sometimes, an empty input)

Operational semantics
The transition system (I)

$$\frac{}{(\phi, \sigma, \mathsf{in}(x).P) \xrightarrow{u} (\phi, \sigma \uplus \{x \mapsto u\}, P)}$$

$$\frac{(\phi, \sigma, P) \xrightarrow{u} (\phi', \sigma', P')}{(\phi, \sigma, \mathsf{if}\ \mathtt{EQ}(s,t)\ \mathsf{then}\ P\ \mathsf{else}\ Q) \xrightarrow{u} (\phi', \sigma', P')}$$

if $[\![s]\!]^{\mathcal{M}}_{\sigma} = [\![t]\!]^{\mathcal{M}}_{\sigma} \notin \mathtt{Err}$

$$\frac{(\phi, \sigma, Q) \xrightarrow{u} (\phi', \sigma', P')}{(\phi, \sigma, \mathsf{if}\ \mathtt{EQ}(s,t)\ \mathsf{then}\ P\ \mathsf{else}\ Q) \xrightarrow{u} (\phi', \sigma', P')}$$

if $[\![s]\!]^{\mathcal{M}}_{\sigma} \neq [\![t]\!]^{\mathcal{M}}_{\sigma}$ or $[\![s]\!]^{\mathcal{M}}_{\sigma} \in \mathtt{Err}$ or $[\![t]\!]^{\mathcal{M}}_{\sigma} \in \mathtt{Err}$

Operational semantics
The transition system (II)

$$\frac{}{(\phi, \sigma, \mathsf{let}\ x = u\ \mathsf{in}\ P) \rightarrow (\phi, \sigma \uplus \{x \mapsto w\}, P)}\ \mathtt{if}\ [\![u]\!]^{\mathcal{M}}_{\sigma} = w \notin \mathtt{Err}$$

$$\frac{}{(\nu \overline{n}.\theta, \sigma, \mathsf{out}(s).P) \rightarrow (\nu \overline{n}.\theta \cdot [\![s]\!]^{\mathcal{M}}_{\sigma}, \sigma, P)}$$

$$\frac{(\phi, \sigma, P) \xrightarrow{u} (\phi', \sigma', P')}{(\phi, \sigma, P \| Q) \xrightarrow{u} (\phi', \sigma', P' \| Q)}$$

$$\frac{}{(\nu \overline{n}.\theta, \sigma, \nu n.P) \rightarrow \nu \overline{n} \uplus n.\theta, \sigma, P)}\ \text{if}\ n \notin \overline{n} \cup \mathsf{freename}(\theta)$$

Example

Restricting the feasible transitions

$$(\phi_1, \sigma_1, P_1) \xrightarrow{u_1} \cdots \xrightarrow{u_{k-1}} (\phi_k, \sigma_k, P_k)$$

is possible w.r.t. model $\mathcal{M}$ and an attacker $\mathcal{A}$ if , for every i,

$$\mathcal{A}([\![\phi_i]\!]^{\mathcal{M}}_{\sigma_i}, P_i) = [\![u_i]\!]^{\mathcal{M}}_{\sigma_i}$$

Note: could include a state in $\mathcal{A}$.

Example DY

There is a DY attacker $\mathcal{A}$ such that $\mathcal{A}(\phi) = [\![u]\!]^{\mathcal{M}_{DY}}_{\sigma}$ iff

$$\phi \vdash_I u\sigma \downarrow$$

where I is defined by:

$$\frac{\phi \vdash u_1 \cdots \phi \vdash u_n}{\phi \vdash f(u_1, \ldots, u_n) \downarrow}$$

For every $f \in \mathcal{F}$

$$\frac{}{\nu \overline{n}.u_1, \ldots, u_n \vdash u_i}$$

$$\frac{}{\nu \overline{n}.\theta \vdash n'}$$

if $n' \in \mathcal{N} \setminus \overline{n}$.

Exercise

In the simple handshake example, describe all feasible transition sequences in the DY model (assume the name extrusion, let, conditionals and outputs are always performed before inputs).

Is the nonce n secret ?

Example computational

$\mathcal{A}$ is a Probabilistic Polynomial Time Turing machine (PPT).

Some inputs that were not possible in the DY model might now be possible.

A typical example

$\mathcal{A}$ might be able to compute (with a significant probability) $[\![\mathsf{aenc}(u, \mathsf{pk}(k_1), r_1)]\!]^{\mathcal{M}_c(\tau,\eta)}$ from $[\![\mathsf{aenc}(v, \mathsf{pk}(k_1), r_1)]\!]^{\mathcal{M}_c(\tau,\eta)}$

$$\exists \mathcal{A}, \quad \mathbf{Prob}\{\tau, \rho : \mathcal{A}([\![\mathsf{aenc}(v, \mathsf{pk}(k_1), r_1)]\!]^{\mathcal{M}_c(\tau,\eta)}) = [\![\mathsf{aenc}(u, \mathsf{pk}(k_1), r_1)]\!]^{\mathcal{M}_c(\tau,\eta)}\} \quad > \epsilon(\eta)$$

ϵ is *non-negligible*: there is a polynomial Pol such that

$$\liminf_{\eta \to +\infty} \epsilon(\eta) \times \mathsf{Pol}(\eta) > 1$$

Confidentiality

In the DY case
Is there a DY attacker $\mathcal{A}$ and a feasible transition sequence

$$(\emptyset, \emptyset, P) \xrightarrow{*} (\phi, \sigma, Q)$$

such that $\mathcal{A}(\phi, Q) = s$? **This problem is in NP**

In the computational case
Is there a PPT $\mathcal{A}$ such that, for every computational model $\mathcal{M}_c(\tau, \eta)$, the probability that there is a feasible sequence

$$(\emptyset, \emptyset, P) \xrightarrow{*} (\phi, \sigma, Q)$$

such that $\mathcal{A}(\phi, Q) = s$ is negligible in η ?
This requires in general assumptions on the libraries

For example, the protocol

$$\nu n \nu s.\mathsf{in}(x).\mathsf{if}\ \mathtt{EQ}(x, n)\ \mathsf{then}\ \mathsf{out}(s) \cdot \mathbf{0}\ \mathsf{else}\ \mathbf{0}$$

satisfies the confidentiality of s in the computational model, as soon as n is uniformly drawn at random. (For any attacker the probability of success is bounded by $\frac{1}{2^\eta}$).

Exercises
In the following cases, give reasonable processes A, B and either give an attack on the confidentiality of s or prove that there is no such attack in the DY model.

1.

$$\begin{array}{ll} A \to B: & \nu n, \nu r.\ \langle \mathsf{pk}(\mathsf{sk}_A), \mathsf{aenc}(s, \mathsf{pk}(\mathsf{sk}_B), r)\rangle \\ B \to A & \nu r'.\ \langle \mathsf{pk}(\mathsf{sk}_B), \mathsf{aenc}(s, \mathsf{pk}(\mathsf{sk}_A), r')\rangle \end{array}$$

$$P = \nu \mathsf{sk}_a, \nu \mathsf{sk}_b.\ \mathsf{out}(\langle \mathsf{pk}(\mathsf{sk}_A), \mathsf{pk}(\mathsf{sk}_B)\rangle)\ \cdot\ (A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_B))\ \|\ B(\mathsf{sk}_b))$$

2.

$$\begin{array}{ll} A \to B: & \nu s, r_1, r_2.\ \mathsf{aenc}(\langle \mathsf{pk}(\mathsf{sk}_A), \mathsf{aenc}(s, \mathsf{pk}(\mathsf{sk}_B), r_1)\rangle, \mathsf{pk}(\mathsf{sk}_B), r_2) \\ B \to A: & \nu r_3, r_4.\mathsf{aenc}(\langle \mathsf{pk}(\mathsf{sk}_B), \mathsf{aenc}(s, \mathsf{pk}(\mathsf{sk}_A), r_3)\rangle, \mathsf{pk}(\mathsf{sk}_A), r_4) \end{array}$$

$$P = \nu \mathsf{sk}_a, \nu \mathsf{sk}_b.\ \mathsf{out}(\langle \mathsf{pk}(\mathsf{sk}_A), \mathsf{pk}(\mathsf{sk}_B)\rangle)\ \cdot\ (A(\mathsf{sk}_a, \mathsf{pk}(\mathsf{sk}_B))\ \|\ B(\mathsf{sk}_b)\ \|\ B(\mathsf{sk}_b))$$

3 Symbolic (Abstract) semantics

Gathering feasability conditions

States of the network are tuples $(\phi, \sigma, P, \theta)$, where

- ϕ, σ, P as before
- θ is a *constraint*: equalities, disequalities and computational constraints of the form $\phi \triangleright u$.

$$\frac{}{(\phi, \sigma, \mathsf{in}(x).P, \theta)\ \to\ (\phi, \sigma, P, \theta \wedge \phi \triangleright x)}$$

$$\frac{}{(\phi, \sigma, \mathsf{if}\ \mathtt{EQ}(s,t)\ \mathsf{then}\ P\ \mathsf{else}\ Q, \theta)\ \to\ (\phi, \sigma, P, \theta \wedge \mathtt{EQ}(s,t))}$$

$$\frac{}{(\phi, \sigma, \mathsf{if}\ \mathtt{EQ}(s,t)\ \mathsf{then}\ P\ \mathsf{else}\ Q, \theta)\ \to\ (\phi, \sigma, P, \theta \wedge \neg\mathtt{EQ}(s,t))}$$

Consequences

Advantages

- A finite transition system (regardless of the model)
- Confidentiality reduces to constraint satisfaction

$$\theta \wedge \phi_f \triangleright s$$

in NP in the DY model

Consequences

Computational case
Specify the assumptions on the libraries: impossibility conditions.

$$\not\vdash n$$

$$S, \mathsf{aenc}(n, \mathsf{pk}(k), r) \triangleright n \quad \Rightarrow \quad S \triangleright n$$

$$S_1 \triangleright x \wedge S_2, x \triangleright y \quad \Rightarrow \quad S_1, S_2 \triangleright y$$

$$S_1 \triangleright x_1 \wedge \ldots \wedge S_n \triangleright x_n \quad \Rightarrow \quad S_1, \ldots, S_n \triangleright f(x_1, \ldots, x_n)$$

S, S_1, S_2 are finite sets of terms. [3mm]

Check the constraint satisfiability, together with $\phi \triangleright s$ and the above axioms (in PTIME !!)

Exercise

Back to the simple handshake protocol. Study its security in the computational model, assuming the properties of the cryptographic libraries that are described in the lecture.